



# Data Analytics Methods for Enterprise-wide Optimization under Uncertainty

Submitted in partial fulfillment of the requirements for  
the degree of  
Doctor of Philosophy  
in  
Chemical Engineering

Bruno A. Calfa

B.S., Chemical Engineering, Pontifical Catholic University of Rio de Janeiro,  
Brazil

Carnegie Mellon University  
Pittsburgh, PA 15213

April, 2015

## Acknowledgments

First, I would like to express my sincere gratitude and appreciation to my advisor, Dr. Ignacio E. Grossmann. Ignacio has not only been instrumental in mentoring me with regards to Process Systems Engineering (PSE) research, but also has been extremely supportive and understanding. His passion for advancing chemical engineering research and education, diligence, charisma, and strong ethical nature have helped shape and will continue to influence my professional career.

I would like to thank the committee members (Larry Biegler, Nick Sahinidis, Alan Scheller-Wolf, and John Wassick) for their patience, guidance, and expertise.

I also would like to acknowledge The Dow Chemical Company not only for the financial support throughout my Ph.D., but also for the very positive and enriching opportunity to work on interesting, industrially-relevant problems. I consider myself privileged for this collaborative effort. I am very thankful for the two internships in Midland, MI that allowed me to experience the role of a Research & Development group in a large chemical company and its interaction with different businesses. I should mention that the discussions with the industrial collaborators were very fruitful and they taught me a great deal about presenting research findings to potential users of the developed approaches. I would like to specially acknowledge the following collaborators with whom I had more direct contact: Anshul Agarwal, John Wassick, Scott Bury, Joe Czyzyk, Ameya Dhaygude, Tao Wu, Laurie Hunt, and Alex Kalos.

I am also very thankful for my colleagues in the PSE group at CMU. In addition to several interesting discussions we had about our research, their friendliness has made my stay at CMU more enjoyable and memorable.

Lastly, but definitely not least, I am eternally grateful for my family for the steadfast emotional and financial support. Naturally, there are many factors that put me on the path to all I have achieved at this point in my life, but among them all, my family played a vital role in my successes. I love you, André, Iara, and Giovanni.

## Abstract

This dissertation primarily proposes data-driven methods to handle uncertainty in problems related to Enterprise-wide Optimization (EWO). Data-driven methods are characterized by the direct use of data (historical and/or forecast) in the construction of models for the uncertain parameters that naturally arise from real-world applications. Such uncertainty models are then incorporated into the optimization model describing the operations of an enterprise. Before addressing uncertainty in EWO problems, [Chapter 2](#) deals with the integration of deterministic planning and scheduling operations of a network of batch plants. The main contributions of this chapter include the modeling of sequence-dependent changeovers across time periods for a unit-specific general precedence scheduling formulation, the hybrid decomposition scheme using Bilevel and Temporal Lagrangean Decomposition approaches, and the solution of subproblems in parallel. Chapters [3](#) to [6](#) propose different data analytics techniques to account for stochasticity in EWO problems. [Chapter 3](#) deals with scenario generation via statistical property matching in the context of stochastic programming. A distribution matching problem is proposed that addresses the under-specification shortcoming of the originally proposed moment matching method. [Chapter 4](#) deals with data-driven individual and joint chance constraints with right-hand side uncertainty. The distributions are estimated with kernel smoothing and are considered to be in a confidence set, which is also considered to contain the true, unknown distributions. The chapter proposes the calculation of the size of the confidence set based on the standard errors estimated from the smoothing process. [Chapter 5](#) proposes the use of quantile regression to model production variability in the context of Sales & Operations Planning. The approach relies on available historical data of actual vs. planned production rates from which the deviation from plan is defined and considered a random variable. [Chapter 6](#) addresses the combined optimal procurement contract selection and pricing problems. Different price-response models, linear and nonlinear, are considered in the latter problem. Results show that setting selling prices in the presence of uncertainty leads to the use of different purchasing contracts.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Enterprise-wide Optimization: Planning, Scheduling, and Decomposition Strategies . . . . .	2
1.2	Probability and Statistics: An Overview . . . . .	5
1.2.1	Basics . . . . .	5
1.2.2	Applications . . . . .	9
1.3	Optimization Under Uncertainty . . . . .	18
1.3.1	Stochastic Programming . . . . .	19
1.3.2	Chance-Constrained Programming . . . . .	19
1.4	Overview of the Thesis . . . . .	20
1.4.1	Chapter 2 . . . . .	20
1.4.2	Chapter 3 . . . . .	21
1.4.3	Chapter 4 . . . . .	21
1.4.4	Chapter 5 . . . . .	22
1.4.5	Chapter 6 . . . . .	22
1.4.6	Chapter 7 . . . . .	22
<b>2</b>	<b>Hybrid Bilevel-Lagrangian Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants</b>	<b>24</b>
2.1	Introduction . . . . .	27
2.2	Problem Definition . . . . .	28
2.3	Problem Formulation . . . . .	30
2.3.1	Planning Model . . . . .	31
2.3.2	Material and Inventory Balances . . . . .	34
2.3.3	Capacities . . . . .	36
2.3.4	Demand Satisfaction . . . . .	37
2.3.5	Modified USGP Scheduling Model . . . . .	38
2.4	Decomposition Strategies . . . . .	43
2.4.1	Bilevel Decomposition . . . . .	43
2.4.2	Lagrangian Decomposition . . . . .	44
2.4.3	Hybrid BD-TLD Scheme . . . . .	47
2.5	Computational Results . . . . .	49
2.5.1	Example 1 . . . . .	50
2.5.2	Example 2 . . . . .	52

2.5.3	Example 3 . . . . .	53
2.6	Conclusions . . . . .	55
<b>3</b>	<b>Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching</b>	<b>57</b>
3.1	Introduction . . . . .	57
3.2	Two-Stage Scenario Tree Generation . . . . .	59
3.2.1	$L^2$ Moment Matching Problem . . . . .	60
3.2.2	$L^1$ and $L^\infty$ Moment Matching Problems . . . . .	63
3.2.3	Remarks on the MMP Formulations . . . . .	67
3.3	Distribution Matching Problem . . . . .	68
3.3.1	Example 1: Uncertain Plant Yield . . . . .	72
3.4	Multi-Stage Scenario Tree Generation . . . . .	78
3.4.1	NLP Approach . . . . .	80
3.4.2	LP Approach . . . . .	81
3.4.3	Example 2: Uncertain Product Demands . . . . .	83
3.5	Conclusions . . . . .	90
<b>4</b>	<b>Data-Driven Individual and Joint Chance-Constrained Optimization via Kernel Smoothing</b>	<b>92</b>
4.1	Introduction . . . . .	92
4.2	Problem Statement . . . . .	94
4.3	Data-Driven Chance Constraints . . . . .	94
4.4	Kernel-Based Reformulation of CCs . . . . .	97
4.4.1	Individual Chance Constraints . . . . .	98
4.4.2	Joint Chance Constraints . . . . .	100
4.4.3	Calculation of $\phi$ -Divergence Tolerance . . . . .	103
4.4.4	Addressing Small Sample Size Cases . . . . .	105
4.5	Initialization Algorithm for JCCs . . . . .	107
4.6	Numerical Examples . . . . .	109
4.6.1	Motivating Example . . . . .	109
4.6.2	Industrial Example . . . . .	117
4.7	Conclusions . . . . .	119
<b>5</b>	<b>Data-Driven Simulation and Optimization Approaches to Incorporate Production Variability in Sales and Operations Planning</b>	<b>121</b>
5.1	Introduction . . . . .	121
5.2	Problem Statement and Methodology . . . . .	123
5.3	Modeling Production Variability with Quantile Regression . . . . .	124
5.4	Simulation and Optimization Frameworks . . . . .	125
5.4.1	Sim-Opt Framework . . . . .	126
5.4.2	Bi-Opt Framework . . . . .	128
5.5	Numerical Examples . . . . .	130
5.5.1	Motivating Example . . . . .	131

5.5.2	Industrial Case Study . . . . .	141
5.6	Conclusions . . . . .	145
<b>6</b>	<b>Optimal Procurement Contract Selection with Price Optimization under Uncertainty for Process Networks</b>	<b>146</b>
6.1	Introduction . . . . .	146
6.2	Literature Review . . . . .	147
6.2.1	Contract Modeling and Selection . . . . .	147
6.2.2	Pricing Analytics . . . . .	148
6.3	Problem Statement . . . . .	149
6.4	Production Planning Model . . . . .	150
6.4.1	Deterministic Model . . . . .	150
6.4.2	Procurement Contract Models . . . . .	152
6.4.3	Price Optimization Models . . . . .	153
6.4.4	Stochastic Programming Model . . . . .	155
6.5	Numerical Examples . . . . .	158
6.5.1	Example 1: Small Process Network . . . . .	159
6.5.2	Example 2: Large Process Network . . . . .	164
6.5.3	Reformulation of MINLP Models . . . . .	168
6.6	Conclusions . . . . .	169
<b>7</b>	<b>Conclusions</b>	<b>172</b>
7.1	Hybrid Bilevel-Lagrangian Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants . . . . .	172
7.2	Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching . . . . .	174
7.3	Data-Driven Individual and Joint Chance-Constrained Optimization via Kernel Smoothing . . . . .	175
7.4	Data-Driven Simulation and Optimization Approaches to Incorporate Production Variability in Sales and Operations Planning	177
7.5	Optimal Procurement Contract Selection with Price Optimization under Uncertainty for Process Networks . . . . .	178
7.6	Contributions of the Thesis . . . . .	179
7.7	Recommendations for Future Work . . . . .	180
	<b>Bibliography</b>	<b>185</b>
	<b>Appendices</b>	<b>200</b>
	Appendix A Hybrid BD-TLD . . . . .	201
	A.1 Data for Example 1 . . . . .	201
	Appendix B Data-Driven Scenario Generation . . . . .	206
	B.1 Data for Example 1 . . . . .	206
	B.2 Data for Example 2 . . . . .	209
	B.3 Time Series Analysis and Scenario Generation . . . . .	209

B.4	Assessing Solution Quality in Stochastic Programming	210
Appendix C	Data-Driven Chance Constraints . . . . .	214
C.1	Motivation of Chance-Constrained Optimization . . . .	214
C.2	Derivation of Bound on ICC Risk Levels for Equivalent JCC . . . . .	217
C.3	Data for Motivating Example . . . . .	218
Appendix D	Procurement Contracts and Pricing . . . . .	221
D.1	Contract Selection Models . . . . .	221
D.2	Concavity of Sales Term when using Logit Demand- Response Model . . . . .	222
D.3	Data for Example 1 . . . . .	223
Appendix E	Future Work: Data-Driven Chance Constraints . . .	226
E.1	Linear Chance Constraints . . . . .	226



# List of Tables

1.1	Definitions of probability mass (PMF), density (PDF), and cumulative distribution (CDF) functions of random variables.	6
1.2	Some kernel functions commonly used in kernel density estimation. . . . .	15
2.1	Cost breakdown for Example 1. . . . .	51
2.2	Problems sizes for Example 1. . . . .	52
2.3	Cost breakdown for Example 2. . . . .	53
2.4	Problems sizes for Example 2. . . . .	53
2.5	Cost breakdown for Example 3. . . . .	54
2.6	Problems sizes for Example 3. . . . .	55
3.1	Objective function values of the DMP formulations in Example 1. The values represent the error of matching the statistical properties. . . . .	76
3.2	Expected profit of the production planning model in Example 1 using the scenario trees from two approaches. . . . .	76
3.3	Average value and upper bound of the optimality gap of the stochastic production planning model in Example 1. . . . .	78
3.4	Expected profit in Example 2 using the scenario trees from heuristic and optimization-based approaches. . . . .	86
3.5	Average value and upper bound of the optimality gap of the stochastic production planning model in Example 2. . . . .	90
4.1	Some divergence functions and the reduced risk level $\alpha'$ for data-driven chance constraints (see <a href="#">Jiang &amp; Guan (2013)</a> for general expression and proofs). . . . .	97
4.2	Problem sizes for motivating example. . . . .	116
4.3	Problem sizes for industrial test case. . . . .	118
5.1	Potential advantages (+) and disadvantages (−) of optimization-based frameworks. Legend: Sim-Opt = Simulation-Optimization framework, Bi-Opt = Bi-Objective Optimization framework, $\Delta Plan$ = deviation conditional on planned values in the context of quantile regression (see Section 5.3), DFO = Derivative-Free Optimization. . . . .	126

5.2	Average profit and its standard deviation in monetary units (m.u.) for Cases 1–3 in the motivating example. The standard deviation of the profit is a measure of its spread, i.e., financial risk. . . . .	134
5.3	Average unmet demand in weight units (w.u.) for each product in Case 2 and Case 3 in the motivating example. . . . .	134
5.4	Average allocated amounts of A in weight units (w.u.) to each downstream plant in Case 2 and Case 3 in the motivating example. The percentage change column is the relative change between the two cases, i.e., (Case 3 – Case 2)/Case 3. . . . .	135
5.5	Computational statistics for optimization models in all cases and subcases in the motivating example. . . . .	140
5.6	Computational statistics for optimization models in all cases and subcases in the industrial case study. CPU and wall times are shown in separate rows. . . . .	144
6.1	Typical price-response models (PRMs). . . . .	153
6.2	Typical demand-response models (DRMs). . . . .	154
6.3	Parameter values for the DRMs in Example 1. . . . .	160
6.4	Computational results for Example 1. C-E stands for Constant-Elasticity. . . . .	161
6.5	Computational results for Example 2. C-E stands for Constant-Elasticity. . . . .	165
6.6	Computational results comparing original and reformulated MINLP models. C-E stands for Constant-Elasticity. Speed-Up is calculated as the ratio between CPU time of original and reformulated models. . . . .	170
A.1	Product-Group mapping. . . . .	201
A.2	Processing data where the acronyms have the following meanings: batch size (BS), batches per hour (BPH), and scale-up cost (SUC). . . . .	201
A.3	Product demands. . . . .	202
A.4	Outbound transportation costs. If product-plant-customer tuple is not listed, then the shipment is not allowed. . . . .	203
A.5	Inbound transportation costs. . . . .	204
A.6	Plant-to-Plant transportation costs. . . . .	204
A.7	Operating costs. . . . .	204
A.8	Plants’ capacities. . . . .	204
A.9	Bill of materials for finished products. The column Unit Ratio represents the parameter $CRR_{ril}$ . If plant-raw material-product-unit ratio tuple is not listed, then the unit ratio is zero. . . . .	205

A.10	Bill of materials for blended products. The column Unit Ratio represents the parameter $BR_{i'j'l}$ . If plant-product-blend tuple is not listed, then the unit ratio is zero. . . . .	206
A.11	Changeover times [hr] (costs [\$]) between groups of products. . . . .	206
B.1	First-stage production yield ( $\theta_f$ at $t = 1$ ) [-]. For facilities $P2$ and $P3$ , the yields remain the same at the second stage. . . . .	207
B.2	Product demand ( $\xi_{m,t}$ ) [t]. . . . .	208
B.3	Maximum inventory ( $w_{m,t}^{inv,max}$ ) [t]. . . . .	208
B.4	Maximum (minimum) capacity ( $w_{f,t}^{min}$ and $w_{f,t}^{max}$ ) [t]. . . . .	208
B.5	Selling price ( $SP_{m,t}$ ) [\$ / t]. . . . .	208
B.6	Operating cost ( $OPC_{f,t}$ ) [\$ / t]. . . . .	208
B.7	Purchase cost ( $PC_{m,t}$ ) [\$ / t]. . . . .	208
B.8	Inventory cost ( $IC_{m,t}$ ) [\$ / t]. . . . .	209
B.9	Raw material availability ( $x_{A,t}^{purch,max}$ ) [t]. . . . .	209
C.1	Cross-validated bandwidths used in the ( $PPJCC$ ) model. . . . .	218
C.2	Reduced risk levels ( $\alpha'$ ) obtained with the K-L formula (Table 4.1) and used in the ( $PPJCC$ ) model. . . . .	219
C.3	Estimated quantiles for nominal risk levels ( $\alpha$ ) used in the ( $PPICC$ ) model. . . . .	219
C.4	Production yield ( $\theta_f$ ) [-]. . . . .	219
C.5	Product demand ( $\gamma_{m,t}$ ) [t]. . . . .	219
C.6	Maximum inventory ( $w_{m,t}^{inv,max}$ ) [t]. . . . .	219
C.7	Maximum (minimum) capacity ( $w_f^{min}$ and $w_f^{max}$ ) [t]. . . . .	219
C.8	Selling price ( $SP_{m,t}$ ) [\$ / t]. . . . .	220
C.9	Operating cost ( $OPC_{f,t}$ ) [\$ / t]. . . . .	220
C.10	Purchase cost ( $PC_{m,t}$ ) [\$ / t]. . . . .	220
C.11	Inventory cost ( $IC_{m,t}$ ) [\$ / t]. . . . .	220
C.12	Raw material availability ( $x_{A,t}^{purch,max}$ ) [t]. . . . .	220
D.1	Deterministic spot market price ( $\alpha_{j,s,t}^{spot}$ ) [\$ / t]. . . . .	223
D.2	Operating cost ( $\delta_{i,s,t}$ ) [\$ / t]. . . . .	223
D.3	Inventory cost ( $\xi_{j,s,t}$ ) [\$ / t]. . . . .	224
D.4	Mass factor ( $\mu_{i,j,s}$ ) [-]. . . . .	224
D.5	Production capacity ( $Q_{i,s,t}$ ) [t]. . . . .	224
D.6	Lower (Upper) raw material availability ( $a_{j,s,t}^L$ and $a_{j,s,t}^U$ ) [t]. . . . .	224
D.7	Upper bound on inventory ( $V_{j,s,t}^U$ ) [t]. . . . .	224
D.8	Purchase price under discount ('d') contract type ( $\psi_{j,s,t}^{d,cs}$ for $j = A$ ) [\$ / t]. . . . .	224
D.9	Purchase threshold under discount ('d') contract type ( $\sigma_{j,s,t}^d$ for $j = A$ ) [t]. . . . .	225
D.10	Purchase price under bulk discount ('b') contract type ( $\psi_{j,s,t}^{b,cs}$ for $j = A$ ) [\$ / t]. . . . .	225
D.11	Purchase threshold under bulk discount ('b') contract type ( $\sigma_{j,s,t}^b$ for $j = A$ ) [t]. . . . .	225

D.12	Purchase price under fixed duration (' $l$ ') contract type ( $\psi_{j,s,t}^{l_p}$ for $j = A$ ) [\$/\$t]. . . . .	225
D.13	Purchase threshold under fixed duration (' $l$ ') contract type ( $\sigma_{j,s,t}^{l_p}$ for $j = A$ ) [t]. . . . .	225
D.14	Purchase spot price for the deterministic base case ( $\alpha_{j,s,t}^{\text{spot}}$ for $j = A$ ) [\$/\$t]. . . . .	225
D.15	Sales spot price for the deterministic base case ( $\psi_{j,t}^{\text{spot}}$ ) [\$/\$t].	226
D.16	Stochastic spot market price ( $\alpha_{j,s,t,k}^{\text{spot}}$ for $j = A$ ) [\$/\$t]. Simulated values from an ARIMA(0,1,1)(0,1,0) <sub>12</sub> time series model.	226
D.17	Stochastic part of demand-response models ( $\epsilon_k$ ) [\$/\$t]. . . .	227

# List of Figures

1.1	Schematic of the set of constraints of a decomposable optimization problem with complicating (a) constraints and (b) variables. . . . .	4
1.2	General case of regression analysis. Classical regression only provides mean output, whereas quantile regression provides any point of the output distribution. . . . .	10
1.3	Illustration of Kernel Density Estimation for an assumed data generating process consisting of a mixture of Gaussian models. . . . .	14
1.4	Illustration of Kernel Regression for conditional quantile estimation of a stochastic process. . . . .	17
1.5	Overview of the thesis. . . . .	20
2.1	Supply-chain schematic of the problem. . . . .	29
2.2	Detailed components of the plant. . . . .	29
2.3	Example of cyclic schedule and breaking of one of the links to determine optimal sequence. . . . .	32
2.4	Breaking the cyclic schedule to determine the first and last groups of the sequence. . . . .	33
2.5	Material and inventory balances schematic. . . . .	35
2.6	Changeovers (dashed cylinder) across time periods for modified USGP. . . . .	40
2.7	Bilevel Decomposition scheme for integrated planning and scheduling. . . . .	43
2.8	Schematic of Temporal Lagrangean Decomposition (TLD). . . . .	45
2.9	Hybrid Bilevel-Temporal Lagrangean Decomposition scheme for integrated planning and scheduling. . . . .	48
2.10	Algorithm for the hybrid Bilevel-Temporal Lagrangean Decomposition scheme. . . . .	49
2.11	Optimal schedule in Example 1. The blocks with a black-and-white downward diagonal pattern represent sequence-dependent changeovers. . . . .	51

2.12	Lower bound (LB), best upper bound (Best UB) and heuristic upper bound (Heuristic UB) on the objective function value using Temporal Lagrangean Decomposition in the Upper Level Planning problem in Example 3. The Heuristic UB is obtained by fixing the assignment variables, $yp_{iult}$ , calculated by each Lagrangean subproblem in the full-space planning model. . . . .	55
3.1	Two-stage scenario tree for one uncertain parameter. . . . .	60
3.2	Network structure for the motivating Example 1. . . . .	72
3.3	Distribution of the historical data for the production yield of facility $P1$ . . . . .	74
3.4	ECDF data of the production yield of facility $P1$ fitted by a simplified GLF. . . . .	74
3.5	Probability profiles for the heuristic and optimization-based (DMP) approaches in Example 1. For reference, a histogram of the uncertain data is depicted in Figure 3.3. . . . .	75
3.6	Optimal inventory levels of product $D$ from using the scenarios obtained from heuristic and DMP approaches. . . . .	77
3.7	Optimal purchase amounts of product $C$ from using the scenarios obtained from heuristic and DMP approaches. . . . .	77
3.8	Alternating forecasting and optimization steps in generating multi-stage scenario trees using the <i>NLP Approach</i> . CI denotes the confidence interval estimated at each forecast and ECDF means Empirical Cumulative Distribution Function. . . . .	81
3.9	Proposed forecasting step in generating multi-stage scenario trees using the <i>LP Approach</i> . $\sigma_e$ , CI, and ECDF denote the standard error, confidence interval estimated at each forecast, and the Empirical Cumulative Distribution Function, respectively. . . . .	82
3.10	Heuristic scenario tree for the demand of products $C$ and $D$ . The percentage deviations are computed based on each base node. The values above and below the arcs are arbitrarily chosen probabilities. . . . .	84
3.11	Time series data of the demand of products $C$ and $D$ . . . . .	84
3.12	Scenario tree obtained with the <i>NLP Approach</i> ( $L^2$ DMP) for Example 2. Top and bottom values inside each node are the calculated demands of products $C$ and $D$ , respectively. . . . .	87
3.13	Scenario tree obtained with the <i>LP Approach</i> ( $L^\infty$ LP DMP) for Example 2. The node values are obtained via forecasting and the probabilities are calculated via optimization. Top and bottom values in each node are the demands of products $C$ and $D$ , respectively. . . . .	88
3.14	Optimal inventory levels of product $D$ from using the scenarios obtained from heuristic and DMP approaches. . . . .	89

3.15	Optimal flow rates out of plant $P2$ from using the scenarios obtained from heuristic and DMP approaches. . . . .	89
4.1	Illustration of the density-based confidence set defined in equation (4.3). The goal is to obtain the estimated distribution $\hat{\mathbb{P}}$ whose “distance” to the unknown true distribution $\mathbb{P}$ is at most $d$ (divergence tolerance). . . . .	95
4.2	Illustration of the accuracy of the estimated quantile function via kernel density estimation or kernel smoothing (KS) for a sample size $n = 250$ . The four distributions are: normal with mean 0 and standard deviation 1, $N(0, 1)$ ; Gamma with both shape and rate parameters equal to 2, $\Gamma(2, 2)$ ; exponential with rate parameter equal to 1, $\text{Exp}(1)$ ; Pearson with mean 0, variance 1, skewness $-1$ , and kurtosis 4; Pearson( $0, 1, -1, 4$ ).100	
4.3	Illustration of the accuracy of the estimated joint CDF via kernel density estimation or kernel smoothing (KS) for a sample size $n = 400$ . The straight line corresponds to the true predictions of each bivariate distribution. . . . .	103
4.4	Illustration of point-wise confidence intervals constructed by the estimated standard errors from kernel smoothing. . . .	104
4.5	Illustration of point-wise binomial confidence intervals for an empirical CDF for small sample sizes. . . . .	106
4.6	Algorithm for the initialization of problems with JCCs. . . .	109
4.7	Network structure for the motivating Example 1. . . . .	110
4.8	<i>Top</i> : histograms for the randomly generated historical data for the maximum capacity of plants $P2$ and $P3$ (sample size $n = 365$ each) in the motivating example. <i>Bottom</i> : estimated empirical and kernel-based quantiles (EQ and KS, respectively). . . . .	112
4.9	Estimated joint cumulative distribution function for the maximum capacity of plants $P2$ and $P3$ (sample size $n = 365$ ) in the motivating example. The straight line corresponds to the true predictions of the bivariate normal distribution. . .	114
4.10	Comparison of the objective function value of the motivating example for different models (exact vs. kernel-based (KS) reformulations). . . . .	115
4.11	Comparison of the production levels of plants $P2$ and $P3$ of the motivating example for different models (exact vs. kernel-based (KS) reformulations). . . . .	116
4.12	Histograms depicting the real data for the maximum production capacities of three key plants in the network: (a) plant $P1$ , (b) plant $P2$ , and (c) plant $P3$ . . . . .	117
4.13	Pareto efficient frontier for the industrial test case. For the ICC model, the same reduced risk levels $\alpha'$ were used for the the three key plants. . . . .	118

4.14	Solution times of the JCC model with and without initialization from the solution of ICC models according to algorithm in Subsection 4.5. For a nominal confidence of 99%, the solver could not find a feasible solution without applying the initialization algorithm. . . . .	119
5.1	Typical stages in the S&OP process. Adapted from <a href="#">Ling &amp; Goddard (1988)</a> . . . . .	122
5.2	Overall strategy to account for historical production variability when generating production plans. . . . .	124
5.3	Example of modeling production variability with quantile regression, where $\Delta = \text{Plan} - \text{Actual}$ . Legend: (a) $\Delta$ vs. Plan plot, (b) estimated quantiles conditional on Plan = 10 w.u. (weight units), and (c) estimated quantiles conditional on Plan = 1.5 w.u.. . . . .	125
5.4	Schematic of the Sim-Opt framework. The DFO solver sets the production targets based on which the $\Delta$ values are estimated and form scenarios for the simulator. The simulator evaluates the current production target and returns a distribution of financial performance values (profit, cost etc.). . .	127
5.5	Process network structure of the motivating example. Plant reliability is related to the spread of the deviation between historical planned and actual production rates around the origin. Specifically in this example, the more reliable a plant is, the lower its margin is. . . . .	132
5.6	Pareto efficient frontier for Subcase 4.A in the motivating example. “m.u.” stands for monetary units. Error bars represent 95% confidence intervals on the average values. . . .	136
5.7	Pareto efficient frontier and average overall service level for Subcase 4.A in the motivating example. “m.u.” stands for monetary units. . . . .	137
5.8	Allocation scheme for two downstream plants in Subcase 4.A in the motivating example. Plant B is the less reliable and highest-margin, whereas plant G is the most reliable and lowest-margin of the downstream plants. The $\epsilon$ value denotes the variance of the profit (financial risk). “w.u.” and “m.u.” stand for weight and monetary units, respectively. . .	138
5.9	Pareto efficient frontier for Subcase 4.B in the motivating example. Error bars represent 95% confidence intervals on the average values. “m.u.” stands for monetary units. . . .	139
5.10	Effect of increasing average service level of product C on average overall service level in Subcase 4.B in the motivating example. . . . .	140



5.11	Illustration of the three proposed approaches to account for dependence of the $\Delta$ values for directly connected plants in the network. The relationships between the $\Delta$ values of the upstream and downstream plants are captured by: (a) linear regression (LR), (b) kernel regression (KR), (c) estimated joint distribution, and (d) bootstrap resampling. “w.u.” stands for weight units. . . . .	143
6.1	Supply chain structure considered in this work. . . . .	149
6.2	Schematic of contract types. <b>Discount:</b> pay higher price for quantity up to threshold value $\sigma$ , and pay discounted price for any amount beyond $\sigma$ . <b>Bulk Discount:</b> if purchased amount exceeds threshold value $\sigma$ , then pay discounted price; otherwise, pay higher price. <b>Fixed Duration:</b> pay discounted price according to contract term duration. . . . .	152
6.3	Time series model simulation for scenario generation. . . . .	156
6.4	Deterministic and stochastic components of a regression-based demand-response model (DRM). The region delimited by the dashed lines on the left plot represents possible realizations of price values according to the probabilistic nature of the residuals, $\epsilon$ . The distribution of the residuals can be sampled to generate scenarios. . . . .	156
6.5	Process network for Example 1. . . . .	159
6.6	Simulated scenarios (red dotted lines in the shaded region) for the spot market prices of raw material A. Mean forecast is represented by blue solid line in the shaded region. . . . .	160
6.7	Illustration of DRMs for products C (a) and D (b). . . . .	161
6.8	Selling price and sales for products C (a) and D (b) in Example 1. “Det” and “Stoch” stand for deterministic and stochastic model solutions, respectively. . . . .	162
6.9	Purchase amounts from contracts and spot market (average) for the deterministic and stochastic problems in Example 1. . . . .	163
6.10	Process network in Example 2 (Iyer & Grossmann, 1998). . . . .	165
6.11	Summary of results in Example 2 (linear DRMs). RP and EV stand for recourse problem (proposed method) and expected value problem (stochastic model with first-stage decisions fixed to the solution of the deterministic base case). . . . .	167
6.12	Selling price and sales for products 13 (a) and 23 (b) in Example 2 (linear DRMs). RP and EV stand for recourse problem and expected value problem solutions, respectively. . . . .	168
B.1	Using time series forecasting to generate a scenario tree for a stochastic process. CI denotes the confidence interval estimated at each forecast, respectively. . . . .	210

B.2	Schematic of the Multiple Replications Procedure (MRP) to assess the quality of two-stage stochastic programming solutions. . . . .	212
B.3	Schematic of the policy-generation Procedure $\mathcal{P}_2$ to assess the quality of multi-stage stochastic programming solutions. In this figure, $\Gamma_t^{(r)}$ denotes the set of subtrees conditionally sampled from stage $t$ at step $r$ . . . . .	214
C.1	Feasible region (grey area delimited by polytope in solid black lines) and optimal solution (red filled circle, $x^* = (3, 3)$ ) for the deterministic linear optimization model in equation (C.1). Dashed blue lines are the contours of the objective function. . . . .	215
C.2	Schematic of deterministic feasible region (grey area), optimal solutions (filled circles), and the new feasible regions (polytopes delimited by solid black lines) of the optimization model with ICCs for different values of individual risk levels: (a) $\alpha_1 = \alpha_2 = 0.1$ , (b) $\alpha_1 = \alpha_2 = 0.05$ , and (c) $\alpha_1 = \alpha_2 = 0.01$ . . . . .	216
C.3	Schematic of deterministic feasible region (grey area), optimal solutions (filled circles), and the new feasible regions (delimited by curved solid black lines) of the optimization model with JCC for different values of joint confidence levels: (a) $1 - \alpha = 0.9$ , (b) $1 - \alpha = 0.95$ , and (c) $1 - \alpha = 0.99$ . . . . .	217

# Chapter 1

## Introduction

This thesis primarily proposes data analytics methods for handling uncertainty in problems related to Enterprise-wide Optimization (EWO) ([Grossmann, 2005](#)) with applications based on chemical process networks. Data analytics methods are data-driven in the sense that they operate directly on available historical and/or forecast data to extract useful information and model variability or uncertainty. The constructed data models or uncertainty models are then incorporated into mathematical optimization models for EWO problems.

An additional contribution of this thesis, even though not directly related to data analytics methods, is decomposition approaches for the integration of deterministic planning and scheduling models of an integrated chemical site. In addition to developing significant enhancements of a previously proposed continuous-time scheduling formulation, we apply a hybrid decomposition scheme to solve a large-scale industrial problem and make use of parallel computing to achieve good solutions in practical computing time.

The objectives of this thesis are as follows:

1. Develop enhancements of a continuous-time scheduling formulation (Unit-Specific General Precedence or USGP) to account for the batching problem and sequence-dependent changeovers across time periods;
2. Effectively solve in parallel a hybrid decomposition scheme based on Bilevel Decomposition and Temporal Lagrangean Decomposition to solve large-scale deterministic integrated planning and scheduling problems.
3. Generate scenarios using a robust data-driven method via statistical-property matching of historical and forecast data in the context of stochastic programming;
4. Reformulate data-driven individual and joint chance constraints with right-hand side uncertainty using kernel smoothing;
5. Develop a data-driven method to incorporate production variability in the operations planning stage of the Sales & Operations Planning process

of a manufacturer;

6. Integrate optimal procurement contract selection and selling price optimization with general price-response models in a unified optimization model under uncertainty.

The remainder of this chapter contains a review of the basic concepts that are dealt with in the rest of the thesis. In particular, [Section 1.1](#) provides a brief review of the research efforts in EWO by focusing on planning and scheduling formulations of chemical processes as well as decomposition strategies to tackle large-scale instances. [Section 1.2](#) gives an overview of fundamental concepts and applications of Probability and Statistics that are relevant for optimization under uncertainty and data-driven methods. It is intended to be a first-read for non-mathematicians specialized in either areas. [Section 1.3](#) defines mathematical optimization under uncertainty in general terms. Two special cases of the general formulation (Stochastic and Chance-Constrained Programming) are briefly discussed. This introductory chapter ends with a brief overview of the chapters of the thesis in [Section 1.4](#).

## 1.1 Enterprise-wide Optimization: Planning, Scheduling, and Decomposition Strategies

The Process Systems Engineering (PSE) and Operations Research (OR) communities have contributed to the development of models and solution strategies to tackle the decision-making on three different time scales in Enterprise-Wide Optimization (EWO) problems. The long-term decisions (years) belong to the strategic level and are usually related to investments, plant capacity expansion and retrofit; the medium-term decisions (months, years) pertain to planning and commonly refer to production planning and inventory control; the short-term decisions (days, weeks) represent the scheduling of operations, i.e., determining the detailed timing and sequencing of operations. Therefore, the integration of different time scales within a mathematical optimization framework becomes valuable for the decision-making process in an enterprise. In order to make this integration efficient for solving real-world problems, special-purpose models and decomposition algorithms need to be investigated. Reviews by [Grossmann \(2005\)](#) and [Varma \*et al.\* \(2007\)](#) provide more details on the relevance of EWO activities in academic research and industrial practice.

Production planning and scheduling problems in the chemical, petrochemical, and pharmaceutical industries are reviewed in the works by [Sung & Maravelias \(2007\)](#) and [Verderame \*et al.\* \(2010\)](#). Planning formulations tend to be Linear Programming (LP), or sometimes Mixed-Integer Linear Programming (MILP), models. Constraints mainly include material and inventory balances that account for raw material purchases, production amounts, and sales. Decision variables typically include production rates, inventory levels, and raw material, intermediate, and finished product flows. The objective function is usually a financial performance indicator, such as profit or total costs, which

include revenue from sales, and operating, transportation, and inventory costs [Kallrath \(2002\)](#). For scheduling, the objective function may correspond to minimizing makespan or tardiness. Note that, in general, the planning model is not concerned with the sequencing of operations in each plant or reactor.

However, when applied to multi-product plants, the lack of a rigorous treatment of the production sequencing may underestimate the total costs and yield a plan that is not feasible. Without adding considerable complexity to the planning model, [Erdirik-Dogan & Grossmann \(2007\)](#) proposed a Detailed Planning (DP) model that utilizes Traveling Salesman Problem (TSP) constraints to generate a cyclic schedule, which is broken in one link to yield an optimal sequence with minimum changeover times. The formulation allows sequence-dependent changeovers across time periods. The planning model used in this work is based on the DP model. [Liu, Pinto, & Papageorgiou \(2008\)](#) have also investigated the estimation of the sequencing using TSP constraints.

At the scheduling level, a number of models that use either discrete- or continuous-time representation of events have been proposed. A review of both representations including their mathematical formulations can be found in [Floudas & Lin \(2004\)](#). We will focus on continuous-time representation models in this work. For sequential batch/continuous processes, two event representations have received a great deal of attention: time slots and precedence-based.

The main idea in the use of time slots is that each product can be assigned to a specific slot that has variable length. In some cases, especially in continuous processes, the number of time slots to be allocated for each reactor is known *a priori*. However, in batch processes that may not be true and additional time slots have to be allocated, thus increasing the size of the problem ([Erdirik-Dogan & Grossmann, 2008](#)). Likewise in the aforementioned DP formulation, the works by [Lima, Grossmann, & Jiao \(2011\)](#) and [Kopanos, Puigjaner, & Maravelias \(2011\)](#) have the desirable feature of sequence-dependent changeovers across adjacent time periods, which even though adds more complexity to the model due to the larger number of binary and continuous variables, renders more realistic plans by not imposing a “hard” barrier for changeovers across time periods.

Unlike time slot models, precedence-based scheduling models effectively model changeovers by means of Big-M constraints. The two types of precedence relationships are local (immediate) and global (general) that are used to track the relative position of products in the production sequence. Four types of precedence-based models have been proposed: Unit-Specific Immediate Precedence (USIP) ([Cerdá, Henning, & Grossmann, 1997](#)), Immediate Precedence (IP) ([Méndez, Henning, & Cerdá, 2000](#)), General Precedence (GP) ([Méndez, Henning, & Cerdá, 2001](#)), and Unit-Specific General Precedence (USGP) ([Kopanos, Laínez, & Puigjaner, 2009](#)). Briefly, IP and USIP differ in that the latter only takes into account the immediate precedence of products that are assigned to the same processing unit, whereas the former does

not. The GP model generalizes the concept of precedence by accounting for all precedence relations (immediate or not) and requires fewer binary variables than the immediate precedence models, but cannot account for changeovers costs. To overcome that limitation, the USGP model was proposed.

When attempting to solve large-scale industrial problems, some authors have identified the need to decompose the problem into subproblems (Conejo *et al.*, 2006). Decomposition approaches take advantage of specific structures of the optimization problems. Two such cases arise in practice: the complicating constraint and the complicating variable structures, as shown in Figure 1.1. Note that complicating constraints involve variables from different blocks, and complicating variables link constraints pertaining to different blocks.

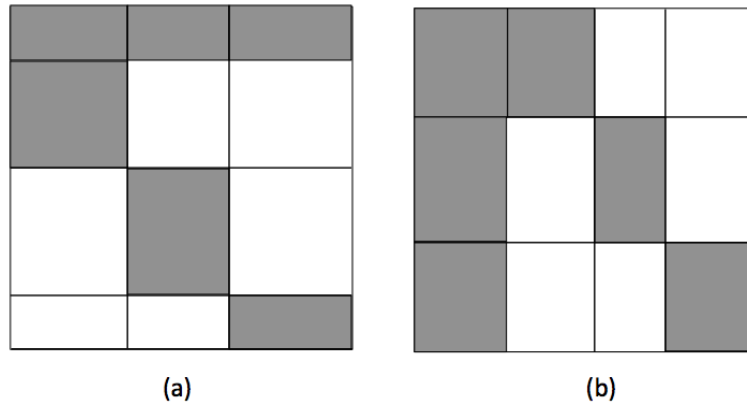


Figure 1.1: Schematic of the set of constraints of a decomposable optimization problem with complicating (a) constraints and (b) variables.

Different decomposition approaches have been proposed depending on the nature of the optimization problem (e.g., linear vs. nonlinear, purely continuous vs. mixed-integer). For LP and MILP problems with complicating constraints, the Dantzig-Wolfe and Branch-and-Price have been applied. When complicating variables are present, Benders decomposition is typically used although it is restricted to having only integer variables in the first-stage decisions for stochastic programming. It should be noted that problems with complicating variables can always be converted to problems with complicating constraints, which are then solved by Lagrangean Decomposition (a particular case of Lagrangean Relaxation (Guignard, 2003)). The main idea of such approach is to create *copies* of a subset of the variables (called copy variables) and add equality constraints that equate the original and the copy variables. This yields a problem with complicating constraints, which are dualized to allow decomposition. Another decomposition approach is the Bilevel Decomposition proposed by Iyer & Grossmann (1998). It involves solving an aggregate formulation in the upper level and a detailed formulation in the lower level with some decisions fixed by the upper level. A review of methods and decomposition approaches for the integration of planning and scheduling can be found in the work by Maravelias & Sung (2009).

## 1.2 Probability and Statistics: An Overview

In this section, we provide a basic overview of concepts in Probability and Statistics that are inherent in modeling optimization problems under uncertainty. These concepts are even more extensively used when applying data analytics or data-driven methods to construct uncertainty models, which in turn are incorporated in EWO models (e.g., planning and scheduling). We do not attempt to provide a thorough account of probability and measure theories. Several standard references are available, including textbooks (e.g., Papoulis (1991); Bauer (2001); Forbes *et al.* (2011); Hogg, McKean, & Craig (2012)) and lecture notes (e.g., <http://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/>).

### 1.2.1 Basics

#### 1.2.1.1 Random Variables

We start this overview by defining random variables. A random variable (r.v.) can take on values that are subject to variations. More formally, a r.v.  $X : \Omega \rightarrow E$  is a measurable function from the set of possible outcomes  $\Omega$  to some set  $E$ . If the image of  $X$  is countably infinite (i.e., it has the same cardinality or number of elements as a subset of the set of natural or integer numbers), then it is called a *discrete random variable*. Otherwise, if  $E$  is uncountably infinite, then  $X$  is a *continuous random variable* (also, usually,  $\Omega = \mathbb{R}$ , which we assume throughout this overview).

**Remark 1.** It is important to note that a r.v. is in fact a real-valued point function. Without delving into details of probability spaces (see references in the beginning of this section), a r.v.  $X(\omega)$  assigns a real scalar value to each point (outcome)  $\omega \in \Omega$ , denoted as  $X(\omega) = x$ . For all problems dealt with in this thesis (and typically in EWO applications),  $\Omega = \mathbb{R}^n$  with  $\omega$  being any point (vector) in  $\mathbb{R}^n$ ; therefore, a convenient mapping is  $X(\omega) = \omega$ . An implication of this choice of mapping is that one can use the terms *outcomes* (domain) and *expressions* (image) of a r.v. interchangeably.

#### 1.2.1.2 Mass, Density, and Distribution Functions

The distribution of a discrete r.v. can be described by a probability mass function (PMF), whereas the distribution of an *absolutely* continuous r.v. can be described by a probability density function (PDF). All r.vs. can be described by their cumulative distribution function (CDF), which expresses the probability that the r.v. will be less than or equal to a certain value. We note that the terms “density” and “distribution” are typically used in the literature to denote PDF and CDF, respectively. The commonly used notation for PMF and PDF is  $f_X(x)$ , and for CDF is  $F_X(x)$ . From its definition,  $F_X(x) = \mathbb{P}(X \leq x)$ , where  $\mathbb{P}(\cdot)$  is a probability measure and reads “probability that”. A summary of mathematical definitions is given in [Table 1.1](#).

Table 1.1: Definitions of probability mass (PMF), density (PDF), and cumulative distribution (CDF) functions of random variables.

Discrete	Continuous
PMF	PDF
$f_X(x) = \mathbb{P}(X = x) = p_X(x)$	$\mathbb{P}(a \leq X \leq b) = \int_a^b f_X(x)dx$ also, $f_X(x) = \frac{dF_X(x)}{dx}$
CDF	CDF
$F_X(x) = \sum_{x_i \leq x} p_X(x_i)$	$F_X(x) = \int_{-\infty}^x f_X(u)du$

Examples of discrete distributions include Poisson, Bernoulli, and multinomial. Examples of continuous distributions include normal (Gaussian), uniform, and Pearson. These are also called parametric population probability distributions. When dealing with data samples, the sample or *empirical* version of the distribution (ECDF) is given by

$$ECDF(t) = \frac{1}{n} \sum_{i=1}^N \mathbf{1}_{\{x_i \leq t\}}(t) \quad (1.1)$$

where  $n$  is the sample size and  $\mathbf{1}_{\{A\}}(\cdot)$  is the indicator function of event  $A$ , that takes the value of one if event  $A$  is true, or zero otherwise. Therefore, given a value  $t$ , the ECDF returns the ratio between the number of elements in the sample that are less than or equal to  $t$  and the sample size.

### 1.2.1.3 Moments

The distribution of a r.v. can be characterized by scalar quantities that have practical interpretations. One example of such quantitative measures are moments. The first moment is called the mean, average, or *expected value* of a r.v. and is usually denoted by  $\mu_X = \mathbb{E}[X]$ , where  $\mathbb{E}[\cdot]$  is the expectation operator. For a discrete r.v.,  $\mu_X = \sum_{x \in \Omega} x p_X(x)$ , and for a continuous r.v.,  $\mu_X = \int_{-\infty}^{\infty} x f_X(x) dx$ . We distinguish *central* from *raw* moments. Central moments are defined relative to the first moment. The  $r$ -th central moment,  $\mu_r = \mathbb{E}[(X - \mu_X)^r]$ , is defined by

$$\text{Discrete: } \mu_r = \sum_{x \in \Omega} (x - \mu_X)^r p_X(x) \quad (1.2)$$

$$\text{Continuous: } \mu_r = \int_{-\infty}^{\infty} (x - \mu_X)^r f_X(x) dx \quad (1.3)$$



whereas the  $r$ -th raw moment,  $\mu'_r = \mathbb{E}[X^r]$ , is given by

$$\text{Discrete: } \mu'_r = \sum_{x \in \Omega} x^r p_X(x) \quad (1.4)$$

$$\text{Continuous: } \mu'_r = \int_{-\infty}^{\infty} x^r f_X(x) dx \quad (1.5)$$

Some central moments receive special names:  $\mu_2$  is the variance (also commonly denoted by  $\sigma^2$ ), which is a measure of the spread of a distribution,  $\mu_3$  is the skewness, which is a measure of the asymmetry of a distribution, and  $\mu_4$  is the kurtosis, which is a measure of the “peakedness” or the “weight” of the tails of a distribution.

Sample versions of moments and their estimates can also be calculated (Klemens, 2009). For instance, for  $N$  data points, the unbiased sample estimate of the mean and variance are given by:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (1.6)$$

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (1.7)$$

**Remark 2.** The concept of expectation includes that of probability as a special case. Let  $\mathbf{1}_A$  be the indicator r.v. (or function) of event (or set of outcomes, subset of sample space)  $A$ . Therefore, in a discrete probability space, we have that  $\mathbb{E}[\mathbf{1}_A] = 1 \times \mathbb{P}(A) + 0 \times \mathbb{P}(A^c) = \mathbb{P}(A)$ , where  $A^c$  is the complement of  $A$ . The same result can be obtained in a continuous probability space.

#### 1.2.1.4 Quantiles

The interpretation and mathematical definition of quantiles varies according to the nature of the r.v. and when dealing with data samples (Parzen, 2004). For continuous random variables (r.v.s.), quantiles are values taken at regular intervals from the inverse of the distribution. The *quantile function* for a continuous r.v.  $X$ ,  $F_X^{-1}(\cdot)$ , returns a threshold value  $x$  below which random draws from the given distribution would fall  $p$  percent of the time. It is defined as follows,

$$F_X^{-1}(p) = \inf_{x \in \mathbb{R}} \{p \leq F_X(x)\} \quad (1.8)$$

where  $p \in [0, 1]$  is a probability value. Different expressions are given for the discrete and sample versions.

The case of data samples is particularly relevant when employing data-driven methods. Given a data sample, a quantile is the value that divides a data set into two subsets. The 50<sup>th</sup> 100-quantile (also called 50<sup>th</sup> percentile or median) separates the higher half of a data set from the lower half. In other words, there is at most 50% probability that a random variable will be less than the median. The 4-quantiles are called quartiles, the 5-quantiles are called quintiles and so on.

### 1.2.1.5 Multivariate Distributions and Independence

Similarly for a vector of real numbers and multivariate real functions, a vector of r.vs. (i.e., random vector) with  $n$  elements can be used to construct a multivariate distribution, which can be denoted by  $F_{X_1, X_2, \dots, X_n}(x_1, x_2, \dots, x_n)$ . The concept of independence of r.vs. is connected to the one for events in elementary probability theory: two events  $A_1$  and  $A_2$  are *independent* if and only if their joint probability (i.e., the probability of their intersection) equals the product of their individual (or *marginal*) probabilities. Mathematically,  $\mathbb{P}(A_1 \cap A_2) = \mathbb{P}(A_1)\mathbb{P}(A_2)$ . Now consider two independent r.vs.,  $X_1$  and  $X_2$ , and the two events  $\{X_1 \leq x_1\}$  and  $\{X_2 \leq x_2\}$ ; thus, we have that  $F_{X_1, X_2}(x_1, x_2) = F_{X_1}(x_1)F_{X_2}(x_2)$ . The same relationship applies to densities and can be extended to  $n$  dimensions.

The notion of moments for single r.vs. can be expanded to more variables, and they are called co-moments. The case of two r.vs. is particularly important, specially for the second co-moment, also called covariance,  $\sigma_{XY} = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)]$ , which is defined as follows:

$$\text{Discrete: } \sigma_{XY} = \sum_{x \in \Omega_X} \sum_{y \in \Omega_Y} (x - \mu_X)(y - \mu_Y) p_{X,Y}(x, y) \quad (1.9)$$

$$\text{Continuous: } \sigma_{XY} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - \mu_X)(y - \mu_Y) f_{X,Y}(x, y) dx dy \quad (1.10)$$

The sample version of the covariance is given by:

$$\hat{\sigma}_{XY}^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (1.11)$$

where  $N$  is the sample size, and  $\bar{x} = N^{-1} \sum_{i=1}^N x_i$  and  $\bar{y} = N^{-1} \sum_{i=1}^N y_i$  are sample means.

### 1.2.1.6 Marginal and Conditional Distributions and Moments

As previously mentioned, marginal distributions can be obtained from the joint distribution. For instance, given a bivariate density  $f_{X_1, X_2}(x_1, x_2)$ , the marginal density with respect to  $X_1$  is calculated by

$$f_{X_1}(x_1) = \int_{-\infty}^{\infty} f_{X_1, X_2}(x_1, x_2) dx_2 \quad (1.12)$$

From the marginal density, one can obtain its associated marginal moments using the formulas aforementioned. Conditional distributions are defined when certain r.vs. of the random vector take on specific values. For example, consider the joint distribution  $F_{X_1, X_2}(x_1, x_2)$  where both r.vs. are continuous. The *conditional* distribution of  $X_1$  *given*  $X_2 = x_2$  is denoted by  $F_{X_1|X_2}(x_1|x_2)$ . The following relationship follows:

$$F_{X_1|X_2}(x_1|x_2) = \frac{F_{X_1, X_2}(x_1, x_2)}{F_{X_2}(x_2)} \quad (1.13)$$

Similarly to marginal moments, conditional moments can be obtained from conditional distributions. One frequently used conditional moment is the conditional expected value, or regression function, which will be discussed in the next subsection.

**Remark 3.** As an extension of the property of the expectation discussed in **Remark 2**, we have that the conditional probability of an event  $A$  given r.v.  $X$  is a special case of the conditional expected value. Mathematically,  $\mathbb{E}[\mathbf{1}_A|X] = \mathbb{P}(A|X)$ .

## 1.2.2 Applications

In this subsection, we provide a brief discussion on some applications of Probability and Statistics that form the basis of the data-driven methods proposed in this thesis. In particular, the scenario generation framework proposed in [Chapter 3](#) is aided by time series analysis, which is based on regression analysis. In [Chapter 4](#), data-driven chance constraints are reformulated to algebraic constraints via a nonparametric technique called kernel smoothing. Production variability in [Chapter 5](#) is modeled using quantile regression. Lastly, in [Chapter 6](#), the relationship between selling price and demand can be modeled via price-response models, which are usually classical regression models.

### 1.2.2.1 Classical and Quantile Regression

Let  $X$  and  $Y$  denote the predictor (or covariate or input) and the response (or output) r.v.s. whose values are denoted by  $x$  and  $y$ , respectively. We restrict the discussion to continuous random variables only.

In regression analysis, the regression model is generally written as,

$$Y = f(X) + \epsilon \quad (1.14)$$

where  $f(\cdot)$  is a mathematical formula that expresses the relationship between  $X$  and  $Y$ , and  $\epsilon$  is the random error term assumed to have mean zero, homoskedastic (i.e., its variance is constant over the range of  $X$  values), and uncorrelated with  $X$  ([Montgomery & Runger, 2003](#)). In linear (classical) regression, for example,  $f(X) = \beta_0 + \beta_1 X$ , where  $\beta_0$  and  $\beta_1$  are parameters to be estimated. Nonlinear parametric and nonparametric functions can also be used to model the relationship between  $X$  and  $Y$  (see overview on “Kernel Smoothing” below).

We want to predict  $Y$  values for given  $X$  values. In classical regression, we write,

$$\hat{Y} = \mathbb{E}[Y|X = x] \quad (1.15)$$

where  $\hat{Y}$  denotes the predicted response variable and  $\mathbb{E}[\cdot]$  is the expectation operator. In other words,  $\hat{Y}$  is the *mean* of  $Y$  values conditional on  $X = x$ . Therefore, for a distribution of  $X$ , the result of a classical regression analysis is a single point (the mean) of the distribution of  $Y$ .

A more general approach to regression analysis is quantile regression ([Koenker, 2005](#)). A quantile is the value that divides a data set in two subsets. The 50<sup>th</sup>

100-quantile (also called 50<sup>th</sup> percentile or median) separates the higher half of a data set from the lower half. In other words, there is at most 50% probability that a random variable will be less than the median. The 4-quantiles are called quartiles, the 5-quantiles are called quintiles and so on. In this thesis, we will use quantile and 100-quantile interchangeably.

In quantile regression, the predicted  $Y$  values correspond to quantiles of the distribution of  $Y$  conditional on  $X = x$ . Mathematically,

$$\hat{Y} = Q_\alpha[Y|X = x] \quad (1.16)$$

where  $Q_\alpha[\cdot]$  denotes the 100 $\alpha$ -th quantile and  $\alpha \in [0, 1]$  is the probability level. Similarly to classical regression, quantile regression can be performed parametrically (linear and nonlinear models such as smoothing splines) as well as nonparametrically (e.g., kernel smoothing (Li & Racine, 2007)).

Figure 1.2 illustrates regression analysis in the general case. In the general case of regression analysis, the objective is to model the relationship between the distribution of  $X$  and the distribution of  $Y$ , i.e., to predict  $F_Y(y)$  from  $F_X(x)$ , where  $F_Y(\cdot)$  and  $F_X(\cdot)$  are the cumulative distribution functions of  $Y$  and  $X$ , respectively. Classical regression provides the mean whereas quantile regression provides *any* quantile of the distribution of  $Y$  conditional on  $X = x$ .

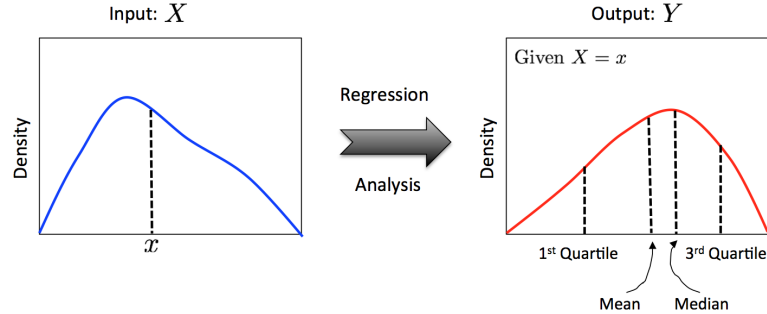


Figure 1.2: General case of regression analysis. Classical regression only provides mean output, whereas quantile regression provides any point of the output distribution.

### 1.2.2.2 Time Series Analysis

Building good time series models is a very complex task. Much effort goes into identifying exogenous or external drivers (what economic or other factors affect the output variables?) as well as seasonal, trend, and cyclical factors in the data in order to obtain good forecasts. In addition, the “big data” problem is also a reality for time-stamped data recorded at a particular frequency due to the availability of numerous databases, such as Global Insight, Chemical Markets Associates Inc. (CMAI), government sources, and more. Therefore, data mining techniques applied to time series data have demonstrated significant value regarding inventory cost reduction as well as a revenue optimization (Rey, Kordon, & Wells, 2012). The literature on time series analysis is vast;

there are standard reference books and dedicated journals. Excellent and influential books are: [Box, Jenkins, & Reinsel \(2008\)](#); [Brockwell & Davis \(2002\)](#); [Fan & Yao \(2002a\)](#); [Tong \(1993\)](#). We provide a very brief introduction to time series analysis, which comprises theoretical and applied concepts in modeling and forecasting.

We begin with some definitions. A *stochastic process* represents the evolution of a system over time and constitutes of a collection of random variables. A particular case of stochastic process when *discrete* time is considered is *time series*. A time series is a set of observations, each one being recorded at a specific time  $t$ . Differently from static data types, time series data can be related to themselves over time, i.e., they are *serially correlated* in time. In other words, it means that there is a dependency between a observations at time periods  $t$  and  $t + h$ , where  $h$  is a time lag. However, this dependency typically decays as  $h \rightarrow \infty$ .

A fundamental concept in time series analysis is *stationarity*. More detailed definitions of the types of stationarity are given in the aforementioned references, but for the purposes of this brief overview a time series is stationary if its statistical properties do not change over time. This is an important characteristic of modeling time series since in order to make predictions *something* should not vary with time.

Time series models can be linear or nonlinear, parametric or nonparametric, and the process of fitting a model to historical data is a parameter estimation problem. Some advantages of using linear models are: (1) linear difference equations are simple and a complete theory is available; (2) it is generally computationally tractable to obtain parsimonious linear models; (3) much experience has been accumulated in their application. However, some of the shortcomings of linear models are: (1) they are not ideally suited for data exhibiting strong asymmetry (these models are often stationary Gaussian, thus having symmetric joint distributions); (2) alternative preferable models exist for data exhibiting sudden bursts of large amplitude at irregular times; (3) they may fail to represent *time irreversible* stochastic processes.

Two popular models are presented as follows. An extensive treatment of the many models that have been proposed is beyond the scope of this overview, but can be found in the references aforementioned.

**Linear Models.** A general linear model is the Autoregressive Integrated Moving Average, ARIMA( $p, d, q$ ), model. The model parameters  $p$ ,  $d$ , and  $q$  correspond to the autoregressive order or how far back in time values of the output variable influence its current value, the number of differences necessary to render the time series stationary (de-trend the series), and the number of lagged errors, respectively. The mathematical expression for an ARIMA model is as follows:

$$\left(1 - \sum_{i=1}^p \phi_i B^i\right) (1 - B)^d Y_t = \left(1 - \sum_{i=1}^q \theta_i B^i\right) \epsilon_t \quad (1.17)$$

where  $Y_t$  is the output variable representing the stochastic process,  $B$  is the backshift or lag operator ( $B^h Y_t = Y_{t-h}$ ),  $\epsilon_t$  are the error terms and also called residuals or innovations, and  $\phi_i$  and  $\theta_i$  are parameters to be estimated. When the innovations are assumed to follow a Normal distribution, this is generally called a linear Gaussian model.

If external driver variables, also known as covariates, are considered, then we have an ARIMAX model. For example, if product demand is the output variable, then a possible external driver variable may be an economic indicator such as the Gross Domestic Product (GDP). Thus, it may be relevant to include time series data of GDP in order to improve the accuracy of demand forecasts for the product of interest. When multiple output variables are considered simultaneously, then the model name is prefixed by the word “Vector” and we have VARMA and VARMAX models (assuming that the series has been previously differenced so the ‘I’ is dropped).

A key result of time series forecasting is that the predictor or optimal point forecast is the mean conditional on past observations. Mathematically, the estimated forecast value  $h$  time steps ahead the current time  $t$  is given by:  $\hat{Y}_{t+h} = \mathbb{E}[Y_{t+h} | \mathbf{Y}_t]$ . In other words, the conditional mean of an ARIMA model is readily available via forecasting. The variance is assumed to be constant over time.

**Nonlinear Models.** Many nonlinear time series models have been proposed in the literature. Some are used in conjunction with linear ARIMA models, such as the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model that considers a time-varying (heteroskedastic) conditional variance. There are several variations of the GARCH model; for instance, the Exponential GARCH (EGARCH) model was used by [Deniz & Luxhøj \(2011\)](#) to generate a multi-stage scenario tree for a portfolio management problem.

We focus on a nonparametric model called Nonlinear Additive Autoregressive (NAAR or sometimes simply AAR). A NAAR model of order  $p$ , or NAAR( $p$ ), is given by:

$$Y_t = \sum_{i=1}^p f_i(Y_{t-i}) + \epsilon_t \quad (1.18)$$

where  $f_i(\cdot)$  are univariate functions and can be estimated. Typically, cubic splines are used, thus adding considerable flexibility to the model. Note that an AR( $p$ ) model is a special case of a NAAR( $p$ ) model.

Likewise for linear models, predicting a value  $h$  steps ahead, or  $\hat{Y}_{t+h}$ , is a regression problem and the outcome is the conditional expectation. Heteroskedastic conditional variance can also be modeled with, for example, a Generalized Additive Model (GAM) fitted to the residuals of the NAAR( $p$ ) model. Moreover, exogenous variables may be included in the model yielding a NAAR model with exogenous variables (NAARX) ([Chen & Tsay, 1993](#)).

In addition to the models mentioned in this section, several other nonlinear, parametric and nonparametric, models have been proposed and adapted to time series modeling, to name a few: Artificial Neural Networks (ANN),

Threshold Autoregressive (TAR), and Kernel Density Estimation (KDE).

**Simulation of Time Series.** Once a time series model, parametric or non-parametric, has been chosen, it can be simulated to generate different paths with specified number of observations. From the simulated paths, one can estimate statistics of the distribution of the generated data. Examples of statistics include quantiles and (empirical) cumulative distribution function.

Parametric models, such as the linear ARIMA models discussed in [paragraph 1.2.2.2](#), can be simulated by randomly generating residuals or innovations,  $\epsilon_t$ , according to some probability distribution model (usually Normal), and then computing the values of the dependent variable using the regressed model.

A simulation approach that does not rely on assuming the distribution of the innovations is based on the bootstrap method originally proposed by [Efron \(1979\)](#). Bootstrapping falls in the class of resampling methods and consists of estimating properties of an estimator by measuring those properties when sampling from an approximating distribution, such as the empirical distribution. Additional care must be taken when using bootstrap methods with time series data – they contain dependent observations in time – in order to preserve the time-series effects.

Two basic approaches for resampling in the time domain are ([Davison & Hinkley, 1997](#)): model-based and block resampling. The idea in model-based resampling is to fit a suitable model to the data, to construct residuals from the fitted model, and then to generate new series by incorporating random samples from the residuals into the fitted model. Block resampling, on the other hand, operates on blocks of consecutive observations instead of innovations. The data set is divided into blocks with specified length, which can be fixed or geometrically-distributed with given mean, and they comprise the bootstrap samples. The idea underlying this approach is that if the blocks are long enough, then enough of the original dependence will be preserved.

**Software.** Statistical procedures, many of the popular models discussed in the literature as well as forecasting capabilities, are available in programming languages and software packages. A few examples are: R programming language ([R Core Team, 2015](#)), MATLAB® ([The MathWorks Inc., 2015](#)), SAS® software ([SAS Institute, 2015](#)), Stata ([StataCorp LP, 2015](#)), and EViews ([IHS Inc., 2013](#)).

### 1.2.2.3 Kernel Smoothing

Kernel smoothing falls into the category of nonparametric statistical methods, i.e., they do not require the user to specify functional forms for random variables being estimated. Examples of kernel smoothing techniques include Kernel Density/Distribution Estimation (KDE) and Kernel Regression (KR). The literature on kernel smoothing is extensive, such as [Silverman \(1986\)](#); [Scott \(1992\)](#); [Pagan & Ullah \(1999\)](#); [Koenker \(2005\)](#); [Li & Racine \(2007\)](#) to name a few. Most of the material presented in this overview is taken from

[Racine \(2008\)](#), which is an excellent short introductory book on kernel methods.

The notation used in this section is as follows. Let  $X$  be a continuous random variable (r.v.) whose unknown true probability density function (PDF or simply density) and cumulative distribution function (CDF or simply distribution) are denoted by  $f(x)$  and  $F(x)$ , respectively. Here,  $x$  is a possible realization of the r.v.  $X$ . Given a series of  $n$  independent and identically distributed (i.i.d.) data sample (this assumption is weakened when dealing with autocorrelated data, see [paragraph 1.2.2.3](#)), the main goal of KDE is to obtain an estimate  $\hat{f}(x)$  of the unknown true density. It does so without resorting to simple parametric reductions of the problem, and this makes it a nonparametric method.

**Univariate KDE.** As an illustration, suppose  $X$  is governed by a data generating process (DGP) characterized by a mixture of Gaussian models ( $N(30, 4/9)$  and  $N(32, 4/9)$  with weights 0.6 and 0.4, respectively). Note that, in reality, we may not know the true DGP. [Figure 1.3](#) shows the histogram and the density estimated with KDE for a sample size of  $n = 500$ . Note that the estimated density captures the bimodal behavior of the data as well as tail effects (extreme realizations). The main components of a kernel density estimator—weights and bandwidth—are also illustrated. Briefly, the kernel (weighting) function assigns more weight to data points that are close to a given  $x$ , i.e., within a window or bandwidth.

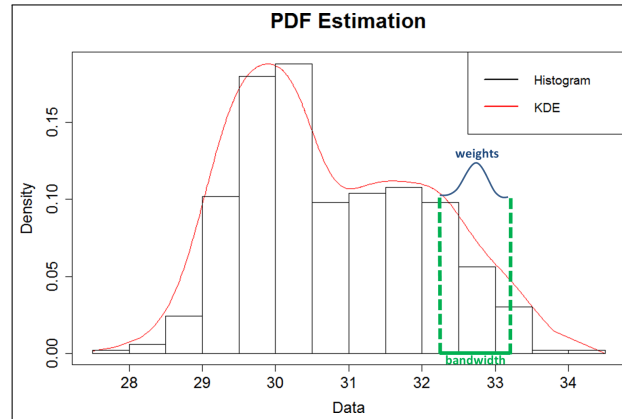


Figure 1.3: Illustration of Kernel Density Estimation for an assumed data generating process consisting of a mixture of Gaussian models.

In mathematical terms, the kernel density estimator is given by

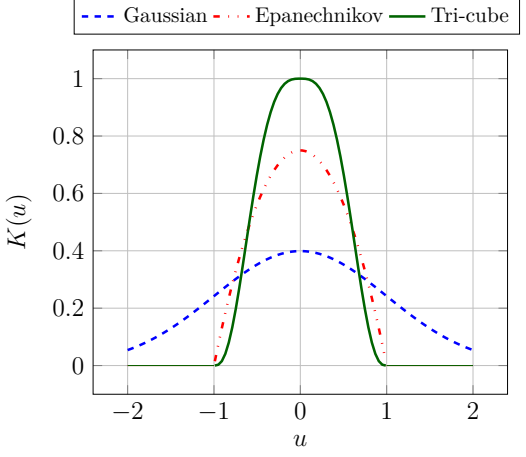
$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right) \quad (1.19)$$

where  $X_i$  is an observation (data point) of  $X$ ,  $h$  is the bandwidth (scaling factor that controls the smoothness or roughness of a density estimate), and  $K(\cdot)$  is



the kernel or weighting function. Several kernel functions have been proposed and used. Table 1.2 shows a few examples of kernel functions of which the Gaussian kernel is the only one that does not have a compact support.

Table 1.2: Some kernel functions commonly used in kernel density estimation.

Name	Formula, <sup>†</sup> $K(u)$	Illustration
Gaussian	$\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}$	
Epanechnikov	$\frac{3}{4}(1 - u^2)\mathbf{1}_{\{ u \leq 1\}}(u)$	
Tri-cube	$(1 -  u ^3)^3\mathbf{1}_{\{ u \leq 1\}}(u)$	

<sup>†</sup>  $\mathbf{1}_A(\cdot)$  is the indicator function, which equals to 1 if  $A$  is true, and 0 otherwise.

It can be shown that the choice of the bandwidth, rather than the kernel, is crucial to obtain accurate results with KDE. Four methods have been proposed to select the optimal bandwidth: rule-of-thumb, plug-in, least squares cross-validation, and likelihood cross-validation. The cross-validation-based methods are fully automatic or “data-driven” in the sense that they are tailored to the sample data under consideration. Cross-validation is a powerful model validation statistical technique for assessing how the results of a statistical analysis will generalize to an independent data set. When cross-validation is applied to the selection of the optimal bandwidth in the least squares method, for example, the objective function to be minimized can be written as follows (see Li & Racine (2007) for details):

$$CV(h) = \frac{1}{n^2 h} \sum_{i=1}^n \sum_{j=1}^n \bar{K}\left(\frac{X_i - X_j}{h}\right) - \frac{2}{n(n-1)h} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n K\left(\frac{X_i - X_j}{h}\right) \quad (1.20)$$

where  $\bar{K}(v) = \int K(u)K(v-u)du$  is the two-fold convolution kernel derived from  $K(\cdot)$ .

Distribution functions,  $F(x)$ , can also be estimated using kernel functions and selecting optimal bandwidths. The expression of the kernel estimator for a distribution function is given by:

$$\hat{F}(x) = \frac{1}{n} \sum_{i=1}^n \mathcal{K}\left(\frac{x - X_i}{h}\right) \quad (1.21)$$

where  $\mathcal{K}(u) = \int K(u)du$  is the integrated kernel function. For instance, the integrated kernel function of the Gaussian kernel  $\phi(\cdot)$  is the standard normal or Gaussian distribution function  $\Phi(\cdot)$ :

$$\mathcal{K}_{\text{Gaussian}}(u) = \Phi(u) = \int_{-\infty}^u \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt = \int_{-\infty}^u \phi(t) dt = \frac{1}{2} \left[ 1 + \operatorname{erf} \left( \frac{u}{\sqrt{2}} \right) \right] \quad (1.22)$$

where  $\operatorname{erf}(u) = \frac{1}{\sqrt{\pi}} \int_{-u}^u e^{-t^2/2} dt$  is the error function.

**Multivariate KDE.** When considering multiple r.vs., a standard, but not the only approach, is to use *product kernels*. A product kernel consists of the product of univariate kernels, each one for each random variable under consideration. Each univariate kernel has its associated bandwidth. Suppose we are interested in estimating the joint PDF and joint CDF of  $m$  r.vs. The kernel joint density and distribution can be written as follows:

$$\hat{f}(x_1, \dots, x_m) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^m \frac{1}{h_j} K_j \left( \frac{x_j - X_{j,i}}{h_j} \right) \quad (1.23)$$

$$\hat{F}(x_1, \dots, x_m) = \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^m \mathcal{K}_j \left( \frac{x_j - X_{j,i}}{h_j} \right) \quad (1.24)$$

Cross-validation techniques can also be used to estimate each  $h_j$ ,  $j = 1, \dots, m$ , for a given choice of univariate kernels  $K_j(\cdot)$  and  $\mathcal{K}_j(\cdot)$ .

Conditional densities and distributions can also be estimated via KDE. For example, suppose  $X_1$  and  $X_2$  are two continuous r.vs. for which we want to estimate the conditional density of  $X_2$  given  $X_1$ , i.e.,  $\hat{f}_{X_2|X_1}(x_2|x_1)$ . By definition, we have,

$$\hat{f}_{X_2|X_1}(x_2|x_1) = \frac{\hat{f}_{X_1, X_2}(x_1, x_2)}{\hat{f}_{X_1}(x_1)} \quad (1.25)$$

where  $\hat{f}_{X_1, X_2}(x_1, x_2)$  is the estimated joint PDF of  $X_1$  and  $X_2$  and  $\hat{f}_{X_1}(x_1)$  is the estimated *marginal* density of  $X_1$ . Once again, cross-validation methods can be used to estimate conditional densities and distributions, and have the property of detecting which components of the random variables are irrelevant for the conditional dependency relationship among them. By extension, conditional quantiles can also be estimated, which is useful when applying Kernel Regression (KR) to stochastic processes as discussed next.

**Conditional Quantiles of Autocorrelated Data.** When the uncertain parameter is a stochastic process, i.e., the collection of r.vs. that represent the evolution of a system over time, then the autocorrelation between observations in different time points may not be ignored. In other words, we cannot treat the observations as i.i.d. data. Typical examples of stochastic processes are product demand and selling price, which can be modeled and forecast using (non)parametric time series models. It is beyond the scope of this article to

review time series analysis, and standard references are available (Brockwell & Davis, 2002; Fan & Yao, 2002a).

To be more specific, we consider the case of *weakly* dependent data (see Chapter 18 in Li & Racine (2007)). That is, we assume that the dependence of an observation at time period/point  $t$  goes to zero for observations sufficiently far in the past, say prior to  $t - d$ . Consider the following simple autoregressive model:

$$Y_t = g(Y_{t-1}, \dots, Y_{t-d}) + u_t, \quad t = 1, \dots, T \quad (1.26)$$

where  $Y_t$  is a r.v. of interest (e.g., product demand),  $g(\cdot)$  is the regression function, and  $u_t$  is the model error term, which is assumed to be uncorrelated with  $Y_t$ , i.e.,  $E(u_t | Y_1, \dots, Y_T) = 0$ .

For a linear regression function, e.g.,  $g(Y_{t-1}, \dots, Y_{t-d}) = c_0 + \sum_{\tau=t-d}^t c_\tau Y_\tau$  where  $c_0$  and  $c_\tau$ ,  $\tau = t - d, \dots, t$ , are the model parameters, classical linear regression is used to obtain the *mean* of the output/response variable ( $Y_t$ ) conditional on the input variables or covariates ( $Y_{t-1}, \dots, Y_{t-d}$ ) by estimating the model parameters (denoted by  $\hat{c}_0$  and  $\hat{c}_\tau$ ,  $\tau = t - d, \dots, t$ ). Nonlinear parametric models as well as nonparametric techniques can also be used in classical regression. Quantile regression (Koenker, 2005), on the other hand, gives a more detailed description of the underlying distribution, since the output of the regression is a specified conditional quantile (e.g., median). Nonparametric conditional quantile regression is an optimization-based approach to estimate quantiles of a regression model where the regression function is conditionally estimated using a kernel function. Figure 1.4 illustrates kernel regression applied to conditional quantile estimation of a stochastic process. For more details, see Chapter 6 in Li & Racine (2007), Li, Liu, & Zhu (2007).

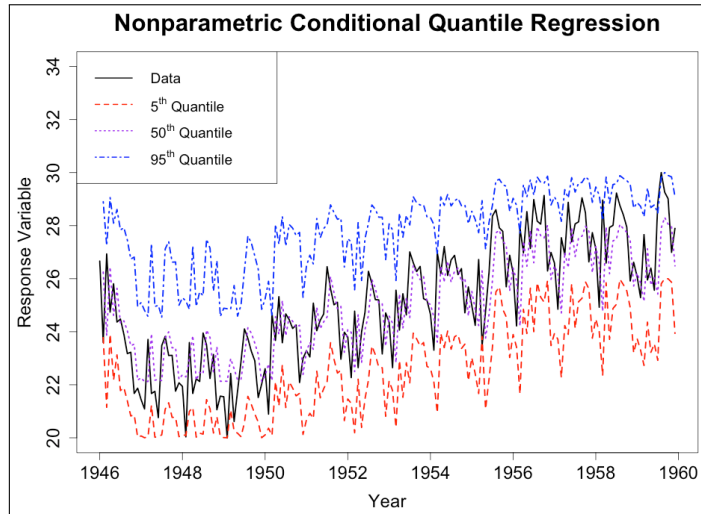


Figure 1.4: Illustration of Kernel Regression for conditional quantile estimation of a stochastic process.

**Available Software.** Several software packages and programming languages offer kernel smoothing features, functions, and routines. Different computer

codes take different approaches for the calculation of optimal bandwidths (plug-in vs. cross-validation methods) and estimation of quantiles (PDF-based vs. CDF-based). Some examples include R programming language (R Core Team, 2015) (packages `np`, `ks`, `KernSmooth`), MATLAB® (The MathWorks Inc., 2015) (function `ksdensity`), SAS® software (SAS Institute, 2015) (procedure KDE), Stata (StataCorp LP, 2015) (command `ksdensity`).

### 1.3 Optimization Under Uncertainty

In this section, we first give a general formulation for mathematical optimization under uncertainty (Georghiou, Wiesemann, & Kuhn, 2011), then address two special cases: Stochastic Programming and Chance-Constrained Programming. The former is used to model problems in Chapters 3, 5 and 6, whereas the latter is the topic of Chapter 4.

In optimization under uncertainty, the nature of the decisions is related to the knowledge of the realization of the uncertainty at a given *stage* in the decision-making process. Decisions are classified as *here-and-now*, when they are taken before the realization of the uncertainty, and *wait-and-see*, when taken after the values of the uncertain parameters (random variables) are revealed (i.e., recourse actions). This two-stage decision-making process can be generalized to account for multiple stages. In multi-period problems, usually, even though not necessarily, a stage corresponds to a single time period or a cluster of time periods.

Before presenting the general formulation, we describe its structure and notation. A decision maker first observes a vector of uncertain parameters  $\tilde{\xi}_1 \in \mathbb{R}^{k_1}$  and then takes decisions  $x_1(\tilde{\xi}_1) \in \mathbb{R}^{n_1}$ . Subsequently, the value of a second vector of uncertain parameters  $\tilde{\xi}_2 \in \mathbb{R}^{k_2}$  is revealed, and the recourse actions taken by the decision maker are denoted by  $x_2(\tilde{\xi}_1, \tilde{\xi}_2) \in \mathbb{R}^{n_2}$ . This process is repeated for  $T$  stages, where at any stage  $t = 1, \dots, T$  the decision maker observes a vector of uncertain parameters  $\tilde{\xi}_t$  and takes decisions  $x_t(\tilde{\xi}_1, \dots, \tilde{\xi}_t)$ . It is important to note that decisions taken at stage  $t$  depend on the whole history of past observations of  $\tilde{\xi}_1, \dots, \tilde{\xi}_t$ , but they may *not* depend on future observations of  $\tilde{\xi}_{t+1}, \dots, \tilde{\xi}_T$ . This feature is the so-called *non-anticipativity condition* (NAC), which ensures causality of events. To simplify notation, we define the random variables of the history of observations up to time  $t$  as  $\tilde{\xi}^t = (\tilde{\xi}_1, \dots, \tilde{\xi}_t) \in \mathbb{R}^{k^t}$ , where  $k^t = \sum_{s=1}^t k_s$ . Moreover, let  $x = [x_1(\tilde{\xi}_1), \dots, x_T(\tilde{\xi}^T)] \in \mathbb{R}^n$  and  $\tilde{\xi} = [\tilde{\xi}_1, \dots, \tilde{\xi}_T] \in \mathbb{R}^k$  denote the vector concatenations of all decision variables and uncertain parameters, respectively, where  $n = \sum_{t=1}^T n_t$  and  $k = k^T$ .

The general formulation is given as follows:

$$\begin{aligned} \min_x \quad & \mathbb{E}_{\tilde{\xi}} \left[ \sum_{t=1}^T c_t(\tilde{\xi}^t)^\top x_t(\tilde{\xi}^t) \right] \\ \text{s.t.} \quad & \left. \begin{aligned} \mathbb{E}_{\tilde{\xi}} \left[ \sum_{s=1}^T A_{t,s} x_s(\tilde{\xi}^s) \mid \tilde{\xi}^t \right] &\geq b_t(\tilde{\xi}^t) \\ x_t(\tilde{\xi}^t) &\geq 0 \end{aligned} \right\} \forall \tilde{\xi} \in \Xi, t = 1, \dots, T \end{aligned} \quad (1.27)$$

it is assumed that the vector of cost coefficients in the objective function depends linearly on the observation history, i.e.,  $c_t(\tilde{\xi}^t) = C_t \tilde{\xi}^t$ , where  $C_t \in \mathbb{R}^{n_t \times k^t}$  is some matrix. The expectation operator  $\mathbb{E}_{\tilde{\xi}}[\cdot]$  is applied to the random vector  $\tilde{\xi}$ ,  $\Xi$  is the set of all *expressions* of the uncertain parameters (recall from [Section 1.2](#) that a random variable is a measurable function; thus, in general,  $\Xi$  corresponds to the Cartesian product of the images of each element of the random vector). The constraints contain the deterministic matrix  $A_{t,s} \in \mathbb{R}^{m_t \times n_s}$  and the uncertain right-hand side vector  $b_t(\tilde{\xi}^t) \in \mathbb{R}^{m_t}$ , which is assumed to be of the form  $b_t(\tilde{\xi}^t) = B_t \tilde{\xi}^t$  for some matrix  $B_t \in \mathbb{R}^{m_t \times k^t}$ . The conditional expectation  $\mathbb{E}_{\tilde{\xi}}[\cdot \mid \tilde{\xi}^t]$  treats the individual values of  $\tilde{\xi}_1, \dots, \tilde{\xi}_t$  as realized and performs the expectation operation only with respect to future observations of  $\tilde{\xi}_{t+1}, \dots, \tilde{\xi}_T$ .

### 1.3.1 Stochastic Programming

The general formulation in equation (1.27) can be specialized to a linear, multi-stage stochastic programming (MSSP) problem with recourse ([Birge & Louveaux, 2011](#)). In order to satisfy the NAC, we must set  $A_{t,s} = 0$  for all  $t < s$ , which has the effect that the term inside the conditional expectation of the  $t$ -th stage constraint becomes independent of  $\tilde{\xi}_{t+1}, \dots, \tilde{\xi}_T$ . Thus, the conditional expectation becomes redundant as the term that is conditioned on,  $\tilde{\xi}$ , is treated as a constant. The MSSP problem formulation is given by,

$$\begin{aligned} \min_x \quad & \mathbb{E}_{\tilde{\xi}} \left[ \sum_{t=1}^T c_t(\tilde{\xi}^t)^\top x_t(\tilde{\xi}^t) \right] \\ \text{s.t.} \quad & \left. \begin{aligned} \sum_{s=1}^T A_{t,s} x_s(\tilde{\xi}^s) &\geq b_t(\tilde{\xi}^t) \\ x_t(\tilde{\xi}^t) &\geq 0 \end{aligned} \right\} \forall \tilde{\xi} \in \Xi, t = 1, \dots, T \end{aligned} \quad (1.28)$$

### 1.3.2 Chance-Constrained Programming

Chance-constrained programming (CCP) was originally proposed by [Charnes & Cooper \(1959\)](#) and its theory is expanded in [Prékopa \(1995\)](#). Let  $\mathbb{P}_{\tilde{\xi}}$  be the (joint) distribution of  $\tilde{\xi}$ . The multi-stage generalization of CCP problems can

be expressed as follows:

$$\begin{aligned}
\min_x \quad & \mathbb{E}_{\tilde{\xi}} \left[ \sum_{t=1}^T c_t(\tilde{\xi}^t)^\top x_t(\tilde{\xi}^t) \right] \\
\text{s.t.} \quad & \mathbb{P}_{\tilde{\xi}} \left( \sum_{t=1}^T a_{i,t} x_t(\tilde{\xi}^t) \geq b_i(\tilde{\xi}) \right) \geq 1 - \alpha_i \quad \forall i = 1, \dots, I \\
& x_t(\tilde{\xi}^t) \geq 0 \quad \forall \tilde{\xi} \in \Xi, t = 1, \dots, T
\end{aligned} \tag{1.29}$$

where  $a_{i,t} \in \mathbb{R}^{n_t}$ ,  $b_i(\tilde{\xi}) \in \mathbb{R}$ , and  $\alpha_i \in (0, 1]$ . The probabilistic or chance constraint requires that the argument of  $\mathbb{P}_{\tilde{\xi}}(\cdot)$  should be satisfied with probability at least  $1 - \alpha_i$ , which is called the confidence level.  $\alpha_i$  is called the risk level. It can be shown that the formulation in equation (1.29) has a tight conservative approximation to the general formulation in equation (1.27).

## 1.4 Overview of the Thesis

Figure 1.5 shows a graphic overview of the topics in Chapters 2 to 6. The application discussed in Chapter 4 motivated the novel work proposed in Chapter 5. The following subsections provide an abstract of all the chapters.

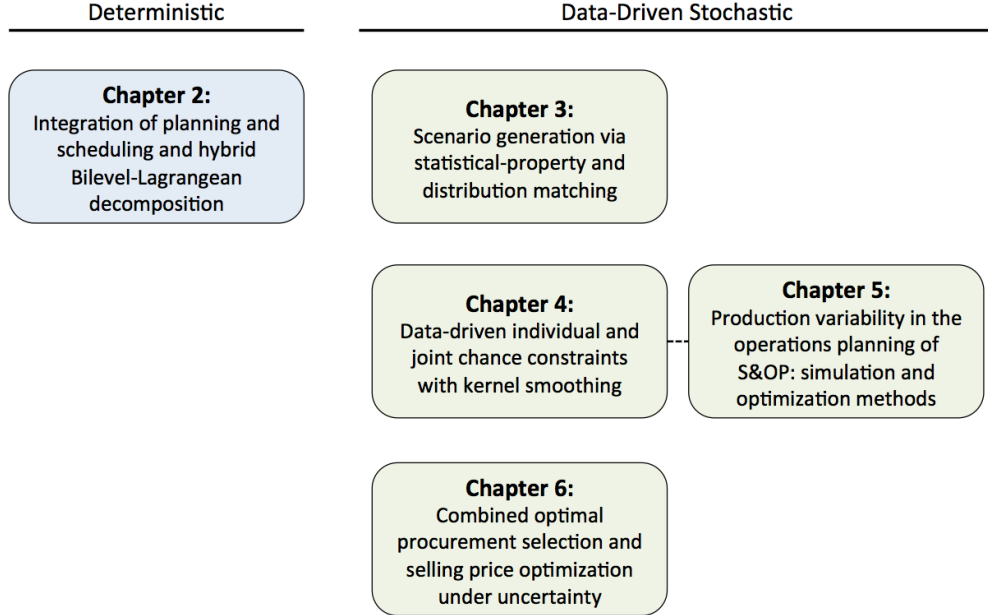


Figure 1.5: Overview of the thesis.

### 1.4.1 Chapter 2

Motivated by a real-world industrial problem, Chapter 2 deals with the integration of planning and scheduling in the operation of a network of batch plants. The network consists of single-stage, multiproduct batch plants located in different sites, which can exchange intermediate products in order to blend them to obtain finished products. The time horizon is given and divided into

multiple time periods, at the end of which the customer demands have to be exactly satisfied. The planning model is a simplified and aggregate formulation derived from the detailed precedence-based scheduling formulation. Traveling Salesman Problem (TSP) constraints are incorporated at the planning level in order to predict the sequence-dependent changeovers between groups of products, within and across time periods, without requiring the detailed timing of operations, which is performed at the scheduling level. In an effort to avoid solving the full-space, rigorous scheduling model, especially for large problem sizes, two decomposition strategies are investigated: Bilevel and Temporal Lagrangean. We demonstrate that Bilevel Decomposition is efficient for small to medium problem instances and that further decomposition of the planning problem, yielding a hybrid decomposition scheme, is advantageous for tackling a large-scale industrial test case.

### 1.4.2 Chapter 3

In this chapter, we bring systematic methods for scenario tree generation to the attention of the Process Systems Engineering community. We focus on a general, data-driven optimization-based method for generating scenario trees that does not require strict assumptions on the probability distributions of the uncertain parameters. Using as a basis the Moment Matching Problem (MMP) originally proposed by [Høyland & Wallace \(2001\)](#), we propose matching marginal (Empirical) Cumulative Distribution Function information of the uncertain parameters in order to cope with potentially under-specified MMP formulations. The new method gives rise to a Distribution Matching Problem (DMP) that is aided by predictive analytics. We present two approaches for generating multi-stage scenario trees by considering time series modeling and forecasting. The aforementioned techniques are illustrated with a production planning problem with uncertainty in production yield and correlated product demands.

### 1.4.3 Chapter 4

We propose a data-driven, nonparametric approach to reformulate (conditional) individual and joint chance constraints with right-hand side uncertainty into algebraic constraints. The approach consists of using kernel smoothing to approximate unknown true continuous probability density/distribution functions. Given historical data for continuous univariate or multivariate random variables (uncertain parameters in an optimization model), the inverse cumulative distribution function (quantile function) and the joint cumulative distribution function are estimated for the univariate and multivariate cases, respectively. The approach relies on the construction of a confidence set that contains the unknown true distribution. The distance between the true distribution and its estimate is modeled via  $\phi$ -divergences. We propose a new way of specifying the size of the confidence set (i.e., the  $\phi$ -divergence tolerance) based on point-wise standard errors of the smoothing estimates. The approach is illustrated with a motivating and an industrial production planning problem

with uncertain plant production rates.

#### 1.4.4 Chapter 5

We propose two data-driven, optimization-based frameworks (simulation-optimization and bi-objective optimization) to account for production variability in the operations planning stage of the Sales and Operations Planning (S&OP) of an enterprise. Production variability is measured as the deviation between historical planned (target) and actual (achieved) production rates. A statistical technique, namely, quantile regression is used to model the distribution of deviation values given planned production rates. Scenarios are constructed by sampling from the distribution of deviation values and used as inputs to the proposed optimization-based frameworks. Advantages and disadvantages of the two proposed frameworks are discussed. The applicability of the proposed methodology is illustrated with a detailed analysis of the results of a motivating example and a real-world production planning problem from a chemical company.

#### 1.4.5 Chapter 6

In this chapter, we propose extending the production planning decisions of a chemical process network to include optimal contract selection under uncertainty with suppliers and product selling price optimization. We use three quantity-based contract models: discount after a certain purchased amount, bulk discount, and fixed duration contracts. With regards to pricing models, we argue that general regression models can be used to describe the relationship between selling price, demand, and possibly other predictors, such as economic indicators. For illustration purposes, we consider three demand-response models (i.e., selling price as a function of demand) that are typically encountered in the literature: linear, constant-elasticity, and logit. Uncertainty is considered and modeled in both supply (e.g., raw material spot market price) and demand (random nature of the residuals of the regression models). The proposed method is illustrated with two numerical examples of chemical process networks.

#### 1.4.6 Chapter 7

[Chapter 7](#) provides a critical review of the work in this thesis, along with a summary of its contributions and suggestions for future work. This thesis has led to the following papers:

- Calfa, B. A., Agarwal, A., Grossmann, I. E. & Wassick, J. M. (2013) *Hybrid Bilevel-Lagrangian Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants*. **Industrial & Engineering Chemistry Research**. 52(5): 2152-2167.
- Calfa, B. A. (2014) *A Memory-Efficient Implementation of Multi-Period Two- and Multi-Stage Stochastic Programming Models*. Carnegie Mellon



University. Technical Report. Available at: <http://repository.cmu.edu/cheme/246/>.

- Calfa, B. A., Agarwal, A., Grossmann, I. E., & Wassick, J. M. (2014) *Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching*. **Computers & Chemical Engineering**. 68(1): 7-23.
- Calfa, B. A., Grossmann, I. E., Agarwal, A., & Wassick, J. M. (2015) *Data-Driven Individual and Joint Chance-Constrained Optimization via Kernel Smoothing*. **Computers & Chemical Engineering**. Submitted for publication.
- Calfa, B. A., Agarwal, A., Wassick, J. M. & Grossmann, I. E. (2015) *Data-Driven Simulation and Optimization Approaches to Incorporate Production Variability in Sales and Operations Planning*. **Industrial & Engineering Chemistry Research**. Submitted for publication.
- Calfa, B. A., & Grossmann, I. E. (2015) *Optimal Procurement Contract Selection with Price Optimization under Uncertainty for Process Networks*. **Computers & Chemical Engineering**. Submitted for publication.

# Chapter 2

## Hybrid Bilevel-Lagrangian Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants

### Notation

#### Indices

$g, g'$	Products groups
$i, i'$	Products (finished, intermediate, and blended products)
$l, l'$	Locations (plants and customers)
$r$	Raw materials
$t$	Time periods
$u$	Processing units

#### Sets

$G_{ul}$	Set of groups that can be processed in unit $u$ of plant $l$
$I_g$	Set of products that belong to group $g$
$I_{ul}$	Set of products that can be processed in unit $u$ of plant $l$
$IB_i$	Set of blended products $i$
$IFP_i$	Set of finished products $i$
$II_i$	Set of finished and intermediate products $i$

$ILL_{il}$	Set of products $i$ processed in plant $l$
$IPM_i$	Set of intermediate products $i$
$LCI_{li}$	Set of customers $l$ that demand product $i$
$LP_l$	Set of plants
$R$	Set of raw materials $r$
$T$	Set of time periods
$U_{gl}$	Set of units of plant $l$ that can process group $g$
$U_{il}$	Set of units of plant $l$ that can process product $i$
$U_l$	Set of blended units of plant $l$

### Parameters

$BINVC_{ilt}$	Blending inventory cost of product $i$ in plant $l$ at time period $t$
$BINVU_{ilt}$	Upper bound for blending inventory for product $i$ in plant $l$ at time period $t$
$BPH_{iul}$	Batches per hour of product $i$ in unit $u$ in plant $l$
$BR_{ii'l}$	Blending ratio between individual products $i$ and $i'$ in plant $l$
$BS_{iul}$	Batch size of product $i$ in unit $u$ in plant $l$
$CC_{gg'ul}$	Changeover cost between group $g$ and group $g'$ in unit $u$ in plant $l$
$CRP_{ii'l}$	Conversion ratio from intermediate product $i$ to finished product $i'$ in plant $l$
$CRR_{ril}$	Conversion ratio from raw material $r$ to product $i$ in plant $l$
$CT_{gg'ul}$	Changeover time between group $g$ and group $g'$ in unit $u$ in plant $l$
$DL_{ilt}$	Demand of product $i$ for customer $l$ at the end of time period $t$
$H_t$	Duration of time period $t$
$INVC_{ilt}$	Dedicated inventory cost of product $i$ in plant $l$ at time period $t$
$INVU_{ilt}$	Upper bound for dedicated inventory for product $i$ in plant $l$ at time period $t$
$IRPC_{rl}$	Inbound transportation cost of raw material $r$ to plant $l$
$MPPA_i$	Maximum number of plants that can produce product $i$
$OPC_{ult}$	Operating cost of unit $u$ in plant $l$ at time period $t$
$OTPC_{ill't}$	Outbound transportation cost of product $i$ from plant $l$ to customer $l'$
$PALB_{il}$	Minimum production capacity of product $i$ in plant $l$
$PAUB_{il}$	Maximum production capacity of product $i$ in plant $l$

$PCAP_{lt}$	Maximum production capacity of plant $l$ at time period $t$
$SUC_{iul}$	Penalty cost for introducing a new product $i$ in unit $u$ of plant $l$ that does not normally process product $i$
$SUUB_l$	Upper bound on the number of scale-up decisions in plant $l$
$SUUBT$	Upper bound on the number of scale-up decisions for all plants
$TCBP_{ll'}$	Transportation cost from plant $l$ to another plant $l'$

### Common Variables

$BA_{ilt}$	Production amount of blended product $i$ in plant $l$ at time period $t$
$BINV_{ilt}$	Inventory levels of blended product $i$ in plant $l$ at time $t$ for blending storage vessels
$INV_{ilt}$	Inventory levels of product $i$ in plant $l$ at time $t$ for dedicated storage vessels
$INVslk_{ilt}$	Slack variable for maximum storage violation of product $i$ in plant $l$ at time period $t$
$IST_{ill't}$	Inter-site transported amounts of product $i$ between plants $l$ and $l'$ at time period $t$
$NB_{iult}$	Number of batches of product $i$ in unit $u$ in plant $l$ at time period $t$
$PA_{iult}$	Production amount of product $i$ produced in unit $u$ in plant $l$ at time period $t$
$PRM_{ilt}$	Production amount of product $i$ that is also a raw material produced in unit $u$ in plant $l$ at time period $t$
$PRMI_{ilt}$	Intermediate product $i$ that is used as a raw material produced in plant $l$ at time period $t$
$PRMO_{ilt}$	Intermediate product $i$ that is used as a raw material used in plant $l$ at time period $t$
$RM_{rlt}$	Amount of raw material $r$ needed in plant $l$ at time period $t$
$SL_{ill't}$	Sales of product $i$ from plant $l$ to customer $l'$ at time $t$
$STR_{ilt}$	Amount stored of product $i$ in plant $l$ at time period $t$
$TA_{ii'lt}$	Amount of product $i$ transferred to blending inventory that makes up blend $i'$ in plant $l$ at time period $t$
$TAI_{ii'lt}$	Amount of intermediate product $i$ transferred to blending inventory that makes up blend $i'$ in plant $l$ at time period $t$
$TAO_{ii'lt}$	Amount of intermediate product $i$ transferred to blending inventory that makes up blend $i'$ used in plant $l$ at time period $t$

$ypsc_{iult}$	0-1 variable to denote the assignment of product $i$ at time period $t$ to unit $u$ of plant $l$ that does not normally produce it (associated with scale-up costs)
$ysl_{ill't}$	0-1 variable to force only one plant $l$ to ship product $i$ to customer $l'$ at time period $t$

### Planning Model Variables

$yg_{gult}$	0-1 variable to denote the assignment of group $g$ to unit $u$ of plant $l$ at time period $t$
$ygf_{gult}$	0-1 variable to denote the first group $g$ assigned to unit $u$ of plant $l$ at time period $t$
$ysl_{gult}$	0-1 variable to denote the last group $g$ assigned to unit $u$ of plant $l$ at time period $t$
$zgg'_{ult}$	0-1 variable to denote if group $g$ is followed by group $g'$ in unit $u$ of plant $l$ within time period $t$
$zzgg'_{ult}$	0-1 variable to denote if link between group $g$ and group $g'$ in unit $u$ of plant $l$ within time period $t$ is broken
$zzzgg'_{ult}$	0-1 variable to denote a changeover between group $g$ and group $g'$ in unit $u$ of plant $l$ across time period $t$

### Unit-Specific General Precedence Model Variables

$Te_{iult}$	End time of product $i$ in unit $u$ of plant $l$ at time period $t$
$Ts_{iult}$	Start time of product $i$ in unit $u$ of plant $l$ at time period $t$
$wpii'_{ult}$	0-1 variable to denote changeover between product $i$ and product $i'$ in unit $u$ of plant $l$ across time period $t$
$xpii'_{ult}$	0-1 variable to denote local precedence between product $i$ and product $i'$ in unit $u$ of plant $l$ within time period $t$
$yp_{iult}$	0-1 variable to denote the assignment of product $i$ to unit $u$ of plant $l$ at time period $t$
$ypf_{iult}$	0-1 variable to denote the first product $i$ assigned to unit $u$ of plant $l$ at time period $t$
$ypl_{iult}$	0-1 variable to denote the last product $i$ assigned to unit $u$ of plant $l$ at time period $t$
$zpii'_{ult}$	0-1 variable to denote global precedence between product $i$ and product $i'$ in unit $u$ of plant $l$ within time period $t$

## 2.1 Introduction

The focus of this work is on integrated planning and scheduling for batch processes. Excellent reviews on each of these levels are available in the literature, for instance [Méndez \*et al.\* \(2006\)](#), [Kallrath \(2002\)](#), and [Maravelias \(2012\)](#). Planning models are usually posed as Linear Programming (LP) or Mixed-Integer Linear Programming (MILP) problems. For instance, [Pochet](#)

& Wolsey (2006) present several MIP formulations of the lot-sizing production planning problem. The main characteristics of such models are captured in constraints related to production targets, production and inventory costs, and material/inventory balances. Usually, those considerations are satisfactory at a planning level, which may span a time horizon of months to years.

In this work, we deal with the integration of planning and scheduling of a network of batch plants motivated by a real-world industrial problem. The network consists of single-stage, multiproduct batch plants located in different sites, which can exchange intermediate products in order to blend them to obtain final products. Sequence-dependent changeover data are given in terms of groups or families of products, which are applied at the planning level. At the scheduling level a detailed model to obtain the timing and sequencing of batches of individual products is applied. It should be noted that at the scheduling level there can also be sequence-dependent changeovers between products belonging to the same group that are accounted for but these are of smaller magnitude than changeovers among different products. Hence, changeovers among products belonging to the same group are not considered at the planning level. The time horizon is given and divided into multiple time periods, at the end of which the customer demands have to be satisfied. We show that Bilevel Decomposition, Lagrangean Decomposition and a hybrid of those methods can be efficient when solving large-scale problems.

The remainder of the chapter is organized as follows. In [Section 2.2](#), we define the manufacturing problem. In [Section 2.3](#), we present the complete planning and scheduling formulations and outline the major assumptions in our models. In [Section 2.4](#), we discuss how the Bilevel Decomposition, Lagrangean Decomposition and a hybrid of those schemes are integrated. [Section 2.5](#) contains computational results of three problem instances of increasing size. [Section 2.6](#) summarizes the main conclusions of the chapter.

## 2.2 Problem Definition

The problem addressed in this work concerns the planning and scheduling of a network of multi-product, single-stage batch plants as shown in [Figure 2.1](#). The problem statement is as follows: a set of raw materials is purchased and transported to each plant, which transforms them into products to fulfill specified customer demands. Shipments of *intermediate* products between plants are allowed, hence the network structure depicted in [Figure 2.1](#). Products are classified into *groups* or *families*, and we identify a product  $i$  belonging to group  $g$  through the subset  $I_g$ . Sequence-dependent changeover data are given between groups of products, which are considered at the planning level. For the scheduling level, changeover times and costs associated with the transition from one product to another of the same group are also given. As indicated above, the sequence-dependent changeovers between different groups of products are significantly higher than the ones from one product to another in the same group. Additionally, the assignment of a product to a plant that does

not normally produce it incurs a *scale-up cost*, which for example can be associated with the testing of new batches of the product in order to meet the quality standards set by the customer. The goal is to minimize total costs over a time horizon given by time periods of months in which the demands are specified at the end of each period.

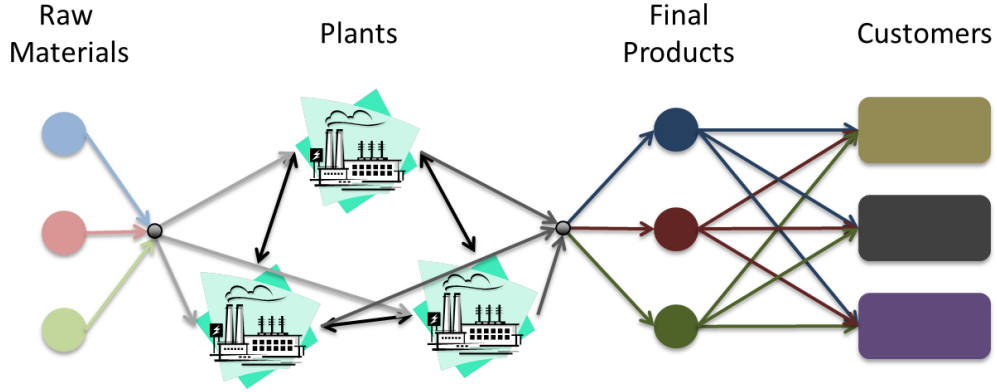


Figure 2.1: Supply-chain schematic of the problem.

Figure 2.2 shows the configuration of each plant. Each plant contains a number of reactors (processing units) operating in parallel that can process a subset of the raw materials and transform them into finished and intermediate products. After a batch is completed, the product is transferred to dedicated storage vessels and becomes a part of the inventory of the plant. In addition, a subset of blended products are formed by transferring finished products from their individual inventories to blend inventories. Finally, finished and blended products are shipped to customers.

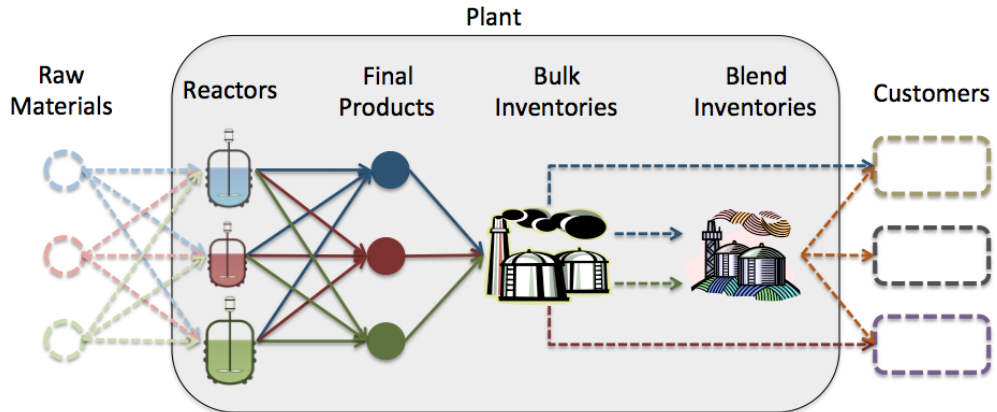


Figure 2.2: Detailed components of the plant.

Given batch sizes and fixed processing times, sequence-dependent changeover times and costs, transportation and inventory costs, and demand forecasts over a time horizon consisting of several time periods, the objectives are to determine:

- (a) The amounts of products to be produced in the plant in each time period;
- (b) The allocation of products to batch units in each time period;
- (c) The detailed timing of operations and sequencing of products in each unit, taking into consideration sequence-dependent changeover times and costs.

The goal is to minimize total cost – including operating, transportation, inventory, changeover costs – in order to meet customer demands at the end of each time period.

In the next section, we present the planning and scheduling models. The planning formulation not only contains material and inventory balances for products (production planning), but also estimates the sequencing of different groups of products through the use of TSP constraints. The scheduling formulation contains equivalent material and inventory balances for products and obtains the detailed production schedule of products, including the timing of operations and accounting for all possible changeovers. The two formulations are integrated through decomposition schemes (Bilevel and Lagrangean) in [Section 2.4](#).

## 2.3 Problem Formulation

The plants' topology is sequential, i.e., single-staged plant with parallel units. The following assumptions are made:

- i. Inventories are stored in dedicated vessels with finite capacity;
- ii. Times for material transfer throughout the plant are considered to be negligible compared to processing and changeover times;
- iii. Batch sizes are fixed and given;
- iv. Batch processing times are given and unit-dependent;
- v. Demands that must be exactly satisfied are specified at the end of each period;
- vi. Sequence-dependent changeovers are group- and unit-dependent;
- vii. All data are deterministic.

The kernel of the planning formulation is based on the Detailed Planning model proposed by [Erdirik-Dogan & Grossmann \(2007\)](#), which involves the classical Traveling Salesman Problem (TSP) sequencing constraints. The scheduling model is based on a modification of the Unit-Specific General Precedence (USGP) model proposed by [Kopanos, Laínez, & Puigjaner \(2009\)](#). Also, sequence-dependent changeovers between products families or groups have been included at the planning level ([Kopanos, Puigjaner, & Maravelias, 2011](#)).



## 2.3.1 Planning Model

### 2.3.1.1 Assignment and Sequencing within Time Periods

The assignments of group  $g$  to unit  $u$  in plant  $l$  in time period  $t$  are given by binary variables  $yg_{ult}$  defined as:

$$yg_{ult} = \begin{cases} 1, & \text{if group } g \text{ is assigned to unit } u \text{ in plant } l \text{ in time period } t \\ 0, & \text{otherwise} \end{cases}$$

Likewise for products, the assignments of product  $i$  to unit  $u$  in plant  $l$  in time period  $t$  are denoted by binary variables  $yp_{iult}$  defined as:

$$yp_{iult} = \begin{cases} 1, & \text{if product } i \text{ is assigned to unit } u \text{ in plant } l \text{ in time period } t \\ 0, & \text{otherwise} \end{cases}$$

Therefore, constraints (2.1) state that if product  $i$  is assigned to unit  $u$  at time period  $t$ , then the group to which it belongs is also assigned to the same unit and time period for plant  $l$ . In addition, according to constraints (2.2),  $yg_{ult}$  is forced to zero if no product  $i$  of group  $g$  is assigned to any unit  $u$  at time period  $t$  in plant  $l$ .

$$yg_{ult} \geq yp_{iult} \quad \forall i \in I_g, g \in G_{ul}, u \in U_{il}, l \in LP_l, t \in T \quad (2.1)$$

$$yg_{ult} \leq \sum_{\substack{i \in I_g \\ i \in I_{ul}}} yp_{iult} \quad \forall g \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.2)$$

Constraints (2.3) activate the binary variables responsible for identifying the products that have been assigned to unit  $u$  of plant  $l$  that does not normally produce it. When this happens, it will incur scale-up costs that for example can be related to the testing phase of new batches of the product in order ensure its quality.

$$yp^{sc}_{iult} \geq yp_{iult} \quad \forall i \in II_i, i \notin IIL_{il}, u \in U_{il}, l \in LP_l, t \in T \quad (2.3)$$

The main idea for the sequencing of products groups is to generate a cyclic schedule for each time period that minimizes changeover times among the assigned groups, and then to determine the optimal sequence of groups by *breaking* one of the links in the cycle as illustrated in Figure 2.3.

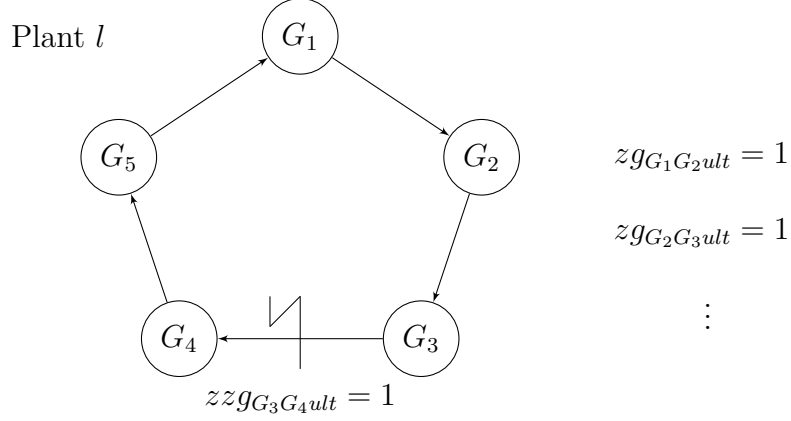


Figure 2.3: Example of cyclic schedule and breaking of one of the links to determine optimal sequence.

The 0-1 variables  $z_{gg'ult}$  represent the changeovers between groups  $g$  and  $g'$  in unit  $u$  in plant  $l$  in time period  $t$ . The cyclic schedules are generated with constraints (2.4) and (2.5), the assignment constraints of the TSP constraints, which state that for each plant  $l$ , group  $g$  is assigned to unit  $u$  during period  $t$  if and only if there is exactly one transition from group  $g$  to product  $g'$  in unit  $u$  in time period  $t$ , and group  $g'$  is assigned to unit  $u$  in period  $t$  if and only if there is exactly one transition from any group  $g$  to group  $g'$  in unit  $u$  in time period  $t$ , respectively.

$$y_{gult} = \sum_{g' \in G_{ul}} z_{gg'ult} \quad \forall g \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.4)$$

$$y_{g'ult} = \sum_{g \in G_{ul}} z_{gg'ult} \quad \forall g' \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.5)$$

Only one link within a cyclic schedule is allowed to be broken ( $zz_{gg'ult} = 1$ ) and that is represented by constraints in (2.6). However, a link can only be broken if the corresponding pair is selected in the cycle as according to constraints (2.7).

$$\sum_{g \in G_{ul}} \sum_{g' \in G_{ul}} zz_{gg'ult} = 1 \quad \forall u \in U_l, l \in LP_l, t \in T \quad (2.6)$$

$$zz_{gg'ult} \leq z_{gg'ult} \quad \forall g \in G_{ul}, g' \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.7)$$

In equations (2.4) and (2.5) the groups  $g$  and  $g'$  can refer to the same group. In order to properly allow self changeovers constraints (2.8) – (2.10) have to be considered. These constraints state that for each plant  $l$ , if group  $g$  is assigned to unit  $u$  in time period  $t$ , and none of the groups  $g'$  different from  $g$  are assigned to the same unit in the same time period, then batches belonging to group  $g$  can be followed by another set of batches also belonging to the same group. Also, if batches of group  $g$  are followed by other batches of the same group in unit  $u$  in period  $t$ , then only group  $g$  is assigned to unit

$u$  for period  $t$  and none of the groups  $g'$  other than  $g$  are assigned to the same unit in the same time period.

$$yg_{gult} \geq zg_{gult} \quad \forall g \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.8)$$

$$zg_{ggult} + yg_{g'ult} \leq 1 \quad \forall g \in G_{ul}, g' \in G_{ul}, g \neq g', u \in U_l, l \in LP_l, t \in T \quad (2.9)$$

$$zg_{ggult} \geq yg_{gult} - \sum_{\substack{g' \in G_{ul} \\ g' \neq g}} yg_{g'ult} \quad \forall g \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.10)$$

### 2.3.1.2 Sequencing of Groups across Time Periods

In order to model sequence-dependent changeovers across time periods, it is necessary to identify the first and last products groups in each sequence. Therefore, two binary variables are introduced:  $ygf_{gult}$  and  $ysl_{gult}$  (see [Figure 2.4](#)).

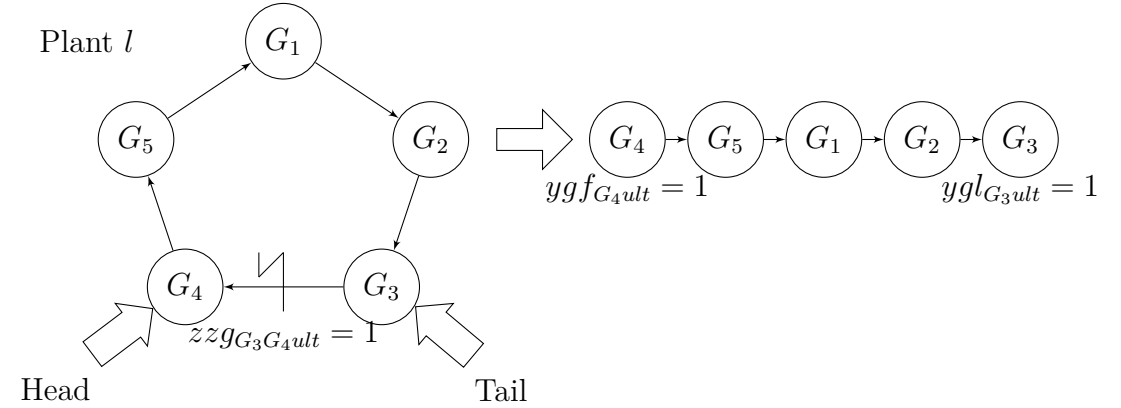


Figure 2.4: Breaking the cyclic schedule to determine the first and last groups of the sequence.

If at least one of the links between groups  $g$  and  $g'$  is broken, then group  $g'$  becomes the *first* group in the optimal sequence for unit  $u$  during time period  $t$  as stated in constraints (2.11). Similarly, constraints (2.12) imply that if at least one of the links between group  $g$  to any group  $g'$  is broken, then group  $g$  becomes the *last* group in the optimal sequence for unit  $u$  during time period  $t$ .

$$ygf_{g'ult} \geq \sum_{g \in G_{ul}} zzg_{gg'ult} \quad \forall g' \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.11)$$

$$ysl_{gult} \geq \sum_{g' \in G_{ul}} zzg_{gg'ult} \quad \forall g \in G_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.12)$$

Moreover, exactly one group can be the first and the last group to be

processed in each unit as represented in equations (2.13) and (2.14).

$$\sum_{g \in G_{ul}} ygf_{gult} = 1 \quad \forall u \in U_l, l \in LP_l, t \in T \quad (2.13)$$

$$\sum_{g \in G_{ul}} ygl_{gult} = 1 \quad \forall u \in U_l, l \in LP_l, t \in T \quad (2.14)$$

Sequence-dependent changeovers across time periods are modeled as follows. For each plant  $l$ , equations (2.15) state that exactly one changeover occurs from group  $g$  to group  $g'$  at the end of time period  $t$  in unit  $u$  if and only if group  $g$  is produced last in time period  $t$ . Similarly, according to equations (2.16), exactly one changeover from group  $g$  to group  $g'$  occurs at the beginning of time period  $t+1$  in unit  $u$  if and only if group  $g'$  is produced first in unit  $u$  in time period  $t+1$ .

$$ygl_{gult} = \sum_{g' \in G_{ul}} zzzg_{gg'ult} \quad \forall g \in G_{ul}, u \in U_l, l \in LP_l, t \in T \setminus \{\bar{t}\} \quad (2.15)$$

$$ygf_{g'ul(t+1)} = \sum_{g \in G_{ul}} zzzg_{gg'ult} \quad \forall g' \in G_{ul}, u \in U_l, l \in LP_l, t \in T \setminus \{\bar{t}\} \quad (2.16)$$

where  $\bar{t}$  is the last time period.

### 2.3.1.3 Time Balances

The time balance constraints (2.17) enforce that the total allocation of production times plus the total changeover times do not exceed the length of each time period. The usage time of each unit  $u$  in plant  $l$  in time period  $t$  is then calculated by adding the batch time to the changeover times, which in turn comprise the changeovers within and across time periods minus the changeover time of the broken link in the cyclic schedule.

$$\sum_{i \in I_{ul}} \frac{NB_{iult}}{BPH_{iul}} + \sum_{g \in G_{ul}} \sum_{g' \in G_{ul}} CT_{gg'ul}(zg_{gg'ult} - zzzg_{gg'ult} + zzzg_{gg'ult}) \leq H_t \quad \forall u \in U_l, l \in LP_l, t \in T \quad (2.17)$$

## 2.3.2 Material and Inventory Balances

Upper and lower bounds on the production amounts for each product are enforced in constraints (2.18) and (2.19).

$$PA_{iult} \geq BS_{iul}yp_{iult} \quad \forall i \in II_i, u \in U_{il}, l \in LP_l, t \in T \quad (2.18)$$

$$PA_{iult} \leq PAUB_{il}yp_{iult} \quad \forall i \in II_i, u \in U_{il}, l \in LP_l, t \in T \quad (2.19)$$

Figure 2.5 shows the schematic for the material and inventory balances. Inventory levels are monitored at the end of each time period.

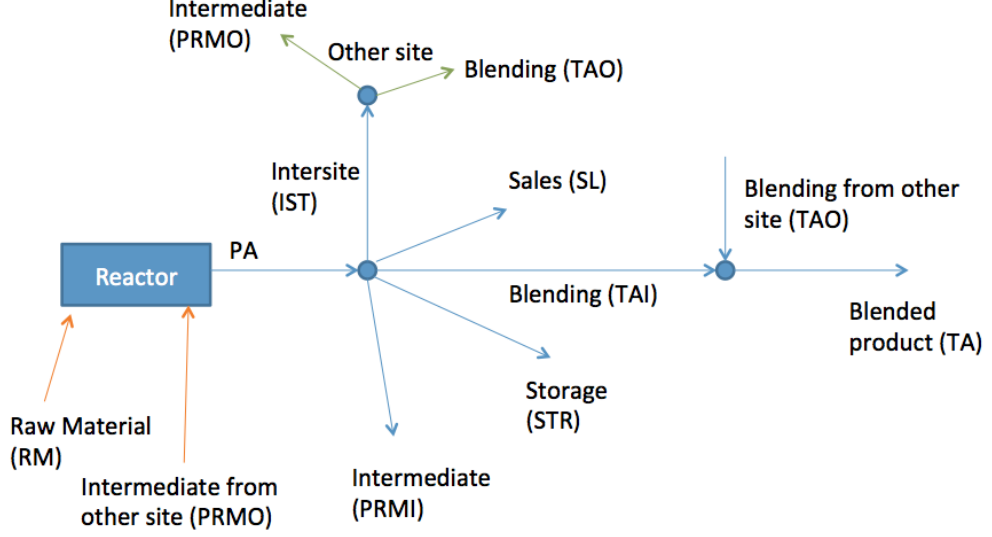


Figure 2.5: Material and inventory balances schematic.

The production amounts of finished, intermediate, and blended products and the amounts of raw materials needed are computed in constraints (2.20) – (2.24). If an intermediate product  $i$  is not to be blended with other intermediate products to form blend  $i'$ , then its blending ratio is set to zero, i.e.,  $BR_{ii'} = 0$ , which in the implementation is represented with a subset with which the corresponding constraint is eliminated from the model.

$$PA_{iult} = NB_{iult}BS_{iul} \quad \forall i \in I_{ul}, u, l \in LP_l, t \in T \quad (2.20)$$

$$BR_{ii'l}BA_{i'lt} = TA_{ii'lt} \quad \forall i \in II_i, i' \in IB_{i'}, l \in LP_l, t \in T \quad (2.21)$$

$$PRM_{ilt} = PRMI_{ilt} + PRMO_{ilt} \quad \forall i \in IPM_i, l \in LP_l, t \in T \quad (2.22)$$

$$PRM_{ilt} = \sum_{i' \in IFP_{i'}} \sum_{u \in U_{i'l}} CRP_{ii'l} PA_{i'ult} \quad \forall i \in IPM_i, l \in LP_l, t \in T \quad (2.23)$$

$$RM_{rlt} = \sum_{i' \in II_{i'}} \sum_{u \in U_{i'l}} CRR_{ri'l} PA_{i'ult} \quad \forall r \in R, l \in LP_l, t \in T \quad (2.24)$$

The material balances on the three nodes in Figure 2.5 for finished and

intermediate products are expressed in constraints (2.25) – (2.27).

$$\begin{aligned} \sum_{u \in U_{il}} PA_{iult} &= \sum_{\substack{l' \in LP_{l'} \\ l' \neq l}} IST_{ill't} + \sum_{l' \in LCI_{l'i}} SL_{ill't} + \\ &\quad \sum_{i' \in IB_{i'}} TAI_{ii'lt} + PRMI_{ilt} + STR_{ilt} \quad \forall i \in II_I, l \in LP_l, t \in T \end{aligned} \quad (2.25)$$

$$\sum_{\substack{l' \in LP_{l'} \\ l' \neq l}} IST_{ill't} = PRMO_{ilt} + \sum_{i' \in IB_{i'}} TAO_{ii'lt} \quad \forall i \in II_i, l \in LP_l, t \in T \quad (2.26)$$

$$TA_{ii'lt} = TAI_{ii'lt} + TAO_{ii'lt} \quad \forall i \in II_i, i' \in IB_{i'}, l \in LP_l, t \in T \quad (2.27)$$

The inventory balances for all products are performed and their capacities are enforced in constraints (2.28) – (2.31).

$$INV_{ilt} = INV_{il(t-1)} + STR_{ilt} \quad \forall i \in II_i, l \in LP_l, t \in T \quad (2.28)$$

$$INV_{ilt} \leq INVU_{ilt} + INVslk_{ilt} \quad \forall i \in II_i, l \in LP_l, t \in T \quad (2.29)$$

$$BINV_{ilt} = BINV_{il(t-1)} + BA_{ilt} - \sum_{l' \in LCI_{l'i}} SL_{ill't} \quad \forall i \in IB_i, l \in LP_l, t \in T \quad (2.30)$$

$$BINV_{ilt} \leq BINVU_{ilt} + INVslk_{ilt} \quad \forall i \in IB_i, l \in LP_l, t \in T \quad (2.31)$$

### 2.3.3 Capacities

Each plant has capacity limitations for individual products for the entire time horizon as stated in constraints (2.32).

$$\sum_{u \in U_l} \sum_{i \in I_{ul}} PA_{iult} \leq PCAP_{lt} \quad \forall l \in LP_l, t \in T \quad (2.32)$$

Upper bounds on the number of scale-up decisions per each plant and for all plants are imposed by constraints (2.33) and (2.34).

$$\sum_{t \in T} \sum_{\substack{i \in II_i \\ i \notin IIL_{il}}} \sum_{u \in U_{il}} ypsc_{iult} \leq SUUB_l \quad \forall l \in LP_l \quad (2.33)$$

$$\sum_{t \in T} \sum_{l \in LP_l} \sum_{\substack{i \in II_i \\ i \notin IIL_{il}}} \sum_{u \in U_{il}} ypsc_{iult} \leq SUUBT \quad (2.34)$$

The number of assignments of products to plants is limited based on operational constraints through constraints (2.35).

$$\sum_{l \in LP_l} \sum_{u \in U_{iul}} yp_{iult} \leq MPPA_i \quad \forall i \in II_i, t \in T \quad (2.35)$$

### 2.3.4 Demand Satisfaction

The demands of each customer  $l$  are met according to constraints (2.36) – (2.38), which enforce that exactly one plant can sell a certain product to a customer.

$$\sum_{l \in LP_l} SL_{ill't} = DL_{ilt} \quad \forall i \in II_i \cup IB_i, l' \in LCI_{l'i}, t \in T \quad (2.36)$$

$$SL_{ill't} \leq ysl_{ill't} DL_{ilt} \quad \forall i \in II_i \cup IB_i, l \in LP_l, l' \in LCI_{l'i}, t \in T \quad (2.37)$$

$$\sum_{l \in LP_l} ysl_{ill't} = 1 \quad \forall i \in II_i \cup IB_i, l' \in LCI_{l'i}, t \in T \quad (2.38)$$

#### 2.3.4.1 Objective Function

The objective is to minimize the total cost, which includes the following terms:

- i. **Operating Costs**, given the unit costs,  $OPC_{ult}$ , for each time period;

$$OPC_{\text{Costs}} = \sum_{t \in T} \sum_{l \in LP_l} \sum_{i \in II_i} \sum_{u \in U_{il}} OPC_{ult} PA_{iult}$$

- ii. **Inbound Transportation Costs**, given the costs,  $ITPC_{il}$ ;

$$ITC_{\text{Costs}} = \sum_{t \in T} \sum_{l \in LP_l} \sum_{r \in R} ITPC_{rl} RM_{rlt}$$

- iii. **Outbound Transportation Costs**, given the costs,  $OTPC_{ill'}$ ;

$$OTC_{\text{Costs}} = \sum_{t \in T} \sum_{l \in LP_l} \sum_{i \in II_i} \sum_{l' \in LCI_{l'i}} OTPC_{ill'} SL_{ill't}$$

- iv. **Shipments Between Plants**, given the transportation costs,  $TCBP_{ll'}$ ;

$$PPC_{\text{Costs}} = \sum_{t \in T} \sum_{l \in LP_l} \sum_{i \in II_i} \sum_{l' \in LP_{l'}} TCBP_{ll'} IST_{ill't}$$

- v. **Individual Inventory Costs**, given the costs to store individual products,  $INVC_{ilt}$ ;

$$INVC_{\text{Costs}} = \sum_{t \in T} \sum_{l \in LP_l} \sum_{i \in II_i} INVC_{ilt} INV_{ilt}$$

- vi. **Blending Inventory Costs**, given the costs to store blended products,  $BINVC_{ilt}$ ;

$$BINVC_{\text{Costs}} = \sum_{t \in T} \sum_{l \in LP_l} \sum_{i \in IB_i} BINVC_{ilt} BINV_{ilt}$$

- vii. **Changeover Costs**, given the sequence-dependent changeover costs between groups,  $CC_{gg'ul}$ , and which are calculated using the same argument as in the changeover times, i.e., costs of changeovers within and across time periods minus the cost of the changeover represented by the broken link in the cyclic schedule;

$$CCosts = \sum_{t \in T} \sum_{l \in LP_l} \sum_{u \in U_l} \sum_{g \in G_{ul}} \sum_{g' \in G_{ul}} CC_{gg'ul} (zg_{gg'ult} - zzg_{gg'ult} + zzzg_{gg'ult})$$

- viii. **Scale-Up Costs**, given for example the penalty costs of having to introduce new products in a plant that does not normally produce them,  $SUC_{iul}$ ;

$$SUCosts = \sum_{t \in T} \sum_{l \in LP_l} \sum_{\substack{i \in II_i \\ i \notin IIL_{il}}} \sum_{u \in U_{il}} SUC_{iul} y p s c_{iult}$$

- ix. **Penalty for Maximum Storage Violation**, given the parameter  $PENINV$  that penalizes the maximum storage capacity;

$$PenInv = \sum_{t \in T} \sum_{l \in LP_l} \sum_{i \notin IR_i} PENINV \cdot INVslk_{ilt}$$

Therefore, the objective function is given by:

$$\begin{aligned} PlanCost = & OPCosts + ITCosts + OTCosts + PPCosts + INVCosts + \\ & BINVCosts + CCosts + SUCosts + PenInv \end{aligned} \quad (2.39)$$

### 2.3.5 Modified USGP Scheduling Model

We present two main enhancements to the Unit-Specific General Precedence (USGP) model proposed by [Kopanos, Láinez, & Puigjaner \(2009\)](#): the treatment of the number of batches as variable and the introduction of binary variables that model sequence-dependent changeovers across time periods. As opposed to the planning model, the scheduling model provides more detailed information concerning the timing and sequencing of individual products batches by accounting for all possible changeovers.

#### 2.3.5.1 Assignment Constraints

Constraints (2.40) imply that every unit in a given plant and time period must process at least one product.

$$\sum_{i \in I_{ul}} y p_{iult} \geq 1 \quad \forall u \in U_l, l \in LP_l, t \in T \quad (2.40)$$



### 2.3.5.2 Sequence-Dependent Changeovers Constraints

The planning model is an aggregate formulation based on the rigorous scheduling model and is only concerned with sequence-dependent changeovers between *groups* of products within and across adjacent periods. At the scheduling level, we compute the sequence of products rigorously. Therefore, the product-by-product changeover times,  $CT_{ii'ul}$ , and costs,  $CC_{ii'ul}$ , are computed based on the corresponding group-by-group parameters,  $CT_{gg'ul}$  and  $CC_{gg'ul}$ . The original USGP model defines two sets of 0-1 variables for sequence-dependent changeovers within time periods:  $zp_{ii'ult}$  (global or general precedence) and  $xp_{ii'ult}$  (local or immediate precedence). In this work, we introduce the 0-1 variables for sequence-dependent changeovers across time periods,  $wp_{ii'ult}$ .

**Sequence-Dependent Changeovers within Time Periods** The big-M constraints (2.41) ensure that the start time of a product is at least the end time of another product plus the changeover time between them.

$$Te_{iult} + xp_{ii'ult}CT_{ii'ul} \leq Ts_{i'ult} + H_t(1 - zp_{ii'ult}) \quad \forall (i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.41)$$

**Sequence-Dependent Changeovers across Time Periods** In order to model changeovers across time periods, we introduce two sets of 0-1 variables that identify the first,  $ypf_{iult}$ , and last,  $ypl_{iult}$ , products assigned to a given unit in a plant. Constraints (2.42) and (2.43) guarantee that a product can be the first or last of a unit if it is assigned to the same unit. Moreover, there can be exactly one first product and exactly one last product assigned to a given unit of a plant as represented in equations (2.44) and (2.45), respectively.

$$ypf_{iult} \leq yp_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.42)$$

$$ypl_{iult} \leq yp_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.43)$$

$$\sum_{i \in I_{ul}} ypf_{iult} = 1 \quad \forall u \in U_l, l \in LP_l, t \in T \quad (2.44)$$

$$\sum_{i \in I_{ul}} ypl_{iult} = 1 \quad \forall u \in U_l, l \in LP_l, t \in T \quad (2.45)$$

With respect to timing of operations, the main idea is to obtain the start and end times of each unit in each plant,  $TsU_{ult}$  and  $TeU_{ult}$  respectively, and ensure that the changeover time from the last product in a time period and the first product in the subsequent time period is taken into consideration (see Figure 2.6).

unit  $u$ , plant  $l$

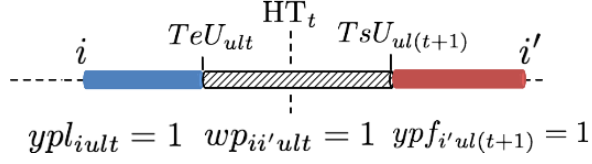


Figure 2.6: Changeovers (dashed cylinder) across time periods for modified USGP.

Constraints (2.46) ensure that the start time of product  $i'$  in time period  $t + 1$  is at least the end time in the previous time period  $t$  plus a changeover time in unit  $u$ . Similarly, constraints (2.47) guarantee that the start time in unit  $u$  in time period  $t + 1$  is at least the end time of product  $i$  in time period  $t$  plus a changeover time.

$$TeU_{ult} + wp_{ii'ult}CTp_{ii'ul} \leq Ts_{i'ul(t+1)} + H_{t+1}(1 - ypf_{i'ul(t+1)})$$

$$\forall(i, i') \in I_{ul}, u \in U_l, l \in LP_l, t \in T \setminus \{\bar{t}\} \quad (2.46)$$

$$TsU_{ul(t+1)} \geq Te_{iult} + wp_{ii'ult}CTp_{ii'ul} - H_t(1 - ypl_{iult})$$

$$\forall(i, i') \in I_{ul}, u \in U_l, l \in LP_l, t \in T \setminus \{\bar{t}\} \quad (2.47)$$

### 2.3.5.3 Sequencing-Allocation Constraints

The global sequencing variables,  $zp_{ii'ult}$ , are activated by the following constraints. The logic proposition (2.48) states that product  $i$  precedes product  $i'$  in unit  $u$  of plant  $l$  and time period  $t$  or product  $i'$  precedes product  $i$  in the same unit, plant, and period if and only if both products are assigned to the same unit.

$$ZP_{ii'ult} \vee ZP_{i'iult} \iff YP_{iult} \wedge YP_{i'ult}$$

$$\forall(i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.48)$$

The constraints correspondent to the above proposition are as follows:

$$zp_{ii'ult} \leq yp_{iult} \quad \forall(i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.49)$$

$$zp_{ii'ult} \leq yp_{i'ult} \quad \forall(i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.50)$$

$$zp_{i'iult} \leq yp_{iult} \quad \forall(i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.51)$$

$$zp_{i'iult} \leq yp_{i'ult} \quad \forall(i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.52)$$

$$zp_{ii'ult} + zp_{i'iult} \geq yp_{iult} + yp_{i'ult} - 1 \quad \forall(i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.53)$$

The local sequencing variables,  $xp_{ii'ult}$ , are activated through the Big-M constraints (2.54) and (2.55), which state that two products  $i$  and  $i'$  are consecutive only in the case that the binary variable  $zp_{ii'ult} = 1$  and when there

is no other product  $i''$  between products  $i$  and  $i'$ , and vice versa. The idea is to count how many products  $i''$ , where  $i'' \neq i \neq i'$ , are in between products  $i$  and  $i'$  and, thus, track the relative position of products that are assigned to a unit.

$$pos_{ii'ult} = \sum_{\substack{i'' \in I_{ul} \\ i'' \neq (i, i')}} (zp_{ii''ult} - zp_{i'i''ult}) + M(1 - zp_{ii'ult})$$

$$\forall (i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.54)$$

$$pos_{ii'ult} + xp_{ii'ult} \geq 1 \quad \forall (i, i') \in I_{ul}, i \neq i', u \in U_l, l \in LP_l, t \in T \quad (2.55)$$

where  $pos_{ii'ult}$  is a nonnegative continuous auxiliary “position” variable and  $M$  is a Big-M value, which could be the total number of products assigned to the respective unit, plant, and time period. Therefore, if  $pos_{ii'ult} = 0$ , that is there are no products between products  $i$  and  $i'$ , then we force  $xp_{ii'ult} = 1$  to ensure that product  $i$  immediately precedes product  $i'$ .

Lastly, the assignment variables relative to the first and last product for each time period are related to the sequencing variables for the changeovers across time periods as follows:

$$ypf_{i'ul(t+1)} = \sum_{i \in I_{ul}} wp_{ii'ult} \quad \forall i' \in I_{ul}, u \in U_l, l \in LP_l, t \in T \setminus \{\bar{t}\} \quad (2.56)$$

$$ypl_{iult} = \sum_{i' \in I_{ul}} wp_{ii'ult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \setminus \{\bar{t}\} \quad (2.57)$$

#### 2.3.5.4 Time Balance Constraints

Equations (2.58) state that the end time of product  $i$  in unit  $u$  of plant  $l$  in time period  $t$  is the sum of its start time and processing time.

$$Te_{iult} = Ts_{iult} + PT_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.58)$$

where  $PT_{iult}$  can be modeled as the following set of disjunctions:

$$\left[ \begin{array}{c} YP_{iult} \\ PT_{iult} = NB_{iult}/BPH_{iul} \end{array} \right] \vee \left[ \begin{array}{c} \neg YP_{iult} \\ PT_{iult} = 0 \end{array} \right] \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.59)$$

Given that there is a one-to-one correspondence between  $YP_{iult}$  (logical variables) and  $yp_{iult}$  (binary variables), the convex hull (Balas, 1985) reformulation yields the following linear constraints:

$$NB_{iult} = NB1_{iult} + NB2_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.60)$$

$$PT_{iult} = NB1_{iult}/BPH_{iul} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.61)$$

$$NB1_{iult} \leq yp_{iult}NBULP_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.62)$$

$$NB2_{iult} \leq (1 - yp_{iult})NBULP_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.63)$$

where  $NB1_{iult}$  and  $NB2_{iult}$  are disaggregated variables and  $NBULP_{iult}$  represents the number of batches of products obtained in the Upper Level Planning problem.

In addition, constraints (2.64) – (2.67) force the start and end times of a given unit in a time period to coincide with the start and end times of the first and last products, respectively.

$$TsU_{ult} \leq Ts_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.64)$$

$$TsU_{ult} \geq Ts_{iult} - H_t(1 - ypf_{iult}) \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.65)$$

$$TeU_{ult} \geq Te_{iult} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.66)$$

$$TeU_{ult} \leq Te_{iult} + H_t(1 - ypl_{iult}) \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.67)$$

Finally, constraints (2.68) and (2.69) ensure that the start and end times lie within the current time period length.

$$Ts_{iult} \geq HT_{t-1} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \setminus \{1\} \quad (2.68)$$

$$Te_{iult} \leq HT_t \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.69)$$

#### 2.3.5.5 Changeover Costs within Time Periods

Given the changeover costs for each unit, the costs associated with the changeovers within time periods are given by:

$$CC1 = \sum_{t \in T} \sum_{l \in LP_l} \sum_{u \in U_l} \sum_{i \in I_{ul}} \sum_{i' \in I_{ul}} xp_{ii'ult} CCp_{ii'ul} \quad (2.70)$$

#### 2.3.5.6 Changeover Costs across Adjacent Time Periods

Similarly, the changeover costs across adjacent time periods are calculated as follows:

$$CC2 = \sum_{t \in T} \sum_{l \in LP_l} \sum_{u \in U_l} \sum_{i \in I_{ul}} \sum_{i' \in I_{ul}} wp_{ii'ult} CCp_{ii'ul} \quad (2.71)$$

#### 2.3.5.7 Material Balances

The constraints for material and inventory balances are equivalent for the planning model, equations (2.20) and (2.25) – (2.38), with the only difference being the variable number of batches, i.e., equation (2.20) becomes:

$$PA_{iult} = NB1_{iult} BS_{iul} \quad \forall i \in I_{ul}, u \in U_l, l \in LP_l, t \in T \quad (2.72)$$

#### 2.3.5.8 Objective Function

The objective function considers the same terms as in the ULP model except for the changeover costs and it can be written as follows (see [Subsubsection 2.3.4.1](#) for definitions of other terms):

$$\begin{aligned} \text{SchedCost} = & \text{OPCosts} + \text{ITCosts} + \text{OTCosts} + \text{PPCosts} + \text{INVCosts} + \\ & \text{BINVCosts} + CC1 + CC2 + \text{SUCosts} + \text{PenSales} \end{aligned} \quad (2.73)$$

## 2.4 Decomposition Strategies

### 2.4.1 Bilevel Decomposition

In order to solve both the planning and scheduling problems simultaneously for medium-to-large problem instances in practical time, a decomposition scheme can be employed. One approach for decomposing this problem is the Bilevel Decomposition (BD) consisting of an Upper Level Planning (ULP) and a Lower Level Scheduling (LLS) problems that yield lower and upper bounds on the total cost, respectively (Erdirik-Dogan & Grossmann, 2008). The problems are solved iteratively until the relative difference between the lower and upper bounds is less than a pre-specified tolerance. Integer inequalities or cuts are added to the upper level problem to ensure the generation of new schedules, and/or to avoid infeasible ones at the lower level problem. A schematic of this strategy is shown in Figure 2.7.

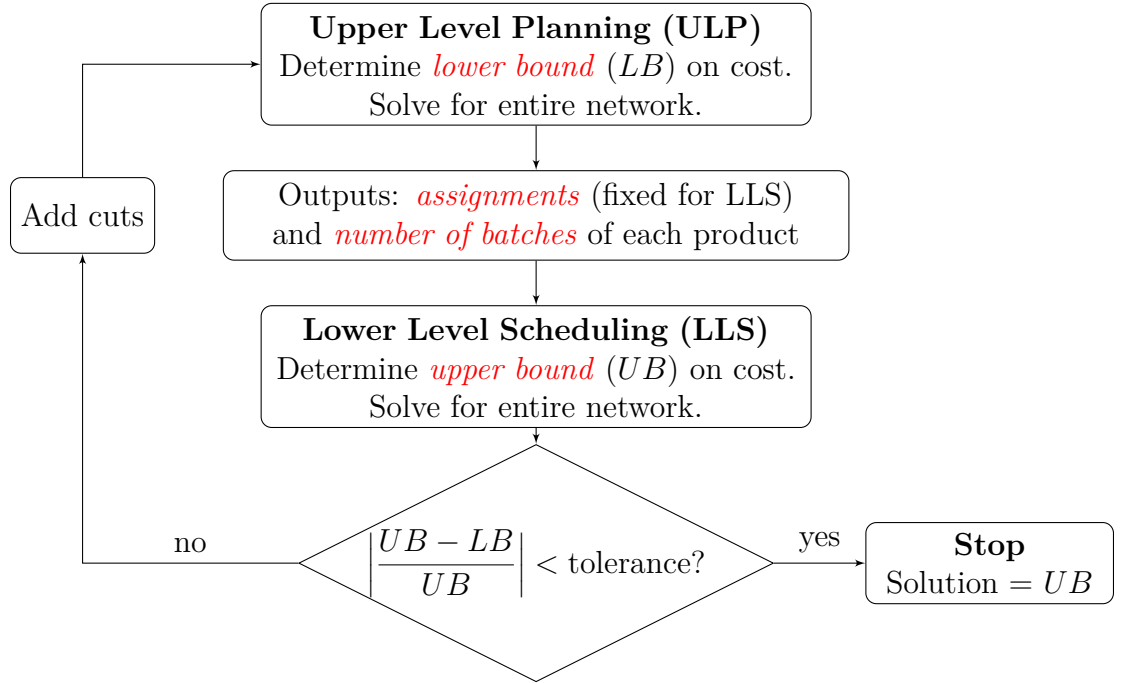


Figure 2.7: Bilevel Decomposition scheme for integrated planning and scheduling.

The main motivation behind the BD is that the ULP problem is less complex and, thus, easier to solve than the LLS problem. Therefore, by solving the aggregate and simplified planning problem and iteratively attempting to match its solution with the one obtained with a rigorous and detailed scheduling model, we strive to arrive at a feasible production plan without the burden of solving the *full-space* scheduling model.

In addition to computing a lower bound on the total cost, the ULP problem determines which products may be processed in a given unit and also the

number of batches of each product on every unit and time period, which will be inputs to the LLS problem. Hence, the scheduling problem at the lower level is solved only for the product assignments predicted by the upper level, which may significantly reduce the number of constraints and variables in the LLS problem.

It may be advantageous to add different integer cuts to the ULP problem after each iteration, such as superset and subset cuts (You, Grossmann, & Wassick, 2011) and symmetry-breaking cuts (Erdirik-Dogan & Grossmann, 2008). However, from our experience gained from solving the present problem, at most two iterations were required to converge the ULP and LLS problems to less than 1% relative error even for large problems. Hence, the integer cuts to exclude the assignments at the previous BD iteration take the form:

$$\sum_{(i,u,l,t) \in B_{iult}^k} yp_{iult} - \sum_{(i,u,l,t) \in N_{iult}^k} yp_{iult} \leq |B_{iult}^k| - 1 \quad k \in BD_k \quad (2.74)$$

where  $k$  is the index of BD iterations that are contained in set  $BD_k$  (maximum of 10 iterations were used in all computational experiments),  $B_{iult}^k = \{(i, u, l, t) \in I_{iul} \times U_l \times LP_l \times T : yp_{iult} = 1 \text{ at iteration } k - 1\}$  and  $N_{iult}^k = \{(i, u, l, t) \in I_{iul} \times U_l \times LP_l \times T : yp_{iult} = 0 \text{ at iteration } k - 1\}$ .

## 2.4.2 Lagrangean Decomposition

Problems with multiperiod formulations or with variables associated with spatially distributed entities, such as plants and customers, are candidates that can undergo Lagrangean Decomposition (LD), which is a special case of Lagrangean Relaxation (LR) where the original problem can be decomposed into subproblems with common variables by splitting them first, and then *dualizing* the copy constraints. In this work, since the ULP can become the bottleneck as it has to be solved for the entire network, unlike the LLS that can be solved for each plant, we decompose the ULP problem using LD inside the BD loop discussed in the previous section. This proved to be computationally advantageous when solving the large-scale industrial problem.

Two types of LD have been explored in the planning and scheduling literature: Temporal Lagrangean Decomposition (TLD) and Spatial Lagrangean Decomposition (SLD). As the names imply, the former is characterized by decomposing the original problem into subproblems corresponding to each time period, whereas the latter is decomposed by the spatially distributed sites. Due to the presence of binary variables in MILP problems, i.e., a source of non-convexity, there is a *duality gap* between the solution of the Lagrangean dual problem and the primal problem. Moreover, it has been shown by Terrazas-Moreno, Trotter, & Grossmann (2011) that the temporal dual bound is at least as tight as the spatial dual bound. Therefore, in this work we focus on the TLD only.

Figure 2.8 shows a schematic of TLD. Each time period corresponds to an independent subproblem that can be solved in parallel with the other subproblems. The variables  $z_t$  in this figure represent inventory variables,  $INV_{ilt}$  and

$BINV_{ilt}$ , and the assignment variables  $ygf_{gult}$  in the ULP formulation because they appear in terms in time period  $t$  and  $t + 1$ , that is they link consecutive time periods. Since we need to separate time periods into unique subproblems, we introduce the *copy variables*  $\overline{INV}_{ilt} = INV_{ilt}$ ,  $\overline{BINV}_{ilt} = BINV_{ilt}$ , and  $\overline{ygf}_{gult} = ygf_{gult}$ . The next step is to dualize these equations by multiplying each of them by the respective Lagrange multipliers  $\lambda INV_{ilt}$ ,  $\lambda BINV_{ilt}$  and  $\lambda ygf_{gult}$ , respectively, and adding the terms to the objective function.

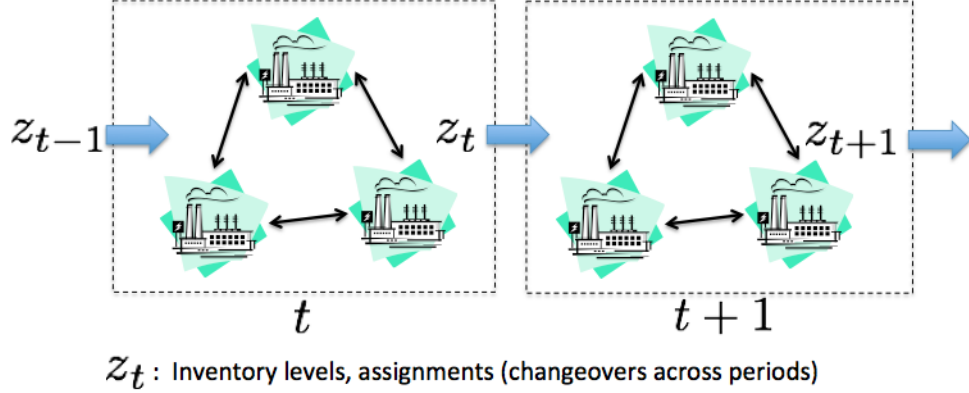


Figure 2.8: Schematic of Temporal Lagrangean Decomposition (TLD).

The objective function of the planning problem is augmented as follows (see equation (2.39) for the other individual terms in the objective function):

$$\begin{aligned} \text{PlanCost}^{\text{TLD}}(\lambda) = & \sum_{t \in T} \text{PlanCost}_t + \sum_{i \in II_i} \sum_{l \in LP_l} \sum_{t \in T} \lambda INV_{ilt} \cdot INV_{ilt} + \\ & \sum_{i \in IB_i} \sum_{l \in LP_l} \sum_{t \in T} \lambda BINV_{ilt} \cdot BINV_{ilt} + \\ & \sum_{l \in LP_l} \sum_{u \in U_l} \sum_{g \in G_{ul}} \sum_{t \in T} \lambda ygf_{ilt} \cdot ygf_{gult} \end{aligned} \quad (2.75)$$

For fixed multipliers, the planning model is separable into time periods and can be written as follows:

$$\begin{aligned} \min \text{PlanCost}_t^{\text{TLD}}(\lambda) = & \text{PlanCost}_t + \\ & \sum_{i \in II_i} \sum_{l \in LP_l} \left[ \lambda INV_{ilt} \cdot INV_{ilt} + \lambda INV_{il(t-1)} \cdot \overline{INV}_{il(t-1)} \right] \\ & \sum_{i \in IB_i} \sum_{l \in LP_l} \left[ \lambda BINV_{ilt} \cdot BINV_{ilt} - \lambda BINV_{il(t-1)} \cdot \overline{BINV}_{il(t-1)} \right] + \\ & \sum_{l \in LP_l} \sum_{u \in U_l} \sum_{g \in G_{ul}} \left[ \lambda ygf_{gul(t+1)} \cdot \overline{ygf}_{gul(t+1)} - \lambda ygf_{gult} \cdot ygf_{gult} \right] \end{aligned} \quad (2.76)$$

and subject to constraints (2.1) – (2.15), (2.17) – (2.27), (2.29), (2.31) – (2.35), and the following constraints, which are rewritten by including the copy vari-

ables:

$$\begin{aligned}
\overline{ygf}_{g'ul(t+1)} &= \sum_{g \in G_{ul}} zzzg_{gg'ult} & \forall g' \in G_{ul}, u \in U_l, l \in LP_l, \\
& & t \in T \setminus \{\bar{t}\} \quad (2.77) \\
INV_{ilt} &= \overline{INV}_{il(t-1)} + STR_{ilt} & \forall i \in II_i, l \in LP_l, t \in T \\
& & (2.78) \\
BINV_{ilt} &= \overline{BINV}_{il(t-1)} + BA_{ilt} - \sum_{\nu \in LCI_{l'i}} SL_{ill't} & \forall i \in IB_i, l \in LP_l, t \in T \\
& & (2.79)
\end{aligned}$$

At the end of a TLD iteration, the multipliers must be updated for the next iteration. Different strategies have been proposed, such as cutting plane (Kelly, 1960), subgradient (Held, Wolfe, & Crowder, 1974), boxstep (Marsten, Hogan, & Blankenship, 1975), bundle (Lemaréchal, 1974), analytic center cutting plane methods (Goffin, Haurie, & Vial, 1992), and volume algorithm (Barahona & Anbil, 2000). In this work, we use the subgradient method in which the update formula for a multiplier  $\lambda X^p$  at iteration  $p \in TLD_p$  of the TLD and associated with the equality constraint of generic variable  $X$  is given by:

$$\lambda X^{p+1} = \lambda X^p + \epsilon^p \frac{(UB^{\text{TLD}} - LB^{\text{TLD}})}{den^p} (X^p - \bar{X}^p) \quad (2.80)$$

where  $\epsilon^p$  is the step size that can be modified at each iteration according to some criterion and usually lies in the interval  $(0, 2]$ ,  $UB^{\text{TLD}}$  is the best upper bound in the TLD scheme up to iteration  $p$  obtained from solving the original planning problem in the reduced space (i.e., by fixing variables from the solution of the Lagrangean subproblems),  $LB^{\text{TLD}}$  is the best lower bound in the TLD scheme up to iteration  $p$  obtained from summing all the optimal objective function values of the Lagrangean subproblems, and  $den^p$  is the sum of squared of the differences between the original variable  $X^p$  and its copy  $\bar{X}^p$  and for the planning formulation is defined as follows:

$$\begin{aligned}
den^p &= \sum_{t \in T} \left[ \sum_{i \in II_i} \sum_{l \in LP_l} (INV_{ilt}^p - \overline{INV}_{ilt}^p)^2 + \sum_{i \in IB_i} \sum_{l \in LP_l} (BINV_{ilt}^p - \overline{BINV}_{ilt}^p)^2 + \right. \\
&\quad \left. \sum_{l \in LP_l} \sum_{u \in U_l} \sum_{g \in G_{ul}} (ygf_{gult}^p - \overline{ygf}_{gult}^p)^2 \right] \quad \forall p \in TLD_p
\end{aligned}$$

Lastly, the multipliers may be initialized prior to the first TLD iteration to zero or to other values, for example the marginal values of the respective equations from the relaxed TLD model (integrality conditions are relaxed). From our experience, the best choice of initial values may vary among different problems. The next section contains the full description of the algorithm implemented when TLD is applied to the ULP problem within the BD loop.



### 2.4.3 Hybrid BD-TLD Scheme

The two decomposition methods, BD and TLD, are combined in an effort to solve large-scale industrial problems. Figure 2.9 shows the schematic of the hybrid approach.

The main steps of the hybrid decomposition scheme is given in Figure 2.10. We underscore the advantage of creating smaller and independent subproblems through TLD, because they can be solved in shorter time as compared to the full problem and also in parallel. Particularly in this work, since each subproblem corresponds to a time period, it means that it is possible to solve all subproblems at the same time in a multi-core computer instead of solving each subproblem one at a time. To our knowledge, parallel “solve statements” have become supported by two of the major modeling platforms, GAMS (Brooke, Kendrick, & Meeraus, 2015) and AIMMS (Roelofs & Bisschop, 2015). We used GAMS to implement our models and algorithms. For more information on how to solve problems in parallel in GAMS, see the website: [http://interfaces.gams-software.com/doku.php?id=the\\_gams\\_grid\\_computing\\_facility](http://interfaces.gams-software.com/doku.php?id=the_gams_grid_computing_facility).

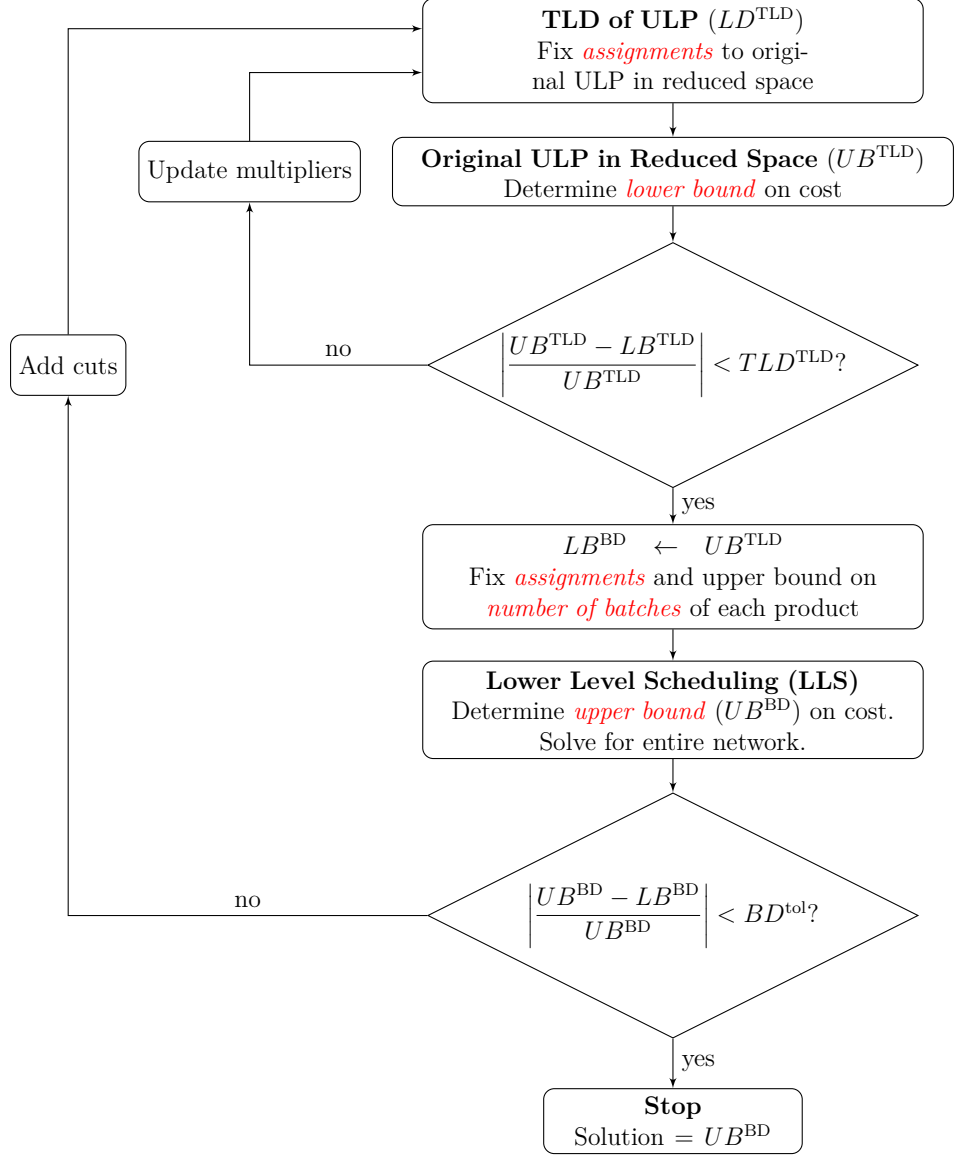


Figure 2.9: Hybrid Bilevel-Temporal Lagrangean Decomposition scheme for integrated planning and scheduling.

```

Data:  $BD_k^{\max}$ : maximum number of iterations in BD loop
         $TLD_k^{\max}$ : maximum number of iterations in TLD loop
         $BD^{\text{tol}}$ : convergence tolerance for BD loop
         $TLD^{\text{tol}}$ : convergence tolerance for TLD loop

/* BD loop */
while  $k \leq BD_k^{\max}$  do
    /* TLD loop */
    while  $p \leq TLD_k^{\max}$  do
        Solve TLD subproblems (in parallel);
        Solve original ULP problem with fixed decisions from TLD
        subproblems;
        Update the best upper ( $UB^{\text{TLD}}$ ) and lower ( $LB^{\text{TLD}}$ ) bounds if
        necessary;
        Calculate gap between best upper and lower bounds:
         $gap^{\text{TLD}} = \left| \frac{UB^{\text{TLD}} - LB^{\text{TLD}}}{UB^{\text{TLD}}} \right|$ ;
        if  $gap^{\text{TLD}} > TLD^{\text{tol}}$  then
            Change step size if necessary, e.g.:  $\epsilon^{p+1} = 0.95\epsilon^p$ ;
            Update multipliers using the subgradient method;
        else
            TLD converged to given tolerance;
            break;
        end
    end
    Store ULP problem objective function value  $\rightarrow U\text{sol}_k$ ;
    Solve LLS problem only for assignments predicted by ULP
    problem;
    Store LLS problem objective function value  $\rightarrow L\text{sol}_k$ ;
    Calculate gap between ULP and LLS objective function values:
     $gap^{\text{BD}} = \left| \frac{L\text{sol}_k - U\text{sol}_k}{L\text{sol}_k} \right|$ ;
    if  $gap^{\text{BD}} > BD^{\text{tol}}$  then
        Add cuts to ULP problem;
    else
        BD converged to given tolerance;
        break;
    end
end

```

Figure 2.10: Algorithm for the hybrid Bilevel-Temporal Lagrangean Decomposition scheme.

## 2.5 Computational Results

Three example problems of increasing size and complexity were solved to demonstrate the efficiency of the decomposition approaches discussed in the previous section. *Example 1* is a small-scale problem for which we provide

all data as well as present the optimal Gantt chart obtained. *Example 2* is a medium-scale problem and *Example 3* is a large-scale industrial problem. All examples were solved using BD and the full-space scheduling model. We used the hybrid BD-TLD algorithm only to solve Example 3.

All models were implemented in GAMS 23.8.2 and solved with Gurobi 4.6.1. All computational experiments were performed in a Dell PowerEdge T410 server with 6 Intel® Xeon® 2.67 GHz CPUs (total 12 threads), 16 GB of RAM, and running Ubuntu Server 12.04 LTS. The Gurobi's option `threads 0` was enabled for all computational runs, which means that all threads were used for parallel processing. The maximum allowed wall time for all problems was 24 hours.

### 2.5.1 Example 1

Example 1 considers a small-scale problem, and it allows us to analyze not only the computational benefit of Bilevel Decomposition (BD), but also to gain insight on the optimal schedule represented by a Gantt chart. All the data are given in [Appendix A.1](#) and the characteristics of Example 1 are as follows:

- 2 plants
  - Plant P1: Units U11 and U12
  - Plant P2: Unit U21
- 9 products (4 blended)
- 3 raw materials
- 3 customers
- Time horizon: 4 weeks
- Time period: weekly
- All models were solved to optimality

[Figure 2.11](#) shows the optimal schedule represented by a Gantt chart for the first time period (week) only, where the letters inside the colored blocks represent the products and the numbers within parentheses denote their number of batches. The same result was obtained after solving both the LLS and the full-space scheduling problems. In both units U12 and U21, the last sequence-dependent changeovers occur across the first time period. We emphasize that the number of batches (integer variables) were optimized simultaneously with the planning and scheduling decisions in each model.

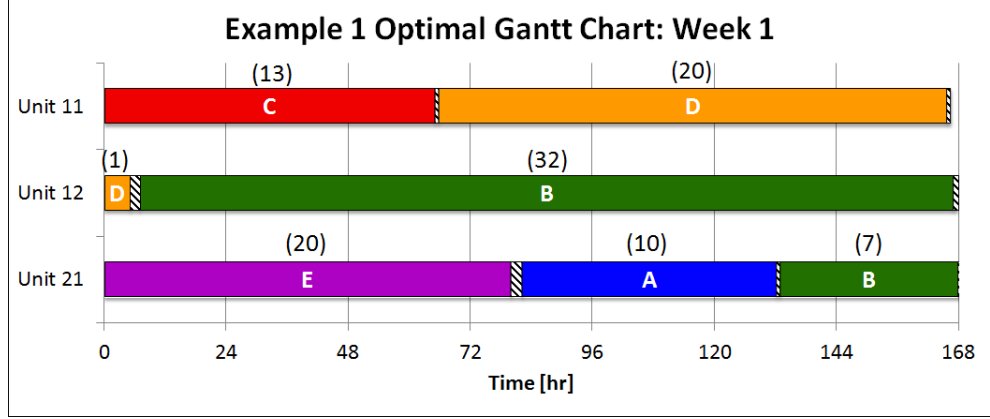


Figure 2.11: Optimal schedule in Example 1. The blocks with a black-and-white downward diagonal pattern represent sequence-dependent changeovers.

Table 2.1 shows the objective function breakdown for all models. Notice that the ULP problem predicted lower changeover costs than the scheduling models due to the aggregation of products into groups of products in the former model so that changeovers between products belonging to the same group are neglected. Moreover, the solution obtained for the LLS model was numerically the same as the one obtained for the full-space scheduling model denoted as “FS”, which may be expected for small and “well-posed” problems. The BD gap between the LLS and the ULP solutions was 0.071% and only one BD iteration was needed to achieve convergence.

Table 2.1: Cost breakdown for Example 1.

Costs	BD		FS
	ULP	LLS	
<b>Total</b>	<b>833,828.23</b>	<b>834,416.23</b>	<b>834,416.23</b>
Operating	38,003.57	38,003.57	38,003.57
Inbound	409,179.52	409,179.52	409,179.52
Outbound	100,022.50	100,022.50	100,022.50
Plant-to-Plant	12,809.52	12,809.52	12,809.52
Inventory	9,347.11	9,347.11	9,347.11
Changeovers	1,966.00	2,554.00	2,554.00
Scale-Up	262,500.00	262,500.00	262,500.00

The problems sizes as well as computational statistics are shown in Table 2.2. The row “Nodes” indicates how many tree nodes in the branch-and-cut method were explored by the solver. Using BD, the total wall time was 1.17 seconds as opposed to solving the full-space (FS) scheduling model, which took 44.98 seconds. Notice that the LLS problem contains fewer variables and constraints than the FS problem, which facilitated its solution yielding only 29 nodes explored in the reduced space as opposed to nearly 95,000 nodes in the full space.

Table 2.2: Problems sizes for Example 1.

	BD		FS
	ULP	LLS	
Discrete Variables	528	507	936
Continuous Variables	925	1,039	1,201
Constraints	1,412	1,726	2,924
Non-Zero Elements	4,537	5,049	9,113
Nodes	5,015	29	94,929
Wall Time [s]	0.99	0.18	44.98

### 2.5.2 Example 2

Example 2 is a medium-scale problem. We compared the effectiveness of applying only the BD with solving the FS problem. The computational benefit of using BD becomes more evident as the problem grows in size and complexity as will be shown in the results below. The characteristics of Example 2 are as follows:

- 3 plants
  - Plant P1: 2 units
  - Plant P2: 1 unit
  - Plant P3: 3 units
- 66 products (16 blended)
- 20 raw materials
- 99 customers
- Time horizon: 6 months
- Time period: monthly
- All models were solved to 0.5% optimality gap

Table 2.3 shows the objective function breakdown for all models. Notice that the final solution of the BD, i.e., the solution to the LLS problem yields a lower total cost than the one obtained solving the FS problem. Also, no shipments between plants were observed in any of the problems. The BD gap between the LLS and the ULP solutions was 0.2% and only one BD iteration was needed to achieve such convergence.

Table 2.3: Cost breakdown for Example 2.

Costs	BD		FS
	ULP	LLS	
<b>Total</b>	<b>5,202,979.79</b>	<b>5,212,454.79</b>	<b>5,222,056.24</b>
Operating	233,495.60	233,495.60	234,284.40
Inbound	3,736,349.69	3,736,349.69	3,744,609.18
Outbound	507,477.50	507,477.50	505,980.67
Plant-to-Plant	—	—	—
Inventory	41,782.00	41,782.00	41,782.00
Changeovers	8,875.00	18,350.00	20,400.00
Scale-Up	675,000.00	675,000.00	675,000.00

The problems sizes as well as computational statistics are shown in [Table 2.4](#). Using BD, the total wall time was 3.96 seconds for a gap of 0.2% as opposed to solving the full-space (FS) scheduling model, which took approximately 3 hours and 25 minutes for 0.5% optimality gap. The difference in terms of number of variables and constraints between performing the BD and not decomposing the problem becomes considerably more expensive as the problem instance increases. That is directly translated into 0 nodes necessary to solve the LLS problem compared to nearly 60,000 nodes explored in the solution of the FS problem.

Table 2.4: Problems sizes for Example 2.

	BD		FS
	ULP	LLS	
Discrete Variables	6,328	4,412	128,400
Continuous Variables	52,783	53,047	95,563
Constraints	43,169	45,378	437,649
Non-Zero Elements	145,009	145,831	3,998,885
Nodes	57	0	57,536
Wall Time [s]	2.34	1.62	12,228.94

### 2.5.3 Example 3

Example 3 is a large-scale industrial problem. We identified the need to decompose the ULP problem and use the hybrid BD-TLD algorithm in order to get a solution in practical time. The following characteristics provide an idea of the problem size of Example 3. Exact characteristics are not provided due to confidentiality reasons:

- More than 5 plants
- More than 10 reactor units with more multiple reactor units in each plant
- More than 200 products and raw materials

- Hundreds of customers
- Time horizon: 12 months
- Time period: monthly
- All models were solved to 2% optimality gap

Table 2.5 shows the objective function breakdown for all models except the FS problem, which could not be solved due to the excessive RAM required. A maximum of 30 TLD iterations were enforced. The BD gap between the LLS and the ULP solutions was 0.13% and only one BD iteration was needed to achieve such convergence.

Table 2.5: Cost breakdown for Example 3.

Costs	BD-TLD	
	ULP	LLS
<b>Total</b>	<b>9,532,583.65</b>	<b>9,545,149.23</b>
Operating	1,696,826.08	1,691,107.90
Inbound	5,658,562.23	5,656,563.11
Outbound	840,872.58	844,039.17
Plant-to-Plant	24,345.55	24,520.93
Inventory	191,110.56	185,084.79
Changeovers	12,533.33	31,333.33
Scale-Up	1,108,333.33	1,112,500.00

The problems sizes as well as computational statistics are shown in Table 2.6. Using hybrid BD-TLD, the total wall time was around 1 hour and 15 minutes. Using only the BD, the wall times were 6 hours and 12 minutes to solve the ULP problem and 1 hour and 26 minutes to solve the LLS problem (7 hours and 38 minutes total). It can be noted that the decomposition in the ULP problem allowed us to obtain a solution with significantly less computation time. We show the size of the FS problem even though we could not solve it. It is clear that decomposition approaches enable tackling real-world problems.



Table 2.6: Problems sizes for Example 3.

	BD-TLD		FS
	ULP*	LLS	
Discrete Variables	119,397	228,701	6,726,779
Continuous Variables	834,195	898,119	3,138,985
Constraints	590,810	1,140,007	22,895,121
Non-Zero Elements	2,206,546	6,836,510	648,785,966
Nodes	0	0	N/A
Wall Time [s]	4,070.48**	452.53	N/A

\* Last ULP problem solved in the TLD loop

\*\* Total time for the TLD loop (30 iterations)

Figure 2.12 shows the evolution of the upper and lower bounds on the objective function value obtained when applying TLD to the ULP problem. The best lower and upper bounds after 30 iterations were \$9,363,323.00 and \$9,440,580.00, which represent a final TLD gap of 0.82%.

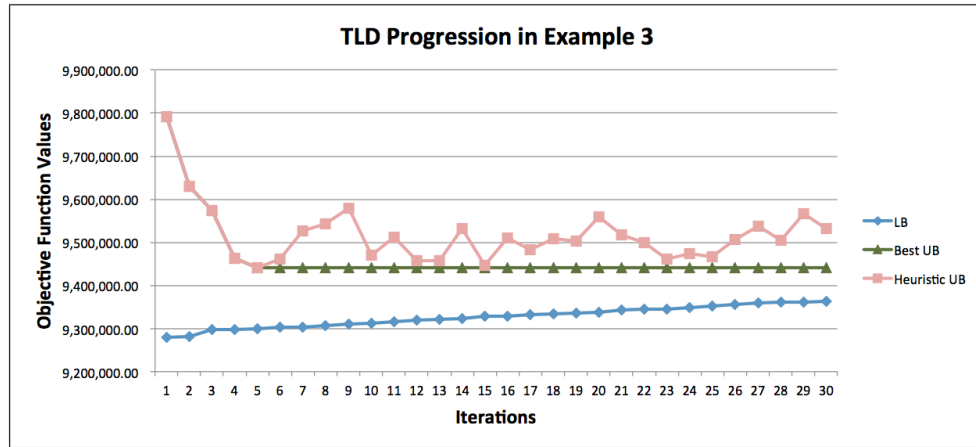


Figure 2.12: Lower bound (LB), best upper bound (Best UB) and heuristic upper bound (Heuristic UB) on the objective function value using Temporal Lagrangean Decomposition in the Upper Level Planning problem in Example 3. The Heuristic UB is obtained by fixing the assignment variables,  $yp_{iult}$ , calculated by each Lagrangean subproblem in the full-space planning model.

## 2.6 Conclusions

In this work, we developed a model for the integration of planning and scheduling in the operation of a network of multiproduct batch plants. Each plant contains single-stage processing units in parallel and the products are classified into groups or families. The planning model not only incorporates production and capacity constraints, but also estimates the sequencing of groups of products through Traveling Salesman Problem (TSP) constraints. The scheduling formulation employs continuous time representation in a precedence-based

framework. It solves the batching and the scheduling problem simultaneously, i.e., the number of batches is a decision variable in the model, and allows sequence-dependent changeovers across time periods. Both models are multi-period Mixed-Integer Linear Programs (MILPs).

The integration of planning and scheduling was performed via two decomposition methods: Bilevel and Lagrangean. For small to medium problems, it was observed that Bilevel Decomposition (BD) represents an attractive and rigorous decomposition strategy to solve them in practical computational time. However, when attempting to solve a real-world problem, the planning level became computationally expensive and, thus, became amenable to further decomposition. We performed Temporal Lagrangean Decomposition (TLD) in the planning level in an inner loop of a BD framework to obtain a solution in a reasonable time.

The computational results of the examples have shown that detailed model formulations applied to large problems can be tackled through efficient modeling decomposition strategies and advances in computing, such as in optimization solvers capabilities, multi-core computer architecture and parallel computing.

# Chapter 3

## Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching

### 3.1 Introduction

The importance of accounting for uncertainty in mathematical optimization was recognized in its early days in the seminal and influential paper by George B. Dantzig ([Dantzig, 1955](#)). Two of the current popular optimization frameworks that incorporate uncertainty in the modeling stage are Robust Optimization ([Ben-Tal, Ghaoui, & Nemirovski, 2009](#)) and Stochastic Programming ([Birge & Louveaux, 2011](#)). In this chapter, we focus on Stochastic Programming (SP) and address the issue of scenario generation.

To illustrate the many possible sources of uncertainty in Process Systems Engineering (PSE), consider as an example a production planning problem for a network of chemical plants. Planning decisions usually span multiple time periods and generally involve, but are not limited to determining the amount of raw materials to be purchased by each plant, the production and inventory levels at each plant, the transportation of intermediate and finished products between different locations, and meeting the forecast demand. It is clear that all those decisions may be subject to some kind of uncertainty. For instance, the availability of a key raw material may be uncertain, and/or the demands of the products. Another example is the possibility of mechanical failure of pieces of equipment in a plant or its complete unplanned shutdown, which affects the entire network. A review on optimization methods with exogenous uncertainties can be found in [Sahinidis \(2004\)](#).

A central aspect of Stochastic Programming is the definition of *scenarios*, which describe possible values that the uncertain parameters or stochastic processes may take. Applications in PSE that make explicit use of scenarios expand multiple areas and time scales. Some representative examples are: dy-

namic optimization (Abel & Marquardt, 2000), scheduling (Guillén, Espuña, & Puigjaner, 2006; Colvin & Maravelias, 2009; Pinto-Varela, Barbosa-Povoa, & Novais, 2009), planning (Sundaramoorthy, Evans, & Barton, 2012; Li & Ierapetritou, 2011; You, Wassick, & Grossmann, 2009; Gupta & Grossmann, 2012), and synthesis and design (Kim, Realff, & Lee, 2011; Chen, Adams II, & Barton, 2011). The most common assumption made in the works listed before is that the scenario tree is *given* (probabilities and values of uncertain parameters at every node are known). That is, the “true” probability distributions are known, and the uncertainty typically is characterized by arbitrary deviations from some average value based on minimum and maximum values (for instance: low, medium, and high values with probabilities arbitrarily chosen).

Researchers have also developed decomposition algorithms to tackle large-scale and real-world instances that originate from explicitly considering scenarios in optimization problems. We argue that it is equally important to generate scenario trees that satisfactorily capture the uncertainty in a given problem, as the quality of the solution to the SP problem is directly influenced by the accuracy of the scenarios. Therefore, it is important to apply systematic scenario generation methods instead of making assumptions that may be questionable. King & Wallace (2012) wrote an excellent book on the challenges of optimization modeling under uncertainty. The authors also discuss the importance of generating meaningful scenarios (see Chapter 4), as modeling with SP results in a framework with practical and robust decision-making capability.

These data-driven approaches to optimization problems have become common in the Operations Research and Management Science communities, and are an example of what is called *Business Analytics* (BA) (Bartlett, 2013). After the data collection and management phase, BA leverages data analysis to make analytics-based decisions that can be divided into three general layers: descriptive (querying and reporting, databases), predictive (forecasting and simulation), and prescriptive (deterministic and stochastic optimization) (Davenport & Harris, 2007). The data-driven scenario generation method described in this chapter can be linked with the descriptive and predictive layers, and then used for decision making in the prescriptive layer.

It is worth noting that, even though not usually regarded as a scenario generation method, the Sample Average Approximation (SAA) method (Kleywegt, Shapiro, & Homem-de-Mello, 2001; Shapiro, 2006) can be used to approximate the continuous probability distribution assumed for the uncertain parameters. Specifically, the distributions are sampled, for instance via Monte Carlo sampling, and the expected value function is approximated by the corresponding sample average function, which is repeatedly solved until some convergence criterion is met. The size of the sample must be such that a degree of confidence on the final objective function value is satisfied. In addition, the sampling step becomes more complicated in Multi-Stage SP (MSSP), as *conditional* sampling is required for the SAA method to produce *consistent* estimators. Conditioning on previous events also plays a key role in the moment

matching method as discussed later in the chapter.

The main contribution of this chapter is the consideration of the (empirical) cumulative distribution function as additional information to be matched when generating scenario trees. This mitigates the typical *under-specification shortcoming* of the Moment Matching Problem (MMP) that has been proposed (Høyland & Wallace, 2001). We call this extended formulation the Distribution Matching Problem (DMP). We first present the  $L^2$ -,  $L^1$ -, and  $L^\infty$ -norm MMP formulations, and then describe their extended versions and how to use them for the generation of two- and multi-stage scenario trees with the aid of time series forecasting.

This chapter is organized as follows. Section 3.2 introduces the moment matching method as a systematic method to generate two-stage scenario trees. In Section 3.3, we propose extensions to each MMP formulation by considering (empirical) cumulative distribution function data in order to mitigate the under-specification shortcoming of MMP models. The new formulations (DMP) and methodology are illustrated with a numerical example for the optimal production planning of a network of chemical plants. Section 3.4 extends the methodology to the multi-stage case; the role of modeling stochastic processes is emphasized and two approaches are described based on nonlinear and linear programming (NLP and LP) statistical property matching formulations for generating multi-stage scenario trees. The approaches are illustrated with a numerical example, and conclusions are drawn in Section 3.5.

## 3.2 Two-Stage Scenario Tree Generation

Scenario trees are an *approximate discretized* representation of the uncertainty in the data (Kaut, 2003). They are based on discretized probability distributions to model the stochastic processes. The scenario trees are approximate because they contain a restricted number of outcomes in order to avoid the integration of continuous distribution functions. However, the size of the scenario trees directly impacts the computational complexity of SP models.

The concerns raised in the above paragraph have motivated the search for methodologies that can be used to systematically generate two-stage scenario trees. Two main classes of methods can be identified: scenario generation and scenario reduction. In this section, we focus on scenario generation methods, in particular, the **moment matching method**, which was originally proposed by Høyland & Wallace (2001) and is described as follows. Given an initial structure of the tree, i.e., number of nodes per stage, it determines at each node the values for the random variables and their probabilities by solving a nonlinear programming (NLP) problem. The NLP problem minimizes the weighted squared error between statistical properties calculated from the outcomes or nodes, and the same properties calculated directly from the data. Thus, it is based on an  $L^2$ -norm formulation. If the absolute deviations from the target properties are minimized as proposed by Ji *et al.* (2005), then an  $L^1$ -norm formulation can be employed, which has the advantage that it can

be cast as a Linear Programming (LP) problem. A new formulation of the MMP based on the  $L^\infty$ -norm. can also be developed. Examples of statistical properties are the first four moments (expected value, variance, skewness, and kurtosis), covariance or correlation matrix, quantiles, etc.

It is important to note that the scenario tree generated by the moment matching method is used as an “input” to the SP model. Therefore, the uncertain parameters as well as the probabilities of the outcomes (or scenarios) in the SP model become decision variables in the scenario generation problem.

In this section, we review MMP formulations for two-stage problems.

### 3.2.1 $L^2$ Moment Matching Problem

In the Moment Matching Problem (MMP), the uncertain parameters of the SP model and the probabilities of the outcomes become variables in a nonlinear optimization formulation. The purpose of the MMP is to find the optimal values for the random variables  $x_j$  and probabilities  $p_j$  (see Figure 3.1) of a pre-specified structure for the scenario tree that minimize the error between the statistical properties calculated from the tree and the ones calculated directly from the data.

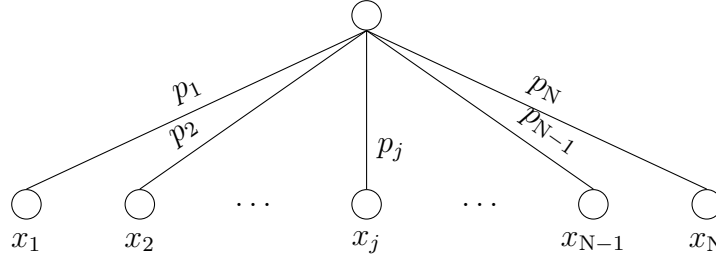


Figure 3.1: Two-stage scenario tree for one uncertain parameter.

In the  $L^2$  formulation, the squared error is employed in the objective function. Hence, the NLP formulation can be generically written as follows:

$$\begin{aligned}
\min_{x, p} \quad & \sum_{s \in S} w_s \cdot (f_s(x, p) - \text{Sval}_s)^2 \\
\text{s.t.} \quad & \sum_{j=1}^N p_j = 1 \\
& p_j \in [0, 1] \quad \forall j = 1, \dots, N
\end{aligned} \tag{3.1}$$

where  $x$  is a vector of random variables (uncertain parameters of the SP model),  $p$  is a vector of probabilities of outcomes,  $s \in S$  is a statistical property to be matched (target),  $w_s$  is the weight for statistical property  $s$ ,  $f_s(\cdot, \cdot)$  is the mathematical expression of statistical property  $s$  calculated from the tree,  $\text{Sval}_s$  is the value of statistical property  $s$  (target value) that characterizes the distribution of the data.

Any statistical property that somehow describes the data can be used to measure how well the scenario tree represents them. Descriptive statistics

provides measures that can be used to summarize and inform us about the probability distribution of the data. Four of these measures, called moments, (Papoulis, 1991), are the following: mean or expectation, and the central moments variance, skewness, and kurtosis. The mean or expectation tells us about the *average* value in a data set, the variance is a measure of the *spread* of the data about the mean, the skewness is a measure of the *asymmetry* of the data, and the kurtosis is a measure of the *thickness* of the tails of the shape of the distribution of the data.

A more detailed definition of the  $L^2$  MMP is as follows. The uncertain data are indexed by  $i \in I$ , which denotes the entity of an uncertain parameter (for example, a product).  $N$  denotes the number of outcomes per node at the second stage,  $j \in J = \{1, 2, \dots, N\}$  denotes the branches (outcomes) from the root node, and  $k \in K = \{1, 2, 3, 4\}$  is the index of the first four moments. The decision variables are the uncertain parameters of the stochastic programming problem,  $x_{i,j}$ , with corresponding probabilities of outcomes,  $p_j$ . The moments calculated from the tree are denoted by variables  $m_{i,k}$  and the ones calculated from the data are denoted by parameters  $M_{i,k}$ . Finally, the second co-moment, i.e., covariance, calculated between entity  $i$  and  $i'$  from the tree and the data are denoted by  $c_{i,i'}$  and  $C_{i,i'}$ , respectively. The  $L^2$  MMP formulation is given as follows (see Gülpınar, Rustem, & Settergren (2004)). The goal is to generate a tree in which we determine the values of  $x_{i,j}$  and  $p_j$  whose properties match those calculated from the data ( $M_{i,k}$  and, if applicable,  $C_{i,i'}$ ).

( $L^2$  MMP):

$$\min_{x, p} z_{\text{MMP}}^{L^2} = \sum_{i \in I} \sum_{k \in K} w_{i,k} (m_{i,k} - M_{i,k})^2 + \sum_{\substack{(i, i') \in I \\ i < i'}} w_{i,i'} (c_{i,i'} - C_{i,i'})^2 \quad (3.2a)$$

s.t.

$$\sum_{j=1}^N p_j = 1 \quad (3.2b)$$

$$m_{i,1} = \sum_{j=1}^N x_{i,j} p_j \quad \forall i \in I \quad (3.2c)$$

$$m_{i,k} = \sum_{j=1}^N (x_{i,j} - m_{i,1})^k p_j \quad \forall i \in I, k > 1 \quad (3.2d)$$

$$c_{i,i'} = \sum_{j=1}^N (x_{i,j} - m_{i,1})(x_{i',j} - m_{i',1}) p_j \quad \forall (i, i') \in I, i < i' \quad (3.2e)$$

$$x_{i,j} \in [x_{i,j}^{\text{LB}}, x_{i,j}^{\text{UB}}] \quad \forall i \in I, j = 1, \dots, N \quad (3.2f)$$

$$p_j \in [0, 1] \quad \forall j = 1, \dots, N \quad (3.2g)$$

where the weighted squared error between the statistical properties calculated from the tree and inferred from the data is minimized in (3.2a). Constraints

(3.2b) ensure that the probabilities of outcomes add up to 1, (3.2c) represent the calculation of the first moment (mean), constraints (3.2d) represent the calculation of higher-order central moments, constraints (3.2e) are the expressions for the covariance, and  $w_{i,k} = w'_{i,k}/M_{i,k}^2$  and  $w_{i,i'} = w'_{i,i'}/C_{i,i'}^2$ , where  $w'_{i,k}$  and  $w'_{i,i'}$  are weights, which can be chosen arbitrarily. The bounds on the decision variables  $x$  and  $p$  are represented in constraints (3.2f) and (3.2g), respectively.

**Remark 1.** Skewness,  $Skew$ , and kurtosis,  $Kurt$ , are by definition normalized properties:

$$Skew_i = \frac{\sum_{j=1}^N (x_{i,j} - m_{i,1})^3 p_j}{\sigma_i^3} \quad \forall i \in I$$

$$Kurt_i = \frac{\sum_{j=1}^N (x_{i,j} - m_{i,1})^4 p_j}{\sigma_i^4} \quad \forall i \in I$$

where  $\sigma_i^2 = m_{i,2}$  is the variance as defined in equation (3.2d) for  $k = 2$ . Therefore, in order to use constraints (3.2d) for  $k > 2$  in the  $L^2$  MMP, the statistical properties calculated from the data have to be denormalized.

**Remark 2.** Before solving the  $L^2$  MMP, the number of branches or outcomes from the root node,  $N$ , is pre-specified. Høyland & Wallace (2001) suggest the rule  $(|I| + 1)N - 1 \sim \text{number of statistical specifications}$ , where  $|I|$  is the number of random variables. The authors also discuss potential over- and under-specification that may arise from choosing a value for  $N$ . The other inputs or parameters to the  $L^2$  MMP are the values of the statistical properties to be matched. They directly affect the quality of the tree obtained. Hence, care should be exercised to obtain those properties in a meaningful way, so that the scenario tree effectively captures the uncertainty in the data.

**Remark 3.** The use of covariance or correlation information enables one to capture the *linear* dependence between multiple sources of uncertainty. More sophisticated and rigorous ways, such as copulas, to model dependency of distributions in a multivariate structure have been employed in a few papers, for instance, Sutiene & Pranevicius (2007); Kaut (2013).

The NLP problem in equation (3.2) is nonconvex and its degree of nonlinearity and nonconvexity increases when attempting to match higher moments. As expected, initialization plays an important role in such optimization problems. Therefore, local NLP solvers may encounter numerical difficulties and get trapped in *poor* local solutions. Systematic *multi-start methods* can be used with local NLP solvers to help overcome the problems aforementioned by sampling multiple starting points in the feasible region and solving the NLP problem using each different starting point; however, it must be recognized that multi-start methods are not a panacea and there is no guarantee of systematically obtaining a global (or near global) solution to the MMP. Finally,



deterministic global optimization solvers, such as BARON (Tawarmalani & Sahinidis, 2005), SCIP (Achterberg, 2009), and COUENNE (Belotti *et al.*, 2009), can also be used although at considerable computational expense if the NLP is not small in size.

### 3.2.2 $L^1$ and $L^\infty$ Moment Matching Problems

If the absolute value of the deviations from the target moments and covariances are minimized, then the MMP becomes an  $L^1$ -norm model as proposed by Ji *et al.* (2005). A well-known reformulation of the nondifferentiable absolute value function in the definition of the objective function consists in splitting the variable in its argument into two non-negative variables, which correspond to the positive and negative values of the original variable.

The  $L^1$  formulation of the MMP is then as follows. Partition the moment and covariance variables,  $m_{i,k}$  and  $c_{i,i'}$ , respectively, into their positive and negative parts  $m_{i,k}^+$ ,  $m_{i,k}^-$ ,  $c_{i,i'}^+$ , and  $c_{i,i'}^-$ . Thus, the  $L^1$  MMP is given by:

( $L^1$  MMP):

$$\min_{x, p} z_{\text{MMP}}^{L^1} = \sum_{i \in I} \sum_{k \in K} w_{i,k} (m_{i,k}^+ + m_{i,k}^-) + \sum_{\substack{(i, i') \in I \\ i < i'}} w_{i,i'} (c_{i,i'}^+ + c_{i,i'}^-) \quad (3.3a)$$

s.t.

$$\sum_{j=1}^N p_j = 1 \quad (3.3b)$$

$$\sum_{j=1}^N x_{i,j} p_j + m_{i,1}^+ - m_{i,1}^- = M_{i,1} \quad \forall i \in I \quad (3.3c)$$

$$\sum_{j=1}^N (x_{i,j} - \sum_{j'=1}^N x_{i,j'} p_{j'})^k p_j + m_{i,k}^+ - m_{i,k}^- = M_{i,k} \quad \forall i \in I, k > 1 \quad (3.3d)$$

$$\sum_{j=1}^N (x_{i,j} - \sum_{j'=1}^N x_{i,j'} p_{j'}) (x_{i',j} - \sum_{j'=1}^N x_{i',j'} p_{j'}) p_j + c_{i,i'}^+ - c_{i,i'}^- = C_{i,i'} \quad \forall (i, i') \in I, \quad i < i' \quad (3.3e)$$

$$m_{i,k}^+, m_{i,k}^- \geq 0 \quad \forall i \in I, k \in K \quad (3.3f)$$

$$c_{i,i'}^+, c_{i,i'}^- \geq 0 \quad \forall (i, i') \in I, \quad i < i' \quad (3.3g)$$

$$x_{i,j} \in [x_{i,j}^{\text{LB}}, x_{i,j}^{\text{UB}}] \quad \forall i \in I, \quad j = 1, \dots, N \quad (3.3h)$$

$$p_j \in [0, 1] \quad \forall j = 1, \dots, N \quad (3.3i)$$

where the weighted absolute deviations between the statistical properties calculated from the tree and inferred from the data are minimized in (3.3a). Constraints (3.3b) ensure that the probabilities of outcomes add up to 1, (3.3c) attempts to match the first moment (mean), constraints (3.3d) represent the matching of higher-order central moments, constraints (3.3e) attempt to match the covariance, and  $w_{i,k} = |w'_{i,k}/M_{i,k}|$  and  $w_{i,i'} = |w'_{i,i'}/C_{i,i'}|$ , where  $w'_{i,k}$  and  $w'_{i,i'}$  are weights that can be arbitrarily chosen. The bounds on the variables  $x, p, m^+, m^-, c^+$ , and  $c^-$  are represented by constraints (3.3f) – (3.3i).

Another way of formulating the MMP is through the minimization of the  $L^\infty$ -norm of the deviations with respect to the targets.

( $L^\infty$  MMP):

$$\min_{x, p} \quad z_{\text{MMP}}^{L^\infty} = \mu + \gamma \quad (3.4a)$$

s.t.

Constraints (3.3b) – (3.3i)

$$\mu \geq w_{i,k} m_{i,k}^+ \quad \forall i \in I, k \in K \quad (3.4b)$$

$$\mu \geq w_{i,k} m_{i,k}^- \quad \forall i \in I, k \in K \quad (3.4c)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^+ \quad \forall (i, i') \in I, i < i' \quad (3.4d)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^- \quad \forall (i, i') \in I, i < i' \quad (3.4e)$$

where  $\mu$  and  $\gamma$  are scalar variables that account for the maximum deviations in the moments and covariances, respectively.

### 3.2.2.1 Linear Programming $L^1$ and $L^\infty$ MMPs

The  $L^2$ ,  $L^1$ , and  $L^\infty$  MMPs shown in equations (3.2) to (3.4), respectively, are nonlinear and nonconvex due to the mathematical expressions for the moments since both probabilities and node values are decision variables. [Ji et al. \(2005\)](#) used ideas from Linear Goal Programming and proposed an LP formulation for the  $L^1$  MMP in which only probabilities are decision variables. In this LP formulation, the node values are generally obtained via some simulation approach. For time-dependent data, such as asset returns in financial portfolio management applications, a time-series model is used to forecast future expected values and possibly higher moments. Multiple values above and below the forecast expected value can be used as the node values or outcomes in the  $L^1$  and  $L^\infty$  LP MMP formulations and the probabilities of each outcome are left as the decision variables. In PSE applications, uncertain parameters that typically have a time component are product demand and market price.

Let  $x_{i,j}$  be a parameter with the value of the uncertain parameter that can be arbitrarily chosen or calculated from some simulation procedure, for example simulation of time-series forecasting models. As long as there are at least two values, for example  $x_{i,j}$  and  $x_{i,j'}$ , that are *symmetric* with respect to the mean, then the expected value can always be matched (see Proposition 1 in [Ji et al. \(2005\)](#)) and the  $L^1$  LP MMP is given as follows:

( $L^1$  LP MMP):

$$\min_p z_{\text{LP MMP}}^{L^1} = \sum_{i \in I} \sum_{k \in K \setminus \{1\}} w_{i,k} (m_{i,k}^+ + m_{i,k}^-) + \sum_{\substack{(i, i') \in I \\ i < i'}} w_{i,i'} (c_{i,i'}^+ + c_{i,i'}^-) \quad (3.5a)$$

s.t.

$$\sum_{j=1}^N p_j = 1 \quad (3.5b)$$

$$\sum_{j=1}^N x_{i,j} p_j = M_{i,1} \quad \forall i \in I \quad (3.5c)$$

$$\sum_{j=1}^N (x_{i,j} - M_{i,1})^k p_j + m_{i,k}^+ - m_{i,k}^- = M_{i,k} \quad \forall i \in I, k > 1 \quad (3.5d)$$

$$\sum_{j=1}^N (x_{i,j} - M_{i,1})(x_{i',j} - M_{i',1}) p_j + c_{i,i'}^+ - c_{i,i'}^- = C_{i,i'} \quad \forall (i, i') \in I, i < i' \quad (3.5e)$$

$$m_{i,k}^+, m_{i,k}^- \geq 0 \quad \forall i \in I, k \in K \quad (3.5f)$$

$$c_{i,i'}^+, c_{i,i'}^- \geq 0 \quad \forall (i, i') \in I, i < i' \quad (3.5g)$$

$$p_j \in [0, 1] \quad \forall j = 1, \dots, N \quad (3.5h)$$

Likewise, the  $L^\infty$  LP MMP can be formulated as follows:

( $L^\infty$  LP MMP):

$$\min_p z_{\text{LP MMP}}^{L^\infty} = \mu + \gamma \quad (3.6a)$$

s.t.

Constraints (3.5b) – (3.5h)

$$\mu \geq w_{i,k} m_{i,k}^+ \quad \forall i \in I, k \in K \quad (3.6b)$$

$$\mu \geq w_{i,k} m_{i,k}^- \quad \forall i \in I, k \in K \quad (3.6c)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^+ \quad \forall (i, i') \in I, i < i' \quad (3.6d)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^- \quad \forall (i, i') \in I, i < i' \quad (3.6e)$$

We note that the LP formulations do not necessarily yield the same solution as the NLP formulations. The NLP formulations have more degrees of freedom (both vectors  $x$  and  $p$  are simultaneously optimized), whereas the LP formulations are used to calculate the vector of probabilities for fixed values of outcomes.

Obviously, it may be more advantageous to solve an LP problem instead of a nonconvex NLP problem. For multi-stage stochastic problems with time-dependent uncertain parameters, the solution strategy is much more complex when applying the NLP model instead of the LP formulation. Sections 3.4.1 and 3.4.2 contain details of multi-stage scenario tree generation using the aforementioned NLP and LP formulations, respectively.

### 3.2.3 Remarks on the MMP Formulations

Regardless of the  $L^2$ ,  $L^1$ , and  $L^\infty$  MMP formulations, it has been our experience that it is common to have *under-specified* NLP and LP problems when only moments are matched. This is due to the fact that not enough information to be matched (statistical properties) is provided to achieve non-degenerate solutions. The consequences are that multiple choices for the node values and/or probabilities yield the same objective function value. In other words, multiple trees with the same number of nodes and having very different node values and (sometimes zero) probabilities satisfy the specifications. In addition, we observed that the Lagrange multipliers associated with all constraints in the models are zero or very small at the optimal solution obtained by local and global solvers. Moreover, the distribution obtained from solving the MMPs does not exhibit a similar shape as the distribution of the data even when up to four moments were matched.

More formally, we can perform an analysis on the well-posedness of the MMP formulations for which both node values ( $x_{i,j}$ ) and probabilities ( $p_j$ ) are variables. The number of variables is  $|I| \cdot N + N - 1 = N(|I| + 1) - 1$ . The number of data points ( $|K|$  moments and covariances, or conditions to be matched) is  $|I| \cdot |K| + \frac{|I|(|I|-1)}{2}$ . The MMP is well-posed if the number of data points is at least the number of variables, i.e.,

$$|I| \cdot |K| + \frac{|I|(|I| - 1)}{2} \geq N(|I| + 1) - 1 \quad (3.7)$$

For example, for one uncertain parameter ( $|I| = 1$ ) and four moments ( $|K| = 4$ ), we have that  $N \leq 2.5$ , meaning we can only have a tree with two scenarios for a well-posed MMP formulation. If we increased  $N$  to 3 or more, then this would result in an under-specified MMP model.

To mitigate the under-specification of MMP formulations, we propose including additional statistical properties to be matched in order to avoid solving an under-specified MMP formulation, and to ensure that the shape of the distribution of the data is captured in the solution. This is also motivated by the fact that in certain applications it may not be practical to obtain accurate estimates of higher moments as a large amount of data is needed. Consequently, fewer moments may be matched based on their availability, while still capturing the shape of distribution of data with the scenario tree. Lastly, our numerical experiments demonstrate that the same solution vector is achieved by local and global solvers. That is, only one tree satisfies the specifications,

although theoretically there is no guarantee that this property holds true due to nonconvexity in the NLP models.

### 3.3 Distribution Matching Problem

A new formulation—Distribution Matching Problem (DMP)—based on the MMP is proposed that not only attempts to match marginal moments, but also the marginal Empirical Cumulative Distribution Function (ECDF) of the data as explained in this section. Specifically, we propose to extend the  $L^2$ ,  $L^1$ ,  $L^\infty$  MMPs, and the  $L^1$  and  $L^\infty$  LP MMPs in order to also match an approximation to the Empirical Cumulative Distribution Function (ECDF) of the data. Before describing the steps of the algorithm to incorporate the ECDF information into the optimization models, some definitions are presented.

For a given random variable (r.v.)  $Z$ , the probability of  $Z$  to take on a value, say  $z$ , less than or equal to some value  $t$  is given by the Cumulative Distribution Function (CDF), or mathematically  $CDF(t)$ . A CDF is associated with a specific Probability Density Function (PDF), for continuous random variables (r.vs.), or Probability Mass Function (PMF), for discrete r.vs. In order to avoid making assumptions about the distribution model, an estimator of the CDF can be used, the Empirical CDF (ECDF), which is defined as follows (van der Vaart, 1998):

$$ECDF(t) = \frac{1}{n} \sum_{i=1}^N \mathbf{1}\{z_i \leq t\} \quad (3.8)$$

where  $n$  is the sample size and  $\mathbf{1}\{A\}$  is the indicator function of event  $A$ , that takes the value of one if event  $A$  is true, or zero otherwise. Therefore, given a value  $t$ , the ECDF returns the ratio between the number of elements in the sample that are less than or equal to  $t$  and the sample size.

Every CDF has the following properties:

- It is monotonically non-decreasing;
- It is right-continuous;
- $\lim_{x \rightarrow -\infty} CDF(x) = 0$ ; and
- $\lim_{x \rightarrow +\infty} CDF(x) = 1$ .

We note that most CDFs are sigmoidal. Therefore, the ECDF, as an estimator of the CDF, is also “S-shaped” in most cases. Hence, in order to incorporate the ECDF data in the optimization models in a *smooth* way, we propose fitting the Generalized Logistic Function (GLF) (Richards, 1959), also known as Richards’ Curve, or a simplified version (for instance, the Logistic Function is a special case of the GLF). The GLF is defined as follows:

$$GLF(x) = \beta_0 + \frac{\beta_1 - \beta_0}{(1 + \beta_2 e^{-\beta_3 x})^{1/\beta_4}} \quad (3.9)$$

where  $\beta_0$ ,  $\beta_1$ ,  $\beta_2$ ,  $\beta_3$ , and  $\beta_4$  are parameters to be estimated. When fitting the GLF to ECDF data, the GLF can be simplified by setting  $\beta_0 = 0$  and  $\beta_1 = 1$  as these parameters correspond to the lower and upper asymptotes, respectively. Analytical expressions for the partial derivatives of  $GLF(x)$  with respect to its parameters can be derived and used to form the Jacobian matrix for least-squares fitting purposes.

The algorithm for generating a two-stage scenario tree, where the uncertain parameters have no time-series effect, by matching moments and ECDF is described as follows:

**Step 1:** Collect data for the (independent) uncertain parameters and obtain individual ECDF curves for each data set.

**Step 2:** Approximate each ECDF curve obtained by fitting the Generalized Logistic Function (GLF) or a simplified version.

**Step 3:** Solve a Distribution Matching Problem (DMP) defined in equations (3.10), (3.11), or (3.12).

**Remark.** We note that if a particular probability distribution family is assumed, i.e., a parametric approach is taken, then CDF information rather than ECDF data can be used in the DMP. This avoids the extra step of fitting a smooth curve to the ECDF data. However, very few distribution families have closed-form expressions for the CDF. Thus, approximate formulas have to be used in order to avoid evaluating integrals in the DMP.

Extended versions of the three MMP formulations for Step 3 are presented as follows. Note that since ECDF information is taken into account, we add  $|I| \cdot N$  conditions that reduce the problem of under-specification. Furthermore, we note that we must ensure that the values of the nodes in the tree are ordered, i.e., order statistics (Hogg, McKean, & Craig, 2012). The convention adopted is the following:  $x_{i,1} \leq x_{i,2} \leq \dots \leq x_{i,N}$ , which is ensured via additional inequalities in each extended NLP model. Because the node values are ordered, the summation  $\sum_{j'=1}^j p_{j'}$  represents the cumulative probability of the node value  $x_{i,j}$ .

( $L^2$  DMP):

$$\min_{x, p} \quad z_{\text{DMP}}^{L^2} = z_{\text{MMP}}^{L^2} + \sum_{i \in I} \sum_{j=1}^N \omega_{i,j} \delta_{i,j}^2 \quad (3.10a)$$

s.t.

Constraints (3.2b) – (3.2g)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j} \quad \forall i \in I, j = 1, \dots, N \quad (3.10b)$$

$$x_{i,j} \leq x_{i,j+1} \quad \forall i \in I, j = 1, \dots, N-1 \quad (3.10c)$$

where the variables  $\delta_{i,j}$  represent the deviations with respect to the ECDF data, which in turn are approximated by, for example, the GLF and is represented by the expression  $\widehat{ECDF}(x_{i,j})$ . In addition to minimizing the weighted square errors from matching (co-)moments, the sum of squares of the deviations  $\delta_{i,j}$  is also minimized with given weights  $\omega_{i,j}$  that can be chosen relative to the weights for the term involving the moments. Thus, the weights represent a trade-off between matching sample (co-)moment data and a smooth representation of the (E)CDF.

( $L^1$  DMP):

$$\min_{x, p} \quad z_{\text{DMP}}^{L^1} = z_{\text{MMP}}^{L^1} + \sum_{i \in I} \sum_{j=1}^N \omega_{i,j} (\delta_{i,j}^+ + \delta_{i,j}^-) \quad (3.11a)$$

s.t.

Constraints (3.3b) – (3.3i)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (3.11b)$$

$$x_{i,j} \leq x_{i,j+1} \quad \forall i \in I, j = 1, \dots, N-1 \quad (3.11c)$$

where the variables  $\delta_{i,j}^+$  and  $\delta_{i,j}^-$  represent the positive and negative deviations with respect to the ECDF data, respectively. The expression  $\widehat{ECDF}(x_{i,j})$  represents the approximation to the ECDF data obtained by, for example, fitting the GLF. The weights to the deviations are given by  $\omega_{i,j}$ .

( $L^1$  LP DMP):

$$\min_p \quad z_{\text{LP DMP}}^{L^1} = z_{\text{LP MMP}}^{L^1} + \sum_{i \in I} \sum_{j=1}^N \omega_{i,j} (\delta_{i,j}^+ + \delta_{i,j}^-) \quad (3.12a)$$

s.t.

Constraints (3.5b) – (3.5h)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (3.12b)$$

The *constant* expression  $\widehat{ECDF}(x_{i,j})$  represents the approximation to the ECDF data obtained by, for example, fitting the GLF. Note that it is required that the vector of node values is ordered, that is,  $x_{i,j} \leq x_{i,j+1}$  for  $j = 1, \dots, N-1$ .



( $L^\infty$  DMP):

$$\begin{aligned} \min_{x, p} \quad & z_{\text{DMP}}^{L^\infty} = z_{\text{MMP}}^{L^\infty} + \xi \\ \text{s.t.} \end{aligned} \quad (3.13a)$$

Constraints (3.3b) – (3.3i) and (3.4b) – (3.4e)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (3.13b)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^+ \quad \forall i \in I, j = 1, \dots, N \quad (3.13c)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (3.13d)$$

$$x_{i,j} \leq x_{i,j+1} \quad \forall i \in I, j = 1, \dots, N-1 \quad (3.13e)$$

( $L^\infty$  LP DMP):

$$\begin{aligned} \min_p \quad & z_{\text{LP DMP}}^{L^\infty} = z_{\text{LP MMP}}^{L^\infty} + \xi \\ \text{s.t.} \end{aligned} \quad (3.14a)$$

Constraints (3.5b) – (3.5h) and (3.6b) – (3.6e)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (3.14b)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^+ \quad \forall i \in I, j = 1, \dots, N \quad (3.14c)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (3.14d)$$

where  $\xi$  is a scalar variable that accounts for the maximum deviations in the ECDF information.

If a parametric approach to the distribution family is taken, then the term  $\widehat{ECDF}(\cdot)$  can be substituted by an exact closed-form expression, represented by  $CDF(\cdot)$ , or an approximate formula, denoted by  $\widetilde{CDF}(\cdot)$ , and no curve fitting is needed.

**Remark.** We compare the well-posedness of the DMP formulations in this section to the expression derived in equation (3.7) for the MMP formulations in Section 3.2. The number of variables is the same,  $N(|I| + 1) - 1$ , but the number of data points (conditions to be matched) is increased by  $|I| \cdot N$  to  $|I| \cdot |K| + \frac{|I|(|I|-1)}{2} + |I| \cdot N$ . Thus, for the case where both the node values and

probabilities are variables, the DMP formulations are well-posed when

$$|I| \cdot |K| + \frac{|I|(|I| - 1)}{2} \geq N - 1 \quad (3.15)$$

In other words, the right-hand side of equation (3.7) for MMP formulations is reduced by  $|I| \cdot N$ , meaning that the number of outcomes  $N$  (or scenarios) can be increased. For example, for one uncertain parameter ( $|I| = 1$ ) and four moments ( $|K| = 4$ ), we have that  $N \leq 5$  instead of  $N \leq 2.5$  for MMP formulations. Thus, the additional information matched by the DMP model eliminates the under-specification present in the MMP formulations. Consequently, the proposed extension allows the modeler to specify more outcomes without leading to under-specified or ill-posed problems.

The distribution matching method is illustrated in the following example in which the objective is to determine the optimal production plan of a network of chemical facilities or plants. For simplicity, the only uncertain parameter considered is the production yield of one facility in the network. The example demonstrates the impact that selecting a scenario tree has on the quality of the solution of the stochastic model.

### 3.3.1 Example 1: Uncertain Plant Yield

Figure 3.2 shows the network of the example used throughout the chapter. It consists of a raw material  $A$ , an intermediate product  $B$ , finished products  $C$  and  $D$  (only product  $D$  can be stored), and facilities (plants)  $P1$ ,  $P2$ , and  $P3$ . Product  $C$  can also be purchased from a supplier, or in the case of multiple sites, it could be transferred from another site that also produces it.

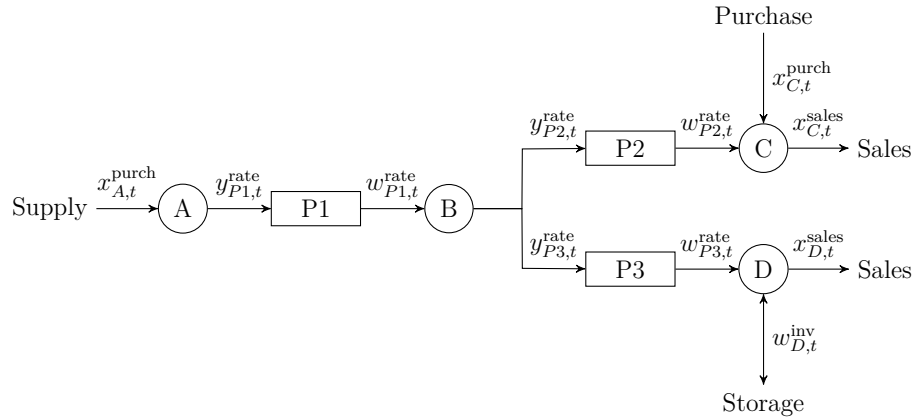


Figure 3.2: Network structure for the motivating Example 1.

The production planning Linear Programming (LP) model has the following main elements: variables corresponding to the inlet/outlet flow rates to/from facility  $f$  in time period  $t$ ,  $y_{f,t}^{rate}$  and  $w_{f,t}^{rate}$  respectively; production yields for each facility,  $\theta_f$ ; and demands for each finished product  $m \in FP$  in

time period  $t$ ,  $\xi_{m,t}$ . The deterministic multiperiod optimization model is given as follows:

$$\max w^{\text{profit}} \quad (3.16a)$$

$$\text{s.t. } w_{f,t}^{\text{rate}} = \theta_f y_{f,t}^{\text{rate}} \quad \forall f \in F, t \in T \quad (3.16b)$$

$$x_{C,t}^{\text{sales}} = w_{P2,t}^{\text{rate}} + x_{C,t}^{\text{purch}} \quad \forall t \in T \quad (3.16c)$$

$$w_{D,t}^{\text{inv}} = w_{D,t-1}^{\text{inv}} + w_{P3,t}^{\text{rate}} - x_{D,t}^{\text{sales}} \quad \forall t \in T \quad (3.16d)$$

$$w_{P1,t}^{\text{rate}} = y_{P2,t}^{\text{rate}} + y_{P3,t}^{\text{rate}} \quad \forall t \in T \quad (3.16e)$$

$$x_{A,t}^{\text{purch}} = y_{P1,t}^{\text{rate}} \quad \forall t \in T \quad (3.16f)$$

$$x_{m,t}^{\text{sales}} + \text{slack}_{m,t}^{\text{sales}} = \xi_{m,t} \quad \forall m \in FP, t \in T \quad (3.16g)$$

$$w_{f,t}^{\text{rate}} \leq w_{f,t}^{\text{rate,max}} + \text{slack}_{f,t}^{\text{max,cap}} \quad \forall f \in F, t \in T \quad (3.16h)$$

$$w_{f,t}^{\text{rate}} \geq w_{f,t}^{\text{rate,min}} - \text{slack}_{f,t}^{\text{min,cap}} \quad \forall f \in F, t \in T \quad (3.16i)$$

$$w_{D,t}^{\text{inv}} \leq w_{D,t}^{\text{inv,max}} \quad \forall t \in T \quad (3.16j)$$

$$x_{A,t}^{\text{purch}} \leq x_{A,t}^{\text{purch,max}} \quad \forall t \in T \quad (3.16k)$$

where constraints (3.16b) relate the output flows with the input flows through the yield of each facility  $f$ , constraints (3.16c) – (3.16f) represent material and inventory balances, equations (3.16g) represent the demand satisfaction and slack variables are employed to account for possible unmet demand, constraints (3.16h) – (3.16k) are limitations in the flows, storage, raw material availability, and capacity violations, respectively, and the profit is calculated as follows:

$$\begin{aligned} w^{\text{profit}} = & \sum_{t \in T} \left[ \sum_{m \in FP} \text{SP}_{m,t} x_{m,t}^{\text{sales}} - \sum_{f \in F} \text{OPC}_{f,t} w_{f,t}^{\text{rate}} - \right. \\ & \sum_{m \in M: MPUR=1} \text{PC}_{m,t} x_{m,t}^{\text{purch}} - \sum_{m \in M: MINV=1} \text{IC}_{m,t} w_{m,t}^{\text{inv}} - \sum_{m \in FP} \text{PEN}_{m,t} \text{slack}_{m,t}^{\text{sales}} \\ & \left. - \sum_{f \in F} \text{PEN}_{f,t} (\text{slack}_{f,t}^{\text{max,cap}} + \text{slack}_{f,t}^{\text{min,cap}}) \right] \end{aligned}$$

where  $\text{SP}_{m,t}$  is the selling price of material  $m$  in period  $t$ ,  $\text{OPC}_{f,t}$  is the operating cost of facility  $f$  in period  $t$ ,  $\text{PC}_{m,t}$  is the purchase cost of material  $m$  in period  $t$ ,  $\text{IC}_{m,t}$  is the inventory cost of material  $m$  in period  $t$ , and  $\text{PEN}_{m,t}$  denotes the penalty associated with unmet demand.

Consider the historical data showing the variability of the production yield of facility  $P1$ ,  $\theta_{P1}$ , with 120 data points, which represent monthly records of  $\theta_{P1}$  for a period of ten years. The distribution of  $\theta_{P1}$  is depicted in a histogram as shown in Figure 3.3. Only the first two moments and ECDF data were estimated from the randomly generated production yield values. The simplified GLF ( $\beta_0 = 0$  and  $\beta_1 = 1$ ) fit to ECDF data and the estimated parameters are shown in Figure 3.4. Details of the procedure for generating the historical data for  $\theta_{P1}$ , fitting the simplified GLF, and the remaining parameters for the production planning model are given in Appendix B.1.

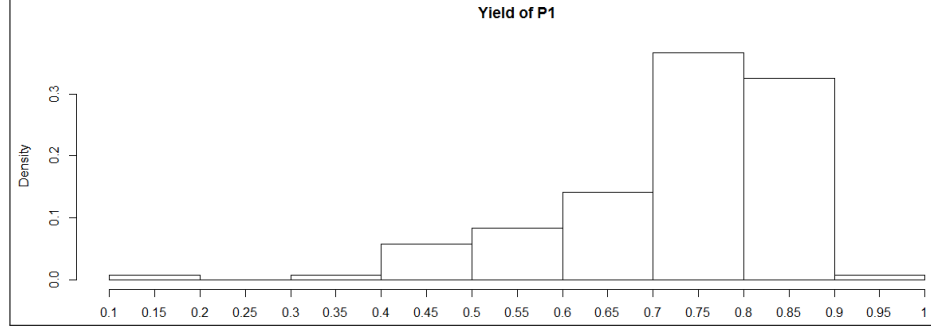


Figure 3.3: Distribution of the historical data for the production yield of facility  $P1$ .

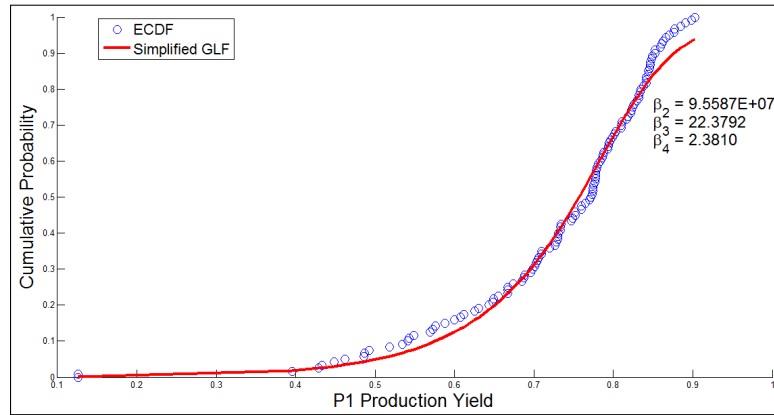


Figure 3.4: ECDF data of the production yield of facility  $P1$  fitted by a simplified GLF.

To simplify the analysis, we model this multiperiod production planning problem as a Two-Stage Stochastic Programming (TSSP) problem. There are four time periods that correspond to a quarterly production plan problem over the course of one year time horizon. The first stage, or *here-and-now* variables, are all the variables in the model at the first time period,  $t = 1$ , whereas the second stage, or *wait-and-see* variables, are all the variables at the remaining time periods,  $t > 1$ . All the DMPs and the deterministic equivalent of the TSSP models are implemented in AIMMS 3.13 (Roelofs & Bisschop, 2015). The DMPs were solved with IPOPT 3.10.1 using the Multi-Start Module in AIMMS and the TSSP model was solved with Gurobi 5.1. The model sizes are small; therefore, CPU times are not reported. In the DMPs, the yield variables were bounded below by the minimum data point, and above by the maximum data point.

Assuming we match for the DMP the first four moments (i.e.,  $|K| = 4$ ) and ECDF data, then it follows from inequality (3.15) that  $N \leq 5$ . Therefore, five scenarios are selected for the two-stage scenario tree. Two approaches are compared: heuristic and DMP, which includes the optimization models described in Section 3.3. The heuristic approach represents an arbitrary way to

construct a scenario tree that does *not* consider the distribution of the historical data. From minimum and maximum data values (*e.g.*, production yield that can vary between 0 and 1), their arithmetic mean is calculated (center node) and the values of the other nodes are calculated by fixed deviations of  $\pm 20\%$  and  $\pm 40\%$  from the mean node. Therefore, the tree in this example has five nodes in the second stage. Also, the probabilities are arbitrarily chosen. Notice that by not visualizing the distribution of the uncertain parameter, choice of outcomes and their probabilities may not satisfactorily characterize the shape of the distribution of the actual data. In other words, the heuristic scenario tree does not represent the actual problem data and the production plan obtained may not be very meaningful. The DMP approach calculates the probabilities (both LP and NLP formulations) and values of the nodes (only NLP formulations) in order to match statistical properties that describe the distribution of the yield data.

Figure 3.5 shows the probabilities and yield values obtained for the five-scenario tree in each approach.

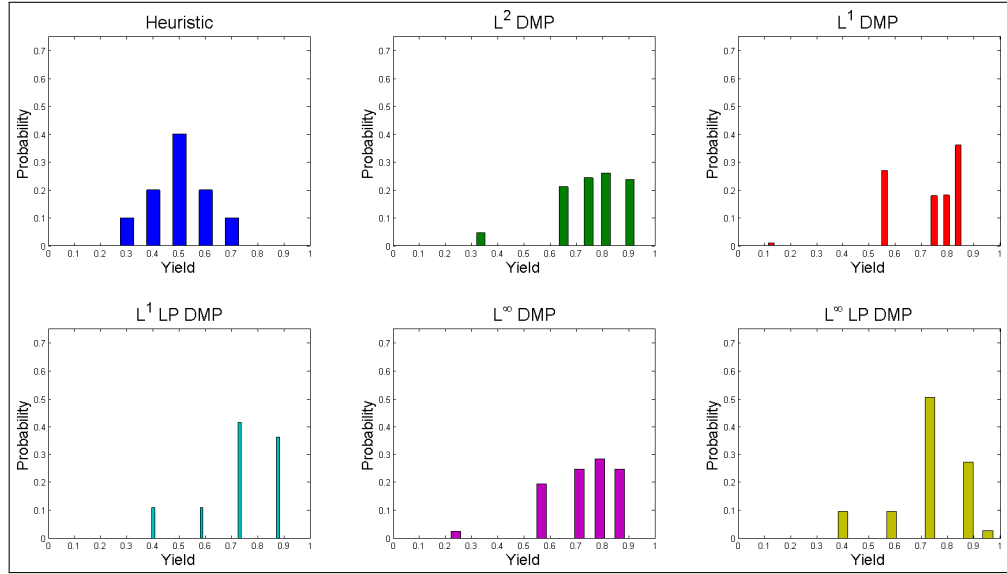


Figure 3.5: Probability profiles for the heuristic and optimization-based (DMP) approaches in Example 1. For reference, a histogram of the uncertain data is depicted in Figure 3.3.

The yield distribution as shown in Figure 3.3 is skewed to the right, which results in higher probabilities assigned to node values that are slightly higher than the mean yield (0.7301). Such characteristic is not captured in the heuristic approach, which does not satisfactorily represent the actual data. It was observed that the probabilities obtained with the  $L^1$  LP DMP and  $L^\infty$  LP DMP formulations are strongly dependent on the node values chosen and this fact affects the remaining results shown below. The objective function value of the three DMP formulations are shown in Table 3.1. Note that the extra

number of variables associated with considering the node values as variables (NLP formulations) results in smaller deviations in the matching procedure for the choices of weights (see [Appendix B.1](#)).

Table 3.1: Objective function values of the DMP formulations in Example 1. The values represent the error of matching the statistical properties.

Model	Objective Function
$L^2$ DMP	0.0030
$L^1$ DMP	0.0403
$L^1$ LP DMP	0.4739
$L^\infty$ DMP	0.0129
$L^\infty$ LP DMP	0.2112

[Table 3.2](#) shows the optimal expected profit of the stochastic production planning model in equation (3.16). The relatively low expected profit by using the heuristic approach can be explained due to high probabilities placed on production yields below the mean of the actual yield data. In other words, the scenario tree in the heuristic approach is *pessimistic* for the values chosen for the production yield of facility  $P1$ . Ultimately, the tree in the heuristic approach is an inaccurate representation of the yield data. However, note that the magnitude of the expected profit of the TSSP problem is not an assessment of the quality of each solution with respect to the “true” solution that would be obtained if the true distributions were known and were not approximated by finite discrete outcomes. The LP deterministic equivalent of the two-stage stochastic program has 273 constraints, 305 variables, 832 nonzeros, and was solved in not more than 0.02 seconds for all approaches.

Table 3.2: Expected profit of the production planning model in Example 1 using the scenario trees from two approaches.

Approach	Expected Profit [\$]
Heuristic	65.43
$L^2$ DMP	76.13
$L^1$ DMP	74.40
$L^1$ LP DMP	76.38
$L^\infty$ DMP	73.82
$L^\infty$ LP DMP	76.42

Figures 3.6 and 3.7 show the different production plans obtained for each approach. Specifically, the solution obtained with the heuristic approach predicted higher inventory levels of product  $D$  for the first time period (here-and-now decisions). Moreover, using the tree obtained with the heuristic approach incurred higher purchase amounts of product  $C$  for the first period.

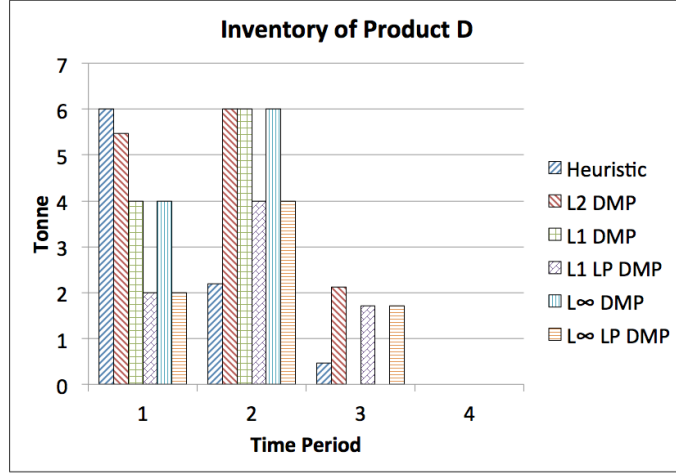


Figure 3.6: Optimal inventory levels of product  $D$  from using the scenarios obtained from heuristic and DMP approaches.

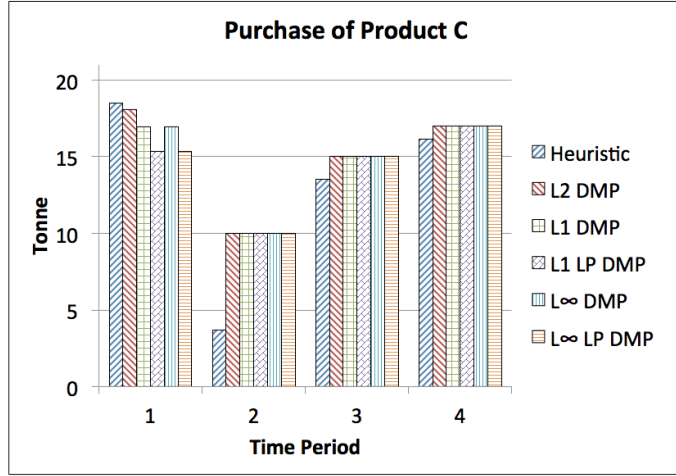


Figure 3.7: Optimal purchase amounts of product  $C$  from using the scenarios obtained from heuristic and DMP approaches.

Finally, the quality of the stochastic solutions was assessed using a simulation-based Monte Carlo sampling scheme that provides statistical bounds on the optimality gap (Bayraksan & Morton, 2006). The optimality gap is defined as the difference between an approximation of the “true” stochastic solution (very large tree to approximate the continuous distribution of the yield) and the candidate stochastic solution. In this context, a candidate stochastic solution refers to the first-stage decisions of the TSSP when using the scenario trees from either heuristic or DMP approaches. The Multiple Replications Procedure (MRP) was used with twenty replications, and each replication contains one hundred independent scenarios. In each replication, the gap between the approximate true solution and the candidate solution is calculated (see Subsubsection B.4.1 in Appendix B.4). Table 3.3 shows the average gap

of all replications plus one-sided confidence intervals for 95% confidence. The results suggest that the stochastic production planning solutions obtained by using the trees generated by the  $L^2$  DMP and  $L^1$  DMP approaches are closer to the true solution as seen from the small optimality gap. Note that since the historical data were artificially generated, i.e., the data-generating mechanism (distribution) is known (see [Appendix B.1](#)), a Monte Carlo sampling strategy can be used.

Table 3.3: Average value and upper bound of the optimality gap of the stochastic production planning model in Example 1.

Approach	Avg Gap [\$]	Upper Bound [\$]
Heuristic	0.21	0.29
$L^2$ DMP	0.04	0.10
$L^1$ DMP	0.14	0.19
$L^1$ LP DMP	0.70	0.91
$L^\infty$ DMP	0.04	0.10
$L^\infty$ LP DMP	0.70	0.91

The results in [Table 3.3](#) clearly show that selecting the scenario tree as an input to the stochastic optimization model is crucial to obtain a meaningful solution of an SP formulation. The distribution matching method is a scenario generation method that allows creating scenario trees that satisfactorily represent the distribution of the data, so that the decisions made with the stochastic optimization model are supported by factual probabilistic information.

### 3.4 Multi-Stage Scenario Tree Generation

The MMPs in equations (3.2) – (3.6) and DMPs in equations (3.10) – (3.14) can be applied to both two-stage and multi-stage cases, with the former more limited with issues of under-specification. When generating multi-stage scenario trees, a statistical property matching problem is solved at every node of the tree except at the leaf or terminal nodes. Multi-stage scenario trees can be viewed as a group of two-stage subtrees that are formed by branching out from every node except the leaf nodes. The complication in generating multi-stage scenario trees with interstage dependency lies in the fact that the moments calculated for each path into future stages are dependent on the previous *states* or nodes present in each path. Therefore, prediction of future events (time series forecasting) must be combined with property matching optimization as will be described below.

For stochastic processes, such as time-series data of product demands, statistical properties can be estimated through forecasting models that take into account information that is *conditional* on past events. [Appendix B.3](#) contains more details of using time series forecasting models to estimate the statistical properties.



Time series forecasting models play an essential role in generating scenario trees when there is a time-series effect in the uncertain parameters. Briefly, mathematical models are fit to the historical data and their predictive capabilities provide *conditional* (co-)moments of the uncertain parameters at future stages. The moments are conditional on past events. That is, they take into account the serial dependency of the observed values of the uncertain parameters. Hence, the statistical moments are supplied to the property matching optimization models at each non-leaf node and the scenario tree is consistently generated. Moreover, simulation of the time series models can be used to generate data of which an ECDF can be constructed and approximated with a smooth function, such as the GLF or a simplified version (see [Section 3.3](#)).

In the next two sections, we present two *sequential* solution strategies for generating a multi-stage scenario tree: *NLP Approach* and *LP Approach*. We focus on the DMP formulations, where  $L^2$  DMP,  $L^1$  DMP and  $L^\infty$  DMP are nonlinear (both node values and probabilities are variables), whereas  $L^1$  LP DMP and  $L^\infty$  LP DMP are linear (only probabilities are variables). Each approach comprises two main steps, **forecasting** and **optimization**, which are summarized below:

**Forecasting Step:** After successfully fitting a time series model to the data, forecast future values.

- Input: observed data represented by the nodes
- Output: conditional moments and ECDF information to be matched in the optimization step

**Optimization Step:** Solve a DMP at a given node in the tree.

- Input: conditional moments estimated by the forecasting step
- Output: probabilities of outcomes, and if using the NLP approach, values of the nodes

**Remark 1.** Conditional (co-)moments are readily available via forecasting. ECDF information can be obtained through simulation of time series models, and a brief overview is given in [paragraph 1.2.2.2](#) in [Appendix B.3](#). If a particular family of distribution is assumed for the forecast data, then the CDF or an approximate expression can be used instead as illustrated in Example 2 ([Subsection 3.4.3](#)).

**Remark 2.** Instead of a forecasting step, some authors have used a simulation step to generate the targets (conditional moments) to the optimization step and/or the values of the nodes in the scenario tree. For example, [Høyland, Kaut, & Wallace \(2003\)](#) proposed a heuristic method to produce a discrete joint distribution of the stochastic process that is consistent with specified values of the first four marginal moments and correlations. In addition, due to the independence of the optimization problems to be solved at each node of a given stage, there is an opportunity for parallel algorithms to speed up the solution process ([Beraldi, De Simone, & Violi, 2010](#)).

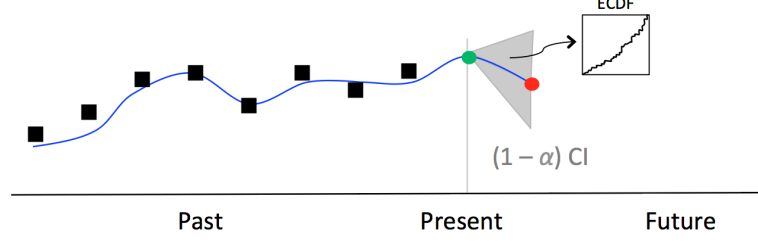
### 3.4.1 NLP Approach

A general solution strategy for generating a multi-stage scenario tree using the NLP formulations of the DMP consists of alternating the two main steps described above in a shrinking-horizon fashion, i.e., marching forward in time, node-by-node and stage-by-stage until the end of the time horizon. [Figure 3.8](#) depicts the sequence of steps in the approach. The black squares (past region) in each subfigure correspond to historical data of the uncertain parameter under consideration, the blue line across the markers represents the time series model used to make predictions of the stochastic process, the red circles (future region) are the possible future states that the stochastic process will visit, and the grey shaded area surrounding the red circles denotes the estimated prediction confidence limits for a given significance level of  $\alpha$ , i.e.,  $\alpha = 0.05$  indicates 95% confidence. Note that by connecting parent nodes to their descendants (from left to right), a scenario tree is obtained.

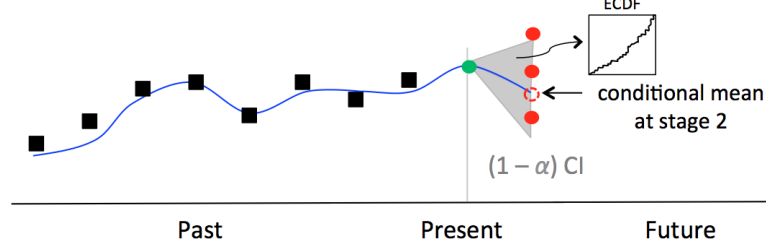
The algorithm can be stated as follows.

- Step 0:** Start at the root (“present”) node whose value is known. Set it as the current node.
- Step 1:** If not the last stage in the time horizon, then perform a one-step-ahead forecast from the current node to estimate conditional moments.
- Step 2:** Simulate the time series model including observations up to the current node, construct the ECDF curve, and approximate it by a smooth function, such as the GLF or a simplified version.
- Step 3:** Solve a nonlinear DMP to determine the node values and their probabilities for the next stage.
- Step 4:** For each node determined, set it as the current node and go to **Step 1**.

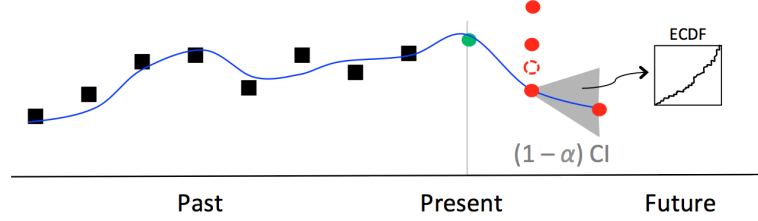
In [Figure 3.8](#), the blue line across the markers represents the time series model, the black squares represent past or historical data, the green circle on the vertical line is the current or present state, and the red circles are the future states. For demonstration purposes, [Figure 3.8\(c\)](#) only shows the forecasting step for the bottom node generated in the first optimization step. Note that some nodes may lie outside the confidence interval predicted by the forecasting step; this allows more extreme events to be captured in the scenario tree, which in turn subject the stochastic programming problem to riskier scenarios and may lead to more “robust” solutions.



(a) First one-step-ahead forecasting step to predict the most likely value of the stochastic process in the next stage as well as possible higher moments and distribution information from simulation.



(b) Optimization step to calculate probabilities and nodes for the next stage. Optionally, the node corresponding to the conditional mean may be fixed in the DMP.



(c) One-step-ahead forecasting step from a given node obtained in the optimization step before. Repeat these steps for every node generated in every stage until the end of the time horizon considered.

Figure 3.8: Alternating forecasting and optimization steps in generating multi-stage scenario trees using the *NLP Approach*. CI denotes the confidence interval estimated at each forecast and ECDF means Empirical Cumulative Distribution Function.

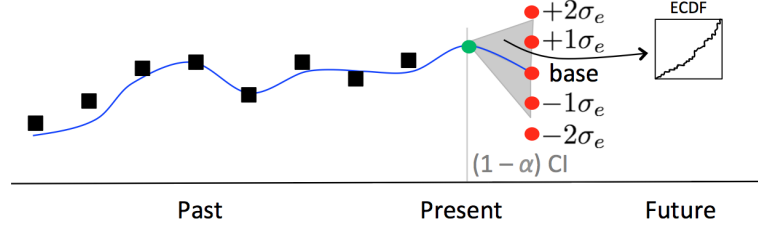
The complexity in implementing this approach in practice is the communication between the forecasting and the optimization steps at every non-leaf node in the tree. On the other hand, the approach using the LP formulations of the property matching problems only alternates between the forecasting and optimization steps once. The next section contains our proposed approach, and we note that there are variants in the literature that for instance use clustering algorithms (Gülpınar, Rustem, & Settergren, 2004; Xu, Chen, & Yang, 2012; Feng & Ryan, 2013).

### 3.4.2 LP Approach

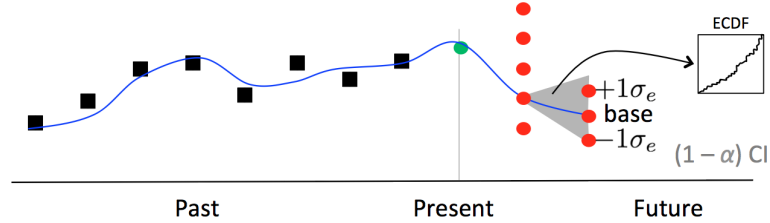
The only decision variables in the LP formulation in equation (3.12) are the probabilities of the outcomes. Therefore, if the node values are known in

advance, then a single optimization problem can be solved for the entire tree to compute their probabilities. Thus, the approach has only two steps: (1) the forecasting step generates the nodes plus the statistical properties to be matched, and (2) an LP DMP is solved for all non-leaf nodes simultaneously. The optimization step is a straightforward solution of an LP problem, whereas the forecasting step contains elements that are particular to a specific strategy.

The strategy for the forecasting step proposed in this chapter is shown in Figure 3.9. As shown in Figure 3.9(a), after performing a one-step-ahead forecast from the present node to the base or most likely node in the second stage, additional nodes are created by adding and subtracting multiples of the standard error of the forecast to the base node. The number of additional nodes above and below the base node is chosen *a priori*. In practice, near-future stages may be more finely discretized than far-future stages, since the prediction is less accurate the further into the future it is made. For ease of exposition, Figure 3.9(b) shows the forecast from the second to the third stage of one of the nodes created in the second stage. The process is repeated for every node in every stage, except the last one of the time horizon considered.



(a) First one-step-ahead forecasting step to predict the most likely value of the stochastic process in the next stage. Create new nodes by adding and subtracting multiples of the standard error,  $\sigma_e$ , of the forecast to the base node.



(b) For each node created, perform a one-step-ahead forecast and create new nodes. Repeat the process until the end of the time horizon considered.

Figure 3.9: Proposed forecasting step in generating multi-stage scenario trees using the *LP Approach*.  $\sigma_e$ , CI, and ECDF denote the standard error, confidence interval estimated at each forecast, and the Empirical Cumulative Distribution Function, respectively.

In summary, the main difference between the *NLP Approach* and the *LP Approach* is that in the latter, the forecasting and optimization steps alternate only once as all the nodes or outcomes of the tree are created in the forecasting step, and then the optimization step is executed to compute the probabilities.

We remark that the nonlinear DMP formulations can also be used in the *LP Approach* by fixing the node values and only optimizing the probabilities. In this case, the forecasting step is followed by the solution of a single nonlinear optimization problem for all non-leaf nodes.

The next example demonstrates how the two approaches can be used to generate a multi-stage scenario tree when product demand is uncertain.

### 3.4.3 Example 2: Uncertain Product Demands

Consider the same network depicted in [Figure 3.2](#) and the deterministic multiperiod production planning model defined in equation (3.16). In this case, the product demands of  $C$  and  $D$  are the uncertain parameters. The planning horizon is one year, which is divided into time periods of quarters. Quarterly historical demand data are given from the years of 2008 to 2012. Thus, the frequency (number of observations per year) of the time series is four. The production planning model is used to obtain the optimal quarterly production plan for the network in the year of 2013. As in Example 1, all optimization models were implemented in AIMMS 3.13. The NLP problems were solved with IPOPT 3.10.1 using the multi-start module in AIMMS with 30 sample points and 10 selected points in each iteration. All LP models were solved with CPLEX 12.5. In the DMPs, the demand variables were bounded below by half the minimum and above by double the maximum historical demand data points.

A common approach for deciding the structure of multi-stage trees is to select more outcomes per node in earlier stages than in later stages, since the uncertainty in the forecasts is much higher in the latter. Thus, it is reasonable to select a finer discretization in earlier stages. Assuming the properties matched are the first two moments (i.e.,  $|K| = 2$ ), covariance, and CDF information, it follows from inequality (3.15) that  $N \leq 6$ . It is then decided that the multi-stage scenario tree has the following structure: 1-5-3-1, which means that the second quarter has five outcomes, the third quarter has three outcomes for each outcome in the second quarter, and the fourth quarter has only one outcome for each outcome of the third quarter, thus, the scenario tree has 15 scenarios as seen in [Figure 3.10](#). As in Example 1, a heuristic approach is compared with the optimization-based DMPs to obtain scenario trees. We consider uncertainty in the demand of both products  $C$  and  $D$ . The tree for each individual product demand is obtained as follows. The center or base node at a given quarter is the arithmetic average of the corresponding quarter of previous years, and the remaining nodes above and below the base node are obtained by fixed deviations. Therefore, the node values *ignore* the serial dependence and time-series effects in the data. The individual heuristic trees for products  $C$  and  $D$  were combined into a single tree with the same structure (1-5-3-1) by overlapping the outcomes for each stage as shown in [Figure 3.10](#). Probabilities of outcomes were arbitrarily chosen and are symmetric with respect to each base node.

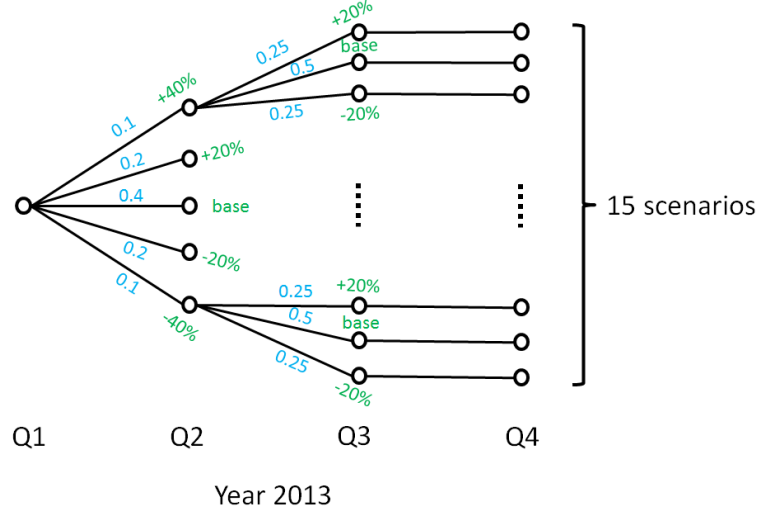


Figure 3.10: Heuristic scenario tree for the demand of products *C* and *D*. The percentage deviations are computed based on each base node. The values above and below the arcs are arbitrarily chosen probabilities.

Figure 3.11 shows the time series demand data for products *C* and *D*. The time series model that best fits the data (see Appendix B.2 for details) is fixed in subsequent forecasts. That is, when executing an approach no refitting is performed prior to forecasting. The root node of the tree, which is the node value at the first quarter in 2013 or “Q12013”, is forecast and assumed to have probability of one. The constant variance for the demand of products *C* and *D* are estimated to be  $1.65 t^2$  and  $1.14 t^2$ , respectively.

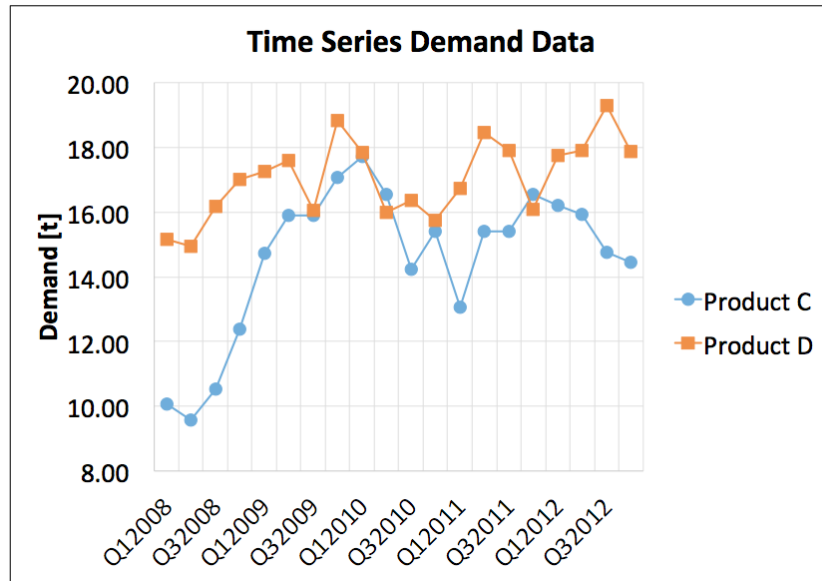


Figure 3.11: Time series data of the demand of products *C* and *D*.

Since the demand data are fitted to a linear Gaussian model (ARIMA), the

forecasts are expected to follow normal distributions. Therefore, an expression for the Cumulative Distribution Function (CDF) of a normal distribution with mean  $\mu$  and standard deviation  $\sigma$  can be used in the constraints involving  $\widehat{ECDF}(\cdot)$ . In particular, the CDF of a normal distribution can be written in terms of the error function as follows ([Abramowitz & Stegun, 1965](#)):

$$CDF(x)^{\text{Normal}} := \Phi\left(\frac{x - \mu}{\sigma}\right) = \frac{1}{2} \left[ 1 + \operatorname{erf}\left(\frac{x - \mu}{\sigma\sqrt{2}}\right) \right]$$

where  $\operatorname{erf}(\cdot)$  is the error function defined as the following integral:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Hence, constraints (3.10b) can be replaced with,

$$\Phi\left(\frac{x_{i,j} - M_{i,1}}{\sqrt{M_{i,2}}}\right) - \sum_{j'=1}^j p_{j'} = \delta_{i,j} \quad \forall i \in I, j = 1, \dots, N$$

constraints (3.11b) and (3.13b) can be substituted by,

$$\Phi\left(\frac{x_{i,j} - M_{i,1}}{\sqrt{M_{i,2}}}\right) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N$$

and finally constraints (3.12b) and (3.14b) are rewritten as,

$$\Phi\left(\frac{x_{i,j} - M_{i,1}}{\sqrt{M_{i,2}}}\right) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N$$

AIMMS offers a native, numerical approximate implementation of the error function, which can be directly used in the implementation of the constraints in the DMPs.

Exclusively for the LP DMPs, it was observed that additional constraints on the probabilities were necessary in order to enforce a normal-like profile, i.e., probabilities monotonically decrease from the center node outward. The additional constraints are given below and are equivalent to the ones proposed by [Ji et al. \(2005\)](#).

$$\begin{aligned} p_j &\geq p_{j'} & \forall j = \left\lceil \frac{N}{2} \right\rceil, \dots, N, j' > j \\ p_j &\geq p_{j'} & \forall j = 1, \dots, \left\lfloor \frac{N}{2} \right\rfloor, j' < j \end{aligned}$$

For illustration purposes, the scenario trees obtained with NLP and LP approaches are shown in [Figure 3.12](#) ( $L^2$  DMP) and [Figure 3.13](#) ( $L^\infty$  LP DMP), respectively. For the *NLP Approach*, the node values in the fourth

time period correspond to the conditional means obtained via forecasting, i.e., no optimization was needed as only one outcome was considered. It should be noted that the total time for solving the six NLPs with multi-start for the tree in Figure 3.12 was 12.35 seconds, while the LP in Figure 3.13 took 0.02 seconds.

The optimal expected profit of the production planning model by using the scenario tree of the proposed approach as the input, and by solving the deterministic equivalent of the multi-stage stochastic programming model is shown in Table 3.4. The heuristic approach underestimates the expected total profit when compared to the NLP DMPs. Again, we note that the scenario probabilities and the solution obtained with the LP DMPs is greatly affected by the node values chosen. The LP deterministic equivalent of the multi-stage stochastic program has 613 constraints, 685 variables, 1,872 nonzeros, and was solved in less than 0.02 seconds for all approaches.

Table 3.4: Expected profit in Example 2 using the scenario trees from heuristic and optimization-based approaches.

Approach	Expected Profit [\$]
Heuristic	79.95
$L^2$ DMP	82.39
$L^1$ DMP	82.65
$L^1$ LP DMP	80.04
$L^\infty$ DMP	82.41
$L^\infty$ LP DMP	80.03

Figures 3.14 and 3.15 show the different production plans obtained for each approach at each quarter. Specifically, the solution obtained with the heuristic approach predicted higher overall inventory levels of product  $D$  for the time horizon under consideration. Moreover, the solution using the heuristic tree shows very different average flowrates out of plant  $P2$  compared to the ones obtained using the DMP formulations. In real life terms, the production quota for a plant affects lower-level operability decisions, such as scheduling and control.



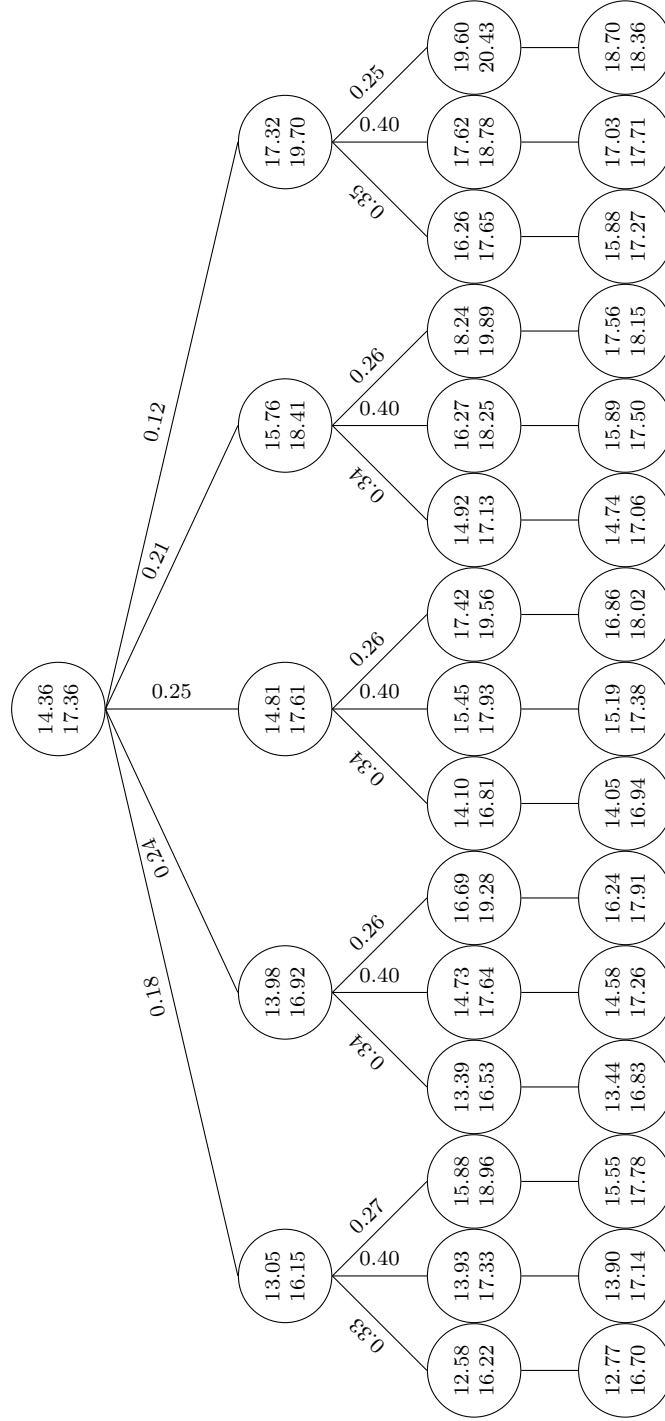


Figure 3.12: Scenario tree obtained with the NLP Approach ( $L^2$  DMP) for Example 2. Top and bottom values inside each node are the calculated demands of products  $C$  and  $D$ , respectively.

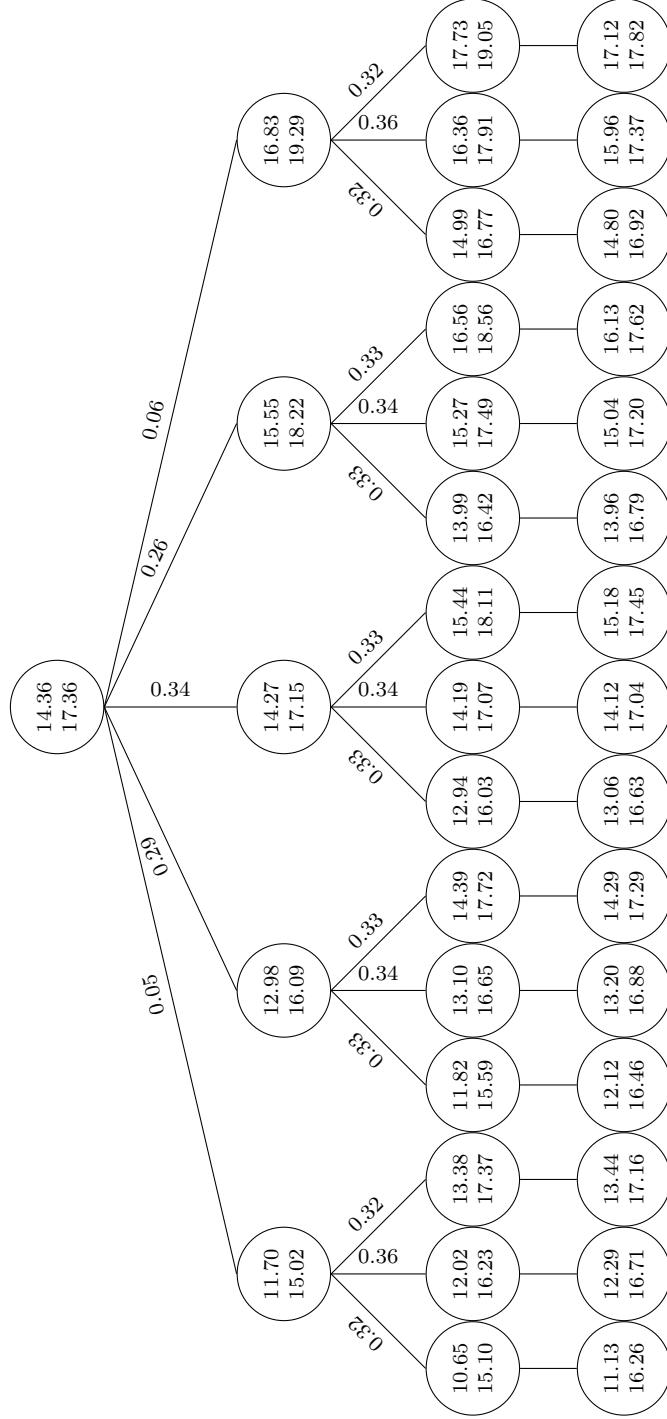


Figure 3.13: Scenario tree obtained with the  $L^{\infty}$  LP Approach for Example 2. The node values are obtained via forecasting and the probabilities are calculated via optimization. Top and bottom values in each node are the demands of products  $C$  and  $D$ , respectively.

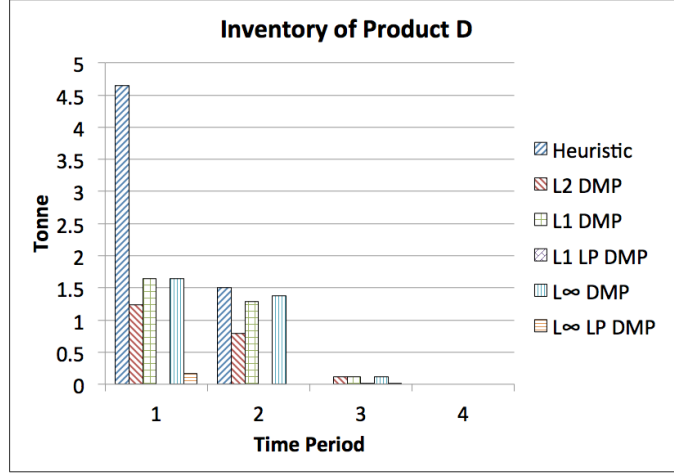


Figure 3.14: Optimal inventory levels of product  $D$  from using the scenarios obtained from heuristic and DMP approaches.

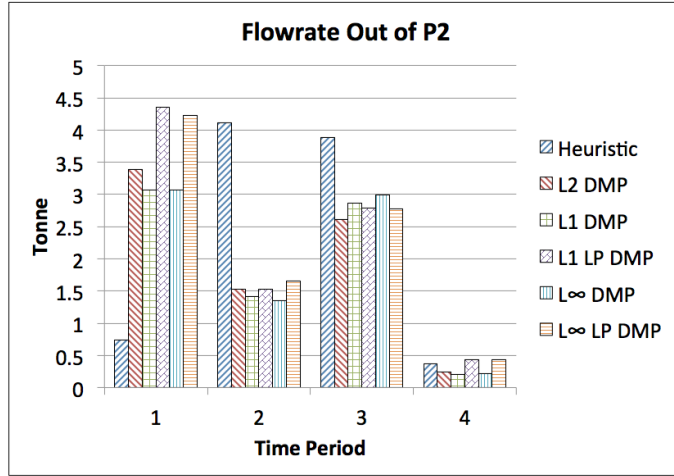


Figure 3.15: Optimal flow rates out of plant  $P2$  from using the scenarios obtained from heuristic and DMP approaches.

Finally, similarly to Example 1, the quality of the stochastic solutions was assessed using a simulation-based Monte Carlo sampling scheme that provides statistical bounds on the optimality gap ([Chiralaksanakul & Morton, 2004](#)). The optimality gap is calculated using a tree-based estimator of the lower bound (candidate solution, maximization problem) and the approximate true solution (upper bound estimator, maximization problem). The simulated trees have the structure 1-10-10-10, which amounts to one thousand scenarios. All subtrees are generated by simulating ARIMA processes (see [paragraph 1.2.2.2](#) in [Appendix B.3](#)). Ten replications of the algorithm (see Procedure  $\mathcal{P}_2$  in the original paper) were performed to obtain the confidence interval on the gap (see [Subsubsection B.4.2](#) in [Appendix B.4](#)). [Table 3.5](#) shows the one-sided confidence intervals for 95% confidence. Note that by modeling the demand

time series as ARIMA processes, the data-generating mechanism is known, and Monte Carlo sampling can be performed by simulating the ARIMA models (see [paragraph 1.2.2.2 in Appendix B.3](#)).

Table 3.5: Average value and upper bound of the optimality gap of the stochastic production planning model in Example 2.

Approach	Avg Gap [\$]	Upper Bound [\$]
Heuristic	2.03	2.16
$L^2$ DMP	0.92	1.00
$L^1$ DMP	0.93	1.01
$L^1$ LP DMP	1.12	1.35
$L^\infty$ DMP	0.93	1.01
$L^\infty$ LP DMP	1.04	1.22

Note that the confidence interval of the gaps obtained for all the DMP formulations are lower than the one obtained for the heuristic approach, which indicates that the scenario trees generated via the optimization-based procedure are good approximations of the “true” distribution. In addition, they contain correlation information between the demands of the two products, thus improving the characterization of the uncertainty.

### 3.5 Conclusions

In this chapter, we have described a systematic method for scenario tree generation that can be used in optimization and simulation solution strategies in Process Systems Engineering (PSE) problems. The distribution matching method is based on an optimization model that can be used to calculate the probabilities and values of outcomes in a scenario tree. This is accomplished by matching statistical properties, such as (co-)moments and Empirical Cumulative Distribution Function (ECDF) information, calculated from the tree to the ones estimated from the data. The motivation behind this approach is first to ensure that the scenarios considered in a stochastic programming framework correspond to the behavior observed in historical and forecast data, as opposed to assigning arbitrary values and probabilities to the nodes in the scenario tree. Secondly, the motivation of the Distribution Matching Problem (DMP) formulations is to overcome the problem of under-specification that is experienced with the Moment Matching Problem (MMP) formulations.

For multi-stage scenario tree generation, we presented two approaches that can be used in conjunction with time series forecasting or simulation. The *NLP Approach* is composed of two alternating steps: the forecasting or simulation step computes conditional (co-)moments and ECDF information to be matched, and the optimization step determines the probabilities and values of the outcomes in order to match those properties. The two steps are alternated until the prescribed time horizon is completed. The *LP Approach*, which is restricted to specified values of the outcomes at each node, also contains the

same two steps, but each step is performed only once. First, the forecasting or simulation step generates all the nodes in the tree and computes the conditional (co-)moments and ECDF information, and then the optimization step calculates the probabilities of the outcomes.

The quality of the solution of the stochastic programming problems was assessed via simulation-based Monte Carlo sampling methods. It was shown that the proposed scenario tree generation methods provide good-quality solutions, and provide a systematic approach for handling multiple sources of uncertainty, thus generating more compact trees with covariance or correlation information.

# Chapter 4

## Data-Driven Individual and Joint Chance-Constrained Optimization via Kernel Smoothing

### 4.1 Introduction

Optimization under uncertainty is an important and challenging topic in the Process Systems Engineering (PSE) community. Its importance stems from the fact that the decision-making capability of optimization models is subject to uncertainty, most commonly by the parameters used in a given model. For instance, forecast product demand is not free of uncertainty since one cannot exactly predict future events. Consequently, optimization models under uncertainty are generally more difficult to model and solve than their deterministic counterpart. A few frameworks to account for uncertainty in optimization have been proposed, such as Stochastic Programming (Birge & Louveaux, 2011), Robust Optimization (Ben-Tal, Ghaoui, & Nemirovski, 2009), Approximate Dynamic Programming (Powell, 2011), and Chance-Constrained Optimization, which is the focus of this article.

Chance- or probabilistic-constrained optimization was introduced in the seminal article by Charnes & Cooper (1959). Briefly, the use of chance constraints (CCs) allows the modeler to specify a *confidence level* or *reliability level* of satisfying some of the constraints in an optimization model. In other words, the solution to the problem satisfies those constraints with at least a given level of probability for all possible realizations of the uncertain parameter present in the respective constraints. The literature considers two types of CCs: *individual* and *joint*. Let  $\tilde{\xi}_j$ ,  $j = 1, \dots, m$ , correspond to continuous random variables, and  $g_j(x)$  be a vector of continuous functions of the vector of decision variables  $x$ . Equations (4.1) and (4.2) correspond to individual and

joint chance constraints with right-hand side uncertainties, respectively,

$$\mathbb{P}\{g_j(x) \geq \tilde{\xi}_j\} \geq 1 - \alpha_j, \quad j = 1, \dots, m \quad (4.1)$$

$$\mathbb{P}\{g_j(x) \geq \tilde{\xi}_j, j = 1, \dots, m\} \geq 1 - \alpha \quad (4.2)$$

where  $\mathbb{P}$  means “probability”,  $1 - \alpha$  is the confidence level or reliability level, and  $\alpha$  is commonly known as the risk level or significance level. Note that each individual CC (ICC) is assigned a confidence level  $1 - \alpha_j$ ; that is, they must be probabilistically satisfied individually. In contrast, in the joint CC (JCC) case, all constraints must be probabilistically satisfied simultaneously; thus, a single confidence level  $1 - \alpha$  is used.

Charnes & Cooper (1959) used CCs to model a multiperiod planning problem under uncertainty in which the uncertain parameter (product demand) was time-dependent. They employed linear decision rules to approximate the solution of the multi-stage stochastic model. This article was followed by a few others (Charnes & Kirby, 1966, 1967; Eisner, Kaplan, & Soden, 1971; Gochet & Padberg, 1974) that proposed analytical approaches for reformulating the (conditional) ICCs and approximately solving the stochastic model with decision rules. The aforementioned articles dealt with the case of *temporal* conditional ICCs, i.e., the realization of a random variable at a given time period may be conditional on realizations at previous time periods.

In this chapter, we only consider CCs with right-hand side uncertainty as shown in equations (4.1) and (4.2). We argue that, even though seemingly restrictive, this represents a very common situation in optimization models in engineering applications. Some examples include: reliable reservoir control under hydrologic uncertainty (Lane, 1973; Simonovic & Srinivasan, 1993; Jairaj & Vedula, 2003), power systems with uncertain power generation (Bienstock, Chertkov, & Harnett, 2012; Mazadi *et al.*, 2013), and process systems engineering with uncertain set-up and operational times, state setpoint tracking, and environmental damage (Orçun, Altinel, & Hortaçsu, 1996; Arellano-Garcia & Wozny, 2009; Guillén-Gosálbez & Grossmann, 2010).

Theoretical properties of CCs that guarantee convexity have been studied (Prékopa, 1970, 1995). In particular, it has been shown that if  $\mathbb{P}$  is log-concave (e.g., normal or Gaussian) and each  $g_j(\cdot)$  is concave (e.g., linear), then the CC admits a convex reformulation, i.e., the CC is reformulated as a convex algebraic constraint. It is beyond the scope of this article to provide an extensive review of the theory of and solution methods for *classical* CCs. Further reading can be found in Nemirovski & Shapiro (2006); Calafiore & Campi (2006); Luedtke & Ahmed (2008).

A common feature of the works mentioned thus far is the strong assumption of a parametric model for probability distributions. In this article, we focus on data-driven rather than axiomatic or model-driven approaches for reformulating CCs into algebraic constraints. The main idea behind the approach discussed by Jiang & Guan (2013) is to construct a confidence set that contains the unknown true distribution. This confidence set is described by

$\phi$ -divergences (Pardo, 2006), which are mathematical expressions that model the distance between two probability distributions. The definition of the confidence set is completed by specifying its size, which in this case is given by the  $\phi$ -divergence tolerance. Jiang & Guan (2013) showed that when using this confidence set, the data-driven CCs are equivalent to classical CCs, but with the estimated distribution and a reduced risk level. We will use this result and provide a new way to specify the  $\phi$ -divergence tolerance in our proposed approach.

The main contributions of this work are two-fold: (1) reformulation of data-driven ICCs and JCCs using kernel smoothing to estimate distributions, and (2) calculation of the divergence tolerance based on point-wise standard errors from the estimation process. Other contributions include: (i) an algorithm for the initialization of optimization models with JCCs, and (ii) reformulation of data-driven chance constraints in small sample size cases. Because kernel smoothing is central to the proposed approach, a brief overview of this statistical technique is given in Subsubsection 1.2.2.3.

This chapter is organized as follows. In Section 4.2, we state the problem addressed in this chapter. Section 4.3 summarizes main results from the literature concerning data-driven CCs. In Section 4.4, we describe the proposed approach to reformulate (conditional) data-driven CCs using kernel smoothing. A new way of specifying the  $\phi$ -divergence tolerance for the density-based confidence set is presented. To cope with the difficulty of solving problems with JCCs, we propose in Section 4.5 an initialization algorithm to decrease the solution time of problems with classical or data-driven JCCs. The kernel-based reformulation approach is illustrated with production planning examples in Section 4.6, and conclusions are drawn in Section 4.7.

## 4.2 Problem Statement

The problem addressed in this work can be stated as follows. Given an optimization model with individual and/or joint chance constraints (ICCs and/or JCCs) with right-hand side uncertainty (e.g., equations (4.1) and (4.2)) and historical data for the uncertain parameter(s), the goal is to convert the probabilistic constraints into algebraic form by using the historical data to estimate probability distributions. That is, instead of using parametric models for PDFs, CDFs, and quantile functions, those functions are estimated from data. A motivation section on chance-constrained optimization with illustrative examples is given in Appendix C.1. Before presenting the proposed approach that uses kernel smoothing to estimate distributions, we review the modeling of data-driven chance constraints in the next section.

## 4.3 Data-Driven Chance Constraints

This section summarizes the literature results concerning the modeling of data-driven chance constraints (Ben-Tal *et al.*, 2011; Jiang & Guan, 2013) that are directly relevant to the reformulation approach proposed in this article.



A central aspect of modeling data-driven chance constraints (CCs) is the definition of a confidence set, which is assumed to contain the unknown or *ambiguous* true distribution within specified tolerance. The construction of such a confidence set is aided by using a distribution that is *estimated* from the data. The accuracy of the estimated distribution can be used to calculate the size of the confidence set; the higher the accuracy, the smaller the confidence set.

Specifically, the density-based confidence set proposed in the literature and illustrated in [Figure 4.1](#) is defined as follows:

$$\mathcal{D} = \{\mathbb{P} \in \mathcal{M}_+ : D(f \parallel \hat{f}) \leq d, f = d\mathbb{P}/d\tilde{\xi}\} \quad (4.3)$$

where  $\mathbb{P}$  is the unknown true probability distribution function (or simply distribution),  $\mathcal{M}_+$  represents the set of all distributions,  $D(\cdot \parallel \cdot)$  is a  $\phi$ -divergence ([Pardo, 2006](#)) that models the distance between distributions and is defined as

$$D(f \parallel \hat{f}) = \int_{\mathbb{R}^m} \phi\left(\frac{f(\tilde{\xi})}{\hat{f}(\tilde{\xi})}\right) \hat{f}(\tilde{\xi}) d\tilde{\xi} \quad (4.4)$$

where  $f(\cdot)$  and  $\hat{f}(\cdot)$  are the true and estimated probability density functions (or simply densities), respectively, and  $\phi(\cdot)$  is a convex function on  $\mathbb{R}^+$  such that  $\phi(1) := 0$ ,  $0 \cdot \phi(0/0) := 0$ , and  $0 \cdot \phi(p/0) := p \lim_{t \rightarrow \infty} \phi(t)/t$  for  $p > 0$ . The  $\phi$ -divergence tolerance  $d$  represents the “size” of the confidence set. Therefore, a complete specification of the confidence set includes the  $\phi$ -divergence function  $\phi(\cdot)$  and the divergence tolerance  $d$ . [Table 4.1](#) contains some examples of divergence functions.

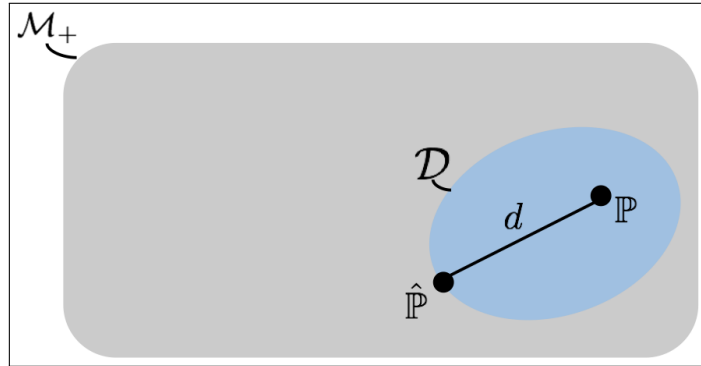


Figure 4.1: Illustration of the density-based confidence set defined in equation (4.3). The goal is to obtain the estimated distribution  $\hat{\mathbb{P}}$  whose “distance” to the unknown true distribution  $\mathbb{P}$  is at most  $d$  (divergence tolerance).

In a data-driven modeling approach, the CCs in equations (4.1) and (4.2) are required to be satisfied for *all* distributions in a given confidence set. Using the definition of the density-based confidence set  $\mathcal{D}$ , the individual and joint

*data-driven* CCs can be represented as follows:

$$\inf_{\mathbb{P} \in \mathcal{D}} \mathbb{P} \left\{ g_j(x) \geq \tilde{\xi}_j \right\} \geq 1 - \alpha_j, \quad j = 1, \dots, m \quad (4.5)$$

$$\inf_{\mathbb{P} \in \mathcal{D}} \mathbb{P} \left\{ g_j(x) \geq \tilde{\xi}_j, \quad j = 1, \dots, m \right\} \geq 1 - \alpha \quad (4.6)$$

where the  $\inf$  operator implies the “worst” distribution in the confidence set  $\mathcal{D}$ , i.e., the distribution that has the largest “distance” from the true distribution within a specified tolerance  $d$ .

A major result by [Jiang & Guan \(2013\)](#) (see Theorem 2) is as follows. The data-driven (individual or joint) CC

$$\inf_{\mathbb{P} \in \mathcal{D}} \mathbb{P} \left\{ A(x)\tilde{\xi} \leq b(x) \right\} \geq 1 - \alpha \quad (4.7)$$

can be reformulated as

$$\hat{\mathbb{P}} \left\{ A(x)\tilde{\xi} \leq b(x) \right\} \geq 1 - \alpha'_+ \quad (4.8)$$

where  $\hat{\mathbb{P}}$  is the estimated distribution,  $\alpha'_+ = \max\{\alpha', 0\}$ , and  $\alpha'$  is a reduced risk level. For CCs with RHS uncertainty, and using the notation in this article,  $b(x) \equiv g(x)$ , and  $A(x) \equiv 1$  and  $A(x) \equiv I_m$  for individual and joint CCs, respectively, where  $I_m$  is an  $m$ -dimensional identity matrix.

In other words, reformulated data-driven CCs are equivalent to classical CCs with the ambiguous probability  $\mathbb{P}$  replaced by its estimate  $\hat{\mathbb{P}}$ , and the risk level  $\alpha$  decreased to  $\alpha'$ . Intuitively, by treating the true probability distribution as unknown or ambiguous, one can reduce  $\alpha$  to make a classical chance constraint more conservative. The calculation of  $\alpha'$  can be formulated as an optimization problem (see Theorem 2 in [Jiang & Guan \(2013\)](#)). Its solution depends on the choice of the  $\phi$ -divergence function, since it is part of the definition of the density-based confidence set in equation (4.3). [Table 4.1](#) shows expressions for  $\alpha'$  (for some divergence functions) that quantify the degree of conservatism. Note that for the Kullback-Leibler (K-L) divergence, a closed-form solution has not been achieved. However, [Jiang & Guan \(2013\)](#) showed that the univariate optimization problem to calculate  $\alpha'$  can be recast as a univariate nonlinear algebraic equation that can be solved with the bisection method.

Table 4.1: Some divergence functions and the reduced risk level  $\alpha'$  for data-driven chance constraints (see [Jiang & Guan \(2013\)](#) for general expression and proofs).

Divergence Name	$\phi(t)$	$\alpha'$
Kullback-Leibler (K-L)	$t \log t - t + 1$	$1 - \inf_{x \in (0,1)} \left\{ \frac{e^{-d} x^{1-\alpha} - 1}{x - 1} \right\}$
Variation Distance	$ t - 1 $	$\alpha - \frac{d}{2}$
$\chi$ Divergence of Order $2^\dagger$	$ t - 1 ^2$	$\alpha - \frac{\sqrt{d^2 + 4d(\alpha - \alpha^2)} - (1 - 2\alpha)d}{2d + 2}$

$^\dagger$  For  $\alpha < \frac{1}{2}$ .

The reformulation approach proposed in this article uses the aforementioned result for the case of distributions estimated with kernel smoothing. We provide a brief overview of this statistical technique in [Subsubsection 1.2.2.3](#).

In summary, the reformulation of data-driven CCs consists in the following:

- Select the  $\phi$ -divergence;
- Estimate the unknown true probability distribution, i.e., obtain  $\hat{\mathbb{P}}$ ;
- Calculate or specify the divergence tolerance  $d$ ;
- Calculate reduced risk level  $\alpha'$ ;
- Solve the optimization problem with the reformulated data-driven CCs.

The novelty of this work lies in the use of kernel smoothing for both the reformulation of data-driven CCs and the calculation of the divergence tolerance  $d$ ; thus, fully specifying the density-based confidence level. However, it is important to note that the reformulation approach proposed in this chapter may only be applied for chance constraints with right-hand side uncertainty. In addition, the use of kernel smoothing to estimate distributions requires a sufficiently large number of data points to achieve satisfactory approximations, which is also true for other nonparametric statistical techniques. The accuracy of the approximation is reflected in the estimation errors, which in turn affect the magnitude of  $d$ .

## 4.4 Kernel-Based Reformulation of Chance Constraints

In this section, we demonstrate how to reformulate data-driven chance constraints (CCs) in which the true and ambiguous distributions are replaced with their kernel estimators. [Subsubsection 1.2.2.3](#) briefly describes kernel

smoothing, which is a powerful, nonparametric statistical technique that can be used to estimate univariate and multivariate (un)conditional densities, distributions, and quantiles. [paragraph 1.2.2.3](#) and [paragraph 1.2.2.3](#) review the univariate and multivariate cases, which are relevant for the reformulation of data-driven unconditional individual and joint CCs (ICCs and JCCs), respectively. [paragraph 1.2.2.3](#) reviews kernel regression for the case of weakly-dependent autocorrelated data, which is relevant for the reformulation of data-driven conditional ICCs.

As reviewed in [Section 4.3](#), the reformulation of data-driven CCs involves reduced risk levels  $\alpha'$ , which can be calculated as shown in [Table 4.1](#). Note that this calculation depends on the nominal risk level  $\alpha$  and the value for the divergence tolerance  $d$ . Hence, the complete implementation of data-driven CCs requires the knowledge of both  $\alpha'$  and  $d$ .

We first discuss the reformulation of individual CCs (ICCs), and then the joint case (JCC). Next, we propose a new way of selecting the  $\phi$ -divergence tolerance  $d$  for the proposed kernel-based reformulation. The case of small sample size is discussed at the end of this section.

#### 4.4.1 Individual Chance Constraints

First, we restate equation (4.8) as an individual data-driven CC using the notation in this article:

$$\hat{\mathbb{P}}\{g_j(x) \geq \tilde{\xi}_j\} \geq 1 - \alpha'_{j,+}, \quad j = 1, \dots, m \quad (4.9)$$

Equation (4.9) can be stated in probabilistic terms as follows: “the probability of the random variable (r.v.)  $\tilde{\xi}_j$  to achieve a value less than or equal to  $g_j(x)$  must be at least  $1 - \alpha'_{j,+}$ , for  $j = 1, \dots, m$ ”. In other words, we require that the estimated CDF of the random variable  $\tilde{\xi}_j$  evaluated at  $g_j(x)$  must be at least  $1 - \alpha'_{j,+}$ , for  $j = 1, \dots, m$ . Mathematically:

$$\hat{F}_{\tilde{\xi}_j}(g_j(x)) \geq 1 - \alpha'_{j,+}, \quad j = 1, \dots, m \quad (4.10)$$

or

$$g_j(x) \geq \hat{F}_{\tilde{\xi}_j}^{-1}(1 - \alpha'_{j,+}), \quad j = 1, \dots, m \quad (4.11)$$

where  $\hat{F}_{\tilde{\xi}_j}^{-1}(\cdot)$  is the estimated inverse CDF (quantile function). Note that the same rationale used to derive equation (4.11) is applicable to the reformulation of classical ICCs.

**Remark 1.** If the original constraints  $g_j(x) \geq \tilde{\xi}_j$ , for  $j = 1, \dots, m$ , are linear in the vector of decision variables  $x$ , i.e.,  $g_j(x) = a_j^\top x$ , where  $a_j$  is a vector of deterministic (fixed) coefficients, then the reformulated data-driven ICC is also linear in the vector of decision variables as can be seen from equation (4.11).

**Remark 2.** Equation (4.11) can also be used to model conditional ICCs ([Charnes & Kirby, 1967](#); [Gochet & Padberg, 1974](#)). Let  $\tilde{\xi}_1$  and  $\tilde{\xi}_2$  be continuous

random variables (r.vs.) with distributions  $F_{\tilde{\xi}_1}(\cdot)$  and  $F_{\tilde{\xi}_2|\tilde{\xi}_1}(\cdot)$ , respectively. Notice that  $\tilde{\xi}_2$  is conditional on  $\tilde{\xi}_1$ . Therefore, the kernel-based reformulation of the data-driven individual CCs involving the two r.vs. can be written as follows:

$$\begin{aligned} g_1(x) &\geq \hat{F}_{\tilde{\xi}_1}^{-1}(1 - \alpha'_{1,+}) \\ g_2(x) &\geq \hat{F}_{\tilde{\xi}_2|\tilde{\xi}_1}^{-1}(1 - \alpha'_{2,+}) \end{aligned}$$

Note that in the case of weakly dependent r.vs (see [paragraph 1.2.2.3](#)), the estimated conditional quantile function in the RHS is obtained with nonparametric conditional quantile regression whose regressors (covariates) are lagged observations of the r.v. under consideration. The reformulated CC can be expressed as follows:

$$g_t(x) \geq \hat{F}_{\tilde{\xi}_t|\tilde{\xi}_{t-1},\dots,\tilde{\xi}_{t-d}}^{-1}(1 - \alpha'_{t,+})$$

**Remark 3.** If the constraint requires the r.v. to be *greater* than or equal to  $g_j(x)$ , that is,

$$\hat{\mathbb{P}}\{g_j(x) \leq \tilde{\xi}_j\} \geq 1 - \alpha'_{j,+}, \quad j = 1, \dots, m \quad (4.12)$$

then we use the fact that  $\mathbb{P}\{Z \geq z\} = 1 - \mathbb{P}\{Z \leq z\}$ , where  $Z$  is a *univariate* continuous r.v., to derive the reformulated data-driven ICC in equation (4.12) as follows:

$$\begin{aligned} 1 - \hat{\mathbb{P}}\{g_j(x) \geq \tilde{\xi}_j\} &\geq 1 - \alpha'_{j,+}, & j = 1, \dots, m \\ \hat{\mathbb{P}}\{g_j(x) \geq \tilde{\xi}_j\} &\leq \alpha'_{j,+}, & j = 1, \dots, m \\ \hat{F}_{\tilde{\xi}_j}(g_j(x)) &\leq \alpha'_{j,+}, & j = 1, \dots, m \end{aligned}$$

or

$$g_j(x) \leq \hat{F}_{\tilde{\xi}_j}^{-1}(\alpha'_{j,+}), \quad j = 1, \dots, m \quad (4.13)$$

In cases where  $g_j(x) \geq \tilde{\xi}_j$  and  $g_{j'}(x) \leq \tilde{\xi}_{j'}$ , for  $j \neq j'$ , the CCs can be naturally reformulated by using the respective expressions of the term involving the integrated kernels in equations (4.11) and (4.13).

The effectiveness of the kernel-based reformulation approach for data-driven ICCs relies on the accuracy of the estimated quantile function, which represents the right-hand side of equations (4.11) and (4.13). [Figure 4.2](#) illustrates the accuracy of the estimated quantile function when compared to the true quantiles obtained from the respective distributions. For each parametric distribution,  $n = 250$  samples were randomly generated, the true or exact quantiles were computed, and the quantile function was estimated via kernel smoothing. Note that the quantiles estimated via kernel smoothing are very accurate. In fact, it can be verified that the accuracy increases with sample size. Consequently,

there is a reduction in the over- and under-estimation of extreme quantiles (far right and left in each graph).

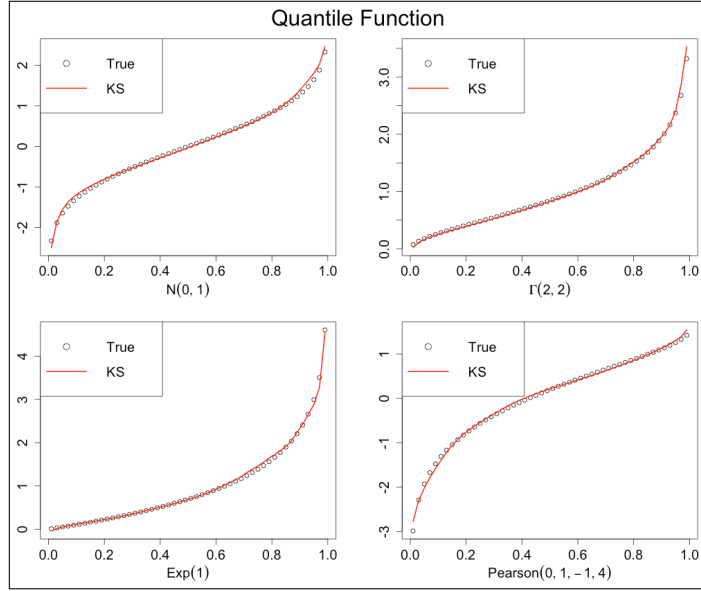


Figure 4.2: Illustration of the accuracy of the estimated quantile function via kernel density estimation or kernel smoothing (KS) for a sample size  $n = 250$ . The four distributions are: normal with mean 0 and standard deviation 1,  $N(0, 1)$ ; Gamma with both shape and rate parameters equal to 2,  $\Gamma(2, 2)$ ; exponential with rate parameter equal to 1,  $\text{Exp}(1)$ ; Pearson with mean 0, variance 1, skewness  $-1$ , and kurtosis 4;  $\text{Pearson}(0, 1, -1, 4)$ .

#### 4.4.2 Joint Chance Constraints

As for the reformulation of data-driven ICCs, we restate equation (4.8) as a joint data-driven CC using the notation in this article:

$$\hat{\mathbb{P}}\{g_j(x) \geq \tilde{\xi}_j, j = 1, \dots, m\} \geq 1 - \alpha'_+ \quad (4.14)$$

Equation (4.14) can be stated in probabilistic terms as follows: “the probability of each random variable (r.v.)  $\tilde{\xi}_j$  to *jointly* achieve values less than or equal to  $g_j(x)$ , for  $j = 1, \dots, m$ , must be at least  $1 - \alpha'_+$ ”. In other words, we require that the estimated joint CDF of the random variables (r.v.s.)  $\tilde{\xi}_j$  evaluated at  $g_j(x)$ , for  $j = 1, \dots, m$ , must be at least  $1 - \alpha'_+$ . Mathematically:

$$\hat{F}_{\tilde{\xi}_1, \dots, \tilde{\xi}_m}(g_1(x), g_2(x), \dots, g_m(x)) \geq 1 - \alpha'_+ \quad (4.15)$$

The joint CDF  $\hat{F}_{\tilde{\xi}_1, \dots, \tilde{\xi}_m}(\cdot)$  estimated with kernel distribution estimation can be expressed using equation (1.24). Therefore, kernel-based reformulated data-driven JCCs can be stated as follows:

$$\frac{1}{n} \sum_{i=1}^n \prod_{j=1}^m \mathcal{K}_j \left( \frac{g_j(x) - X_{j,i}}{h_j} \right) \geq 1 - \alpha'_+ \quad (4.16)$$

where  $X_{j,i}$ , for  $i = 1, \dots, n$ , are data values, and  $h_j$  is the bandwidth of the integrated kernel associated with the  $j$ -th r.v. (see [paragraph 1.2.2.3](#) and [paragraph 1.2.2.3](#)).

**Remark 4.** Even if the original constraints  $g_j(x) \geq \tilde{\xi}_j$ , for  $j = 1, \dots, m$ , are linear in the vector of decision variables  $x$ , i.e.,  $g_j(x) = a_j^\top x$ , where  $a_j$  is a vector of deterministic (fixed) coefficients, the reformulated data-driven JCC is nonlinear in the vector of decision variables. The nonlinearity arises from the mathematical form of the integrated kernel  $\mathcal{K}_j(\cdot)$ . A possible initialization scheme for the resulting nonlinear optimization model is to use the solution to the same problem but with reformulated data-driven ICCs for risk levels smaller than the original model according to equation (4.17) (see derivation in [Appendix C.2](#)).

$$\alpha^{\text{JCC}} \geq \sum_{j=1}^m \alpha_j^{\text{ICC}} \quad (4.17)$$

**Remark 5.** Equation (4.16) can also be used to model conditional JCCs. Let  $\tilde{\xi}_1$  and  $\tilde{\xi}_2$  be continuous r.v.s. with densities  $f_{\tilde{\xi}_1}(\cdot)$  and  $f_{\tilde{\xi}_2|\tilde{\xi}_1}(\cdot)$ , respectively. Notice that  $\tilde{\xi}_2$  is conditional on  $\tilde{\xi}_1$ . Therefore, the kernel-based reformulation of the data-driven JCCs involving the two r.v.s can be written as follows:

$$\begin{aligned} \hat{F}_{\tilde{\xi}_1, \tilde{\xi}_2}(g_1(x), g_2(x)) &\geq 1 - \alpha'_+ \\ \int_{-\infty}^{g_1(x)} \int_{-\infty}^{g_2(x)} \hat{f}_{\tilde{\xi}_1, \tilde{\xi}_2}(\xi_1, \xi_2) d\xi_1 d\xi_2 &\geq 1 - \alpha'_+ \\ \int_{-\infty}^{g_1(x)} \int_{-\infty}^{g_2(x)} \hat{f}_{\tilde{\xi}_1}(\xi_1) \hat{f}_{\tilde{\xi}_2|\tilde{\xi}_1}(\xi_2|\xi_1) d\xi_1 d\xi_2 &\geq 1 - \alpha'_+ \\ \int_{-\infty}^{g_1(x)} \hat{f}_{\tilde{\xi}_1}(\xi_1) d\xi_1 \int_{-\infty}^{g_2(x)} \hat{f}_{\tilde{\xi}_2|\tilde{\xi}_1}(\xi_2|\xi_1) d\xi_2 &\geq 1 - \alpha'_+ \\ \frac{1}{n} \sum_{i=1}^n \mathcal{K}_1\left(\frac{g_1(x) - X_{1,i}}{h_1}\right) \mathcal{K}_2\left(\frac{g_2(x) - X_{2,i}}{h_2}\right) &\geq 1 - \alpha'_+ \end{aligned}$$

where  $\mathcal{K}_1(\cdot)$  and  $\mathcal{K}_2(\cdot)$  are the integrated kernels based on the estimated densities  $\hat{f}_{\tilde{\xi}_1}(\cdot)$  and  $\hat{f}_{\tilde{\xi}_2|\tilde{\xi}_1}(\cdot)$ , respectively. However, it is generally computationally more efficient to directly estimate the joint CDF instead of separately estimating the individual (un)conditional distributions.

**Remark 6.** If the constraint requires each of the r.v.s. to be *greater* than or equal to each  $g_j(x)$ , that is,

$$\hat{\mathbb{P}}\{g_j(x) \leq \tilde{\xi}_j, j = 1, \dots, m\} \geq 1 - \alpha'_+ \quad (4.18)$$

then, for kernel estimation purposes, we use the fact that  $\mathbb{P}\{W \geq w\} = \mathbb{P}\{-W \leq -w\}$ , where  $-W$  is a *multivariate* continuous r.v., and the data values are negated, which is indicated by  $-w$  (note that, in general,  $\mathbb{P}\{W \geq w\} \neq 1 - \mathbb{P}\{W \leq w\}$  in the multivariate case). The kernel-based reformulation is facilitated by the use of product kernels, which are formed by the product of *univariate* kernels. Therefore, for each univariate integrated kernel, we

can use the fact that  $\mathbb{P}\{Z \geq z\} = 1 - \mathbb{P}\{Z \leq z\}$ , where  $Z$  is a *univariate* continuous r.v., as described in **Remark 3** for the case of data-driven ICCs. The reformulated data-driven JCC in equation (4.18) is as follows:

$$\begin{aligned} & \int_{g_1(x)}^{\infty} \cdots \int_{g_m(x)}^{\infty} \hat{f}_{\tilde{\xi}_1, \dots, \tilde{\xi}_1}(\xi_1, \dots, \xi_m) d\xi_1 \cdots d\xi_m \geq 1 - \alpha'_+ \\ & \int_{g_1(x)}^{\infty} \cdots \int_{g_m(x)}^{\infty} \frac{1}{n} \sum_{i=1}^n \prod_{j=1}^m \frac{1}{h_j} K_j \left( \frac{\xi_j - X_{j,i}}{h_j} \right) d\xi_1 \cdots d\xi_m \geq 1 - \alpha'_+ \\ & \frac{1}{n} \sum_{i=1}^n \int_{g_1(x)}^{\infty} \frac{1}{h_1} K_1 \left( \frac{\xi_1 - X_{1,i}}{h_1} \right) d\xi_1 \cdots \int_{g_m(x)}^{\infty} \frac{1}{h_m} K_m \left( \frac{\xi_m - X_{m,i}}{h_m} \right) d\xi_m \geq 1 - \alpha'_+ \\ & \frac{1}{n} \sum_{i=1}^n \left[ 1 - \mathcal{K}_1 \left( \frac{g_1(x) - X_{1,i}}{h_1} \right) \right] \cdots \left[ 1 - \mathcal{K}_m \left( \frac{g_m(x) - X_{m,i}}{h_m} \right) \right] \geq 1 - \alpha'_+ \end{aligned}$$

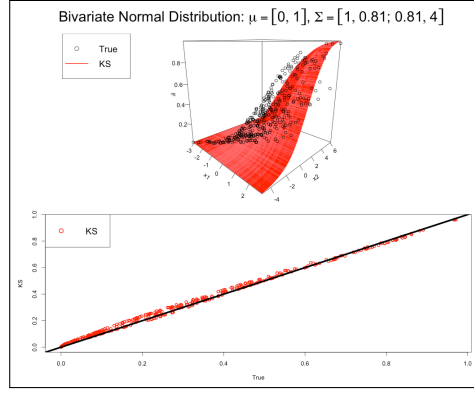
or, in a more compact way,

$$\frac{1}{n} \sum_{i=1}^n \prod_{j=1}^m \left[ 1 - \mathcal{K}_j \left( \frac{g_j(x) - X_{j,i}}{h_j} \right) \right] \geq 1 - \alpha'_+ \quad (4.19)$$

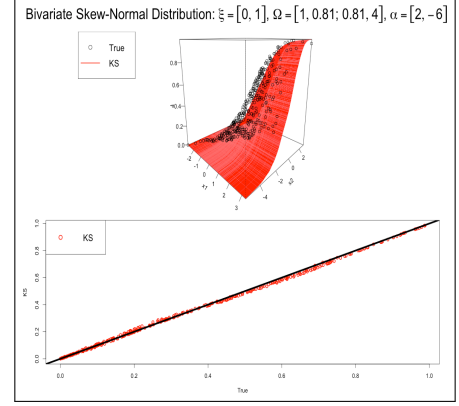
In cases where  $g_j(x) \geq \tilde{\xi}_j$  and  $g_{j'}(x) \leq \tilde{\xi}_{j'}$ , for  $j \neq j'$ , the CCs can be naturally reformulated by using the respective expressions of the term involving the integrated kernels in equations (4.16) and (4.19).

The effectiveness of the kernel-based reformulation approach for data-driven JCCs relies on the accuracy of the estimated joint CDF, which represents the left-hand side of equations (4.16) and (4.19). Figure 4.3 illustrates the accuracy of the estimated joint distribution function when compared to the true values obtained from the respective distributions. For each parametric distribution,  $n = 400$  samples were randomly generated, the true or exact cumulative probabilities were computed, and the joint CDF was estimated via kernel smoothing. The bottom graphs in each subfigure show the accuracy of the estimated cumulative probabilities when compared to the true values (straight line).





(a) Bivariate normal distribution with mean vector  $\mu = [0, 1]$  and covariance matrix  $\Sigma = [1, 0.81; 0.81, 4]$ .



(b) Bivariate skew-normal distribution with location vector  $\xi = [0, 1]$ , scale matrix  $\Omega = [1, 0.81; 0.81, 4]$ , and shape vector  $\alpha = [2, -6]$ .

Figure 4.3: Illustration of the accuracy of the estimated joint CDF via kernel density estimation or kernel smoothing (KS) for a sample size  $n = 400$ . The straight line corresponds to the true predictions of each bivariate distribution.

#### 4.4.3 Calculation of $\phi$ -Divergence Tolerance

In order to use the kernel-based reformulations of data-driven CCs proposed in this chapter, one has to specify the size of the confidence set defined in equation (4.3), that is, the  $\phi$ -divergence tolerance  $d$ . The kernel density estimator is a consistent estimator, i.e., it converges in probability to the true density as the sample size  $n$  increases indefinitely (Wied & Weißbach, 2012). Therefore, it is reasonable to consider  $d$  to be a function of  $n$ , that is,  $d := d(n)$ . Moreover, we assume that  $d$  is nonincreasing as  $n$  increases and tends to zero as  $n$  tends to infinity (Jiang & Guan, 2013).

We propose using point-wise confidence intervals of the predictions of the kernel density/distribution estimator to specify a value for  $d$ . More specifically, given that the data-driven CCs have to be satisfied with nominal confidence level  $1 - \alpha$ , the size of the confidence can be set to the  $100(1 - \alpha)\%$  quantile of the squared point-wise standard errors of the kernel estimator. Mathematically:

$$d = SE_{1-\alpha}^2 \quad (4.20)$$

where  $SE^2$  denotes a vector of length  $n$  of squared point-wise standard errors of the kernel estimator, and its subscript  $1 - \alpha$  corresponds to the quantile probability level. Point-wise standard errors convey the notion of confidence intervals.

The motivation for selecting  $d$  as in equation (4.20) is to guarantee that the “distance” between the true distribution and its kernel estimator is less than or equal to the  $100(1 - \alpha)\%$  largest squared point-wise standard error of the predictions. Note that the standard error is defined as  $SE = \frac{s}{\sqrt{n}}$ , where  $s$  is the

sample standard deviation; thus, the square of the standard error is inversely proportional to the sample size  $n$ . This satisfies the assumption about the dependency of  $d$  with  $n$ , i.e.,  $d$  monotonically decreases as  $n$  increases.

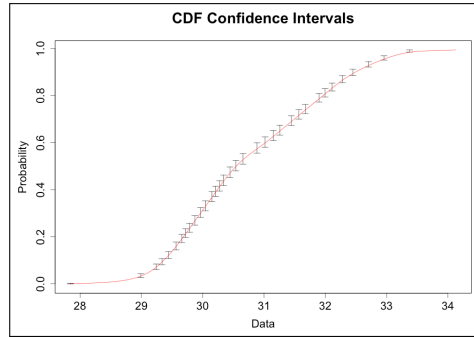
Different approaches to estimate standard errors of kernel estimators have been proposed (Fiorio, 2004; Hall & Horowitz, 2013). They are commonly obtained via bootstrapping or derived formulas based on the asymptotic normality of kernel estimators, such as the following (Racine, 2008):

$$[SE^2]_{\hat{f}_{Y|X}(y|x)} = \frac{f_{Y|X}(y|x)\kappa^q}{n \prod_{j=1}^q h_j f_{Y|X}(x)} + O\left(\frac{1}{n}\right)$$

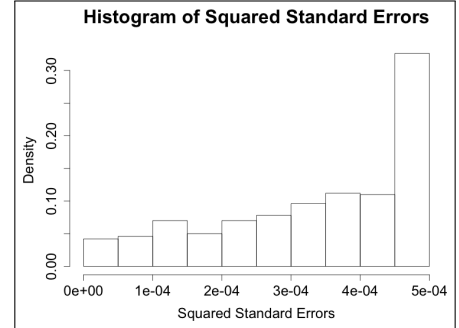
$$[SE^2]_{\hat{F}_{Y|X}(y|x)} = \frac{F_{Y|X}(y|x)[1 - F_{Y|X}(y|x)]\kappa^q}{n \prod_{j=1}^q h_j f_{Y|X}(x)} + O\left(\frac{1}{n}\right)$$

where  $[SE^2]_{\hat{f}_{Y|X}(y|x)}$  and  $[SE^2]_{\hat{F}_{Y|X}(y|x)}$  are the squared point-wise standard errors of the estimated density and distribution, respectively,  $\kappa = \int K^2(u)du$ , and  $q$  is the dimension of the random vector conditioned on (i.e.,  $X$ ).

Figure 4.4 illustrates point-wise confidence intervals for the estimated CDF of the mixture of two Gaussian models presented in paragraph 1.2.2.3 (sample size  $n = 500$ ) and whose estimated PDF is shown in Figure 1.3. The  $\phi$ -divergence tolerance  $d$  corresponds to the  $100(1 - \alpha)\%$  quantile of the squared standard errors shown in Figure 4.4(b).



(a) Estimated kernel CDF with point-wise confidence intervals that correspond to standard errors of the estimates.



(b) Histogram of squared point-wise standard errors from the kernel estimation process.

Figure 4.4: Illustration of point-wise confidence intervals constructed by the estimated standard errors from kernel smoothing.

In the next subsection, we propose a way to handle small sample size cases, i.e., when the lower accuracy of kernel estimators may compromise the estimated quantiles for data-driven ICCs. The proposed expression for  $d$  for small sample sizes uses similar rationale as for the KS-based approach, i.e., we make use of point-wise confidence intervals to specify a value for  $d$ .

#### 4.4.4 Addressing Small Sample Size Cases

In this subsection, we are only concerned with univariate cases since multivariate cases require even larger sample sizes (the more random variables, the larger  $n$  should be for accurate smoothing). Smoothing techniques such as kernel smoothing (KS) are generally not considered suitable for small sample sizes. Even though there does not seem to be an established theoretical minimum sample size to successfully apply KS, a general rule-of-thumb is  $n \geq 100$ . For smaller sample sizes, there can be over- and under-estimation of (especially extreme) quantiles as discussed in [Subsection 4.4.1](#). In addition, quantiles that are over- and under-estimated via KS may violate physical bounds on the historical data, whereas empirical quantiles always lie within the data range. We propose a way to calculate the value of the  $\phi$ -divergence tolerance  $d$  using point-wise confidence intervals obtained from the empirical CDF,  $\hat{F}_n(\cdot)$ , instead of the quantile function estimated with KS.

Recall that the reformulation of a data-driven ICC  $j$  requires the specification of its reduced risk level  $\alpha'_j$  (see equation (4.9)), which in turn depends on the value of the divergence tolerance  $d$  (see [Table 4.1](#)). The main idea of the proposed approach is to evaluate the  $\hat{F}_n(\cdot)$  at the available data points for the uncertain parameter, and construct point-wise confidence intervals around the predictions.

The empirical distribution is defined by

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\{X_i \leq x\}}(x)$$

where  $\mathbf{1}_A(\cdot)$  is the indicator function, which equals to 1 if  $A$  is true, and 0 otherwise. Note that for a fixed point  $x \in \mathbb{R}$  and any sample size  $n$ , the quantity  $n\hat{F}_n(x)$  denotes the number of data points  $X_i$  that are less than or equal to  $x$ , i.e., the number of “successes” in a sequence of  $n$  “trials”, and each success has the probability of  $\hat{F}_n(x)$ , i.e., the ratio of number of successes to the number of trials. This is a well known fact: the quantity  $n\hat{F}_n(x)$  has a binomial distribution with parameters  $n$  and success probability  $\hat{F}_n(x)$ , denoted by  $B(n, \hat{F}_n(x))$ .

The Wilson score-test-based method can be used to compute point-wise confidence intervals of binomial distributions, since it is generally preferred among a few other methods ([Agresti & Coull, 1998](#)). The function `binconf` in the R `Hmisc` package ([Harrell \*et al.\*, 2014](#)) implements the Wilson method and others. It computes point-wise lower and upper bounds of the confidence interval given a risk level  $\alpha$  (i.e.,  $1 - \alpha$  confidence level).

Denote the point-wise lower and upper bounds of a binomial confidence interval with risk level  $\alpha$  by  $BinLo_\alpha(x)$  and  $BinUp_\alpha(x)$ , respectively. The deviations between the upper and lower bounds of the point-wise confidence interval and the predicted cumulative probabilities generate two distributions. The larger the sample size  $n$ , the smaller the deviations. Similarly to the approach proposed in [Subsection 4.4.3](#), we may specify a value for  $d$  that is

greater than a fraction of all the deviations, i.e., a quantile of the distribution of deviations. Since there are two distributions of deviations (i.e., predicted minus lower and upper minus predicted), we propose setting the  $\phi$ -divergence tolerance  $d$  as follows:

$$d = \min \left\{ HD_{1-\alpha} \left( \hat{F}_n(x) - BinLo_\alpha(x) \right), HD_{1-\alpha} \left( BinUp_\alpha(x) - \hat{F}_n(x) \right) \right\} \quad (4.21)$$

where  $HD(\cdot)$  is the Harrell-Davis (H-D) distribution-free (empirical) quantile function (Harrell & Davis, 1982). The argument of the  $HD(\cdot)$  function is an array of values. The subscript  $1 - \alpha$  denotes the quantile probability level. The H-D quantile estimator has demonstrated to be superior to traditional empirical quantile estimators (based on one or two order statistics) for small sample sizes (e.g.,  $n \leq 60$ ). Note that as  $n$  increases, the confidence intervals become smaller; therefore,  $d$  monotonically decreases as  $n$  increases.

The expression in equation (4.21) can be interpreted as follows:  $d$  is the minimum between the  $100(1 - \alpha)\%$  quantile (using the H-D estimator) of the deviations between estimated empirical CDF and the binomial lower and upper bounds for the given confidence level. Figure 4.5 illustrates the binomial confidence intervals for the empirical CDF of the mixture of two Gaussian models presented in paragraph 1.2.2.3, but with sample size  $n = 24$ .

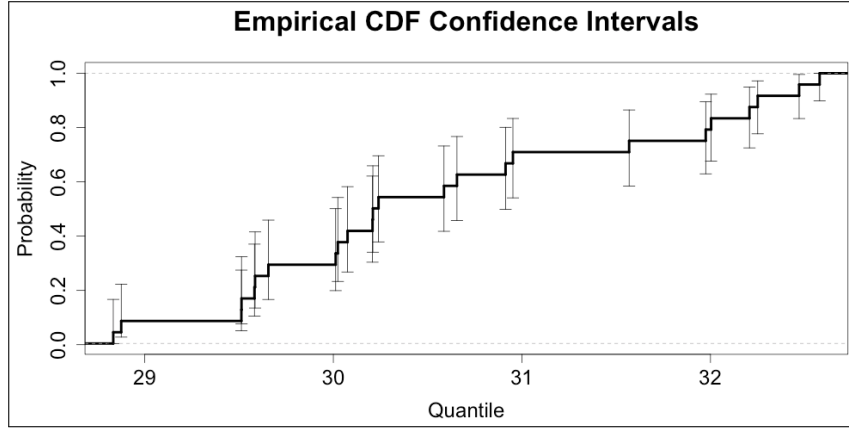


Figure 4.5: Illustration of point-wise binomial confidence intervals for an empirical CDF for small sample sizes.

Since the H-D estimator is used instead of the quantile function estimated via KS, equations (4.11) and (4.13) are rewritten as follows:

$$g_j(x) \geq HD_{1-\alpha'_{j,+}}([X_{j,1}, \dots, X_{j,n}]), \quad j = 1, \dots, m \quad (4.22)$$

$$g_j(x) \leq HD_{\alpha'_{j,+}}([X_{j,1}, \dots, X_{j,n}]), \quad j = 1, \dots, m \quad (4.23)$$

where  $[X_{j,1}, \dots, X_{j,n}]$  represents an array of data values for each random variable  $\tilde{\xi}_j$ , for  $i = 1, \dots, n$ , and  $\alpha'_{j,+}$ , for  $j = 1, \dots, m$ , is obtained in the same way as for the kernel-based reformulation approach (see Section 4.3).

## 4.5 Algorithm for the Initialization of JCC Models

The algorithm described here can be used in the context of both classical and data-driven joint chance constraints. For a cleaner presentation, we use the acronyms JCCs and ICCs to refer to either classical or data-driven approaches.

Optimization problems with JCCs can be computationally more intensive to solve than problems with ICCs. One major contributor to that is the nonlinear reformulation of the JCCs. However, the solution of nonlinear optimization problems can be expedited by proper initialization of the values of the variables. We propose providing a starting point for a problem with JCCs by using the solution to the same problem, but with the ICCs instead. That is, each JCC is converted into separate ICCs whose individual risk levels satisfy the relationship in equation (4.17) (see derivation in Appendix C.2). In case the solution to the problems with ICCs yields one or more infeasible JCCs (especially when the confidence level  $1 - \alpha$  approaches the value of 1), a feasibility problem is solved first, and then its solution is used to initialize the model with JCCs. The initialization algorithm can reduce the total solution time if these constraints represent a relatively small part of all the constraints in the model. Before describing the algorithm, we define the feasibility problem.

The feasibility problem can be used to find a feasible solution to the JCCs alone. A possible objective is to minimize a distance measure between the values of the variables present in those constraints and their respective values from the solution to the problem with ICCs. In this way, the solution to the feasibility problem will be as close as possible to the one from the problem with ICCs, which is a good starting point for the problem with JCCs. Let  $\mathbf{x}^{\text{ICC}}$  be a constant vector initialized to the optimal solution to the problem with ICCs. We may define the feasibility problem as follows:

$$\begin{aligned} \min_x \quad & ||x - \mathbf{x}^{\text{ICC}}||_2^2 \\ \text{s.t.} \quad & \text{JCCs, such as equations (4.14), (4.18), or a combination of both} \end{aligned} \tag{4.24}$$

where  $||\cdot||_2^2$  in the objective function is the  $L^2$  norm squared. For models with few JCCs relative to all other constraints, the feasibility problem is a smaller problem that may be solved more efficiently than the original problem with JCCs.

For ease of exposition of the algorithm, we define the original optimization problem with JCCs and the problem with associated ICCs in equations (4.25) and (4.26), respectively. A possible approach to select the individual risk levels is as follows (see equation (4.17)): let  $\alpha_j^{\text{ICC}} = \alpha^{\text{ICC}}$ , for  $j = 1, \dots, m$ ; thus,

$$\alpha^{\text{ICC}} \leq \frac{1}{m} \alpha^{\text{JCC}}.$$

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & \mathbb{P} \left\{ g_j(x) \geq \tilde{\xi}_j, j = 1, \dots, m \right\} \geq 1 - \alpha^{\text{JCC}} \end{aligned} \tag{4.25}$$

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & x \in \mathcal{X} \\ & \mathbb{P} \left\{ g_j(x) \geq \tilde{\xi}_j \right\} \geq 1 - \alpha_j^{\text{ICC}}, \quad j = 1, \dots, m \end{aligned} \tag{4.26}$$

where  $\mathcal{X}$  is a compact set that contains constraints that are not JCCs nor ICCs.

The initialization algorithm is shown in [Figure 4.6](#). In step 1, the associated problem with ICCs in equation (4.26) is solved yielding  $x^{\text{ICC}}$ . In step 2, each JCC is evaluated at the solution to the problem with ICCs in order to determine its feasibility. If all JCCs are feasible, then the original problem in equation (4.25) is initialized with  $x^{\text{ICC}}$  and solved; the algorithm stops. Otherwise, the feasibility problem is solved in step 4. In step 5, the variables present in the JCCs are fixed in the problem with ICCs, which is resolved using the bounds on the individual risk levels as in step 1. The purpose of step 5 is to guarantee that the solution to the corresponding problem with ICCs is feasible for all JCCs present in the original problem, thus providing a feasible starting point for the original problem. Finally, the solution to this problem with ICCs is used to initialize the original problem with JCCs (step 3), thus concluding the algorithm.

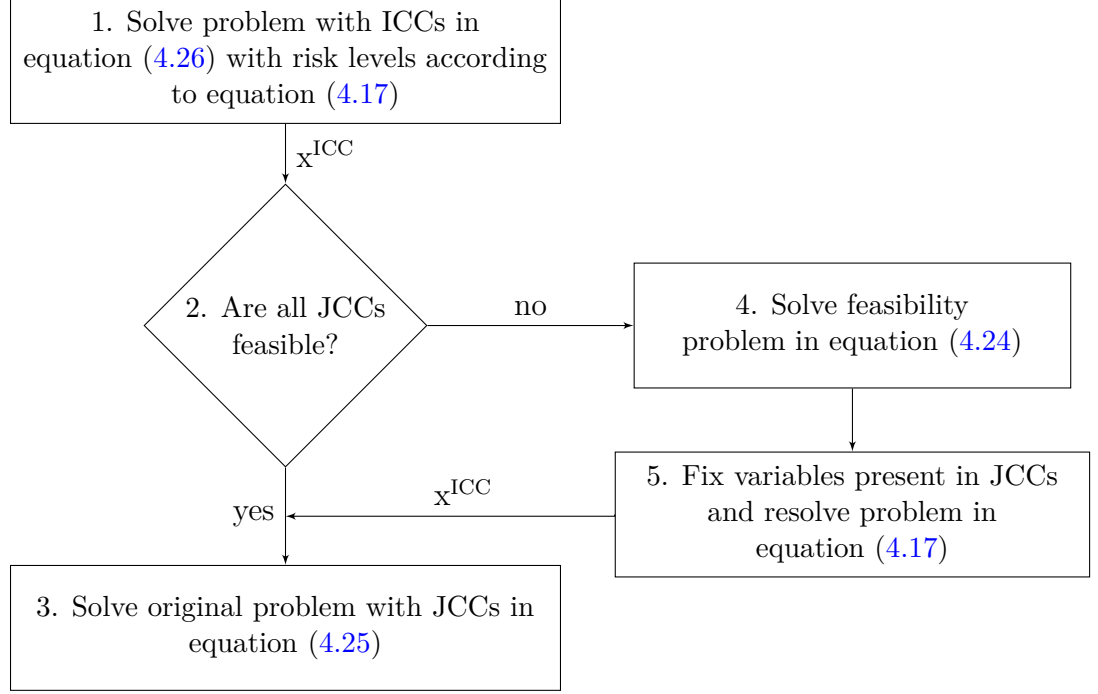


Figure 4.6: Algorithm for the initialization of problems with JCCs.

## 4.6 Numerical Examples

This section contains two production planning examples. The first example is a motivating example to illustrate the proposed kernel-based reformulation approach for data-driven individual and joint CCs. The second example is a real-world test case for a network of chemical plants that are geographically located in different sites. In the industrial test case, we illustrate how chance-constrained optimization models can be interpreted as multi-objective optimization models for which a Pareto efficient frontier can be obtained.

### 4.6.1 Motivating Example

Figure 4.7 shows the network of the motivating production planning example. It consists of a raw material  $A$ , an intermediate product  $B$ , finished products  $C$  and  $D$  (only product  $D$  can be stored), and facilities (plants)  $P1$ ,  $P2$ , and  $P3$ . Product  $C$  can also be purchased from a supplier, or in the case of multiple sites, it could be transferred from another site that also produces it. Both plants  $P2$  and  $P3$  have uncertain production capacity.

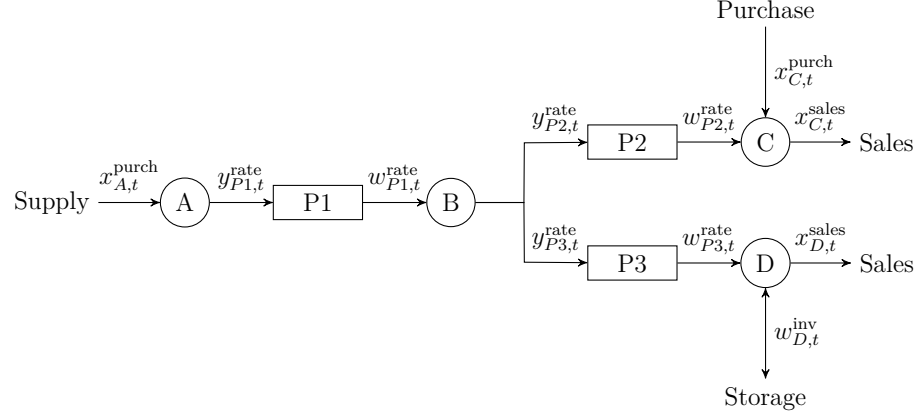


Figure 4.7: Network structure for the motivating Example 1.

The Linear Programming (LP) formulation has the following main elements: variables corresponding to the inlet/outlet flow rates to/from facility  $f$  in time period  $t$ ,  $y_{f,t}^{\text{rate}}$  and  $w_{f,t}^{\text{rate}}$  respectively; production yields for each facility,  $\theta_f$ ; and demands for each finished product  $m \in FP$  in time period  $t$ ,  $\gamma_{m,t}$ . The deterministic multiperiod optimization model, (*PPDet*), is given as follows:

$$\max w^{\text{profit}} \quad (4.27a)$$

$$\text{s.t. } w_{f,t}^{\text{rate}} = \theta_f y_{f,t}^{\text{rate}} \quad \forall f \in F, t \in T \quad (4.27b)$$

$$x_{C,t}^{\text{sales}} = w_{P2,t}^{\text{rate}} + x_{C,t}^{\text{purch}} \quad \forall t \in T \quad (4.27c)$$

$$w_{D,t}^{\text{inv}} = w_{D,t-1}^{\text{inv}} + w_{P3,t}^{\text{rate}} - x_{D,t}^{\text{sales}} \quad \forall t \in T \quad (4.27d)$$

$$w_{P1,t}^{\text{rate}} = y_{P2,t}^{\text{rate}} + y_{P3,t}^{\text{rate}} \quad \forall t \in T \quad (4.27e)$$

$$(PPDet) \quad x_{A,t}^{\text{purch}} = y_{P1,t}^{\text{rate}} \quad \forall t \in T \quad (4.27f)$$

$$x_{m,t}^{\text{sales}} + \text{slack}_{m,t}^{\text{sales}} = \gamma_{m,t} \quad \forall m \in FP, t \in T \quad (4.27g)$$

$$w_{f,t}^{\text{rate}} \leq w_f^{\text{rate,max}} + \text{slack}_{f,t}^{\text{max,cap}} \quad \forall f \in F, t \in T \quad (4.27h)$$

$$w_{f,t}^{\text{rate}} \geq w_f^{\text{rate,min}} - \text{slack}_{f,t}^{\text{min,cap}} \quad \forall f \in F, t \in T \quad (4.27i)$$

$$w_{D,t}^{\text{inv}} \leq w_{D,t}^{\text{inv,max}} \quad \forall t \in T \quad (4.27j)$$

$$x_{A,t}^{\text{purch}} \leq x_{A,t}^{\text{purch,max}} \quad \forall t \in T \quad (4.27k)$$

where constraints (4.27b) relate the output flows with the input flows through the yield of each facility  $f$ , constraints (4.27c) – (4.27f) represent material and inventory balances, equations (4.27g) represent the demand satisfaction and slack variables are employed to account for possible unmet demand, constraints (4.27h) – (4.27k) are limitations in the flows, storage, raw material availability,



and capacity violations, respectively, and the profit is calculated as follows:

$$\begin{aligned}
w^{\text{profit}} = & \sum_{t \in T} \left[ \sum_{m \in FP} \text{SP}_{m,t} x_{m,t}^{\text{sales}} - \sum_{f \in F} \text{OPC}_{f,t} w_{f,t}^{\text{rate}} \right. \\
& - \sum_{m \in M:MPUR=1} \text{PC}_{m,t} x_{m,t}^{\text{purch}} - \sum_{m \in M:MINV=1} \text{IC}_{m,t} w_{m,t}^{\text{inv}} - \sum_{m \in FP} \text{PEN}_{m,t} \text{slack}_{m,t}^{\text{sales}} \\
& \left. - \sum_{f \in F} \text{PEN}_{f,t} (\text{slack}_{f,t}^{\text{max,cap}} + \text{slack}_{f,t}^{\text{min,cap}}) \right]
\end{aligned}$$

where  $\text{SP}_{m,t}$  is the selling price of material  $m$  in period  $t$ ,  $\text{OPC}_{f,t}$  is the operating cost of facility  $f$  in period  $t$ ,  $\text{PC}_{m,t}$  is the purchase cost of material  $m$  in period  $t$ ,  $\text{IC}_{m,t}$  is the inventory cost of material  $m$  in period  $t$ , and  $\text{PEN}_{m,t}$  denotes the penalty associated with unmet demand.

The uncertainty in this example is represented by the historical variability of the maximum production capacities of plants  $P2$  and  $P3$ . In other words, the two random variables (uncertain parameters) are  $\tilde{w}_{P2}^{\text{rate,max}}$  and  $\tilde{w}_{P3}^{\text{rate,max}}$  (note that they are time-independent). All other parameters are considered deterministic for the sake of this example, and four time periods were specified. We assume that the data generating processes for  $\tilde{w}_{P2}^{\text{rate,max}}$  and  $\tilde{w}_{P3}^{\text{rate,max}}$  are the Gaussian models  $N(27, 7)$  and  $N(28.5, 9)$ , respectively. Note that in reality one does not usually know the true distribution of the uncertain parameters. The industrial example in the next subsection considers real plant data. For illustration purposes, we will consider both individual and joint chance constraints (ICCs and JCCs), which will form two different models: an LP model (ICCs) and an NLP model (JCC).

Equation (4.27h), for  $f = \{P2, P3\}$ , can be written as ICCs as follows:

$$\mathbb{P} \left\{ w_{P2,t}^{\text{rate}} \leq \tilde{w}_{P2}^{\text{rate,max}} \right\} \geq 1 - \alpha_{P2} \quad \forall t \in T \quad (4.28)$$

$$\mathbb{P} \left\{ w_{P3,t}^{\text{rate}} \leq \tilde{w}_{P3}^{\text{rate,max}} \right\} \geq 1 - \alpha_{P3} \quad \forall t \in T \quad (4.29)$$

Note that the associated slack variables are removed from the model. Using equation (4.13), the kernel-based reformulation of the data-driven ICCs yields:

$$w_{P2,t}^{\text{rate}} \leq \hat{F}_{\tilde{w}_{P2}^{\text{rate,max}}}^{-1}(\alpha'_{P2,+}) \quad \forall t \in T \quad (4.30)$$

$$w_{P3,t}^{\text{rate}} \leq \hat{F}_{\tilde{w}_{P3}^{\text{rate,max}}}^{-1}(\alpha'_{P3,+}) \quad \forall t \in T \quad (4.31)$$

where  $\hat{F}_{\tilde{w}_{P2}^{\text{rate,max}}}^{-1}(\cdot)$  and  $\hat{F}_{\tilde{w}_{P3}^{\text{rate,max}}}^{-1}(\cdot)$  are quantile functions estimated via KS. Figure 4.8 illustrates the histogram for both uncertain parameters and the estimated quantile functions. The (linear) production planning model with reformulated data-driven ICCs, (*PPICC*), is as follows:

$$\begin{aligned}
& \max w^{\text{profit}} \\
& \text{s.t.} \quad \text{Constraints in equations (4.27b) to (4.27g)} \\
(PPICC) \quad & w_{P1,t}^{\text{rate}} \leq \tilde{w}_{P1}^{\text{rate,max}} + \text{slack}_{P1,t}^{\text{max,cap}} \quad \forall t \in T \\
& w_{P2,t}^{\text{rate}} \leq \hat{F}_{\tilde{w}_{P2}^{\text{rate,max}}}^{-1}(\alpha'_{P2,+}) \quad \forall t \in T \\
& w_{P3,t}^{\text{rate}} \leq \hat{F}_{\tilde{w}_{P3}^{\text{rate,max}}}^{-1}(\alpha'_{P3,+}) \quad \forall t \in T \\
& \text{Constraints in equations (4.27i) to (4.27k)}
\end{aligned}$$

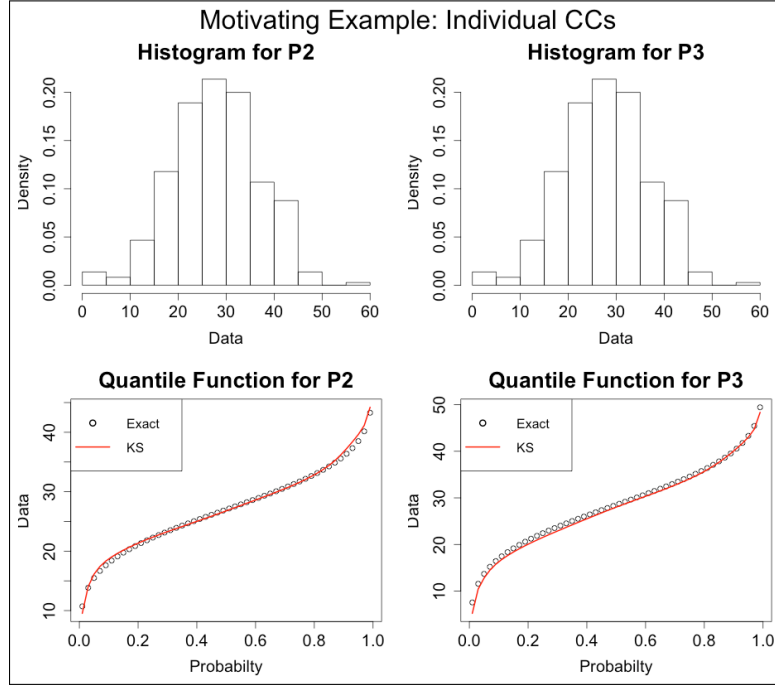


Figure 4.8: *Top*: histograms for the randomly generated historical data for the maximum capacity of plants  $P2$  and  $P3$  (sample size  $n = 365$  each) in the motivating example. *Bottom*: estimated empirical and kernel-based quantiles (EQ and KS, respectively).

In the joint case, we consider that the two Gaussian models for  $\tilde{w}_{P2}^{\text{rate,max}}$  and  $\tilde{w}_{P3}^{\text{rate,max}}$  are independent; thus, they are uncorrelated, which yields a bivariate normal distribution with mean vector  $\mu = [27, 28.5]$  and covariance matrix  $\Sigma = [49, 0; 0, 81]$ . Equation (4.27h), for  $f = \{P2, P3\}$ , can also be written as a JCC as follows:

$$\mathbb{P} \left\{ w_{P2,t}^{\text{rate}} \leq \tilde{w}_{P2}^{\text{rate,max}}, w_{P3,t}^{\text{rate}} \leq \tilde{w}_{P3}^{\text{rate,max}} \right\} \geq 1 - \alpha \quad \forall t \in T \quad (4.33)$$

Using equation (4.19), the kernel-based reformulation of the data-driven JCC

yields:

$$\frac{1}{n} \sum_{i=1}^n \left[ 1 - \mathcal{K}_{P_2} \left( \frac{w_{P_2,t}^{\text{rate}} - w_{P_2,i}^{\text{rate,max}}}{h_{P_2}} \right) \right] \left[ 1 - \mathcal{K}_{P_3} \left( \frac{w_{P_3,t}^{\text{rate}} - w_{P_3,i}^{\text{rate,max}}}{h_{P_3}} \right) \right] \geq 1 - \alpha'_+ \quad \forall t \in T \quad (4.34)$$

where  $w_{P_2,i}^{\text{rate,max}}$  and  $w_{P_3,i}^{\text{rate,max}}$  are data points, and  $\mathcal{K}_{P_2}(\cdot)$  and  $\mathcal{K}_{P_3}(\cdot)$  are integrated kernels with respective bandwidths  $h_{P_2}$  and  $h_{P_3}$ . When implementing the reformulated constraints in an optimization model, we choose the Gaussian integrated kernel (see equation (1.22) in paragraph 1.2.2.3) since it is smooth, i.e., it does not contain discontinuities as the other kernels, such as Epanechnikov and Tri-cube (see Table 1.2). Figure 4.9 illustrates the histogram for both uncertain parameters and the estimated quantile functions. The (non-linear) production planning model with the reformulated data-driven JCC, (*PPJCC*), is as follows:

$$\begin{aligned} & \max w^{\text{profit}} \\ & \text{s.t.} \quad \text{Constraints in equations (4.27b) to (4.27g)} \\ & \quad w_{P_1,t}^{\text{rate}} \leq w_{P_1}^{\text{rate,max}} + \text{slack}_{P_1,t}^{\text{max,cap}} \quad t \in T \\ (PPJCC) \quad & \frac{1}{n} \sum_{i=1}^n \left[ 1 - \mathcal{K}_{P_2} \left( \frac{w_{P_2,t}^{\text{rate}} - w_{P_2,i}^{\text{rate,max}}}{h_{P_2}} \right) \right] \cdot \\ & \left[ 1 - \mathcal{K}_{P_3} \left( \frac{w_{P_3,t}^{\text{rate}} - w_{P_3,i}^{\text{rate,max}}}{h_{P_3}} \right) \right] \geq 1 - \alpha'_+ \quad \forall t \in T \\ & \text{Constraints in equations (4.27i) to (4.27k)} \end{aligned}$$

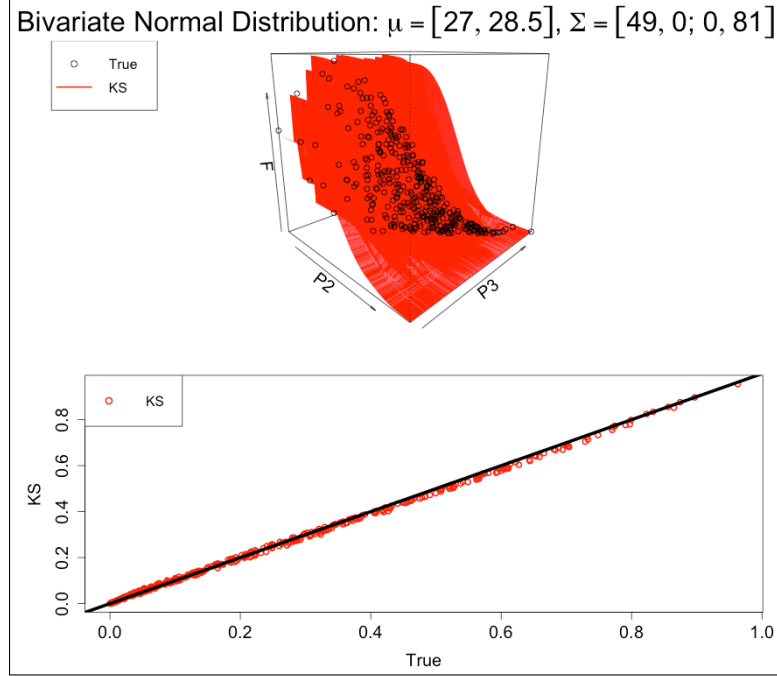
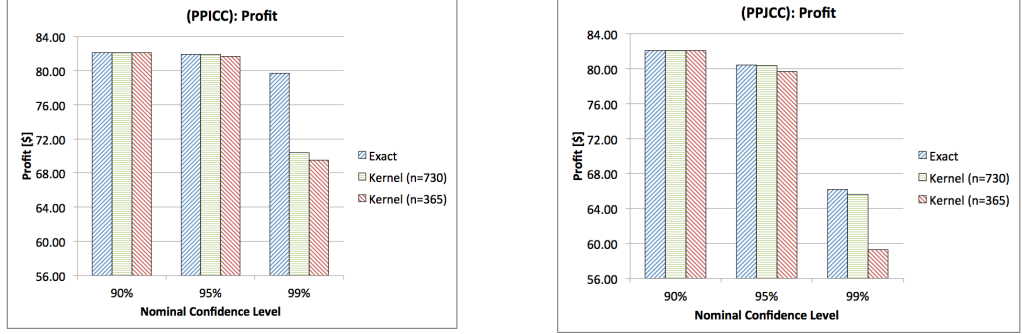


Figure 4.9: Estimated joint cumulative distribution function for the maximum capacity of plants  $P_2$  and  $P_3$  (sample size  $n = 365$ ) in the motivating example. The straight line corresponds to the true predictions of the bivariate normal distribution.

All models were implemented in AIMMS 3.13 and solved on a desktop computer with the following specifications: Dell Optiplex 990 with 4 Intel® Core™ i7-2600 CPUs at 3.40 GHz (total 8 threads), 8 GB of RAM, and running Windows 7 Enterprise. The linear and nonlinear solvers used were GUROBI 5.6 and IPOPT 3.10.1, respectively. The R `np` package was used for all kernel smoothing computations (see [paragraph 1.2.2.3](#)). The reduced risk level  $\alpha'_+$  was calculated using the K-L divergence formula (see [Table 4.1](#)). The complete problem data are available in [Appendix C.3](#).

We begin the analysis of the results with the optimal profit value (penalties not included) for all models at different nominal confidence levels or reliability levels  $1 - \alpha$ . In [Figure 4.10](#), we compare the profit values for the optimization models with exact (known) distributions and with kernel-based reformulated CCs, and the influence of the sample size  $n$ . Note that, for this motivating example, the data were generated from specified Gaussian models; therefore, exact quantiles (for ICCs) and bivariate normal cumulative probabilities (for JCC) were used to assess the quality of the proposed kernel-based reformulation. From [Figure 4.10](#), for 90% nominal confidence level, the kernel-based approach yields precisely the same solution as for the case where exact information for the distribution is used. For higher values of nominal reliability level, the difference increases, but not considerably. The more data points are used for the estimation of the underlying distribution, the more accurate

becomes the estimated distribution. Moreover, the reduced risk level  $\alpha'_+$  becomes closer to the nominal risk level  $\alpha$ . As expected, the larger the sample size  $n$ , the closer the results for the kernel-based approach are to the solution using assumed exact distributions. Note that the JCC problem exhibits lower profits than the ICC problem, since the former is more restrictive than the latter.

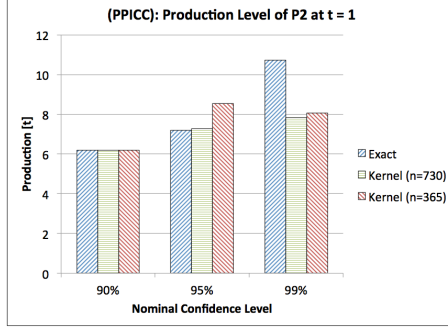


(a) Profit values for exact and kernel-based reformulation with individual chance constraints.

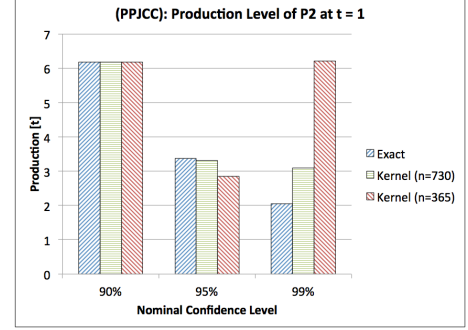
(b) Profit values for exact and kernel-based reformulation with joint chance constraint.

Figure 4.10: Comparison of the objective function value of the motivating example for different models (exact vs. kernel-based (KS) reformulations).

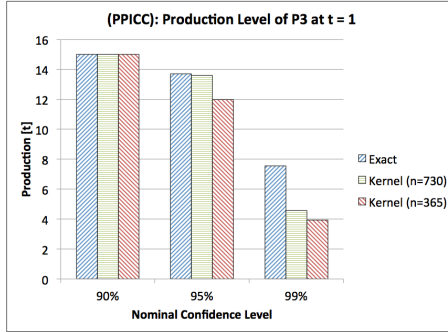
A similar analysis can be performed for the decision variables. For example, in [Figure 4.11](#), the optimal production levels of plants  $P2$  and  $P3$  obtained with kernel-based approach are very similar to those obtained when exact distributions are used. Once again, the larger the sample size  $n$ , the smaller the difference between estimating distributions and using the exact distributions. The increasing flow rates out of  $P2$  at  $t = 1$  for increasing nominal confidence levels might seem counterintuitive at first, since as the confidence levels increase the flow rates are further constrained from above. However, the flow rate out of  $P2$  is still free to increase in order to satisfy the demand of product  $C$ , which is followed by a decrease in its purchased amounts from an external source (not shown). In contrast, the flow rate out of  $P3$  at  $t = 1$  decrease with increasing confidence levels as expected. This is due to the fact that the ICC for  $P3$  is binding (i.e., the flow rate reached its upper bound, which is the estimated quantile shown in [Table C.3](#)), whereas the ICC for  $P2$  is non-binding.



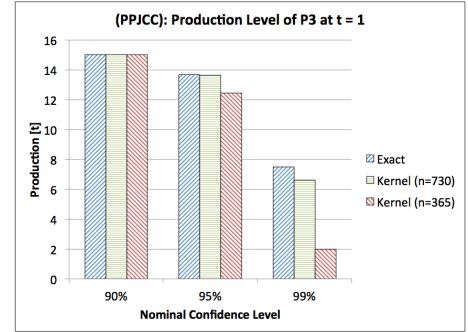
(a) Production levels of plant  $P2$  for exact and kernel-based reformulation with individual chance constraints.



(b) Production levels of plant  $P2$  for exact and kernel-based reformulation with joint chance constraint.



(c) Production levels of plant  $P3$  for exact and kernel-based reformulation with individual chance constraints.



(d) Production levels of plant  $P3$  for exact and kernel-based reformulation with joint chance constraint.

Figure 4.11: Comparison of the production levels of plants  $P2$  and  $P3$  of the motivating example for different models (exact vs. kernel-based (KS) reformulations).

All optimization problems were solved in less than one second each. Table 4.2 summarizes the problem sizes. The additional 8 variables in the deterministic model ( $PPDet$ ) correspond to the slack variables  $slack_{f,t}^{\max, \text{cap}}$ , for  $f = \{P2, P3\}$  and  $t = \{1, 2, 3, 4\}$ , which were removed when modeling constraints in equation (4.27h) as CCs. We note that the model ( $PPJCC$ ) is a nonconvex NLP. It has fewer constraints than the model ( $PPICC$ ), because for each JCC there are two associated ICCs. More specifically, for each time period  $t$ , the two ICCs in equations (4.30) and (4.31) in the model ( $PPICC$ ) are “joined” to form the JCC in equation (4.34) in the model ( $PPJCC$ ). In the computational experiments, the NLP solver (IPOPT) converged to the same solution with or without multi-start.

Table 4.2: Problem sizes for motivating example.

Model	( $PPDet$ )	( $PPICC$ )	( $PPJCC$ )
Constraints	68	68	64
Variables	76	68	68

## 4.6.2 Industrial Example

The industrial test case concerns the optimal production planning of a network of production sites. Each site contains several plants that are highly integrated. The plants can also transfer products between sites. At these multiple production sites, with more than 25 production facilities, several products are manufactured. The time horizon of one year is divided into monthly time periods. The objective of the optimization model is to maximize the net present value (NPV). The LP model is similar to the model used in the motivating example (equation (4.27)) with some additional constraints. That is, its constraints represent material and inventory balances across all chemical sites.

In this test case, we focus on the reliability levels of three key plants in the network that are of interest to this test case. The plants are denoted by  $P1$ ,  $P2$ , and  $P3$ . Similar to the motivating example in Subsection 4.6.1, the uncertain parameters are the maximum production capacities of those three key plants. The historical data comprise 610 observations for each key plant as shown in Figure 4.12. Clearly, the real data cannot be accurately represented by typical Gaussian distributions as is commonly assumed in the literature. For the JCC model (NLP model), the joint CDF was estimated, whereas for the ICC model (LP model), the individual quantile functions were estimated. The reduced risk levels  $\alpha'$  were calculated based on the K-L divergence (see Table 4.1). We used the same algebraic modeling environment and optimization solvers as in the motivating example.

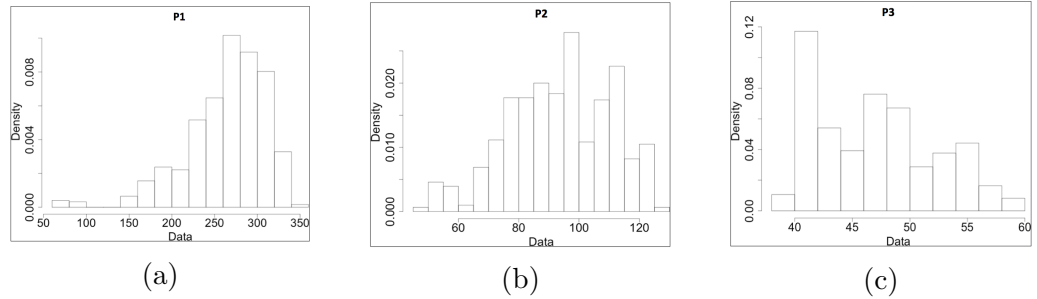


Figure 4.12: Histograms depicting the real data for the maximum production capacities of three key plants in the network: (a) plant  $P1$ , (b) plant  $P2$ , and (c) plant  $P3$ .

The JCC and ICC (when all individual risk levels have the same value) models essentially correspond to bi-objective optimization models in which one objective is to maximize the NPV and the other objective is to maximize the probability of satisfying the chance constraints. By changing the confidence levels of the chance constraints and resolving the optimization model corresponds to applying the  $\epsilon$ -constraint method for multi-objective optimization (Haimes, Lasdon, & Wismer, 1971). The outcome is a Pareto efficiency curve, which represents the trade-off between the two objectives, i.e., the im-

provement of one objective leads to the worsening of the other objective. Figure 4.13 shows the normalized NPV values for different nominal confidence levels  $(1 - \alpha)$ , i.e., the Pareto efficient frontier. Note that the solution to the ICC problem is an upper bound to the solution of the JCC problem, i.e., the JCC model is more conservative than the ICC model for the same nominal confidence level.

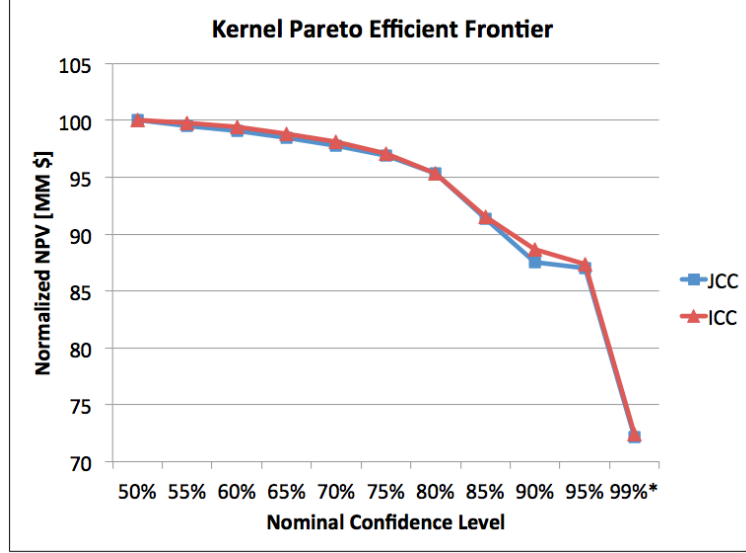


Figure 4.13: Pareto efficient frontier for the industrial test case. For the ICC model, the same reduced risk levels  $\alpha'$  were used for the the three key plants.

The problem sizes are shown in Table 4.3. In this industrial test case, there is one JCC per time period (total of 12 time periods), and each JCC has three variables that correspond to the flow rates out of the three key plants.

Table 4.3: Problem sizes for industrial test case.

Model	Deterministic	ICC	JCC	Feasibility
Constraints	192,240	192,240	192,216	12
Variables	283,217	283,181	283,181	36

In order to speed up the solution of the NLP models, they were initialized with the solution of ICC models for which the individual risk levels satisfy the relationship in equation (4.17). In particular, we set  $\alpha_{P1}^{ICC} = \alpha_{P2}^{ICC} = \alpha_{P3}^{ICC} = \frac{1}{3}\alpha^{JCC}$ , computed the respective individual reduced risk levels  $\alpha'$  with the K-L formula (see Table 4.1), and obtained the estimated quantiles. The effect of properly initializing the JCC models can be seen in Figure 4.14. Speedups ranged from 2 to 6 times. For a nominal confidence level of 99% (i.e.,  $1 - \alpha = 0.99$ ), the JCCs for certain time periods were infeasible when evaluated at the solution of the problem with ICCs. When the initialization algorithm described in Section 4.5 was not applied, the solver IPOPT could not find a



feasible solution and stopped after 1 hour. In contrast, by properly initializing the JCC model using the proposed algorithm, a feasible and (local) optimal solution was found in under 6 minutes for the 99% confidence level case. Thus, the proposed initialization algorithm makes the solution of problems with JCCs more robust and faster. The feasibility problem and each ICC model were each solved in one second or less; thus, their solution times are not reported.

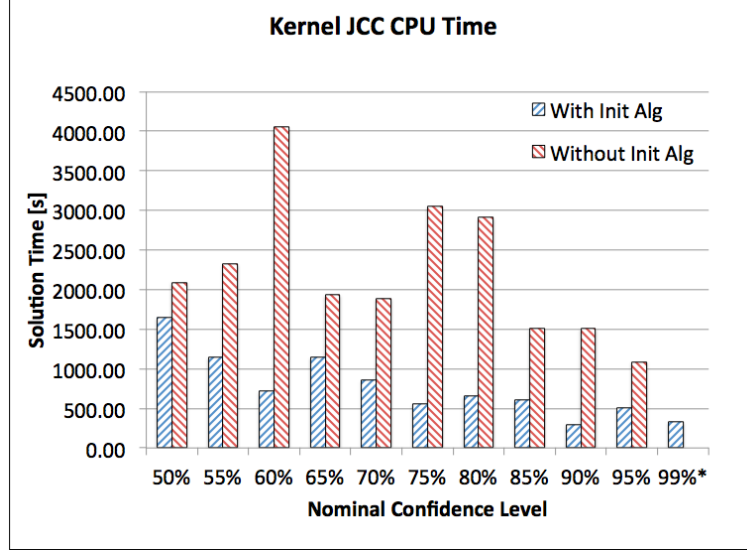


Figure 4.14: Solution times of the JCC model with and without initialization from the solution of ICC models according to algorithm in [Subsection 4.5](#). For a nominal confidence of 99%, the solver could not find a feasible solution without applying the initialization algorithm.

## 4.7 Conclusions

In this work, we have investigated a specific reformulation approach for data-driven conditional and unconditional individual and joint chance constraints (CCs) with right-hand side uncertainty. As opposed to classical CCs that rely on strong assumptions regarding the parametric model for distributions, data-driven CCs use estimated distributions based on available data, i.e., historical or predicted observations of the uncertain parameters. One nonparametric statistical technique to estimate probability densities and distributions as well as to regress quantiles is known as kernel smoothing (KS).

We showed how kernel smoothing can be used to reformulate data-driven CCs into algebraic constraints. In addition, we proposed a novel way of calculating the divergence tolerance (i.e., the size of density-based confidence set that is assumed to contain the unknown true distribution) by using squared point-wise standard errors from the estimation process. For small sample size cases, we proposed a reformulation approach for data-driven individual CCs that use the Harrell-Davis empirical quantile estimator. Another contribution of this work is the proposed algorithm for solving problems with joint CCs

by initializing its variables from the solution of the associated problem with individual CCs.

The proposed reformulation approach and algorithm were applied to numerical examples, including an industrial test case. We showed that the solution to the motivating example when KS was used to estimate distributions was very similar to the solution obtained when exact and assumed to be known distributions were used to describe the uncertainty. For the industrial case study, the proposed initialization procedure enabled the solution of the joint chance-constrained problem for all selected values of the confidence level. Among the problem instances successfully solved with and without the initialization procedure, solution times were up to five times faster when applying the initialization procedure.

# Chapter 5

## Data-Driven Simulation and Optimization Approaches to Incorporate Production Variability in Sales and Operations Planning

### 5.1 Introduction

Sales and Operations Planning (S&OP) is a business and decision-making process through which a company makes certain that tactical plans in every business area balance demand and supply for products. Therefore, S&OP links the corporate strategic plan to daily operations plans. The overall result of the S&OP process is an operating plan to allocate company resources ([Grimson & Pyke, 2007](#)).

Attempts in the literature to systematically survey case studies of the S&OP process adopt the Capability Maturity Model (CMM) ([Paulk, Weber, & Chrissis, 1993](#)). A recent review that surveys some maturity models can be found in [Thomé \*et al.\* \(2012\)](#). Maturity models define stages for the S&OP process that include activities such as meetings, demand forecasting, integration of procurement, production, and distribution plans, and performance measurements. [Figure 5.1](#) illustrates typical stages in the S&OP process. We note that uncertainty and variability affect decisions in both stages 1 (Sales Planning) and 2 (Operations Planning). Forecasting demand for products takes into account future market conditions that are not known exactly (i.e., demand uncertainty), and the operation of plants is subject to unplanned events and imperfect implementation (i.e., production variability).

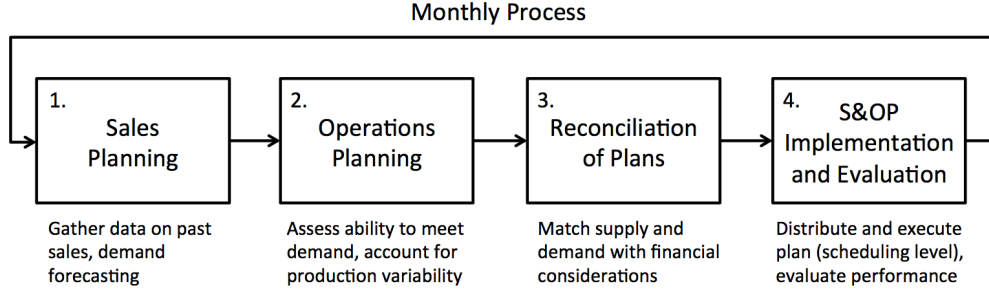


Figure 5.1: Typical stages in the S&OP process. Adapted from [Ling & Goddard \(1988\)](#).

The Operations Research and Management Science (OR&MS) and Process Systems Engineering (PSE) communities have contributed with optimization-based tactical production planning models as well as solution strategies for different industry sectors. The review paper by [Mula \*et al.\* \(2006\)](#) and the book by [Pinedo \(2009\)](#) present an extensive literature survey about models for tactical production planning and scheduling with uncertainty considerations, and classify the literature based on the production planning area and the modeling approach. Production planning and scheduling problems in the chemical, petrochemical, and pharmaceutical industries are reviewed in the works by [Sung & Maravelias \(2007\)](#); [Verderame \*et al.\* \(2010\)](#). The authors identify typical sources of uncertainty in different applications and how they are usually modeled in the context of optimization under uncertainty.

Some works have considered hybrid simulation and optimization to account for uncertainty in generating tactical production plans. For instance, [Li, González, & Zhu \(2009\)](#) studied a dedicated remanufacturing system of electronic products with stochastic batch arrival times. The system is modeled in the discrete-event simulator Arena® 10.0 by Rockwell Software, and its objective is to analyze the effect of operational changes on the profit performance of this dedicated remanufacturing system. The optimization approach was based on Genetic Algorithms. [Lim, Alpan, & Penz \(2014\)](#) propose a simulation-optimization approach for managing S&OP in a problem related to the automotive industry. The stochastic parameter is the demand that is assumed to be uniformly distributed. The simulation part is coded in the Java programming language, and the optimization formulation accounts for multiple criteria through  $\epsilon$ -constraints (see [Subsection 5.4.2](#)). We note that, on the one hand, there has been a modest effort to address the effect of uncertainty in the activities pertaining to sales and procurement planning; on the other hand, there is limited work on incorporating production variability in the operations planning stage.

In this work, we focus on the operations planning stage of the S&OP process (see [Figure 5.1](#)). More specifically, we propose two data-driven optimization-based approaches to account for uncertainty (in this case, production variability) when generating a tactical production plan. The production variability is quantified as the deviation between historical planned and actual production

rates. The statistical technique of quantile regression (Koenker, 2005) is used to model the distribution of deviation values for a given planned rate. This distribution is then sampled from in order to construct scenarios. The main contributions of this work are summarized below.

- Statistical modeling via quantile regression of historical production data to quantify production variability;
- Simulation-optimization and bi-objective optimization frameworks to account for production variability when generating a tactical production plan in the S&OP process; and
- Generated tactical production plan with tradeoff information between average profit and risk (i.e., Pareto efficient frontier).

This paper is organized as follows. Section 5.2 defines the problem and presents the high-level methodology. Section 5.3 provides a brief overview of classical and quantile regression, which are statistical techniques that can be used to model historical production variability. Section 5.4 describes the two proposed optimization-based solution strategies to incorporate production variability in the operations planning stage of the S&OP process. The proposed approach is illustrated by two numerical examples in Section 5.5: motivating example and industrial case study. In the latter, we propose modeling approaches to account for highly integrated networks. Conclusions are drawn in Section 5.6.

## 5.2 Problem Statement and Methodology

The approach proposed in this work is illustrated with an application related to the Chemical Process Industry (CPI). In particular, we deal with Enterprise-wide Optimization (EWO) decisions of highly-integrated chemical production sites, which produce basic chemicals and their downstream derivatives (Wasick, 2009). However, we note that the proposed methodology is general and can be applied to other types of industries.

For this problem, we consider a process network of chemical plants and focus on the effects of production variability in the operations planning stage. We assume that the future monthly demand is given and is deterministic over the planning horizon. Also given is the minimum/maximum installed production capacity of each plant and its production costs. Transportation and inventory holding costs, inventory capacity, and initial inventory are given. Future planned maintenance outages of production plants may also be given. Given a multi-period linear programming (LP) production planning model for the given process network (similar to the model presented in Sahinidis *et al.* (1989), excluding capacity expansion considerations), the objectives are two-fold: (1) propose a production plan incorporating historical production variability data, and (2) measure the performance of the proposed plan in the form of a tradeoff between average profit and risk. Production variability is

incorporated in a two-stage the stochastic programming production planning formulation whose details are given in [Subsection 5.4.2](#).

The overall strategy to generate production plans by taking into account the variability of S&OP data is shown in [Figure 5.2](#). From available historical data that consist of actual and planned production rates, quantile regression models (see [Section 5.3](#)) are built and used to characterize the production variability, i.e., deviation between planned and actual rates. Deviation values are then sampled from these statistical models and used in an optimization-based framework to generate production plans with profit vs. risk tradeoff information (see [Section 5.4](#)). We propose and discuss the advantages and disadvantages of two different frameworks: (1) a simulation-optimization approach, and (2) a bi-objective optimization approach.

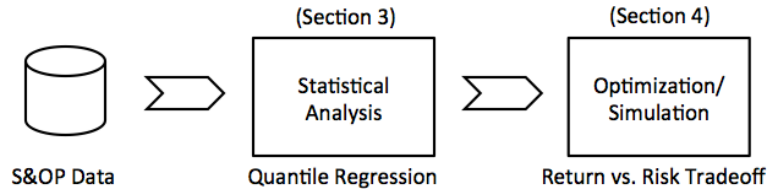


Figure 5.2: Overall strategy to account for historical production variability when generating production plans.

### 5.3 Modeling Production Variability with Quantile Regression

Please see [Subsubsection 1.2.2.1](#) for an overview of classical and quantile regression. In this section, we illustrate how quantile regression can be used to model production variability.

The proposed approach for using quantile regression to model production variability in S&OP data is as follows. From historical planned and actual production rates, the deviation between them is calculated as  $\Delta = \text{Plan} - \text{Actual}$ . The regression analysis consists of regressing  $\Delta$  on Plan, i.e., obtain the regression function  $Q_\alpha[\Delta|\text{Plan} = \text{planned value}]$  for a given probability level  $\alpha$ . Finally, a distribution of  $\Delta$  values given a planned value can be obtained by estimating quantiles for several probability levels (e.g.,  $\alpha = \{0, 0.01, \dots, 1\}$ ). [Figure 5.3](#) shows an example of a  $\Delta$  vs. Plan plot for a given chemical plant and the estimated quantiles conditional on two different planned values. The top plot shows that the distribution of  $\Delta$  values (i.e., conditional quantiles) varies depending on the planned value. This can also be seen from the bottom plots, where the range of  $\Delta$  values is larger for the planned value of 10 w.u. (weight units, bottom left plot) than for the planned value of 1.5 w.u. (bottom right plot).

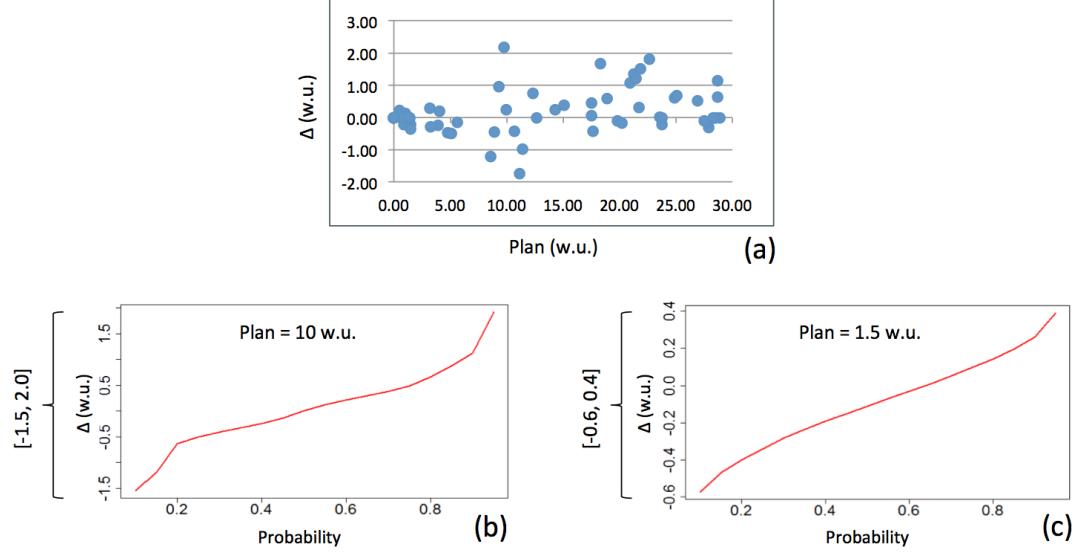


Figure 5.3: Example of modeling production variability with quantile regression, where  $\Delta = \text{Plan} - \text{Actual}$ . Legend: (a)  $\Delta$  vs. Plan plot, (b) estimated quantiles conditional on Plan = 10 w.u. (weight units), and (c) estimated quantiles conditional on Plan = 1.5 w.u..

**Remark.** As will be discussed in the next section, it may be difficult or impossible to employ a quantile regression model to generate samples within an optimization formulation. One possible approximation is to disregard the covariate in the quantile regression (i.e., the Plan values), and generate samples from the distribution of the  $\Delta$  values alone. This is the approach taken in the numerical examples discussed later in this paper.

## 5.4 Simulation and Optimization Frameworks

The objective of the proposed approach is two-fold: (1) account for historical production variability when generating an optimal Sales and Operations Planning (S&OP) production plan, and (2) provide tradeoff information about the generated plan in the form of average profit vs. risk. We describe two optimization-based frameworks whose potential advantages and disadvantages are listed in Table 5.1. The frameworks are detailed in the next two subsections.

Table 5.1: Potential advantages (+) and disadvantages (−) of optimization-based frameworks. Legend: Sim-Opt = Simulation-Optimization framework, Bi-Opt = Bi-Objective Optimization framework,  $\Delta|\text{Plan}$  = deviation conditional on planned values in the context of quantile regression (see [Section 5.3](#)), DFO = Derivative-Free Optimization.

Sim-Opt	Bi-Opt
+ Easy to accommodate for arbitrary $\Delta \text{Plan}$ ;	+ Simultaneous generation of plan and minimization of risk;
− Expensive simulations as number of scenarios increases;	+ Explicit handling of constraint violation;
− No explicit handling of constraint violation;	− Difficult or impossible to accommodate for arbitrary $\Delta \text{Plan}$ ;
− Decrease in efficiency if high-dimensional DFO problem.	− Optimization model may be large and nonlinear.

#### 5.4.1 Simulation-Optimization Framework (Sim-Opt)

The Simulation-Optimization framework (Sim-Opt) consists of alternating between a simulator and a Derivative-Free Optimization (DFO) ([Conn, Scheinberg, & Vicente, 2009](#)) solver as illustrated in [Figure 5.4](#). The purpose of the DFO solver is to set the production target (i.e., generate the S&OP production plan). For a new proposed plan,  $\Delta$  values are generated using quantile regression and used to form scenarios to be evaluated by the simulator. In the simulator, the production rates of the plants that are subject to variability are fixed to the respective plan proposed by the DFO minus the respective  $\Delta$  value, i.e.,  $\text{Production Rate} = \text{Production Target} - \Delta$ . Recall that  $\Delta = \text{Plan} - \text{Actual}$ ; therefore, by subtracting the estimated  $\Delta$  value from the production target, we estimate the actual production rate for a plant.



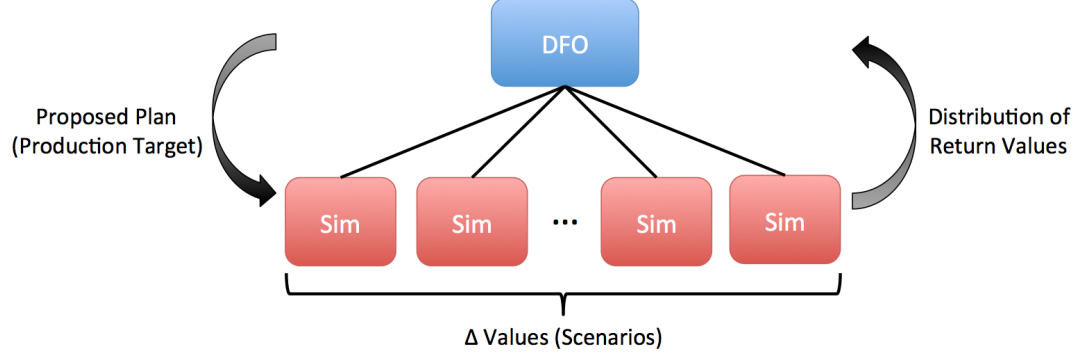


Figure 5.4: Schematic of the Sim-Opt framework. The DFO solver sets the production targets based on which the  $\Delta$  values are estimated and form scenarios for the simulator. The simulator evaluates the current production target and returns a distribution of financial performance values (profit, cost etc.).

The simulator can be a black-box S&OP software or a production planning optimization model that acts as a simulator by fixing certain “input” variables, such as production rates. Different scenarios containing  $\Delta$  values are passed to the simulator, which evaluates the impact of the proposed plan on a performance metric, such as profit or cost. Note that each scenario is independent of the other, which makes this approach amenable to parallelization.

A recent review on DFO solvers and algorithms is given by [Rios & Sahinidis \(2013\)](#). We note that some limitations of DFO include the decrease in efficiency for high-dimensional optimization problems and the number of necessary function evaluations (i.e., calls to the simulator, which may be computationally expensive due to the number of scenarios) in order to achieve significant progress. In the type of problem addressed in this paper, if the production rate of a plant is indexed by time periods and the planning time horizon considered has twelve time periods, then for each plant that is subject to variability, twelve decision variables are needed.

One of the objectives of this paper is to show how a tradeoff curve of average profit vs. risk (Pareto efficient frontier) of the proposed production plan can be used in the analysis of results. This tradeoff curve can be obtained by applying a bi-criterion approach to the Sim-Opt framework as explained as follows. For the case of a black-box simulator, in which no model is available, multi-objective DFO algorithms can be used to construct the Pareto efficient frontier. The literature on multi-objective DFO is generally divided into two classes: Direct Search Methods (DSM) of directional type and Evolutionary Multi-objective Optimization (EMO) algorithms. Reviews are given in [Zhou \*et al.\* \(2011\)](#); [Custódio, Emmerich, & Madeira \(2012\)](#). If an optimization model is used as the simulator, then standard multi-objective optimization techniques can be used, such as the  $\epsilon$ -constraint method (see [Subsection 5.4.2](#)) ([Hwang & Masud, 1979](#); [Miettinen, 1999](#)).

### 5.4.2 Bi-Objective Optimization Framework (Bi-Opt)

The Bi-Objective Optimization framework (Bi-Opt) simultaneously proposes a production plan and minimizes the risk of operating a such plan under production variability consideration. This is accomplished by solving an optimization problem, which requires the production planning optimization model to be fully known. The proposed optimization model is a bi-objective two-stage stochastic program (Birge & Louveaux, 2011) whose first-stage variables are the production targets and second-stage variables are the remaining variables of the model (e.g., flows and inventory). The two objectives are the average profit value to be maximized and a risk measure (e.g., financial risk) to be minimized.

A general deterministic equivalent model of the bi-objective two-stage stochastic program is as follows,

$$\begin{aligned}
& \min_{x_s, PT} \quad \text{Risk}(x_s, PT) \\
& \max_{x_s, PT} \quad f_0(PT) + \sum_{s \in S} p_s f_s(x_s) \\
& \text{s.t.} \quad g(x_s) \leq 0 \quad \forall s \in S \\
& \quad \quad PR_s = PT - \Delta_s \quad \forall s \in S
\end{aligned} \tag{5.1}$$

where the first objective minimizes risk while the second objective maximizes the expected profit. In equation (5.1),  $PT$  are the first-stage production target variables, while  $x_s$  is a vector of two-stage decision variables (including the production rate variables,  $PR_s$ ) that is defined for each scenario  $s \in S$ ,  $p_s$  is a constant vector of probability of scenario  $s$ ,  $f_0(\cdot)$ ,  $f_s(\cdot)$ , and  $g(\cdot)$  are linear functions that define the multi-period LP planning model. In this paper, functions  $f_0(\cdot)$  and  $f_s(\cdot)$  correspond to the first- and second-stage profit terms, respectively, whereas functions  $g(\cdot)$  represent linear material and inventory balances and capacities.

A major contribution of this paper is to model production variability as shown in the last set of equality constraints of equation (5.1). This set of constraints fixes  $PR_s$  to the production target ( $PT$ ) minus the deviation value for a given scenario  $s$  ( $\Delta_s$ ). Recall that subtracting  $\Delta_s$  from the production target (i.e., production plan) results in the estimated actual production rate. Also, note that  $PT$  is not indexed by scenarios, since it is a vector of first-stage variables. The production target is the production plan that is sought to be implemented in practice.

The bi-objective optimization problem in equation (5.1) can be cast as a single-objective model using standard multi-objective optimization techniques as mentioned in the previous section. The  $\epsilon$ -constraint method results in two

possible models,

$$\begin{aligned}
& \max_{x_s, PT} \quad f_0(PT) + \sum_{s \in S} p_s f_s(x_s) \\
& \text{s.t.} \quad \text{Risk}(x_s, PT) \leq \epsilon \\
& \quad \quad g(x_s) \leq 0 \quad \quad \quad \forall s \in S \\
& \quad \quad PR_s = PT - \Delta_s \quad \quad \forall s \in S
\end{aligned} \tag{5.2}$$

or,

$$\begin{aligned}
& \min_{x_s, PT} \quad \text{Risk}(x_s, PT) \\
& \text{s.t.} \quad f_0(PT) + \sum_{s \in S} p_s f_s(x_s) \geq \epsilon \\
& \quad \quad g(x_s) \leq 0 \quad \quad \quad \forall s \in S \\
& \quad \quad PR_s = PT - \Delta_s \quad \quad \forall s \in S
\end{aligned} \tag{5.3}$$

where  $\epsilon$  is a threshold value that represents the maximum risk (equation (5.2)) or minimum average profit (equation (5.3)) the decision maker is willing to have. The Pareto efficient frontier can be constructed by varying the value of  $\epsilon$  and resolving the optimization problem.

If the objective function to be maximized in equation (5.1) is a financial performance indicator (e.g., profit or cost, if minimization), then one possibility is to use a financial risk measure for the expression of  $\text{Risk}(\cdot, \cdot)$ . Different financial risk management strategies have been proposed in the literature (see Sarykalin, Serraino, & Uryasev (2008) for a tutorial). Some of these risk measures are presented below. Note that each measure operates on the distribution of values of the financial performance indicator.

- *Variance*: It is a measure of the spread of a distribution that operates symmetrically on all values with respect to the expected value. Minimization of variance can be interpreted as the minimization of the square of the  $L^2$ -norm between the financial performance in a scenario and the average financial performance over all scenarios.

$$\text{Risk}_2(x_s, PT) = \sum_{s \in S} p_s \left[ f_s(x_s, PT) - \bar{f}(x_s, PT) \right]^2 \tag{5.4}$$

where  $\bar{f}(x_s, PT) = \sum_{s \in S} p_s f_s(x_s, PT)$  is the expected value of the distribution of profit values.

- *Semivariance*: It is a deviation measure similar to the variance, but it operates on values above or below the expected value. It is also similar to downside risk where the threshold is the expected value of a distribution.

$$\text{Risk}_{2+}(x_s, PT) = \sum_{s \in S} p_s \left[ f_s(x_s, PT) - \bar{f}(x_s, PT) \right]_+^2 \tag{5.5}$$

$$\text{Risk}_{2-}(x_s, PT) = \sum_{s \in S} p_s \left[ \bar{f}(x_s, PT) - f_s(x_s, PT) \right]_+^2 \tag{5.6}$$

where  $[a]_+ = \max\{0, a\}$ .

- *Mean Absolute Deviation (MAD)*: The MAD (also known as the average absolute deviation about the mean) also measures the dispersion of a distribution, but in an absolute sense. Analogously to the variance, its minimization can be seen as the minimization of an  $L^1$ -norm.

$$\text{Risk}_1(x_s, PT) = \sum_{s \in S} p_s |f_s(x_s, PT) - \bar{f}(x_s, PT)| \quad (5.7)$$

- *Maximum Absolute Deviation*: It is analogous to the MAD, but its minimization is equivalent to minimizing the  $L^\infty$ -norm (Cai *et al.*, 2000).

$$\text{Risk}_\infty(x_s, PT) = \max_{s \in S} \sum_{s \in S} p_s |f_s(x_s, PT) - \bar{f}(x_s, PT)| \quad (5.8)$$

- *Conditional Value-at-Risk (CVaR)*: It is also called expected shortfall and is a quantile-based risk measure similarly to Value-at-Risk (VaR), which is the quantile of a distribution for a given probability level  $\alpha$ . The minimization of CVaR was first proposed by Rockafellar & Uryasev (2000).

$$\text{Risk}_{\text{CVaR}_\alpha}(x_s, PT) = \gamma - \frac{1}{1 - \alpha} \sum_{s \in S} p_s [-f_s(x_s, PT) - \gamma]_+ \quad (5.9)$$

where  $\gamma \in \mathbb{R}$  (additional variable).

## 5.5 Numerical Examples

The proposed approach to deal with production variability in the operations planning stage of the Sales & Operations Planning (S&OP) process is illustrated with a motivating example and an industrial case study. The two-stage scenario tree for the stochastic models has its node values (outcomes) fixed to the sampled  $\Delta$  values from the quantile regression analysis, and the probabilities of the scenarios,  $p_s$ , were calculated using the data-driven scenario generation approach described in Calfa *et al.* (2014) (see  $L^2$  DMP formulation).

All optimization models were implemented in AIMMS 3.13 (Roelofs & Bisschop, 2015) and solved on a desktop computer with the following specifications: Dell Optiplex 990 with 4 Intel® Core™ i7-2600 CPUs at 3.40 GHz (total 8 threads), 8 GB of RAM, and running Windows 7 Enterprise. All linear programming (LP) and convex quadratically-constrained programming (QCP) models were solved with Gurobi 5.6.

The Sim-Opt approach consists of a main script in MATLAB (The MathWorks Inc., 2015) in which the DFO algorithm `fminsearchbnd`<sup>1</sup> (Nelder-Mead

---

<sup>1</sup>The function `fminsearchbnd` extends MATLAB's built-in function `fminsearch` by considering bounds on the decision variables. The bounds used in the computational experiments were the plant minimum/maximum capacities. See <http://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd--fminsearchcon> for implementation (retrieved on March 22, 2015).

simplex search algorithm) sets the production targets  $PT$  that are fixed in the AIMMS model (the simulator). In other words, the DFO algorithm proposes a production plan, which is evaluated by the simulator (two-stage stochastic programming model). In order to perform a comparison between Sim-Opt and Bi-Opt approaches, we used the same  $\Delta$  values in both, even though the Sim-Opt can accommodate the sampling of  $\Delta$  values conditional on the proposed production plan.

### 5.5.1 Motivating Example

The goal of the motivating example is to demonstrate that different allocation schemes of chemicals in a process network are obtained when production variability is considered and some risk measure is adopted. Typically, only the margin (sales from revenue minus operating cost) of individual products is used as a criterion for deciding their allocation throughout the network. By also accounting for production variability, larger amounts of a feedstock chemical may be allocated to lower-margin, but potentially more reliable plants (to be defined in the next paragraph) than to higher-margin, but less reliable plants that “compete” for the same raw material. Even though this may seem counter-intuitive at first, we show through this example the trade-off between the expected or average *overall* profit and the risk of choosing an allocation scheme, i.e., less risk with allocation favoring low-margin and more reliable plants vs. high risk with allocation favoring high-margin and less reliable plants. Please recall that by reliability we mean the spread of  $\Delta$  values around zero, i.e., the deviation of actual production rates from the respective planned values. A detailed analysis of the results is given to illustrate the applicability of the proposed methodology.

The process network is shown in [Figure 5.5](#). Each plant produces a single product, which receives the same name as the plant that produces it. Therefore, we will use plant and product interchangeably. The main objective is to demonstrate the different allocation schemes of chemical A to the downstream plants (B–G) between deterministic and stochastic (risk neutral and averse) solutions by considering production variability of the downstream plants. Note that the order of plant’s *reliability* is the reverse of the order of plant’s margin (revenue from sales minus operating costs), i.e., the most reliable plant (G) is the lowest-margin plant, whereas the less reliable plant (B) is the highest-margin one. In this context, reliability is represented by the spread of the deviation between historical planned and actual production rates ( $\Delta$  values) around the origin, which is along the lines of a root mean square calculation. In other words, a more reliable plant means  $\Delta \approx 0$ , i.e., it is more likely to actually achieve its planned values proposed in the operations planning stage of the S&OP process. Please note that, in this paper, we do not refer to reliability in the sense of maintainability or probability of failures of a system or component.

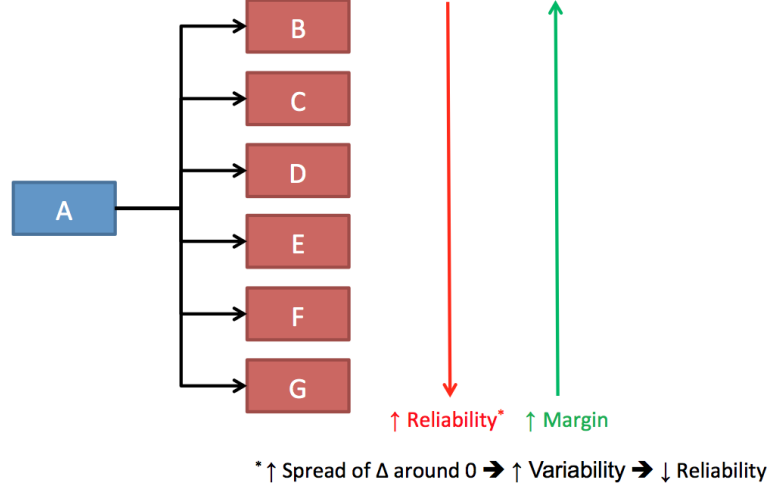


Figure 5.5: Process network structure of the motivating example. Plant reliability is related to the spread of the deviation between historical planned and actual production rates around the origin. Specifically in this example, the more reliable a plant is, the lower its margin is.

The multi-period, LP production planning model is similar to the one presented in [Sahinidis \*et al.\* \(1989\)](#), but excluding capacity expansion decisions. That is, the model consists of simple input-output relationships of material and inventory balances. Possible demand and plant capacity violations are captured with non-negative slack variables added to the respective constraints and penalized in the objective function. When production variability is taken into account, the deterministic equivalent model of the two-stage stochastic production planning model can be generically written as in equation (5.1), where the proposed S&OP production plan or target,  $PT$ , is a vector of first-stage variables. The model also has slack variables that capture unsatisfied demand and capacity violations that are penalized in the objective function. Twenty scenarios (samples from the quantile regression analysis) were considered in all stochastic models. We perform a detailed analysis of the results of four cases defined as follows:

#### Case 1. Deterministic

- No production variability, i.e.,  $\Delta_s = 0$ .
- Identifier: 1. Det

#### Case 2. Risk Neutral Stochastic with Fixed Production Target

- Production variability is considered, i.e.,  $\Delta_s \neq 0$ .
- *Fixed* the values of production targets,  $PT$ , to the optimal production rates obtained by solving Case 1.
- The purpose of this case is to evaluate the performance of the deterministic production plan in an uncertain environment.

- Identifier: 2. Stoch Fix

### Case 3. Risk Neutral Stochastic with Variable Production Target

- Similar to Case 2, but with *variable* production targets,  $PT$ .
- Production targets (first-stage decisions) are optimally set while taking into account the historical variability in production rates.
- Identifier: 3. Stoch Var

### Case 4. Risk Averse Stochastic (Bi-Opt Framework)

- Similar to Case 3, but with two objective functions, one of them measuring risk.
- Identifier: 4. Bi-Opt

We should note that Cases 1–3 give rise LP models since the underlying planning model is linear, while Case 4 gives rise to a convex QCP model, because we use the variance of the profit as the risk measure. The QCP model is convex because the variance is taken with respect to the profit, which is a linear function, and the scenario probabilities are non-negative; thus, the  $\text{Risk}(x_s, PT)$  function is a sum of non-negative quadratic terms. Let us first focus on the results obtained with the Bi-Opt approach ([Subsection 5.4.2](#)), and then comment on its differences with the Sim-Opt approach ([Subsection 5.4.1](#)). We begin with Cases 1–3, and then discuss Case 4, which has Case 3 as a special case.

#### 5.5.1.1 Cases 1, 2, and 3

We start the analysis by comparing average overall margin and its standard deviation for the first three cases. The overall profit (or simply profit) is the difference between the revenue from the sale of all products and total costs (operating and inventory). Operating costs are proportional to production rates, and inventory costs are proportional to the amount of chemicals stored in each period. As it can be seen in [Table 5.2](#), adjusting the production target (Case 3) in the face of production variability yields more profitable and less risky (smaller standard deviation) production plans. In other words, production targets obtained with the deterministic model (Case 1) yield lower average profit with larger spread (higher risk) when production variability is taken into account (Case 2). In addition, some plant capacity violations were observed in the solution of Case 2, which is clearly undesirable. The optimization results convey that the value of simultaneously proposing the production targets and accounting for production variability is  $(320.60 - 272.84) \text{ m.u.} = 47.77 \text{ m.u.}$  (monetary units) on average.

Table 5.2: Average profit and its standard deviation in monetary units (m.u.) for Cases 1–3 in the motivating example. The standard deviation of the profit is a measure of its spread, i.e., financial risk.

	<b>Case</b>		
	1. Det	2. Stoch Fix	3. Stoch Var
Average	375.15	272.84	320.60
Std Dev	–	28.30	25.82

The difference in average profitability between the solution of Case 2 and Case 3 is also explained by the average overall service level (SL) defined in equation (5.10). The overall SL is the complement of the fraction of total demand satisfied from sales of all products.

$$\mathbb{E}[\text{SL}] = 1 - \frac{\sum_{s \in S} p_s \text{Sales}_s}{\text{Total Demand}} \quad (5.10)$$

The overall SL for the deterministic solution (Case 1) is 100% (i.e., all demand is satisfied), and the average overall SL for Case 2 and Case 3 is 82.68% and 92.24%, respectively. The breakdown of unmet demand for each product is given in Table 5.3. From Case 2 to Case 3, the demand satisfaction of high-margin products (B, C, and D) increases relatively more than for the low-margin products (E, F, and G). In fact, the demand satisfaction of product G, which is produced by the most reliable and lowest-margin plant, actually decreased in Case 3. When analyzing the results of Case 4 later on, it will be clear that the solution of Case 3 favors more allocation of product A to the high-margin plants, since the objective function (profit) is not constrained by any risk measure (i.e., more risky condition).

Table 5.3: Average unmet demand in weight units (w.u.) for each product in Case 2 and Case 3 in the motivating example.

<b>Product</b>	<b>Case</b>	
	2. Stoch Fix	3. Stoch Var
B	64	6
C	55	10
D	41	4
E	34	10
F	18	10
G	8	34

The amounts of chemical A allocated to the six downstream plants are shown in Table 5.4, which complements the results shown in the previous table. As it would be expected after the analysis of unmet demand, in Case 3, relatively more amounts of A are allocated to high-margin plants than for



low-margin ones. The negative percentage change for plant G (lowest-margin) means that it receives less A in Case 3 than in Case 2.

Table 5.4: Average allocated amounts of A in weight units (w.u.) to each downstream plant in Case 2 and Case 3 in the motivating example. The percentage change column is the relative change between the two cases, i.e., (Case 3 – Case 2)/Case 3.

Plant	Case		
	2. Stoch Fix	3. Stoch Var	Change
B	160	204	28%
C	96	143	49%
D	131	164	25%
E	147	178	21%
F	343	356	4%
G	148	111	–25%

Let us analyze the results for Case 4. The goal is to evaluate the impact of controlling some measure of risk by including an additional constraint ( $\epsilon$ -constraint) on the allocation scheme of chemical A to downstream plants. Consider two subcases of Case 4: Subcase 4.A uses an explicit financial risk measure (variance of profit) and Subcase 4.B uses the individual expected service level for one of the high-margin products.

#### 5.5.1.2 Subcase 4.A: Variance of Profit

The  $\epsilon$ -constraint in equation (5.1) takes the following form,

$$\text{Risk}(x_s, PT) = \sum_{s \in S} p_s \left[ \text{Profit}_s(x_s, PT) - \overline{\text{Profit}}(x_s, PT) \right]^2 \leq \epsilon \quad (5.11)$$

where  $\text{Profit}_s(\cdot, \cdot)$  denotes only the profit calculation of the objective function, i.e., excluding penalized slack variables, and  $\overline{\text{Profit}}(x_s, PT) = \sum_{s \in S} p_s \text{Profit}_s(x_s, PT)$  is the average profit. In this subcase,  $\epsilon$  is interpreted as the maximum allowed variance of profit and has units of (m.u.)<sup>2</sup>, where “m.u.” stands for monetary units.

Note that Case 3 is a special case of Subcase 4.A in which  $\epsilon$  takes a large enough value so that the constraint is not active at the solution, i.e., the financial risk is unconstrained and the model becomes risk neutral. Thus, the solution to Case 3 represents the condition of maximum variance of the profit, which is the right-most point in the Pareto efficient curve of average profit vs. variance (or standard deviation) of profit in Figure 5.6. In addition to the solution of Case 3, the bi-objective optimization model (convex QCP) was solved ten times for different values of  $\epsilon$ , ranging from 60 to 600 with a stride of 60, and the solutions are represented by points on the Pareto efficient frontier. Note that from left to right the spread of the profit across scenarios increase, i.e., more risky solutions.

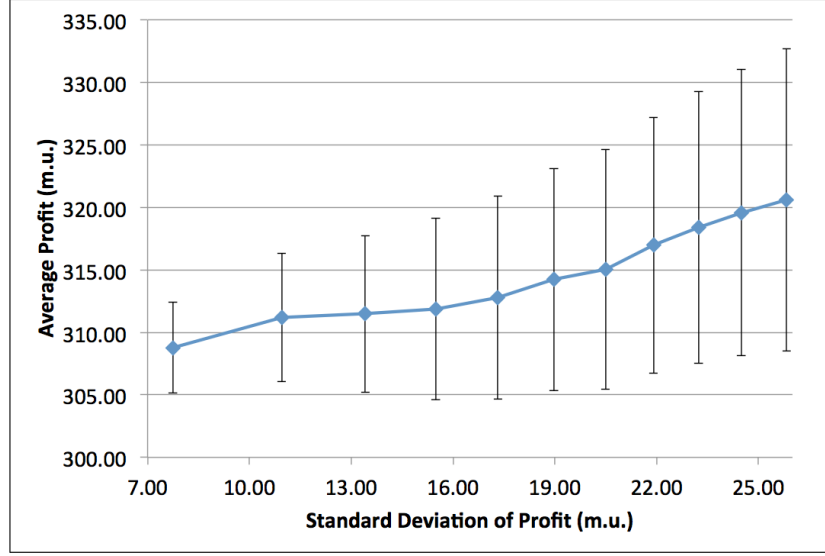


Figure 5.6: Pareto efficient frontier for Subcase 4.A in the motivating example. “m.u.” stands for monetary units. Error bars represent 95% confidence intervals on the average values.

Figure 5.7 shows the same Pareto efficient frontier together with the average overall service level (SL) as defined in equation (5.10). Note that there is an increase in the average overall SL for the solutions from points P1 to P2, and after point P2 until point P3 the average overall SL levels off. Therefore, we classify the solutions as belonging to two regions: Region I (less risky) and Region II (more risky). After point P2, the amount of A allocated to all downstream plants remains practically the same with the exception of the least and most reliable plants, B and G, respectively. As the  $\epsilon$  value increases (more risky condition), there is a shift of the amount of A allocated from G to B.

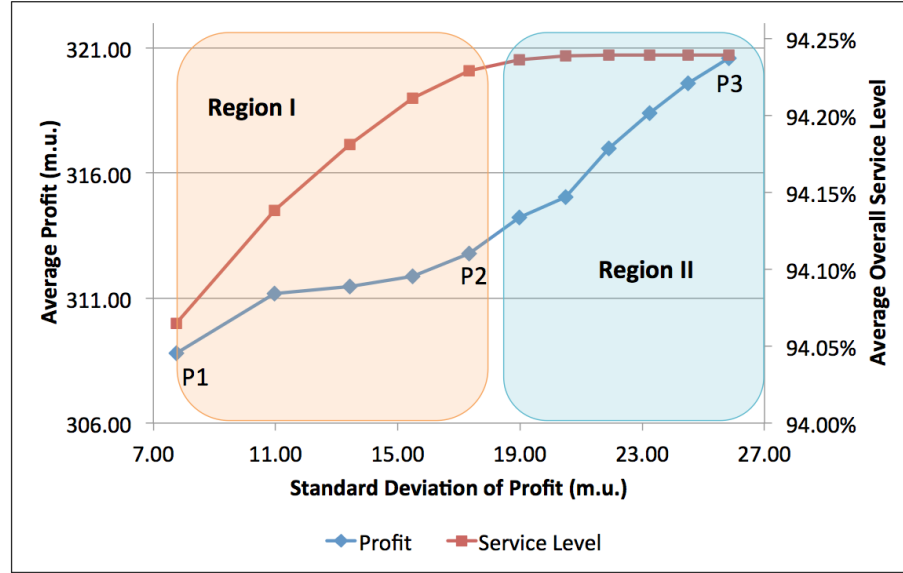


Figure 5.7: Pareto efficient frontier and average overall service level for Subcase 4.A in the motivating example. “m.u.” stands for monetary units.

Figure 5.8 shows the allocation amounts of A to the downstream plants for each point in the Pareto efficient curve for the two plants in the extremes of the reliability-margin scale. The same overall trend is observed for the other plants: more A is allocated to less reliable, high-margin plants (B, C, and D) as risk ( $\epsilon$  value) increases; conversely, less A is allocated to more reliable, low-margin plants (E, F, and G) from left to right in the figure.

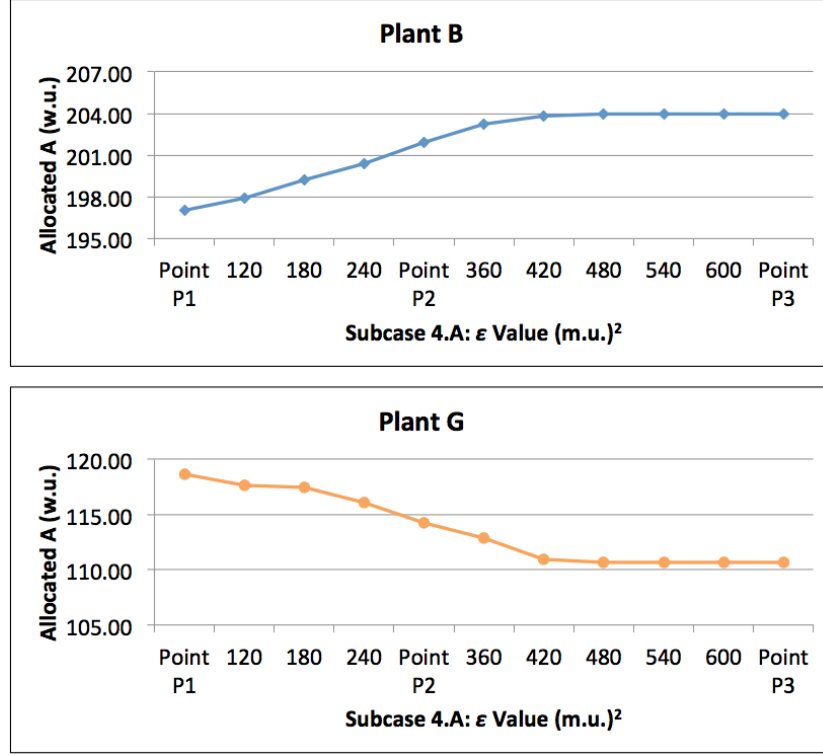


Figure 5.8: Allocation scheme for two downstream plants in Subcase 4.A in the motivating example. Plant B is the less reliable and highest-margin, whereas plant G is the most reliable and lowest-margin of the downstream plants. The  $\epsilon$  value denotes the variance of the profit (financial risk). “w.u.” and “m.u.” stand for weight and monetary units, respectively.

### 5.5.1.3 Subcase 4.B: Average Service Level of Product C

The  $\epsilon$ -constraint in equation (5.1) takes the following form,

$$\text{Risk}(x_s, PT) = 1 - \frac{\sum_{s \in S} p_s \text{Sales}_s^C}{\text{Total Demand}} \leq \epsilon \quad (5.12)$$

where  $\text{Sales}_s^C$  indicates that only sales for product C are considered, thus the  $\epsilon$ -constraint is a calculation of the complement of the average *individual* service level (SL) of product C. In this subcase,  $\epsilon$  is interpreted as the maximum allowed fraction of unmet demand of product C and is dimensionless. It can be expressed as a percentage, e.g.,  $\epsilon = 1\%$  means that at most 1% of the demand of product C can be unmet, or equivalently, at least 99% of the demand of C must be satisfied.

The motivation behind this subcase is two-fold: (1) it uses a non-conventional form of the  $\epsilon$ -constraint for the risk (i.e., not an explicit financial risk measure); (2) the average SL of the high-margin product C in the solution of Case 3 is 93.85%, and it is desired to evaluate the impacts of enforcing a higher average SL of this valuable product on the overall profitability of the production plan.

Figure 5.9 shows the Pareto efficient curve of the average profit vs. average individual service level of product C. In addition to the solution of Case 3, the bi-objective optimization model (LP) was solved four times for different values of  $\epsilon$  (5%, 4%, 3%, and 2%). In this motivating example, 99% (i.e.,  $\epsilon = 1\%$ ) or higher average SL of product C is not feasible. Note that as the average SL of C increases, the overall average profit decreases and its spread (represented by the error bars) increases. Even though more demand of product C is met from left to right, the overall production plan becomes less profitable on average.

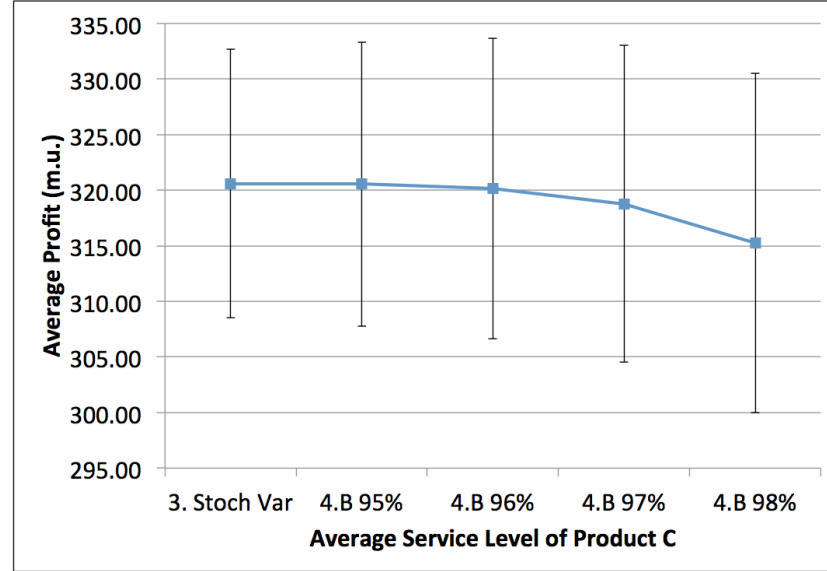


Figure 5.9: Pareto efficient frontier for Subcase 4.B in the motivating example. Error bars represent 95% confidence intervals on the average values. “m.u.” stands for monetary units.

Figure 5.10 helps explain the result in the previous figure. The average overall SL decreases as the average individual SL of product C is forced to increase. In other words, by requiring higher SL for product C, there is a shift of A allocated from other plants to plant C in order to ensure its desired SL. Consequently, the individual demand satisfaction of the other products decreases from left to right in the figure.

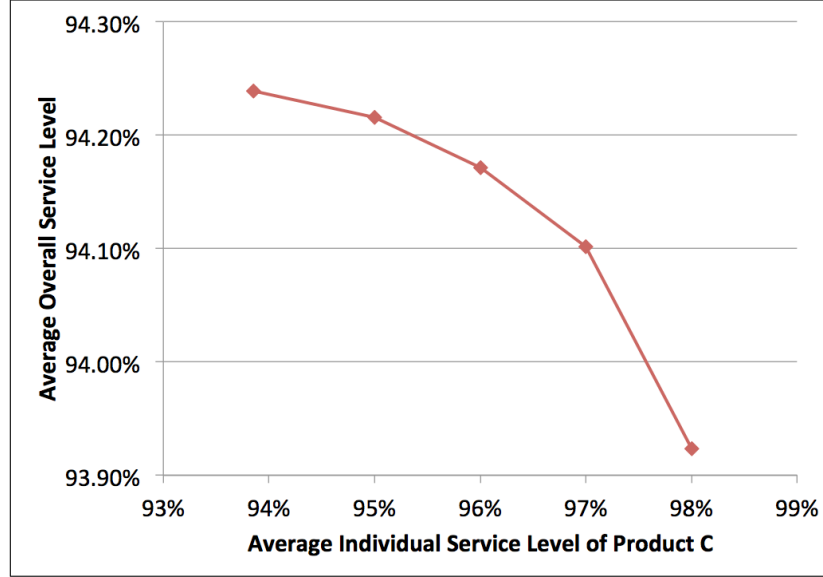


Figure 5.10: Effect of increasing average service level of product C on average overall service level in Subcase 4.B in the motivating example.

#### 5.5.1.4 Computational Statistics

To conclude the analysis of the motivating example, [Table 5.5](#) presents the computational statistics of the optimization models solved in the four cases. The time corresponds to the wall time, including loading and solving the models. The time listed under Subcases 4.A and 4.B corresponds to the average wall time for all points in the Pareto efficient frontier in each subcase. Recall that Cases 1 through 3 are LP models, Subcase 4.A contains convex QCP models, and Subcase 4.B has LP models.

Table 5.5: Computational statistics for optimization models in all cases and subcases in the motivating example.

	Case and Subcase				
	1. Det	2. Stoch Fix	3. Stoch Var	4.A Bi-Opt	4.B Bi-Opt
Variables	5,186	101,778	101,874	101,874	101,874
Constraints	3,602	73,482	73,482	73,483	73,483
Time [s]	2.59	4.38	5.05	5.31	5.12

#### 5.5.1.5 Sim-Opt vs. Bi-Opt

We note that Cases 1 and 2 are the same for both Sim-Opt and Bi-Opt frameworks, since they correspond to the deterministic problem (no production variability) and the stochastic problem with fixed production targets (no proposed production plan), respectively. The Sim-Opt framework for Case 3 took 1,000 iterations (imposed limit), 1,160 function evaluations (i.e., calls to the simulator, which is the two-stage stochastic programming model implemented

in AIMMS), and 10,745 seconds to achieve an expected profit of 294.43 m.u., while the Bi-Opt framework yielded an expected profit of 320.60 m.u. in less than 5 seconds. In addition, the solution of the Sim-Opt framework exhibited 91.36% expected overall service level, which is lower than that obtained with the Bi-Opt framework (94.24%).

The number of decision variables in this case is 72, since they correspond to the monthly production targets of 6 plants for a time horizon of one year. Moreover, every call to the simulator takes approximately 7 seconds, while the time spent by the DFO algorithm per iteration is negligible. These two factors make the Sim-Opt framework more computationally intensive than the Bi-Opt approach, in spite of having more flexibility for sampling  $\Delta$  values conditional on proposed production targets. Similar observations were made for the solution of Case 4 with the Sim-Opt approach.

### 5.5.2 Industrial Case Study

The industrial test case concerns the optimal production planning of chemical sites. Each site contains several plants that are highly integrated. The plants can also transfer products between sites. The chemical sites contain more than 12 production facilities and manufacture several products. The time horizon of one year is divided into monthly time periods. The objective of the optimization model is to maximize the total profit. Due to confidentiality reasons, we only discuss the modeling changes from the motivating example and the computational results.

In Case 2 of the motivating example, the production targets,  $PT$ , are fixed to the corresponding production rates obtained from the solution of the deterministic model (Case 1). The goal is to evaluate the performance of the production targets proposed by the deterministic model when production variability is taken into account. In order to perform a similar study on a highly integrated system, some modeling modifications may be necessary. If the general deterministic equivalent form of the two-stage stochastic programming model (see equation (5.1)) describes a highly integrated system, the equality constraint that captures the production variability,

$$PR_s = PT - \Delta_s \quad \forall s \in S \quad (5.13)$$

for *fixed*  $PT$ , may cause infeasibility. For instance, it may not be feasible to satisfy material and inventory balances in the network by imposing such a constraint on production rates with fixed production targets. To circumvent this potential problem, we propose two modifications to the implementation of Case 2 for a highly integrated system: (1) unfix the production target decision variables,  $PT$ , and (2) penalize the deviation of the production targets set by stochastic model from the corresponding values obtained in the solution of the deterministic model,  $PT^{\text{det}}$  (a constant parameter vector). Therefore, the

general optimization problem for Case 2 is rewritten as follows:

$$\begin{aligned}
& \min_{x_s, PT} \quad \text{Risk}(x_s, PT) \\
& \max_{x_s, PT} \quad f_0(PT) + \sum_{s \in S} p_s f_s(x_s) - \psi \cdot \|PT - PT^{\text{det}}\|_p \\
& \text{s.t.} \quad g(x_s) \leq 0 \quad \forall s \in S \\
& \quad \quad PR_s = PT - \Delta_s \quad \forall s \in S
\end{aligned} \tag{5.14}$$

where  $\psi$  is a penalty factor and  $\|\cdot\|_p$  is an  $L^p$ -norm. In order to preserve the linearity of the production planning model used in this work, we employed the  $L^1$ -norm, which resulted in the following formulation:

$$\begin{aligned}
& \min_{x_s, PT} \quad \text{Risk}(x_s, PT) \\
& \max_{x_s, PT} \quad f_0(PT) + \sum_{s \in S} p_s f_s(x_s) - \psi \cdot (PT^+ + PT^-) \\
& \text{s.t.} \quad g(x_s) \leq 0 \quad \forall s \in S \\
& \quad \quad PR_s = PT - \Delta_s \quad \forall s \in S \\
& \quad \quad PT - PT^{\text{det}} = PT^+ - PT^- \\
& \quad \quad PT^+, PT^- \geq 0
\end{aligned} \tag{5.15}$$

where  $PT^+$  and  $PT^-$  capture the positive and negative deviations.

The high degree of integration in the network also has to be considered in the statistical modeling. If two plants are directly connected (e.g., plant A feeds plant B), then it may not be realistic (and likely lead to infeasibilities) to consider them independent from the point of view of production rates. In other words, deviations from plan in an upstream plant may affect production in a downstream plant, and vice versa.

Different approaches can be used to account for this dependence between plants for which production variability is taken into account. If only  $\Delta$  values are used to characterize production variability (i.e., the covariate Plan values are disregarded), then we propose the following approaches (illustrated in [Figure 5.11](#)) for any two connected plants A and B:

- Assume a *parametric* classical regression model for the  $\Delta$  values between the upstream and downstream plants, and from the generated samples of one of the plants, calculate the corresponding (expected) sample of the other plant. For example, if a linear regression model such as  $\Delta^{\text{Plant B}} = \beta_0 + \beta_1 \cdot \Delta^{\text{Plant A}}$  is considered, then (1) estimate the regression function (in this case, the model parameters  $\beta_0$  and  $\beta_1$ ) by regressing  $\Delta$  values of Plant B on  $\Delta$  values of Plant A, then (2) generate samples of  $\Delta$  values for Plant A, and finally (3) calculate the corresponding  $\Delta$  values for Plant B using the linear model.
- Similar to the previous approach, but instead assume a *nonparametric* classical regression model for the  $\Delta$  values between the upstream and



downstream plants. In other words, the regression model is generally written as  $\Delta^{\text{Plant B}} = g(\Delta^{\text{Plant A}})$ , where  $g(\cdot)$  is the nonparametric regression function (e.g., kernel regression (Li & Racine, 2007)). Follow the same three-step procedure discussed in the previous approach.

- Estimate the joint distribution of  $\Delta$  values for both plants A and B, i.e., obtain  $\hat{F}(\Delta^{\text{Plant A}}, \Delta^{\text{Plant B}})$ , and then sample from this estimated multivariate distribution.
- Perform bootstrap resampling (Efron, 1979), i.e., sample with replacement data points from the original data set. As in the previous approach, the samples constitute the outcomes (i.e., scenarios) in the stochastic programming framework.

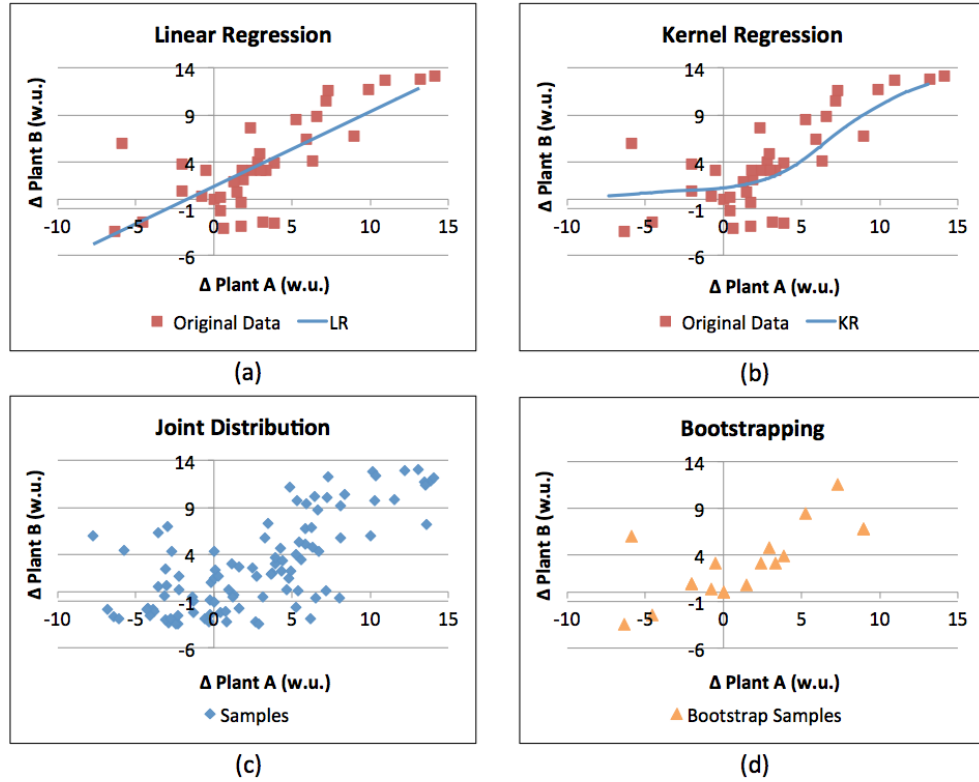


Figure 5.11: Illustration of the three proposed approaches to account for dependence of the  $\Delta$  values for directly connected plants in the network. The relationships between the  $\Delta$  values of the upstream and downstream plants are captured by: (a) linear regression (LR), (b) kernel regression (KR), (c) estimated joint distribution, and (d) bootstrap resampling. “w.u.” stands for weight units.

When the Plan values are taken into account in modeling production variability (i.e., the quantile regression approach discussed in Section 5.3), then the problem essentially becomes modeling the relationship between *conditional* distributions,  $\Delta^{\text{Plant B}} | \text{Plan}^{\text{Plant B}}$  given  $\Delta^{\text{Plant A}} | \text{Plan}^{\text{Plant A}}$ . A general

approach would be to estimate the joint conditional distribution of all random variables involved,  $\hat{F}(\Delta^{\text{Plant A}}, \Delta^{\text{Plant B}} | \text{Plan}^{\text{Plant A}}, \text{Plan}^{\text{Plant B}})$ , and then sample from this estimated multivariate distribution. This general approach may pose computational and algorithmic challenges. An *approximate* approach to generate samples of  $\Delta^{\text{Plant B}} | \text{Plan}^{\text{Plant B}}$  given  $\Delta^{\text{Plant A}} | \text{Plan}^{\text{Plant A}}$  is described in the following steps:

- i. Obtain the estimated distribution of  $\Delta^{\text{Plant A}} | \text{Plan}^{\text{Plant A}}$ ,  $\hat{F}_A(\Delta^{\text{Plant A}} | \text{Plan}^{\text{Plant A}})$ , and sample from this conditional distribution. Let the samples be denoted by  $S_A$ .
- ii. Repeat the previous step replacing “A” with “B”.
- iii. Obtain the estimated distribution  $\hat{F}_S(S_B | S_A)$ , and sample from this conditional distribution. These final “approximate” samples relate  $\Delta^{\text{Plant B}} | \text{Plan}^{\text{Plant B}}$  given  $\Delta^{\text{Plant A}} | \text{Plan}^{\text{Plant A}}$ .

Table 5.6 presents the computational statistics of the optimization models solved in the four cases. We report both solution times (CPU times for the solver) and wall times (includes loading and solving the models), which become significant as the problem instance increases. Similar to the Motivating Example, Cases 1–3 and Subcase 4.B are LP models, while Subcase 4.A contains a convex QCP model.

Table 5.6: Computational statistics for optimization models in all cases and subcases in the industrial case study. CPU and wall times are shown in separate rows.

	Case and Subcase				
	1. Det	2. Stoch Fix	3. Stoch Var	4.A Bi-Opt	4.B Bi-Opt
Variables	124,116	2,447,057	2,446,913	2,446,914	2,446,913
Constraints	88,790	1,777,313	1,777,241	1,777,243	1,777,242
CPU [s]	0.45	9.17	10.22	15.61	17.43
Wall [s]	15.26	50.17	53.77	60.20	62.76

The Sim-Opt framework for Case 3 was run for 100 iterations (imposed limit), 181 function evaluations (i.e., calls to the two-stage stochastic programming model implemented in AIMMS), and 10,130 seconds to achieve an expected profit of 1,293.45 m.u., while the Bi-Opt framework yielded an expected profit of 1,299.81 m.u. in less than 54 seconds of wall time. Moreover, the solution of the Sim-Opt framework yielded 97.92% expected overall service level, which is lower than that obtained with the Bi-Opt framework (98.60%). Similar to the motivating example, the number of decision variables for the DFO algorithm is 72 (monthly production targets for 6 plants and time horizon of one year). Each call to the AIMMS model takes approximately 80 seconds, which contributes significantly to the overall solution time of the Sim-Opt framework. Similar observations were made for the solution of Case 4.

## 5.6 Conclusions

In this paper, we have addressed uncertainty in the operations planning stage of the Sales & Operations Planning (S&OP) process of a manufacturing company. The uncertainty is attributed to production variability, which is caused by unplanned events that can result in actual production rates lower or higher than their planned values. In order to model production variability, we defined  $\Delta$  as the deviation between historical planned and actual production rates, and used quantile regression to predict quantiles of the distribution of  $\Delta$  conditional on planned production rates. The predicted quantiles form samples or scenarios in a two-stage bi-objective stochastic programming production planning model, whose first-stage variables are the production plans or targets that are sought to be implemented in practice. One objective represents the financial performance of the production plan (e.g., profit), whereas the other is a risk measure (explicitly financial or not).

The applicability of the proposed approach was illustrated with a motivating example and an industrial case study. The motivating example consisted of a small process network with a feedstock plant that serves six plants. The downstream plants are sorted in reverse order of reliability and margin, i.e., the most reliable plant is also the lowest-margin one, and the less reliable plant is the highest-margin one. The motivation behind this was to show that depending on the maximum desired level of risk, the optimization model decides to allocate more of the feedstock chemical to more or less reliable plants. In other words, the optimization model considers not only the individual margin of the plants, but also their reliability in order to obtain a solution with maximum *expected* profit. For the industrial case study, we proposed modeling approaches to address the connectivity in the network, which may create dependence in production variability profiles.

## Chapter 6

# Optimal Procurement Contract Selection with Price Optimization under Uncertainty for Process Networks

### 6.1 Introduction

Manufacturing enterprises deal with uncertainty from both internal and external sources. Internally, production variability due to unplanned events may prevent the company to achieve its demand-driven production targets. Externally, fluctuations in supply and demand as well as market economic conditions pose challenges to efficient operation of the supply chain. One way to reduce the level of uncertainty on both the supply and the customer sides, and that is typically used by companies, is by making contractual agreements. In the context of this chapter, a contract is a binding agreement in which the seller provides the specified product and the buyer pays for it under specific terms and conditions.

A different approach to managing uncertainty is pricing analytics, also known as price optimization. In formulating such a problem, selling prices become decision variables, and the demand of a product is modeled as a function of its price. Nonetheless, this still typically does not completely eliminate uncertainty, since stochastic environmental, economic, and market conditions remain uncontrollable by the manufacturing company.

In this chapter, we combine the aforementioned uncertainty management strategies in a two-stage stochastic optimization model for multi-period, multi-site tactical production planning. Uncertain parameters include raw material and finished product price and availability/demand. We propose a manufacturer-centric approach in which the purchase contract structures are set by suppliers, and it is the manufacturer's decision to select which contract, if any, to sign. In addition, the manufacturer sets the selling price of its main products that may be used to design sales contracts with its customers. How-

ever, it is the customer's decision to select the sales contracts, if any, designed by the manufacturer. The contract design problem is not addressed in this chapter, but is discussed as a future work. We demonstrate how different pricing models, linear and nonlinear, can be used within the proposed optimization framework. Throughout this chapter, we refer to price as *unit* price (e.g., \$/kg, \$/t etc.).

The chapter is organized as follows. [Section 6.2](#) provides a literature review of contract modeling and selection and price optimization. [Section 6.3](#) defines the problem under investigation and states the model assumptions. [Section 6.4](#) describes the deterministic production planning model, the contract selection and price optimization elements of the planning model, and the stochastic optimization model. Two numerical examples are discussed in [Section 6.5](#) followed by a discussion on reformulations. Finally, conclusions and suggested extensions of the proposed approach are presented in [Section 6.6](#).

## 6.2 Literature Review

### 6.2.1 Contract Modeling and Selection

[Tsay, Nahmias, & Agrawal \(1999\)](#) provide a literature review on contracts from a modeling perspective. The authors classify the literature on contracts in the context of Supply Chain Management (SCM) by eight contract clauses: (a) specification of decision rights, (b) pricing, (c) minimum purchase commitments, (d) quantity flexibility, (e) buyback or returns policies, (f) allocation rules, (g) lead time, and (h) quality. Moreover, [Höhn \(2010\)](#) discusses some model extensions and other contract types based on finance theories, such as auction and real options.

Even though the modeling of contracts has been addressed by different communities, including operations research and management science, economics, and finance, the optimal selection of contracts has received relatively less attention in the literature. The contract selection problem can be defined as follows: given the structure (price and quantity or volume) of the different contract types, the multi-period optimization model has to select which contracts to use. We present some specific examples of this problem that are relevant to the Process Systems Engineering (PSE) community.

[Park \*et al.\* \(2006\)](#) used disjunctive programming to model three types of quantity-based contracts that a company may sign with suppliers and customers. The production planning and capacity expansion decisions are augmented by the selection of contracts. The disjunctions are reformulated resulting in a tight Mixed-Integer Linear Programming (MILP) formulation. [Bansal, Karimi, & Srinivasan \(2007\)](#) modeled quantity- and price-based contracts and coupled their selection with a multi-period optimization model for the minimization of total procurement costs of a multinational company (buyer). [Khalilpour & Karimi \(2011\)](#) developed a multi-period optimization model to minimize total procurement cost applied to liquefied natural gas (LNG) value chain. The model selects LNG contracts that are modeled with price for-

mulations, flexibility, duration, quality, quantity, commitment, discount, and other terms and conditions. We note that the aforementioned works have only considered deterministic formulations.

Some authors, however, have considered uncertainty in the contract selection problem. For instance, [Rodríguez & Vecchietti \(2009\)](#) addressed the delivery and purchase optimization in a supply chain under provision uncertainty. Quantity-based contracts were modeled with disjunctions and logic restrictions, and the uncertain amount delivered by the suppliers is modeled with a decision tree approach that represents supplier failure discrete probability distributions. [Rodríguez & Vecchietti \(2012\)](#) proposed a mid-term planning model with sales contracts. Piece-wise linear price-response models were used to model demand as a function of selling price. If no sales contract is signed, then safety stock is considered to deal with demand uncertainty. [Feng & Ryan \(2013\)](#) developed a two-stage stochastic programming model with fixed recourse to coordinate contract selection and capacity allocation in a three-tier manufacturing supply chain in the oriented strand board industry. The uncertain parameters include price and quantity (availability and demand) of materials.

### 6.2.2 Pricing Analytics

Pricing decisions have been discussed in the revenue management literature and constitute one set of demand-management decisions that can be made by a firm ([Talluri & van Ryzin, 2005](#)). Some examples of pricing decisions include setting posted prices, individual-offer prices, and reserve prices (in auctions), pricing across product categories, pricing over time, and discounting over the product lifetime.

By treating selling price as an additional degree of freedom, a company can adjust its prices in order to maximize its overall profitability. The typical modeling approach reported in the literature is the use of a price-response model  $d(p)$ , which specifies demand  $d$  for the product of a single seller as a function of the price  $p$  offered by that seller ([Phillips, 2005](#); [Bodea & Ferguson, 2014](#)). More details on pricing models are given in [Subsection 6.4.3](#).

Several papers address pricing problems whose optimization model only contains continuous decision variables, and is sometimes unconstrained (e.g., maximization of profit). For such problems, analytical expressions for the optimal pricing scheme can be derived ([Chong & Cheng, 1975](#); [Pasche, 1998](#); [Lobo & Boyd, 2003](#); [Thiele, 2005](#)). Some counterexamples include the following works. [Kaplan \*et al.\* \(2011\)](#) used a constant price elasticity model in a multi-period, multi-echelon supply chain optimization model. Such pricing model results in a nonlinear, but concave, objective function, and the authors proposed a reformulation to account for unbounded derivatives. [Lin & Wu \(2014\)](#) formulated a two-stage stochastic programming model and used a linear pricing model. The authors observed that as the variance of the demand distribution increases, the manufacturer will increase its inventory to levels that are greater than the anticipated demand to prevent the potential loss of

sales and will simultaneously raise product prices to obtain a greater profit. Gjerdrum, Shah, & Papageorgiou (2001) developed a mixed-integer nonlinear programming (MINLP) model to for the optimal calculation of *transfer prices* for the supply chain optimization involving multiple enterprises. The authors used game theoretical Nash-type models to obtain a fair, optimized profit distribution between members of multienterprise supply chains.

### 6.3 Problem Statement

The supply chain structure considered in this chapter is shown in Figure 6.1. The manufacturer owns one or multiple production sites that may be situated at different geographic locations. The key decisions of the manufacturing company include: (1) selection of procurement contracts, and (2) set selling product prices. The manufacturer-centric approach taken in this work means that the contract structure (prices and quantity thresholds) are specified, and it is ultimately the manufacturer's decision to sign or not a particular type of contract for a raw material. Moreover, the manufacturer can use price- or demand-response models to set the price of its finished products.

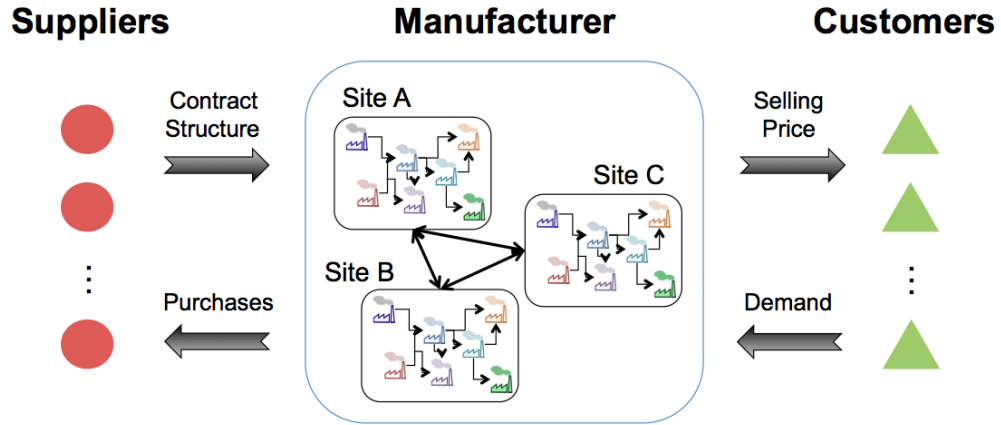


Figure 6.1: Supply chain structure considered in this work.

The following assumptions are considered:

- The contract structure is given and fixed (see Subsection 6.4.2), i.e., no negotiation between supplier and manufacturer is allowed;
- Operating, inventory, and inter-site transfer costs are given and deterministic;
- A price- or demand-response model (see Subsection 6.4.3) is given for pricing decisions.

## 6.4 Production Planning Model

### 6.4.1 Deterministic Model

We begin by describing the deterministic multi-period, multi-site production planning model is described as follows. It extends the short-term planning model described in [Park \*et al.\* \(2006\)](#) by accounting for multiple sites. The procurement contract and pricing models are described in [Subsection 6.4.2](#) and [Subsection 6.4.3](#). Lastly, the stochastic programming model is discussed in [Subsection 6.4.4](#).

## Nomenclature

### Indices

$i$	Process of chemical plants
$j$	Chemical
$s$	Site
$t$	Time period

### Sets

$I$	Set of process of chemical plants
$I_j$	Set of processes that consume chemical $j$
$IS_s$	Set of processes that belong to site $s$
$J_i$	Set of chemicals involved in process $i$
$JM_i$	Set of main products of process $i$
$JP$	Set of products
$JR$	Set of raw materials
$O_j$	Set of processes that produce chemical $j$
$S$	Set of sites
$T$	Set of time periods

### Parameters

$a_{j,s,t}^L, a_{j,s,t}^U$	Lower and upper bounds on availability of raw material $j$ to site $s$ at time period $t$
$\alpha_{j,s,t}^{\text{spot}}$	Spot market price of raw material $j$ to site $s$ at time period $t$
$\delta_{i,s,t}$	Operating cost of process $i$ in site $s$ at time period $t$
$\eta_{j,s,s',t}$	Inter-site transfer cost of chemical $j$ from site $s$ to site $s'$ at time period $t$



$F_{j,s,s',t}^U$	Upper bound on inter-site transfer of chemical $j$ from site $s$ to site $s'$ at time period $t$
$\mu_{i,j,s}$	Mass factor of product $j$ in process $i$ in site $s$
$Q_{i,s,t}$	Production capacity of process $i$ in site $s$ at time period $t$
$V_{j,s,t}^U$	Upper bound on inventory of chemical $j$ in site $s$ at time period $t$
$\xi_{j,s,t}$	Inventory cost of chemical $j$ in site $s$ at time period $t$

## Variables

$F_{j,s,s',t}$	Inter-site transfer amount of product $j$ from site $s$ to site $s'$ at time $t$
$P_{j,s,t}$	Purchase amount of raw material $j$ by site $s$ at time period $t$
$S_{j,t}$	Aggregated sales amount of product $j$ for all sites at time $t$
$SS_{j,s,t}$	Sales amount of product $j$ from site $s$ at time $t$
$V_{j,s,t}$	Inventory level of chemical $j$ in site $s$ at time $t$
$W_{i,j,s,t}$	Amount of chemical $j$ consumed or produced in process $i$ in site $s$ at time $t$

The objective function to be maximized is the profit of the overall production planning decisions as given in equation (6.1). The first term is the revenue obtained from sales for all products  $j$  and time periods  $t$ . The second term is the overall purchase costs from the spot market and/or contracts. These two terms will be defined later. The third term accounts for the operating costs of plant  $i$  in each site  $s$ . The fourth term is the total inventory cost. The final term accounts for the inter-site transfer costs.

$$\begin{aligned} \text{PROFIT} = & \sum_{j \in \text{JP}} \sum_{t \in T} \text{SALES}_{j,t} - \sum_{j \in \text{JR}} \sum_{t \in T} \text{COST}_{j,t} - \sum_{s \in S} \sum_{i \in \text{IS}_s} \sum_{j \in \text{JM}_i} \sum_{t \in T} \delta_{i,s,t} W_{i,j,s,t} \\ & - \sum_{j \in J} \sum_{s \in S} \sum_{t \in T} \xi_{j,s,t} V_{j,s,t} - \sum_{s \in S} \sum_{s' \in S} \sum_{j \in J} \sum_{t \in T} \eta_{j,s,s',t} F_{j,s,s',t} \end{aligned} \quad (6.1)$$

A simple input-output relationship is used to model the chemical processes as shown in equation (6.2). The mass factor  $\mu_{i,j,s}$  (positive for inputs, negative for outputs) is defined for the main products  $j$  of process  $i$ . The process capacity is enforced in equation (6.3).

$$W_{i,j,s,t} = |\mu_{i,j,s}| W_{i,j',s,t} \quad \forall s \in S, i \in \text{IS}_s, j \in J_i, j' \in \text{JM}_i, t \in T \quad (6.2)$$

$$W_{i,j,s,t} \leq Q_{i,s,t} \quad \forall s \in S, i \in \text{IS}_s, j \in \text{JM}_i, t \in T \quad (6.3)$$

Raw material availability is represented by equation (6.4).

$$a_{j,s,t}^L \leq P_{j,s,t} \leq a_{j,s,t}^U \quad \forall s \in S, j \in \text{JR}, t \in T \quad (6.4)$$

A material balance for each chemical  $j$  in every site  $s$  at time period  $t$  is given by equation (6.5). The sales amount for each site,  $SS_{j,s,t}$ , is aggregated

to all sites in equation (6.6).

$$V_{j,s,t-1} + \sum_{i \in O_j} W_{i,j,s,t} + P_{j,s,t} = V_{j,s,t} + \sum_{i \in I_j} W_{i,j,s,t} + \sum_{\substack{s' \in S \\ s' \neq s}} F_{j,s,s',t} + SS_{j,s,t} \quad \forall s \in S, j \in J, t \in T \quad (6.5)$$

$$S_{j,t} = \sum_{s \in S} SS_{j,s,t} \quad \forall j \in J, t \in T \quad (6.6)$$

Upper bounds on inventory levels and inter-site transfers are represented by equations (6.7) and (6.8).

$$V_{j,s,t} \leq V_{j,s,t}^U \quad \forall s \in S, j \in J, t \in T \quad (6.7)$$

$$F_{j,s,s',t} \leq F_{j,s,s',t}^U \quad \forall (s, s') \in S, s \neq s', j \in J, t \in T \quad (6.8)$$

The purchase cost term in the objective function,  $\text{COST}_{j,t}$ , is defined in the next subsection.

#### 6.4.2 Procurement Contract Models

We use the same three contract types described in [Park \*et al.\* \(2006\)](#): discount after a certain purchased amount, bulk discount, and fixed duration contracts. These contract types are modeled with disjunctions and reformulated as mixed-integer linear programming (MILP) constraints. In this work, we employ the same reformulated constraints presented in the original paper with the only difference being that all variables have an additional index  $s$  for sites (see [Appendix D.1](#)). [Figure 6.2](#) illustrates the three contract types considered in this work.

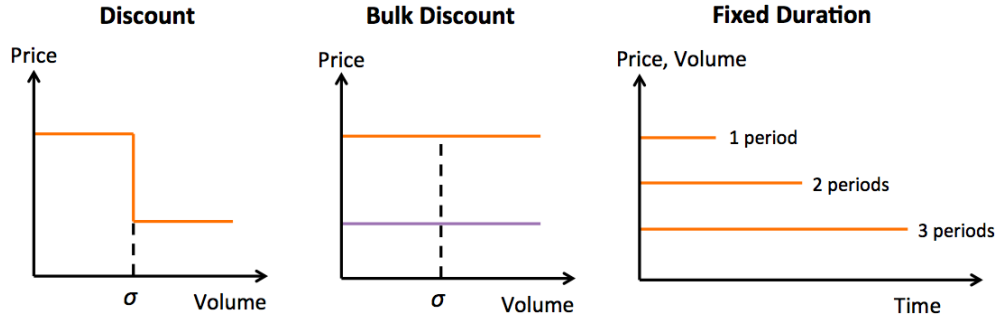


Figure 6.2: Schematic of contract types. **Discount**: pay higher price for quantity up to threshold value  $\sigma$ , and pay discounted price for any amount beyond  $\sigma$ . **Bulk Discount**: if purchased amount exceeds threshold value  $\sigma$ , then pay discounted price; otherwise, pay higher price. **Fixed Duration**: pay discounted price according to contract term duration.

Let  $c \in C$  denote a contract type. The total purchased amount and total procurement cost have two components, market (spot) and contract with supplier. The former is subject to uncertainty as will be discussed later. Equations

(6.9) and (6.10) are used to calculate these two quantities, respectively.

$$P_{j,s,t} = P_{j,s,t}^{\text{spot}} + \sum_{c \in C} P_{j,s,t}^c \quad \forall s \in S, j \in \text{JR}, t \in T \quad (6.9)$$

$$\text{COST}_{j,t} = \sum_{s \in S} \left[ \alpha_{j,s,t}^{\text{spot}} P_{j,s,t}^{\text{spot}} + \sum_{c \in C} \text{COST}_{j,s,t}^c \right] \quad \forall j \in \text{JR}, t \in T \quad (6.10)$$

where  $\text{COST}_{j,s,t}^c$  is the procurement cost of contract type  $c$  for raw material  $j$  in site  $s$  at time period  $t$ . Expressions for  $\text{COST}_{j,s,t}^c$  for each contract type  $c$  are given in [Park et al. \(2006\)](#) and can be found in [Appendix D.1](#).

### 6.4.3 Price Optimization Models

As mentioned in the literature review ([Subsection 6.2.2](#)), the main goal in price optimization is for a company to set and adjust its prices to maximize profitability. Pricing decisions rely on price-response models (PRMs) usually denoted by  $d(p)$ , where  $p$  is the price of a product and  $d$  is the expected demand for that product and its given price. As commonly treated in the literature, we use demand and sales interchangeably as we assume that the manufacturer has enough capacity to meet its customer demand. Three PRMs typically encountered in the literature are given in [Table 6.1](#).

Table 6.1: Typical price-response models (PRMs).

Model	Equation
Linear	$d(p) = \beta_0 - \beta_1 p$
Constant-Elasticity	$d(p) = \beta_2 p^{-E_d}$
Logit	$d(p) = \frac{\beta_3 e^{-(\beta_4 + \beta_5 p)}}{1 + e^{-(\beta_4 + \beta_5 p)}}$

The different  $\beta$ s are estimated parameters (e.g., from a regression analysis), and  $E_d$  is the price elasticity of demand, which is a measure used in economics to show the responsiveness, or elasticity, of the quantity demanded of a good or service to a change in its price.

We note that for typical production planning models, the sales flow decision variable appears in both constraints and objective function. Therefore, the choice of a nonlinear PRM results in nonlinear terms in both constraints (e.g., equation (6.5)) and objective function (in the form of selling price  $\times$  sales). In order to restrict nonlinear terms to the objective function, it may be advantageous to use demand-response models (DRMs), i.e.,  $p(d)$ . The PRMs described in [Table 6.1](#) can be inverted to DRMs as shown in [Table 6.2](#).

Table 6.2: Typical demand-response models (DRMs).

Model	Equation
Linear	$p(d) = \beta'_0 - \beta'_1 d$
Constant-Elasticity	$p(d) = \beta'_2 d^{-1/E_d}$
Logit	$p(d) = \frac{1}{\beta_5} \left[ \ln \left( \frac{\beta_3 - d}{d} \right) - \beta_4 \right]$

The following relationships hold:  $\beta'_0 = \frac{\beta_0}{\beta_1}$ ,  $\beta'_1 = \frac{1}{\beta_1}$ , and  $\beta'_2 = \beta_2^{1/E_d}$ .

Therefore, the sales term (selling price  $\times$  sales flow, i.e.,  $p(d) \times d$ ) in the objective function in equation (6.1) for each DRM is given as follows:

$$\text{SALES}_{j,t} = \beta'_0 S_{j,t} - \beta'_1 S_{j,t}^2 \quad \forall j \in \text{JP}, t \in T \quad (6.11)$$

$$\text{SALES}_{j,t} = \beta'_2 S_{j,t}^{1-1/E_j} \quad \forall j \in \text{JP}, t \in T \quad (6.12)$$

$$\text{SALES}_{j,t} = \frac{1}{\beta_5} \left[ \ln \left( \frac{\beta_3 - S_{j,t}}{S_{j,t}} \right) - \beta_4 \right] S_{j,t} \quad \forall j \in \text{JP}, t \in T \quad (6.13)$$

We note that the linear DRM yields a concave quadratic term in the objective function, the constant-elasticity DRM results in a nonlinear, but concave term for  $E_j > 1$  (elastic products) and nonlinear, nonconvex term for  $0 < E_j < 1$  (inelastic products), and the logit DRM yields, in principle, a general nonlinear, nonconvex term. We show in [Appendix D.2](#) that the logit DRM results in a concave term for specific conditions on its parameters.

**Remark.** Even though the PRMs (DRMs) listed in [Table 6.1](#) ([Table 6.2](#)) are typically used in the literature, we argue that general, and likely more complex regression models can be employed to model the relationship between selling price, demand, and possibly additional predictors. For instance, a more general, multiple linear PRM is given in equation (6.14),

$$d(X) = \beta_0 + \sum_{i=1}^m \beta_i X_i \quad (6.14)$$

where the  $\beta$ s are parameters to be estimated, and  $X_i$ ,  $i = 1, \dots, m$ , are predictors (or regressors). Examples of predictors in addition to the selling price  $p$  may include raw material prices, economic indicators (e.g., gross domestic product), crude oil and other product prices as well as their time-lagged values. Note that the multiple linear PRM can also be inverted to a multiple linear DRM generally denoted by  $p(X)$ .

Nonlinear PRMs or DRMs may capture more accurately the relationship between predictors and the response variable at the expense of yielding a (mixed-integer) nonlinear production planning model. General parametric, semi-parametric, or nonparametric nonlinear models can be employed. An

alternative to pre-specifying regression models is Symbolic Regression (also known as Genetic Programming) (Koza, 1992), which is a technique to generate general regression models from data via evolutionary programming concepts.

#### 6.4.4 Stochastic Programming Model

The previous two subsections presented ways to anticipate the uncertainty in the production planning decision-making process. However, the uncertainty in the market remains regardless if the manufacturer signs contracts with suppliers (and/or customers). Moreover, the predictive accuracy of demand-response models (DRMs) is not guaranteed to be perfect. In this subsection, we extend the deterministic optimization model presented in Subsection 6.4.1 by including sources of uncertainty in both supply and demand sides.

We propose a scenario-based two-stage stochastic programming framework (Birge & Louveaux, 2011) to explicitly account for uncertainty in spot market prices of raw materials and the predictability of DRMs. More specifically, we describe a multi-period, two-stage stochastic programming model whose first-stage decisions are the selection of contracts (discrete decisions) and the selling prices (which determine sales flows), and second-stage decisions include the remaining flows and inventory levels in the network.

Let  $k \in K$  represent the index and discrete set of scenarios used to approximate the possible realizations of the uncertain parameters. From equations (6.9) and (6.10), the total purchase amount and cost may have two contributions: spot market and signed contracts. Contractual agreements are considered deterministic and fixed, and must be honored if selected. The volatility in the market is then represented by raw material prices and, possibly, availability. Therefore, the corresponding scenario-based constraints are given by

$$P_{j,s,t,k} = P_{j,s,t,k}^{\text{spot}} + \sum_{c \in C} P_{j,s,t}^c \quad \forall s \in S, j \in \text{JR}, t \in T, k \in K \quad (6.15)$$

$$\text{COST}_{j,t,k} = \sum_{s \in S} \left[ \alpha_{j,s,t,k}^{\text{spot}} P_{j,s,t,k}^{\text{spot}} + \sum_{c \in C} \text{COST}_{j,s,t}^c \right] \quad \forall j \in \text{JR}, t \in T, k \in K \quad (6.16)$$

Spot market prices of raw material are typically predicted using time series models. The literature on time series analysis is extensive and includes linear, nonlinear, parametric, semi-parametric, and nonparametric models (Brockwell & Davis, 2002; Fan & Yao, 2002b; Box, Jenkins, & Reinsel, 2008). Figure 6.3 illustrates the simulation of time series models that can be used to generate scenarios.

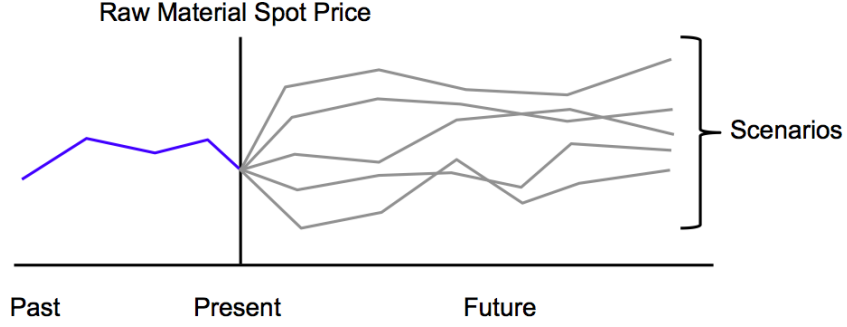


Figure 6.3: Time series model simulation for scenario generation.

For the demand side, the uncertainty in demand or selling price can be quantified by the distribution of residuals of the regression models as illustrated in Figure 6.4. By construction, the residuals must be independent and identically distributed (i.i.d.), have zero mean and constant variance, and be uncorrelated with each other. Additional assumptions are typically used, e.g., the residuals follow a normal distribution with zero mean and known variance  $\sigma^2$ .

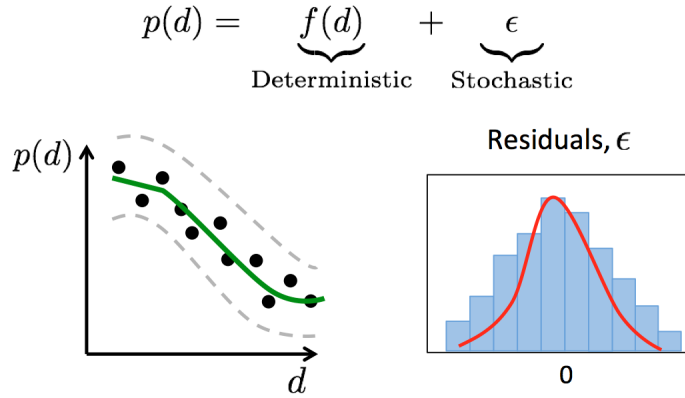


Figure 6.4: Deterministic and stochastic components of a regression-based demand-response model (DRM). The region delimited by the dashed lines on the left plot represents possible realizations of price values according to the probabilistic nature of the residuals,  $\epsilon$ . The distribution of the residuals can be sampled to generate scenarios.

A scenario-based version of a general DRM can be written for a particular realization of the residuals (scenario  $k$ ) as follows:

$$p(d) = f(d) + \epsilon_k$$

Therefore, the scenario-based version of the sales term in the objective function (equation (6.1)) of each DRM listed in Table 6.2 is given as follows:

- Linear:

$$\text{SALES}_{j,t,k} = (\beta'_0 - \beta'_1 S_{j,t} + \epsilon_k) S_{j,t} = \beta'_0 S_{j,t} - \beta'_1 S_{j,t}^2 + S_{j,t} \epsilon_k \quad (6.17)$$

- Constant-Elasticity:

$$\text{SALES}_{j,t,k} = (\beta'_2 S_{j,t}^{-1/E_j} + \epsilon_k) S_{j,t} = \beta'_2 S_{j,t}^{1-1/E_j} + S_{j,t} \epsilon_k \quad (6.18)$$

- Logit:

$$\begin{aligned} \text{SALES}_{j,t,k} &= \left\{ \frac{1}{\beta_5} \left[ \ln \left( \frac{\beta_3 - S_{j,t}}{S_{j,t}} \right) - \beta_4 \right] + \epsilon_k \right\} S_{j,t} \\ &= \frac{1}{\beta_5} \left[ \ln \left( \frac{\beta_3 - S_{j,t}}{S_{j,t}} \right) - \beta_4 \right] S_{j,t} + S_{j,t} \epsilon_k \end{aligned} \quad (6.19)$$

As mentioned in the remark at the end of the previous subsection, when using a general (and possibly nonlinear) DRM denoted by  $p(X)$ , where  $X$  is a vector of predictors including  $S_{j,t}$  and potentially others, the sales term can be written similarly as for the three special cases considered in this chapter. That is, a sample of the residual distribution,  $\epsilon_k$ , is added to the deterministic part of the model  $p(X)$  and the result is multiplied by  $S_{j,t}$ .

Both deterministic and stochastic models are mixed-integer programming (MIP) problems. For linear DRMs, they are concave MIQP models, and for constant-elasticity and logit DRMs, they are concave and separable MINLP models depending on the values of the parameters in the DRMs. The complete deterministic equivalent model for the stochastic formulation is given by the following general MINLP model:

$$\begin{aligned} \max \mathbb{E}[\text{PROFIT}] &= \sum_{k \in K} \pi_k \left[ \sum_{j \in \text{JP}} \sum_{t \in T} \text{SALES}_{j,t,k} - \sum_{j \in \text{JR}} \sum_{t \in T} \text{COST}_{j,t,k} \right. \\ &\quad - \sum_{s \in S} \sum_{i \in \text{IS}_s} \sum_{j \in \text{JM}_i} \sum_{t \in T} \delta_{i,s,t} W_{i,j,s,t,k} - \sum_{j \in J} \sum_{s \in S} \sum_{t \in T} \xi_{j,s,t} V_{j,s,t,k} \\ &\quad \left. - \sum_{s \in S} \sum_{\substack{s' \in S \\ s' \neq s}} \sum_{j \in J} \sum_{t \in T} \eta_{j,s,s',t} F_{j,s,s',t,k} \right] \end{aligned} \quad (6.20)$$

subject to

$$W_{i,j,s,t,k} = |\mu_{i,j,s}| W_{i,j',s,t,k} \quad \forall s \in S, i \in \text{IS}_s, j \in J_i, j' \in \text{JM}_i, t \in T, k \in K \quad (6.21)$$

$$W_{i,j,s,t,k} \leq Q_{i,s,t} \quad \forall s \in S, i \in \text{IS}_s, j \in \text{JM}_i, t \in T, k \in K \quad (6.22)$$

$$a_{j,s,t}^L \leq P_{j,s,t,k} \leq a_{j,s,t}^U \quad \forall s \in S, j \in \text{JR}, t \in T, k \in K \quad (6.23)$$

$$\begin{aligned} V_{j,s,t-1,k} + \sum_{i \in O_j} W_{i,j,s,t,k} + P_{j,s,t,k} &= V_{j,s,t,k} + \sum_{i \in I_j} W_{i,j,s,t,k} + \sum_{\substack{s' \in S \\ s' \neq s}} F_{j,s,s',t,k} + SS_{j,s,t} \\ &\quad \forall s \in S, j \in J, t \in T, k \in K \end{aligned} \quad (6.24)$$

$$(6.25)$$

$$P_{j,s,t,k} = P_{j,s,t,k}^{\text{spot}} + \sum_{c \in C} P_{j,s,t,k}^c \quad \forall s \in S, j \in \text{JR}, t \in T, k \in K \quad (6.26)$$

$$\text{COST}_{j,t,k} = \sum_{s \in S} \left[ \alpha_{j,s,t,k}^{\text{spot}} P_{j,s,t,k}^{\text{spot}} + \sum_{c \in C} \text{COST}_{j,s,t,k}^c \right] \quad \forall j \in \text{JR}, t \in T, k \in K \quad (6.27)$$

$$S_{j,t} = \sum_{s \in S} SS_{j,s,t} \quad \forall j \in J, t \in T \quad (6.28)$$

$$V_{j,s,t,k} \leq V_{j,s,t}^U \quad \forall s \in S, j \in J, t \in T, k \in K \quad (6.29)$$

$$F_{j,s,s',t,k} \leq F_{j,s,s',t}^U \quad \forall (s, s') \in S, s \neq s', j \in J, t \in T, k \in K \quad (6.30)$$

Contract selection constraints: Equations (D.1) to (D.18) ([Appendix D.1](#))

$$W_{i,j,s,t,k}, V_{j,s,t,k}, F_{j,s,s',t,k}, P_{j,s,t,k}, S_{j,t}, SS_{j,s,t} \geq 0 \\ y_{j,s,t}^c = \{0, 1\}$$

where  $\pi_k$  is the probability of scenario  $k$ ,  $\text{SALES}_{j,t,k}$  which involves nonlinear terms, is given by equation (6.17), (6.18), or (6.19), and  $y_{j,s,t}^c$  represent the binary variables for the selection of contracts ([Appendix D.1](#)). For illustration purposes, we will assume in the numerical examples that all scenarios have the same probability. Data-based scenario tree generation techniques can be used to calculate not only the probabilities of scenarios, but also the values of the outcomes in order to match statistical properties of historical and forecast data ([Calfa et al., 2014](#)).

## 6.5 Numerical Examples

To illustrate our proposed approach, we consider the two following examples. The first example involves a single site with three chemical plants, one raw material, one intermediate product, and two finished products. The same demand-response model (DRM) is used for both finished products. The second example considers a larger process network with 38 processes and 28 chemicals. In [Subsection 6.5.3](#), we present a reformulation of the mixed-integer nonlinear programming (MINLP) models that decreases the solution times.

All optimization models were implemented in AIMMS 4.3 ([Roelofs & Bisschop, 2015](#)) and solved on a desktop computer with the following specifications: Dell Optiplex 990 with 4 Intel® Core™ i7-2600 CPUs at 3.40 GHz (total 8 threads), 8 GB of RAM, and running Windows 7 Enterprise. The mixed-integer quadratic (MIQP) solver used was GUROBI 6.0. The mixed-integer nonlinear programming (MINLP) models were solved using the convex AOA



algorithm implemented in AIMMS. AOA stands for AIMMS Outer Approximation, which implements an outer approximation algorithm that alternates between the solution of a master problem (MILP) and a subproblem (NLP) (Duran & Grossmann, 1986). The MILP and NLP solvers used were GUROBI 6.0 and CONOPT 3.14V, respectively. The convex AOA algorithm is based on the work by Quesada & Grossmann (1992) and is described in Hunting (2012). The lower bound for the total sales variables,  $S_{j,t}$ , was set to 0.1. Time series analysis was performed in the R programming language (R Core Team, 2015) with the `forecast` package (Hyndman *et al.*, 2014). We conclude this section by proposing a reformulation of the MINLP models that considerably reduces the solution times.

### 6.5.1 Example 1: Small Process Network

The process network for Example 1 is shown in Figure 6.5. It consists of raw material A, an intermediate product B, finished products C and D (only product D can be stored), and processes (plants) P1, P2, and P3. A time horizon consisting of 6 time periods (months) is considered. All data for this example can be found in Appendix D.3.

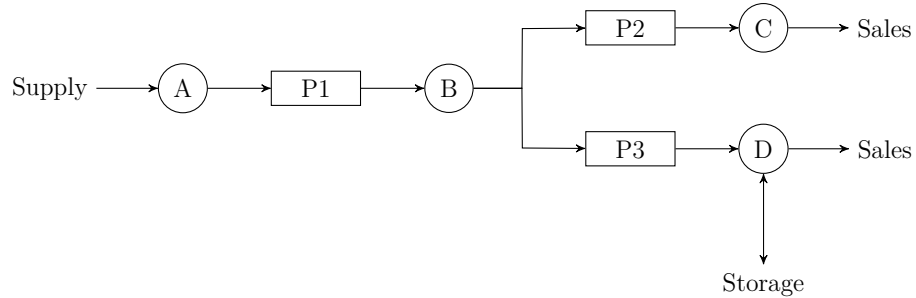


Figure 6.5: Process network for Example 1.

The stochastic programming model has ten scenarios and was solved using its deterministic equivalent formulation. The values of the spot market prices,  $\alpha_{j,s,t,k}^{\text{spot}}$ , for raw material A were generated from a simulation of a seasonal autoregressive integrated moving average (ARIMA) time series model, which is shown in Figure 6.6. Only three scenarios (red dotted lines in the shaded region) are displayed in order to illustrate the simulation results.

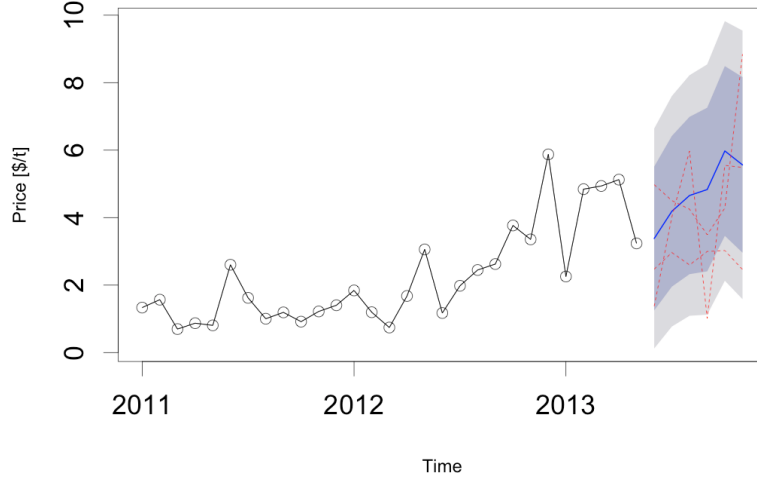


Figure 6.6: Simulated scenarios (red dotted lines in the shaded region) for the spot market prices of raw material A. Mean forecast is represented by blue solid line in the shaded region.

Three demand-response models (DRMs) were considered: linear, constant-elasticity, and logit. The parameters for each DRM are shown in Table 6.3, and the three DRMs are illustrated in Figure 6.7. The stochastic part of the DRMs for products C and D is assumed to be normally distributed,  $N(0, 1)$  and  $N(0, 2)$ , respectively.

Table 6.3: Parameter values for the DRMs in Example 1.

DRM	Product	
	C	D
Linear	$\beta'_0 = 4.00 \text{ \$/t}$	$\beta'_0 = 4.33 \text{ \$/t}$
	$\beta'_1 = 0.07 \text{ \$/t}^2$	$\beta'_1 = 0.06 \text{ \$/t}^2$
Constant-Elasticity	$\beta'_2 = 10.00 \text{ \$/t}^{0.5}$	$\beta'_2 = 21.00 \text{ \$/t}^{0.33}$
	$E_C = 2$	$E_D = 1.5$
Logit	$\beta_3 = 40.00 \text{ t}$	$\beta_3 = 50.00 \text{ t}$
	$\beta_4 = -2.00$	$\beta_4 = -3.00$
	$\beta_5 = 0.70 \text{ t/\$}$	$\beta_5 = 1.00 \text{ t/\$}$

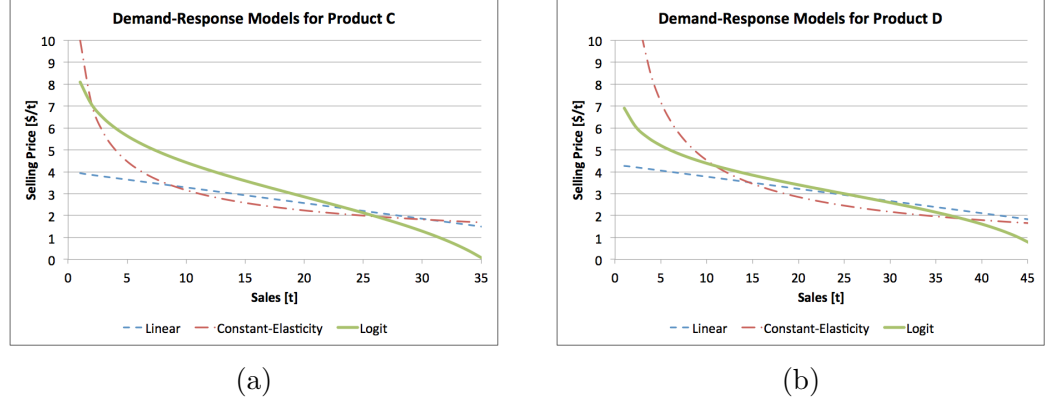


Figure 6.7: Illustration of DRMs for products C (a) and D (b).

A total of six different optimization problems were solved: (i) deterministic with linear DRM, (ii) stochastic with linear DRM, (iii) deterministic with constant-elasticity DRM, (iv) stochastic with constant-elasticity, (v) deterministic with logit DRM, and (vi) stochastic with logit DRM. Problems (i) and (ii) are concave MIQPs, whereas problems (iii) – (vi) are concave MINLPs (maximization). Table 6.4 shows the computational results for all problems. All deterministic (stochastic) problems contain 60 (60) binary variables, 184 (725) continuous variables, and 264 (750) constraints.

Table 6.4: Computational results for Example 1. C-E stands for Constant-Elasticity.

	Deterministic			Stochastic		
	<i>Linear</i>	<i>C-E</i>	<i>Logit</i>	<i>Linear</i>	<i>C-E</i>	<i>Logit</i>
MIQP/MILP Nodes	658	528	3,801	140	1,181	3,868
NLP Solves	–	34	114	–	55	224
Wall Time [s]	0.27	0.56	1.33	0.48	1.33	7.29

We restrict our analysis to the results of problems (i) and (ii) (i.e., with linear DRMs). In other words, we assess the impact of considering uncertainty in both raw material prices and the relationship between selling price and demand captured by linear DRMs. The profit of the deterministic problem is \$94.03, and the expected profit of the stochastic problem is \$259.69. The difference in profit values between the solutions is mainly attributed to the predominantly lower selling prices suggested by the stochastic model in comparison to the deterministic solution. Consequently, the lower prices obtained in the stochastic solution are correlated with more sales as shown in Figure 6.8, and higher overall profit.

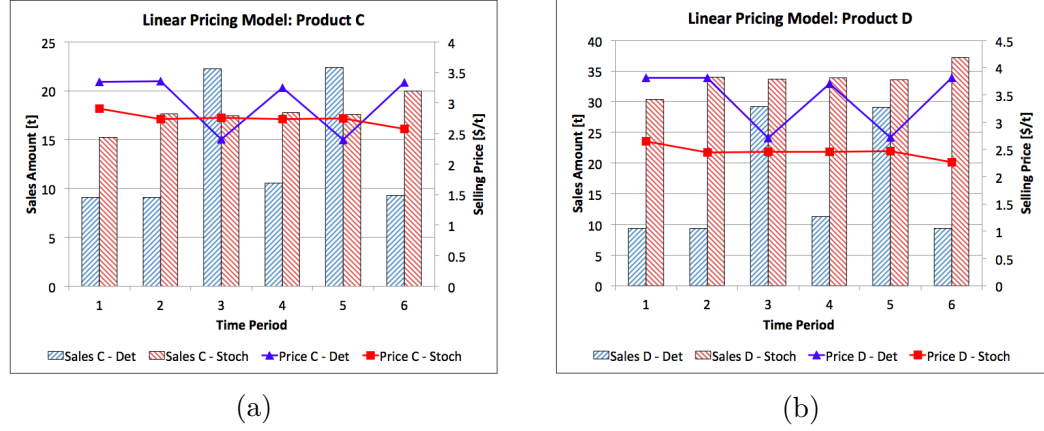


Figure 6.8: Selling price and sales for products C (a) and D (b) in Example 1. “Det” and “Stoch” stand for deterministic and stochastic model solutions, respectively.

The amount of sales quantified by the DRM translates into the required purchase amounts. Figure 6.9 shows these purchases that include both contracts and spot market (average). According to the solution to the deterministic model, the manufacturer not only purchases raw material A from contracts with the supplier, but also from the spot market at certain time periods. The total average purchase amounts from the spot market represent 29% of total raw material purchases. However, the total purchase amount in the deterministic solution (155 t) is lower than what is required by the stochastic model (270 t). From the stochastic solution, the manufacturer does not use the spot market as a source of raw material. The reasons are two-fold: unfavorable market price variability and the need to increase the purchase amounts due to the increase in sales. Thus, the manufacturer opts for signing more contracts as a more cost-effective and less uncertain strategy. We note that similar trends occur in the other problems with the other two DRMs.

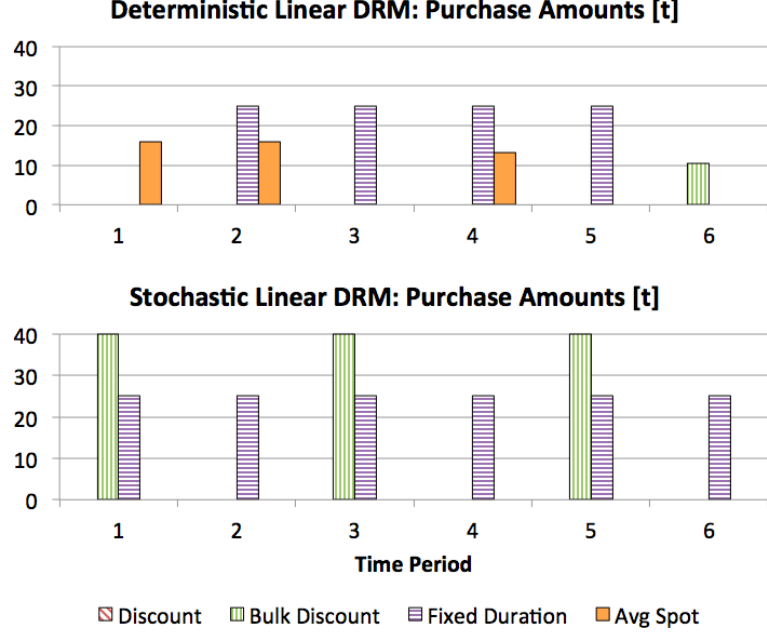


Figure 6.9: Purchase amounts from contracts and spot market (average) for the deterministic and stochastic problems in Example 1.

Even though the expected profit of the stochastic model is higher than the profit of the deterministic model, it is not appropriate to directly compare the objective function values of stochastic and deterministic solutions, since the latter only typically accounts for the average values of the uncertain parameters, and not the different possible outcomes considered by the former. A more meaningful comparison is performed through the calculation of the Value of the Stochastic Solution (VSS) as described in [Birge & Louveaux \(2011\)](#). The VSS is a metric to evaluate the performance of a deterministic solution in a stochastic environment. This is accomplished by solving the deterministic model with average values for the uncertain parameters, and then fixing the first-stage variables in the stochastic programming model to the values of the corresponding variables in the solution of the deterministic model. For a maximization problem, the VSS is calculated as follows:

$$VSS = RP - EV \quad (6.31)$$

where  $RP$  is the objective function value of the *recourse problem* (original stochastic model) and  $EV$  is the *expected value* solution (stochastic model with fixed first-stage decisions from deterministic model with average values for the uncertain parameters). In this motivating example, we have that  $VSS = \$259.69 - \$190.47 = \$69.22$ , which represents the additional expected profit from implementing the stochastic solution over the deterministic one, both with purchase contracts and pricing models. The VSS is approximately 27% of the solution obtained with the proposed method.

The calculation of the VSS can be modified to compare the solution of the proposed method with a *deterministic base case* model, which is typically

used as a first approach in practical applications. The deterministic base case does not account for contract selection, and the demand and selling prices are fixed parameters. We call this modified metric a *restricted* VSS or RVSS. The formula is the same as in equation (6.31) with the difference being the meaning of the term  $EV$ , which now corresponds to the objective function value of the stochastic model with first-stage variables fixed to the solution of the deterministic base case. In this example,  $RVSS = \$259.69 - \$134.10 = \$125.59$ , i.e., the proposed method results in approximately 48% additional expected profits relative to the deterministic base case with no contracts and fixed demand and selling prices.

### 6.5.2 Example 2: Large Process Network

Example 2 is based on the process network shown in Figure 6.10 and discussed in Iyer & Grossmann (1998) (see the third example). The process network can be considered an integrated petrochemical site in which chemicals are produced in separate, dedicated production plants that are situated at the same geographic location due to economies of scale related to utility operations and supply chain cost advantages (Wassick, 2009). We do not include explicit data for this problem because of its size; however, this information is available from the authors upon request.

The process network has 38 processes (plants) and 28 chemicals, in which 17 are products and 11 are raw materials and intermediates. The time horizon is divided into 4 time periods that represent quarters in a year. Purchasing contracts are considered for all raw materials, and the three DRMs considered in this chapter are used for all finished products. Similarly as in Example 1, 10 scenarios are considered. Each scenario contains spot market prices and samples of residuals of the DRMs. The spot market prices for the raw materials in each scenario were generated via simulation of ARIMA models, and for illustration purposes, the regression residuals of the DRMs are assumed to be normally distributed with given means and standard deviations.

Similarly as in Example 1, six different optimization problems were solved. They are the same models solved in Example 1, but with different data. Table 6.5 shows the computational results for all problems. All deterministic (stochastic) problems contain 400 (400) binary variables, 1,349 (6,777) continuous variables, and 2,383 (6,055) constraints. Note that, as the problem size increases relative to the previous example, the convex outer-approximation algorithm requires significantly more time, nodes in the solution of master MILP problems, and calls to the NLP subproblem. This is an indication that as the model size increases decomposition strategies should be required to reduce the computing time.

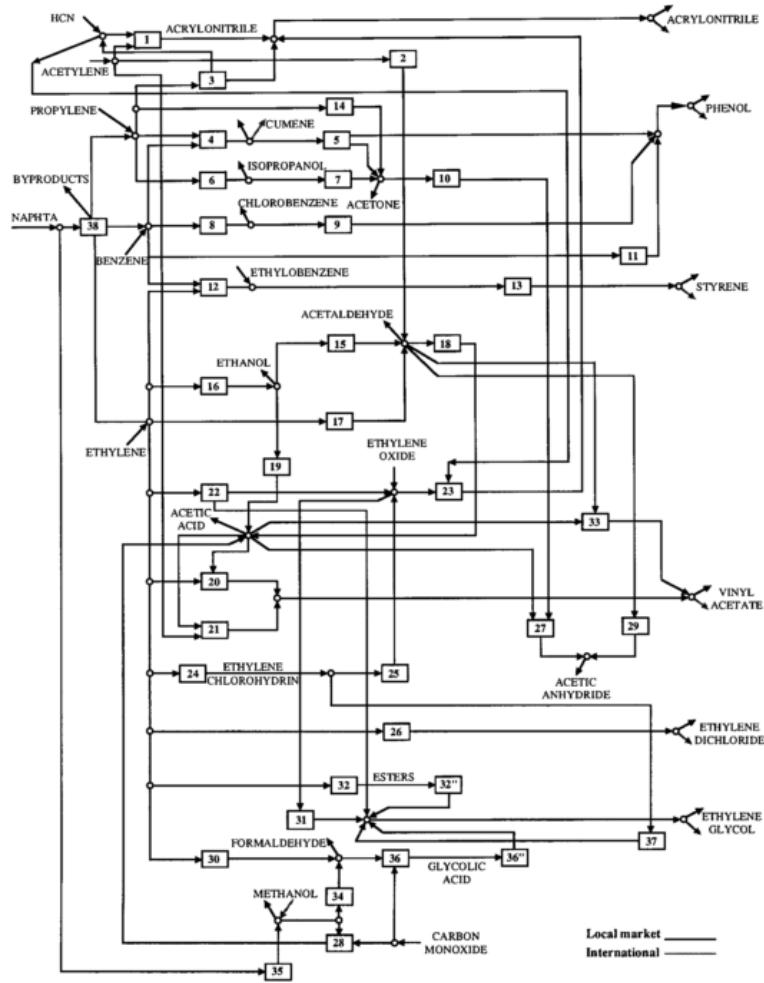


Figure 6.10: Process network in Example 2 (Iyer & Grossmann, 1998).

Table 6.5: Computational results for Example 2. C-E stands for Constant-Elasticity.

	Deterministic			Stochastic		
	<i>Linear</i>	<i>C-E</i>	<i>Logit</i>	<i>Linear</i>	<i>C-E</i>	<i>Logit</i>
MIQP/MILP Nodes	129	31,611	1,177,918	72	350	62,840
NLP Solves	—	220	1,571	—	31	366
Wall Time [s]	1.36	57.25	15,341	2.12	233.86	2,708

We restrict the analysis of the results to the deterministic and stochastic models with linear DRMs. The profit of the deterministic solution is \$33,783.48, and the expected profit of the stochastic solution is \$37,722.05. A key difference between the deterministic and stochastic solutions is the different types of contracts and the respective contracted amounts for certain products at each time period. The total sales influence the production rates, which in turn are connected to the total amount of purchases. Finally, the

combined effect of these decisions and inventory considerations impact the overall profitability of the production plan under uncertain market conditions.

Proceeding as in the previous example, the calculated Value of the Stochastic Solution is  $VSS = \$37,722.05 - \$34,076.98 = \$3,645.07$ , which represents 10% of the solution obtained with the proposed method. However, when compared to the deterministic base case (i.e., no contracts and fixed demand and selling prices), the restricted VSS is  $RVSS = \$37,722.05 - \$19,677.10 = \$18,044.95$ , which amounts to 48% of the solution obtained with the proposed method. A summary of results for the two solutions with respect to total average purchase amounts, total average inventory levels, and individual scenario profit is shown in [Figure 6.11](#). Note that the solution obtained with the proposed method (represented by the label RP) displays lower total average amounts for purchased raw materials and inventory than the expected value solution in all time periods. In fact, a slack variable had to be added to the right-hand side of the material and inventory balance constraint (equation (6.24)) and penalized in the objective function (equation (6.20)) in order to obtain a feasible solution. This indicates that the solution from the deterministic base case fails to balance inventory, production rates, and sales when subject to uncertainty in both supply and demand for certain chemicals. Note also that the profit of each scenario in the RP solution is always higher than in the EV solution.



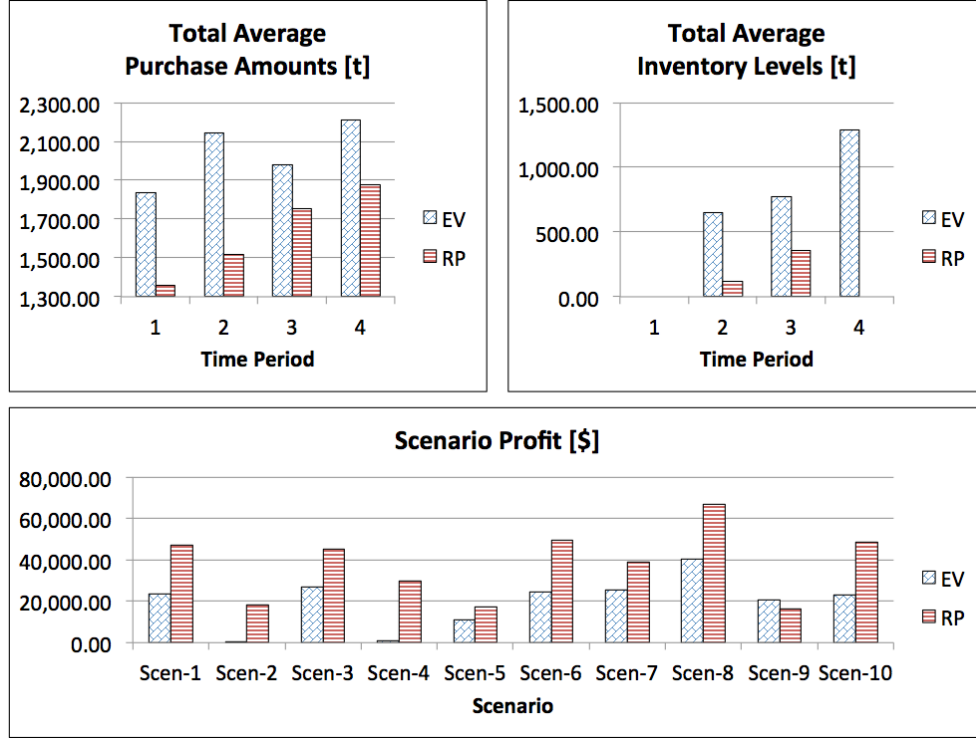


Figure 6.11: Summary of results in Example 2 (linear DRMs). RP and EV stand for recourse problem (proposed method) and expected value problem (stochastic model with first-stage decisions fixed to the solution of the deterministic base case).

The differences between the two solutions may be attributed to the total sales and selling prices. Figure 6.12 shows the optimal values for sales and selling prices for two products, 13 and 23 (acetone and ethylene dichloride, respectively, in the problem discussed in the original paper). Recall that the solution to the EV problem has fixed demand and selling prices. Note that, in RP solution, the sales of product 13 are always larger than in the EV solution, whereas the opposite is true for product 23. In the model used in this chapter, the non-negative sales variables have no upper bounds, but different bounds can be used depending on the application. In addition, simple contract information with suppliers that imposes ranges of the demand can also be used. The design of customer contracts; however, is a more challenging problem, and some suggestions of extensions to the proposed method are given in the conclusion section. Notice that the results shown in the top subfigures of Figure 6.11 are *total* expected amounts, whereas Figure 6.12 shows results for two out of the twenty-eight chemicals.

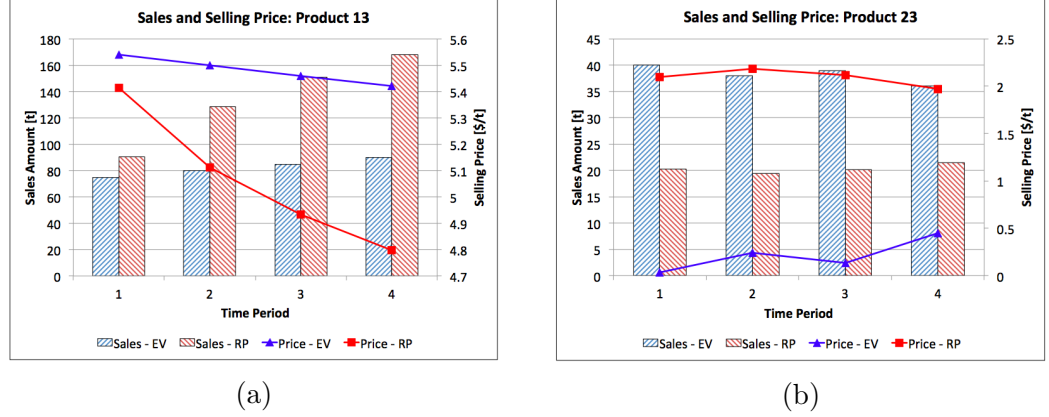


Figure 6.12: Selling price and sales for products 13 (a) and 23 (b) in Example 2 (linear DRMs). RP and EV stand for recourse problem and expected value problem solutions, respectively.

### 6.5.3 Reformulation of MINLP Models

When using constant-elasticity or logit DRMs, the objective function contains nonlinear terms, which are concave (maximization) under certain conditions. We propose transferring the nonlinear terms in the objective function to the set of constraints by introducing non-negative auxiliary variables,  $\text{SALES}_{j,t}^{\text{aux}}$ . Specifically, for the two nonlinear DRMs considered in this work, the sales term of the objective function of the deterministic model in equation (6.1),  $\text{SALES}_{j,t}$ , is simply rewritten as  $\text{SALES}_{j,t}^{\text{aux}}$ , and equivalently in equation (6.20) of the stochastic model, the sales term  $\text{SALES}_{j,t,k}$  becomes  $\text{SALES}_{j,t}^{\text{aux}} + S_{j,t}\epsilon_k$ . The additional set of constraints is given as follows:

- Constant-Elasticity:

$$\text{SALES}_{j,t}^{\text{aux}} \leq \beta_2' S_{j,t}^{1-1/E_j} \quad \forall j \in J, t \in T \quad (6.32)$$

- Logit:

$$\text{SALES}_{j,t}^{\text{aux}} \leq \frac{1}{\beta_5} \left[ \ln \left( \frac{\beta_3 - S_{j,t}}{S_{j,t}} \right) - \beta_4 \right] S_{j,t} \quad \forall j \in J, t \in T \quad (6.33)$$

The advantage of the reformulated model is that more cuts or inequalities (linearizations) are generated and added to the master MILP problem in an outer-approximation algorithm, since  $|J| \times |T|$  cuts are generated (corresponding to the nonlinear constraints) as opposed to a single cut per NLP solve generated by restricting the nonlinear terms to the objective function. One potential disadvantage is the need to solve larger MILP models. However, the increased number of cuts generated in the reformulated model may provide tighter and more accurate outer-approximation of the feasible region, thus decreasing the solution time.

In Table 6.6, the computational results of the proposed reformulation are compared with the results of the original models (nonlinear terms in the objective function only) shown in Tables 6.4 and 6.5. Considerable speed-ups, and decrease in the number of MILP nodes and NLP solves were observed with the reformulated model. Even though the proposed reformulation significantly reduced solution times, decomposition approaches may be necessary as the problem instance increases in size and more general nonlinear DRMs are used.

## 6.6 Conclusions

In this chapter, we have formulated a multi-period, multi-site stochastic programming production planning model that combines optimal procurement contract selection with selling price optimization under supply and demand uncertainty. The proposed approach is manufacturer-centric in the sense that it is the manufacturer’s decision to sign or not contracts with suppliers in order to hedge against spot market supply uncertainty, and the manufacturer can set selling prices for its customers. The possible selection of sales contracts is considered to be the customer’s decision, and was not addressed in this work. With regards to price optimization, the formulation of three price-response models that are typically encountered in the literature was discussed, even though we argued that general regression models can be used. The demand uncertainty is represented by samples from the distribution of regression residuals. We showed that it may be more advantageous to use demand-response models (DRMs), i.e., price as a function of demand, and we provided conditions on the parameters of the logit model that yield a concave revenue term in the objective function.

The proposed approach was illustrated with two numerical examples. In the first example (small process network), the stochastic model set the selling prices predominantly lower than the deterministic model, and had larger sales, and thereby an overall higher expected profit. The larger volume of sales in the stochastic solution relative to the deterministic solution translated into larger purchase amounts in the former, all of which was obtained through contracts (i.e., no purchases from the uncertain spot market). The computational results showed that the optimization models with logit DRMs were more computationally intensive than the models with linear and constant-elasticity DRMs due to the increase in complexity of the MINLP. Similar conclusions were drawn from the second example (large process network), in which the economic advantage of the stochastic solution over the deterministic one was measured by calculating the Value of the Stochastic Solution. The solution time increased considerably in the second example when using the original models. However, transferring the nonlinear terms from the objective function to the set of constraints demonstrated to significantly reduce solution times. Moreover, in both examples, the Value of the Stochastic Solution (VSS) showed a modest improvement of the proposed stochastic method over

Table 6.6: Computational results comparing original and reformulated MINLP models. C-E stands for Constant-Elasticity. Speed-Up is calculated as the ratio between CPU time of original and reformulated models.

<i>Example</i>	<i>Problem</i>	<i>DRM</i>	<b>Original</b>			<b>Reformulation</b>			<i>Speed-Up</i>
			<i>CPU [s]</i>	<i>MILP Nodes</i>	<i>NLP Solves</i>	<i>CPU [s]</i>	<i>MILP Nodes</i>	<i>NLP Solves</i>	
1	Deterministic	C-E	0.56	528	34	0.08	95	6	7x
		Logit	1.33	3,801	114	0.23	1,643	14	6x
	Stochastic	C-E	1.33	1,181	55	0.25	956	9	5x
		Logit	7.29	3,868	224	0.48	3,299	13	15x
2	Deterministic	C-E	57.25	31,611	220	3.28	2,362	24	18x
		Logit	15,341	1,177,918	1,571	6.57	10,902	15	2,335x
	Stochastic	C-E	233.86	350	31	30.61	153	15	8x
		Logit	2,708	62,840	366	19.75	2,002	18	137x

its deterministic solution; however, by comparing the proposed method with a deterministic base case (no contracts and fixed demand and selling price), the calculated restricted VSS showed that the proposed method exhibited significant improvement with regards to expected profitability when uncertainty in both supply and demand is considered.

Possible extensions of the proposed approach include: (1) design of sales contracts with customers, where the decision to sign or not these contracts are made by each customer in a bilevel programming framework ([Shimizu, Ishizuka, & Bard, 1997](#); [Bard, 1998](#)); (2) approximate decomposition approaches when dealing with nonlinear DRMs, for example, via piece-wise linear functions ([Taha, 2010](#)).

# Chapter 7

## Conclusions

This thesis has proposed data-driven methods to handle uncertainty in problems related to Enterprise-wide Optimization (EWO). It also addresses, in [Chapter 2](#), the integration of deterministic planning and scheduling decisions across multiple spatial and temporal scales of a network of batch plants, which is motivated by a real-world industrial problem. [Chapter 3](#) deals with scenario generation via statistical property matching in the context of stochastic programming. A distribution matching problem is proposed that addresses the under-specification shortcoming of the originally proposed moment matching method. [Chapter 4](#) deals with data-driven individual and joint chance constraints with right-hand side uncertainty. Kernel smoothing is used to estimate the true and unknown distributions. [Chapter 5](#) proposes the use of quantile regression to model production variability in the context of Sales & Operations Planning. [Chapter 6](#) addresses the combined optimal procurement contract selection and pricing problems. Different price-response models, linear and nonlinear, are considered.

### 7.1 Hybrid Bilevel-Lagrangian Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants

In [Chapter 2](#), we developed a framework for the integration of planning and scheduling in the operation of a network of multiproduct batch plants. Each plant contains single-stage processing units in parallel and the products are classified into groups or families. The parameters of the deterministic optimization models comprise batch sizes and fixed processing times, sequence-dependent changeover times and costs, transportation and inventory costs, and demand forecasts over a time horizon consisting of several time periods. The main goals include determining production amounts in each plant, as well as the allocation and sequencing of batches in each reactor.

The planning model not only incorporates production and capacity con-

straints, but also estimates the sequencing of groups of products through Traveling Salesman Problem (TSP) constraints. The unit-specific general precedence (USGP) scheduling formulation employs continuous time representation. Both models are mixed-integer linear programs (MILPs).

We proposed two important enhancements to the USGP formulation that make it more flexible and realistic, but at the expense of additional constraints and discrete variables. The first extension is treating the number of batches as a decision variable. In batch scheduling terminology, the calculation of the number of batches required to fulfill the orders is called the *batching problem*. Typical scheduling formulations separate the batching problem from the scheduling/sequencing problem, which may result in suboptimal solutions or even infeasible in reality. Simultaneously solving the batching and scheduling problems gives rise to a product of integer and binary variables, which was linearized using disjunctive programming techniques as shown in equations (2.59) – (2.63). The second enhancement to the original USGP formulation is the modeling of sequence-dependent changeovers *across* time periods (see [Subsection 2.3.5](#)). In order to allow changeovers between batches of two products to occur across the boundary separating two time periods, binary variables were added to detect the first and last products produced in consecutive periods.

The integration of planning and scheduling was performed via two decomposition methods: Bilevel and Lagrangean. For small to medium problems (with up to 95,000 continuous variables, 128,000 discrete variables, and 437,000 constraints), it was observed that Bilevel Decomposition (BD) represents an attractive and rigorous decomposition strategy that allowed the integration of planning and scheduling of Examples 1 and 2 within 5 seconds. However, when attempting to solve Example 3, a real-world problem (with nearly 10,000,000 variables, two thirds are discrete, and 23,000,000 constraints), the planning level became computationally expensive largely due to the presence of the TSP constraints. We employed Lagrangean Decomposition (LD) at the planning level by decomposing it in subproblems for each time period. The Lagrange multipliers were updated via the Subgradient Method (SM). This problem was solved in around 4,000 seconds (wall time).

The SM is a nonsmooth optimization approach that is computationally inexpensive (the multipliers are updated via a single formula). From an implementation point of view, this method presents two major issues: (1) the step size in the updating formula may significantly affect the convergence rate or even the success of the method (as has been our experience), and (2) an upper bounding problem (if minimizing) must be derived. The first point is usually dealt with by trial-and-error, and the second point is typically a “heuristic” approach (e.g., solving the original problem in a reduced space by fixing certain variables).

However, one positive characteristic of LD is that its subproblems are independent of each other and can be solved in parallel. We effectively explored that in Example 3 by using the grid computing features of the modeling platform (GAMS). The 12 subproblems (one per time period) were solved in par-

allel on a server with twelve cores.

## 7.2 Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching

In [Chapter 3](#), we proposed important enhancements to the previously reported moment matching method for scenario generation. We called it the distribution matching method, which is an optimization-based model that can be used to calculate the probabilities and values of outcomes in a scenario tree, whether two- or multi-stage. In addition to matching statistical properties, such as (co-)moments, it also matches marginal (empirical) cumulative distribution function ((E)CDF) information.

The main novelty of the work is two-fold: (1) quantitative recognition of the conditions for well-posedness of the Moment Matching Problem (MMP) (see equation (3.7)), and (2) addition of easy-to-compute target information to be matched in the proposed Distribution Matching Problem (DMP). With regards to the first point, we demonstrated that for one uncertain parameter and four moments (targets), we have that the number of scenarios has to be at most equal to 2.5 to yield a well-posed MMP vs. 5 for the DMP (equation (3.15)). The additional information matched by the DMP model mitigates the under-specification present in the MMP formulations and, consequently, allows the modeler to specify more outcomes without leading to under-specified or ill-posed problems. For the second point, the additional targets come from points of the marginal distribution of each uncertain parameter. The proposed framework is general in that it allows for either assumed models for the marginal distributions (e.g., normal), or completely data-driven as long as they are smooth estimators (e.g., generalized logistic function or kernel smoothing). We note that by increasing the order of moments matched (e.g., higher than fourth-order) in the MMP could potentially and numerically reduce the under-specification issue; however, this would be at the cost of increasing the nonlinearity and nonconvexity of the NLP model, and the need for a vast amount of data to be available in order to obtain meaningful estimates of the high-order moments.

However, the disadvantage of matching marginal distributions is that any dependency structure among the uncertain parameters is not captured. In order to overcome to some extent that limitation, the covariance between two uncertain parameters is matched. Covariance is a measure of the *linear* dependence of two random variables. A more general framework to model dependence between distributions is through copulas as discussed in [Kaut \(2013\)](#), but it comes at additional costs, such as obtaining target copulas from data (empirical vs. parametric given the availability of data) and MIP-based heuristic to generate scenarios.

In [Section 3.4](#), we described two approaches for the generation of multi-



stage scenario tree generation: *NLP Approach* and *LP Approach*, the latter being more restrictive since the outcomes are fixed. Both show a powerful synergy between the optimization-based DMP and time series forecasting/simulation. The former approach is more challenging to be implemented than the latter, but on the other hand it offers more flexibility since both node values and branch probabilities are decision variables. The difficulty in implementing the *NLP Approach* is due to the alternating sequence of steps between forecasting and solving the DMP for each non-leaf node of the tree. In Example 2 (Subsection 3.4.3), this approach was carried out without automating the alternating steps. Complete automation of *NLP Approach* may be facilitated if the software used performs both functions, such as Enterprise Guide and Forecast Studio/Server components in the SAS® suite (SAS Institute, 2015). An alternative, but not necessarily easy way to fully automate the generation of the multi-stage tree, would be to communicate between a forecasting software and an optimization modeling system/solver.

Even if the automation of the *NLP Approach* is successfully accomplished, and despite its flexibility, this approach also has one potential shortcoming. The calculation of node values and branch probabilities is performed *sequentially* (i.e., node by node) instead of *simultaneously* for all non-leaf nodes of the tree. In contrast, in the *LP Approach*, all node values are generated via time series simulation, thus any potential dependence across multiple stages is preserved, and then all probabilities are calculated simultaneously. The advantage of a sequential approach is that the NLP problems are smaller, thus there is a trade-off between optimization complexity and preserving overall dependence in the tree.

The examples solved involved trees with up to 30 nodes from a production planning problem. The CPU times required were quite small.

### 7.3 Data-Driven Individual and Joint Chance-Constrained Optimization via Kernel Smoothing

In Chapter 4, we proposed the use of a nonparametric statistical technique, namely kernel smoothing (KS), to reformulate individual and joint data-driven chance constraints (CCs) into their respective algebraic forms. KS is used to estimate the true, but unknown distributions given data for the uncertain parameters. The modeling of data-driven CCs relies on a confidence set, which is constructed so that it contains both the true distribution and its estimate. We proposed using standard errors of the smoothing process in the calculation of the size of the confidence set.

As with any nonparametric technique, the accuracy of the kernel smoother depends on the availability of a significant amount of data. A general rule-of-thumb for estimating univariate (marginal) distributions is that the sample size should be at least around 100. This number can grow much bigger if

estimating joint (multivariate) distributions. The effect of the sample size on approximating the true distributions is illustrated in the motivating example (Subsection 4.6.1), where the solution (objective function value and decision variables) of the optimization models with reformulated data-driven CCs approaches the solution to the respective problems with exact distributions as shown in Figures 4.10 and 4.11. Moreover, we showed that even if the original CCs are linear, a joint chance-constrained reformulation yields nonlinear models, which can be significantly more challenging to be solved. In order to overcome this additional complexity, we developed an algorithm that uses the solution of the corresponding individual chance-constrained model to initialize the original problem with joint CCs. The proposed algorithm was used to solve a production planning problem with more than 25 production facilities and several products, which gives rise to a large-scale nonlinear programming problem (around 192,000 constraints and 283,000 variables) and achieved speedups ranging from 2 to 6 times depending on the confidence level. For instance, at 60% confidence level, applying the proposed initialization approach to the original nonlinear problem with JCC reduced the solution time from 4,000 to 750 seconds.

Even though a formal proof was not provided, it is intuitive that the proposed kernel-based reformulation methodology results in an inner approximation of the feasible region that would result from using the true distributions (which are usually unknown in practice). In the limit as  $n \rightarrow \infty$  (i.e., infinite number of data points), the kernel estimator asymptotically converges to the true density/distribution function, provided that the bandwidth is appropriately chosen. The kernel-based reformulation contains a finite number of points (smaller feasible region), which is an approximation of the true continuous function (infinite number of points).

A limitation of the approach is that it requires CCs with right-hand side (RHS) uncertainty only. Even though they are very common in different applications ranging from engineering to finance, there are some typical situations where the random variables are involved in nonlinear terms with the decision variables of the model. For instance, the linear CC

$$\mathbb{P}\{\tilde{a}^\top x \leq c\} \geq 1 - \alpha$$

where  $\tilde{a}$  is a random vector,  $x$  is a vector of decision variables,  $c$  is a deterministic parameter, and  $\alpha$  is the risk level, cannot be directly reformulated to a respective algebraic form using the proposed methodology due to the product of uncertain parameters and decision variables.

## 7.4 Data-Driven Simulation and Optimization Approaches to Incorporate Production Variability in Sales and Operations Planning

In [Chapter 5](#), we proposed a data-driven method to model production variability in the operations planning stage of the Sales & Operations Planning (S&OP) process of a manufacturing company. The method uses quantile regression to generate samples of the historical deviation ( $\Delta$ ) between planned and actual production rates conditional on a given plan. The samples correspond to scenarios that quantify uncertainty in two proposed optimization-based frameworks: Simulation-Optimization (Sim-Opt) and Bi-Objective Optimization (Bi-Opt).

Both Sim-Opt and Bi-Opt frameworks have advantages and disadvantages. Perhaps the major advantage of the Sim-Opt approach is the fact that it can easily accommodate any statistical relationship between  $\Delta$  conditional on planned values. The reason for this is that the generation of samples based on given production targets (plans) is performed after the new targets are proposed by the derivative-free optimizer, but before the execution of the “simulator” (which can be a production planning model as discussed in the chapter). Conversely, the Sim-Opt framework may become computationally intensive as the number of scenarios and decision variables (production targets) increase. The Bi-Opt approach offers interesting advantages, such as simultaneous calculation of production plans and minimization of risk, and explicit treatment of possible constraint violations. However, its main shortcoming is that it may be difficult (or even impossible) to explicitly define a functional relationship for  $\Delta$  conditional on production targets within the optimization model.

The proposed modeling framework only considers the operations planning stage of the S&OP process as shown in [Figure 5.1](#). However, the sales planning stage is also affected by uncertainty, for example, from demand forecasting. A more comprehensive model would account for both stages, and also feature a practical metric for performance evaluation at the end of the S&OP process cycle.

The applicability of the proposed approach was illustrated with a motivating example and an industrial case study. The motivating example consisted of a small process network with a feedstock plant that serves 6 downstream plants. The downstream plants are sorted in reverse order of reliability and margin, i.e., the most reliable plant is also the lowest-margin one, and the less reliable plant is the highest-margin one. The motivation behind this was to show that depending on the maximum desired level of risk, the optimization model decides to allocate more of the feedstock chemical to more or less reliable plants. In other words, the optimization model considers not only the individual margin of the plants, but also their reliability in order to obtain a

solution with maximum *expected* profit. The industrial case study consisted of a chemical site with more than 12 highly interconnected plants and several products. We proposed four approaches to account for a possible strong dependence in the historical data of production variability of plants that are directly connected: parametric regression, nonparametric regression, sampling from estimated joint distribution, and bootstrap sampling.

The computational results show that the solution times were not severely affected by the problem size (around 5 seconds for the motivating example and less than 20 seconds for the industrial case study). However, the wall time was more significantly affected (up to 5 times the solution time), which was largely due to the time spent loading the optimization problems into memory.

## 7.5 Optimal Procurement Contract Selection with Price Optimization under Uncertainty for Process Networks

In [Chapter 6](#), we proposed a stochastic optimization modeling framework that includes both optimal procurement contract selection and selling price optimization for process networks. The contract selection, which considers three types of contracts (discount, bulk discount, and fixed duration), is modeled via disjunctive programming techniques (and reformulated as mixed-integer linear constraints), and the pricing is quantified with regression models. The proposed approach is manufacturer-centric in the sense that it is the manufacturer’s decision to sign or not contracts with suppliers in order to hedge against spot market supply uncertainty, and the manufacturer can set selling prices for its customers. The possible selection of sales contracts is considered to be the customer’s decision, and was not addressed in this work.

The supply uncertainty is given by the spot market price of raw materials, even though raw material availability could also be considered as an uncertain parameter. Scenarios of spot market prices are generated by simulating autoregressive time series models, which are assumed to quantify the evolution of raw material prices with time. With regards to price optimization, the formulation of three price-response models (linear, constant-elasticity, and logit) that are typically encountered in the literature was discussed, even though we argued that general regression models can be used. The demand uncertainty is represented by samples from the distribution of regression residuals. We showed that it may be more advantageous to use demand-response models (DRMs), i.e., price as a function of demand, and we provided conditions on the parameters of the logit model that yield a concave revenue term in the objective function (in maximization problems).

The two-stage stochastic programming model has the following structure: the binary variables for contract selection and the sales variable are first-stage variables, and the remaining variables (production rates, inter-site transfers, and inventory levels) are second-stage variables. Production planning models

are typically multi-period in nature, and a multi-stage stochastic programming formulation may be more realistic. For example, in a three-stage formulation, the first-stage variables may be the same as the ones used in the proposed approach, the second-stage variables may include all remaining variables for the first-period (or a cluster of the first periods), and the third-stage variables may include all remaining variables for the remaining periods (or cluster of final periods). This modeling approach could be extended to more than three stages.

From both numerical examples, it was clear that more complex DRMs yielded more computationally intensive MINLP models. In particular, in Example 2 that dealt with the production planning of an integrated petrochemical site with 38 plants and 28 chemicals ([Subsection 6.5.2](#)), it took around 4 hours for the outer approximation-based solver to converge for the deterministic model using the logit demand-response model with 400 binary variables, 1,349 continuous variables, and 2,383 constraints. However, the proposed reformulation, which transfers the nonlinear terms in the objective function to the set of constraints, proved to significantly decrease the solution time. This was primarily due to being able to provide a tighter approximation with several terms of the objective function in the form of several constraints. In addition, different amounts of sales (assumed to be equal to the demand) were observed between deterministic and stochastic solutions and when using different DRMs. In order to ensure that customer demand is satisfied, bounds on sales or contract information setting ranges on demand could be considered in future work.

## 7.6 Contributions of the Thesis

The main contributions of this thesis are the following:

1. A continuous-time, unit-specific general precedence (USGP) scheduling formulation that considers the batching problem (i.e., it simultaneously calculates the number of batches with the scheduling decisions) and allows for sequence-dependent changeovers within and across time periods. These two additional features yield a more realistic and general scheduling model without significantly compromising the solution time.
2. A systematic, data-driven scenario generation method (Distribution Matching Problem or DMP) via statistical property matching, which remedies the under-specification characteristics of a previously proposed method. In addition to marginal (co-)moments, the DMP also includes information from marginal (empirical) cumulative distribution function, thus improving the balance between number of variables and data points (targets).
3. Two methods for the generation of multi-stage scenario trees that combine time series forecasting with the solution of the proposed distribution

matching method. The *NLP Approach* sequentially computes the node values and branch probabilities for each non-leaf node. The *LP Approach* computes all branch probabilities simultaneously for a tree with fixed node values obtained from time series simulation.

4. A method using kernel smoothing to model data-driven individual and joint chance constraints with right-hand side uncertainty. The approach is based on the construction of a confidence set that contains the estimated distributions and is considered to also contain the true, but unknown distributions. In addition, the approach uses the standard errors estimated in the smoothing process to calculate the size of the confidence set, which is necessary to complete the reformulation of the probabilistic constraints into their respective algebraic ones.
5. A framework to model production variability and incorporate it in the operations planning stage of the Sales & Operations Planning (S&OP) process of a manufacturing company. The method consists of defining  $\Delta$  as the deviation between historical planned and actual production rates, and using quantile regression to obtain the quantiles of the distribution of  $\Delta$  conditional on planned values. The model of uncertainty can be used in simulation- and optimization-based approaches, along with risk measures to control the degree of conservatism of the solution.
6. A mixed-integer programming formulation that combines optimal procurement contract selection and selling price optimization in a manufacturer-centric approach. The proposed framework allows for general regression models as demand-response functions. The approach is illustrated with three different contract types previously reported in the literature, and three typical demand-response models (linear, constant-elasticity, and logit). Conditions for the concavity of the logit model (generally regarded as the most realistic of the three models) are derived.

## 7.7 Recommendations for Future Work

1. In the hybrid decomposition scheme implemented in [Chapter 2](#), the Subgradient Method (SM) was used to update the Lagrange multipliers within the Lagrangean Decomposition (LD) framework. As mentioned in [Section 7.1](#), obtaining convergence with the SM proved to be significantly challenging due to its two major issues: choice of step size and erratic behavior of upper bound solution (in minimization problems). Other methods for updating multipliers require additional optimization problems to be solved, which may be prohibitive for large-scale problems. An alternative to LD is based on the method of multipliers and it is called Augmented Lagrangean Decomposition (ALD) ([Rosa & Ruszczyński, 1996](#); [Ruszczyński, 1997](#)). ALD is also an iterative, dual decomposition method (i.e., on the space of the



dual variables), which does not suffer from the same aforementioned issues as LD with SM and still yield independent subproblems that can be solved in parallel. In addition to updating the multipliers, the ALD method also updates the decision variables at each iteration. The main difference between the two approaches is that with ALD one optimization problem is solved instead of two as in LD (lower- and upper-bounding problems). The additional cost of ALD is the presence of quadratic terms in the objective function, which turns a (MI)LP model into a (MI)QP problem. However, MIP solvers, such as Gurobi and CPLEX, are becoming very efficient at solving (MI)QP problems.

2. The Distribution Matching Problem (DMP) proposed in [Chapter 3](#) is an optimization-based approach for generating scenario trees by matching marginal statistical information, such as moments and cumulative distribution function (CDF). Any dependence between random variables (r.vs.) is simply captured by covariance information. However, as mentioned in [Section 7.2](#), the DMP does not account for more general dependence among r.vs. There are some research areas in statistics, machine learning, and digital signal processing that could offer different approaches for the same problem, and that directly use the joint distribution (estimated or not), which contains the complete dependence (or lack thereof) structure among r.vs. Scenario trees are a form of discretized or *quantized* representation of distributions, i.e., they approximate (usually continuous) distributions by considering a finite number of outcomes (scenarios) with given probabilities (dimensionality reduction).

- *Quantization and Information Theory.* In general terms, quantization is the process of mapping a large set of input values (e.g., continuous distribution) to a countable (discrete) smaller set. A typical application includes data compression in communication. If the object (input) to be quantized is a univariate function, then the method is called Scalar Quantization (SQ), and if the input is a multivariate function, it is named Vector Quantization (VQ). Briefly, the problem is to find the optimal set of countable values (called *reproduction points* or *code points*) that represents the original input. The formulation involves concepts of Rate-Distortion Theory ([Cover & Thomas, 2006](#)), in which a distortion function is used to measure the cost of representing the original input in a reduced and discrete space. The modeling of the theory uses the *mutual information*, which is a measure of the mutual dependence between two r.vs. [Colin et al. \(2013\)](#) propose an optimal quantization method of the support of a multivariate distribution using  $\phi$ -mutual information (based on the notion of  $\phi$ -divergence, which is discussed in [Chapter 4](#)).
- *Sampling and Clustering.* Another idea related to quantization is the use of cluster analysis ([Everitt et al., 2011](#)) in data mining. Among

other methods and algorithms,  $k$ -means clustering is a method of VQ that aims to partition  $n$  observations into  $k$  clusters. The observations in this case could be samples generated from the (estimated) joint distribution. The center of each cluster would represent an outcome of the scenario tree, and the collection of centers of clusters would provide a discrete approximation to the continuous joint distribution. An initial algorithm that can be applied to both cases of univariate or multivariate joint distributions is as follows:

- (i) Generate  $n$  samples from the (estimated) joint distribution,  $x_1, \dots, x_j, \dots, x_n$ ;
- (ii) Calculate the density for each sample using the (estimated) continuous joint distribution,  $p(x_1), \dots, p(x_j), \dots, p(x_n)$ ;
- (iii) Cluster the samples into  $k$  groups or clusters;
- (iv) Set the probability of cluster  $i$ , for  $i = 1, \dots, k$ , to be proportional to the sum of  $p(x_j)$ , for every sample  $j$  put in cluster  $i$ . For example,

$$\text{Probability of scenario (cluster) } i = \frac{\sum_{j: j \text{ in cluster } i} p(x_j)}{\sum_j p(x_j)}$$

3. The kernel-based reformulation methodology proposed in [Chapter 4](#) requires chance constraints (CCs) with right-hand side uncertainty only. It is likely that an extension of the approach that accounts for more general CCs will be mathematically challenging. In addition, it will likely not preserve linearity of the constraints as is the case of the reformulation of individual CCs. One research direction could be based on the use of transformations of r.v.s., i.e, the *Method of Transformations* ([Casella & Berger, 2002](#)). Consider the following scalar example. Let the original CC be  $\mathbb{P}_{\tilde{a}} \{\tilde{a}x \leq c\} \geq 1 - \alpha$ , where  $\tilde{a}$  is a continuous r.v.,  $x$  is a continuous decision variable (usually, non-negative),  $c$  is a deterministic parameter, and  $\alpha$  is the risk level. Suppose that  $\tilde{a}$  has PDF  $f(a)$  and CDF  $F(a)$ . In this case, we can apply a *linear transformation* by treating  $x$  as a “coefficient” of the r.v.  $\tilde{a}$ . Introduce the r.v.  $\tilde{b} = \tilde{a}x$  with PDF  $g(b)$  and CDF  $G(b)$ . Therefore, the CC becomes  $\mathbb{P}_{\tilde{b}} \{\tilde{b} \leq c\} \geq 1 - \alpha$ , which seems more tractable than the original CC, since the new r.v. is not involved in nonlinear terms with the decision variable. To reformulate this new CC, we need to ultimately invoke the CDF of  $\tilde{b}$ . It can be shown that

$$G(b) = \frac{1}{|x|} x F\left(\frac{b}{x}\right)$$

The term  $xF(b/x)$  corresponds to the *perspective* of the CDF of the original r.v.,  $\tilde{a}$ . When dealing with data-driven CCs, all true distributions are replaced with their estimates, i.e.,  $f(\cdot)$  and  $F(\cdot)$  are replaced with  $\hat{f}(\cdot)$  and



$\hat{F}(\cdot)$ , respectively. Even though the original CC is linear, the reformulation in this case will not be linear. More details are given in [Appendix E](#).

4. One of the major disadvantages of the Bi-Optimization approach (Bi-Opt) proposed in [Chapter 5](#) is the difficulty (or impossibility) of expressing the deviation from plan conditional on production targets ( $\Delta_s$  for scenario  $s$ ). In the numerical examples,  $\Delta_s$  was constant regardless of the production targets proposed by the optimization model. This may not be a reasonable approximation, since the distribution and range of  $\Delta_s$  values can be very different depending on planned production rates. In order to make the formulation more flexible, the following modeling strategy could be investigated. For each plant with production variability data, do the following steps:

- (i) Obtain Plan vs. Actual historical production rate data, and calculate  $\Delta = \text{Plan} - \text{Actual}$ ;
- (ii) Discretize the range of Plan values into  $n = 1, \dots, N$  intervals, and select one value in each interval (e.g., median) to be the *representation point* of each interval;
- (iii) For each interval, perform quantile regression for  $\Delta$  conditional on Plan being equal to the interval's representation point (i.e., generate the samples or scenarios), and store the output in a parameter  $\Delta_{n,s}$ ;
- (iv) We need to ensure that if the production targets fall into the interval of Plan values, then the correct deviation values are used. The following disjunction can be used to model this condition.

$$\bigvee \left[ \begin{array}{c} Y_n \\ \text{PT}_n^L \leq PT \leq \text{PT}_n^U \\ PR_s = PT - \Delta_{n,s} \end{array} \right] \quad \forall n = 1, \dots, N, s \in S$$

where  $Y_n$  is a Boolean variable that is **True** if  $PT$  belongs to the interval  $[\text{PT}_n^L, \text{PT}_n^U]$ , and **False** otherwise.

However, the increased flexibility suggested above comes at some possible computational expense due to additional binary variables needed to reformulate the disjunctions as mixed-integer linear constraints.

5. In the proposed approach in [Chapter 6](#), the decision to sign or not sales contracts lies with customers, not with the manufacturer. An extension to design sales contracts with customers could be as follows. The manufacturer sets selling prices by incorporating demand-response models (DRMs) in the production planning model; those prices can be used to create the structure of sales contracts (e.g., discount quantity thresholds and price levels); an optimization problem for each customer is formulated that decides which contracts, if any, to sign in order to maximize/minimize an objective (e.g., minimize total costs for the customer from purchasing finished products

from the manufacturer). This will lead to a *bilevel programming* formulation (Shimizu, Ishizuka, & Bard, 1997; Bard, 1998; Labbé & Violin, 2013) in which the upper level problem (ULP) is the model proposed in this work, and the lower level problem (LLP) consists of optimization models for each customer. The objectives of the ULP are to select procurement contracts to sign or not, and to set design sales contracts (by setting selling prices), whereas the objective of the LLP is to select or not sales contracts. A further extension would even consider a separate optimization problem for the suppliers that would design procurement contracts to be selected or not by the manufacturer, thus resulting in a trilevel programming formulation.

6. The use of nonlinear (even if concave) DRMs may considerably increase the solution time as shown in the numerical examples in Chapter 6. If the nonlinear DRM is a univariate function (i.e., if there is only one regressor, thus the model's objective is a *separable function*), then an approximate solution approach could be based on piece-wise linear approximations of the nonlinear DRMs (Taha, 2010). This would require the introduction of additional constraints and variables (Special Ordered Sets of type 2 or SOS2), but the complexity would be expected to decrease since the concave MINLP model would be approximated by an MILP model. In case more general, multivariate nonlinear DRMs are used, then we suggest a decomposition approach that alternates between a master problem (MIQP) and a restricted slave problem (NLP). The master problem uses a linear DRM (with one or more regressors) as an approximation to the nonlinear DRM, and the slave problem is the original optimization model with the nonlinear DRM, but with all discrete variables fixed to the solution of the master problem. At the start of a new iteration, sales variables from the solution to the slave problem are fixed in master problem. A convergence criterion could be a small relative error between the objective function value of the slave problem at the current and previous iterations. Again, it would be expected that solving MIQP and NLP problems would be computationally less intensive than directly solving an MINLP problem.

The data-driven approaches to Enterprise-Wide Optimization proposed in this thesis fit with the recent trend towards Business Analytics (Davenport & Harris, 2007; Stubbs, 2011). The decision-making model (e.g., production planning) is enhanced by a model of uncertainty and variability, which is systematically obtained using advanced data analytics techniques. In other words, the majority of the contributions of this thesis combine predictive analytics (regression, forecasting, and simulation) with prescriptive decision making (deterministic and stochastic optimization). Moreover, the proposed data-driven approaches were developed to support the use of nonparametric statistical techniques for both continuous and categorical (discrete) data. This makes the proposed approaches statistically more *robust* and widely applicable to different problems, since they do not require strong assumptions about the probability distributions of the data generating process.

# Bibliography

- Abel, O., and Marquardt, W. 2000. Scenario-Integrated Modeling and Optimization of Dynamic Systems. *AIChE Journal*. 46(4):803–823.
- Abramowitz, M., and Stegun, I. A. 1965. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Dover Publications.
- Achterberg, T. 2009. SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computation* 1(1):1–41.
- Agresti, A., and Coull, B. A. 1998. Approximate Is Better than “Exact” for Interval Estimation of Binomial Proportions. *The American Statistician*. 52(2):119–126.
- Arellano-Garcia, H., and Wozny, G. 2009. Chance Constrained Optimization of Process Systems Under Uncertainty: I. Strict Monotonicity. *Computers & Chemical Engineering*. 33(10):1568–1583.
- Balas, E. 1985. Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. *SIAM Journal on Algebraic Discrete Methods*. 6(3):466–486.
- Bansal, M.; Karimi, I. A.; and Srinivasan, R. 2007. Optimal Contract Selection for the Global Supply and Distribution of Raw Materials. *Industrial & Engineering Chemistry Research*. 46(20):642–660.
- Barahona, F., and Anbil, R. 2000. The Volume Algorithm: Producing Primal Solutions with a Subgradient Method. *Mathematical Programming*. 87(3):385–399.
- Bard, J. F. 1998. *Practical Bilevel Optimization: Algorithms and Applications*, volume 30 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers. Dordrecht, The Netherlands.
- Bartlett, R. 2013. *A Practitioner’s Guide to Business Analytics: Using Data Analysis Tools to Improve Your Organization’s Decision Making and Strategy*. The McGraw-Hill Companies.
- Bauer, H. 2001. *Measure and Integration Theory*, volume 26 of *Studies in Mathematics*. Walter de Gruyter GmbH & Co. Berlin, Germany.

- Bayraksan, G., and Morton, D. P. 2006. Assessing Solution Quality in Stochastic Programs. *Mathematical Programming*. 108(2-3):495–514.
- Becker, M., and Klößner, S. 2013. *PearsonDS: Pearson Distribution System*. R package version 0.95.
- Belotti, P.; Lee, J.; Liberti, L.; Margot, F.; and Wächter, A. 2009. Branching and Bounds Tightening Techniques for Non-convex MINLP. *Optimization Methods and Software* 24(4-5):597–634.
- Ben-Tal, A.; den Hertog, D.; De Waegenaere, A.; Melenberg, B.; and Rennen, G. 2011. Robust Solutions of Optimization Problems Affected by Uncertain Probabilities. *Optimization Online*. Retrieved from [http://www.optimization-online.org/DB\\_FILE/2011/06/3076.pdf](http://www.optimization-online.org/DB_FILE/2011/06/3076.pdf) on Feb 24, 2014.
- Ben-Tal, A.; Ghaoui, L. E.; and Nemirovski, A. 2009. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton, NJ. USA.
- Beraldi, P.; De Simone, F.; and Violi, A. 2010. Generating Scenario Trees: A Parallel Integrated Simulation-Optimization Approach. *Journal of Computational and Applied Mathematics*. 233(9):2322–2331.
- Bienstock, D.; Chertkov, M.; and Harnett, S. 2012. Chance Constrained Optimal Power Flow: Risk-Aware Network Control under Uncertainty. *Computing Research Repository (CoRR)* abs/1209.5779. Retrieved from <http://arxiv.org/abs/1209.5779> on Feb 24, 2014.
- Birge, J. R., and Louveaux, F. 2011. *Introduction to Stochastic Programming*. Springer Science+Business Media, LLC., second edition. New York, NY. USA.
- Bodea, T., and Ferguson, M. 2014. *Segmentation, Revenue Management and Pricing Analytics*. Routledge – Taylor & Francis. New York, NY. USA.
- Bonferroni, C. E. 1936. Teoria Statistica Delle Classi e Calcolo Delle Probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*. 8:1–62.
- Box, G. E. P.; Jenkins, G. M.; and Reinsel, G. C. 2008. *Time Series Analysis: Forecasting and Control*. Holden-Day, Inc., fourth edition. Hoboken, NJ. USA.
- Brockwell, P. J., and Davis, R. A. 2002. *Introduction to Time Series and Forecasting*. Springer-Verlag New York, Inc., second edition. New York, NY. USA.
- Brooke, A.; Kendrick, D.; and Meeraus, A. 2015. *General Algebraic Modeling System (GAMS)*. <http://www.gams.com/>.

- Cai, X.; Teo, K.-L.; Yang, X.; and Zhou, X. Y. 2000. Portfolio Optimization Under a Minimax Rule. *Management Science*. 46(7):957–972.
- Calafiore, G. C., and Campi, M. C. 2006. The Scenario Approach to Robust Control Design. *IEEE Transactions on Automatic Control*. 51(5):742–753.
- Calfa, B. A.; Agarwal, A.; Grossmann, I. E.; and Wassick, J. M. 2014. Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching. *Computers & Chemical Engineering*. 68(1):7–23.
- Casella, G., and Berger, R. L. 2002. *Statistical Inference*. Duxbury. The Wadsworth Group. Thomson Learning, Inc., second edition. Pacific Grove, CA. USA.
- Cerdá, J.; Henning, G. P.; and Grossmann, I. E. 1997. A Mixed-Integer Linear Programming Model for Short-Term Scheduling of Single-Stage Multiproduct Batch Plants with Parallel Lines. *Industrial & Engineering Chemistry Research*. 36(5):1695–1707.
- Charnes, A., and Cooper, W. W. 1959. Chance-Constrained Programming. *Management Science*. 6(1):73–79.
- Charnes, A., and Kirby, M. 1966. Optimal Decision Rules for the E-Model of Chance-Constrained Programming. *Cahiers du Centre d'Études de Recherche Opérationnelle*. 8:5–44.
- Charnes, A., and Kirby, M. J. L. 1967. Some Special P-Models in Chance-Constrained Programming. *Management Science*. 14(3):183–195.
- Chen, R., and Tsay, R. S. 1993. Nonlinear Additive ARX Models. *Journal of the American Statistical Association*. 88(423):955–967.
- Chen, Y.; Adams II, T. A.; and Barton, P. I. 2011. Optimal Design and Operation of Flexible Energy Polygeneration Systems. *Industrial & Engineering Chemistry Research*. 50(8):4553–4566.
- Chiralaksanakul, A., and Morton, D. P. 2004. Assessing Policy Quality in Multi-stage Stochastic Programming. *Stochastic Programming E-Print Series*. (12):1–36. Retrieved from <http://edoc.hu-berlin.de/docviews/abstract.php?id=26770>.
- Chong, C.-Y., and Cheng, D. C. 1975. Multistage Pricing under Uncertain Demand. *Annals of Economic and Social Measurement*. 4(2):311–323.
- Colin, B.; Dubeau, F.; Khreibani, H.; and de Tibeiro, J. 2013. Optimal Quantization of the Support of a Continuous Multivariate Distribution based on Mutual Information. *Journal of Classification*. 30(3):453–473.

- Colvin, M., and Maravelias, C. T. 2009. Scheduling of Testing Tasks and Resource Planning in New Product Development using Stochastic Programming. *Computers & Chemical Engineering*. 33(5):964–976.
- Conejo, A. J.; Castillo, E.; Mínguez, R.; and García-Bertrand, R. 2006. *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications*. Springer Science+Business Media. New York, NY. USA.
- Conn, A. R.; Scheinberg, K.; and Vicente, L. N. 2009. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics (SIAM) and the Mathematical Programming Society (MPS). Philadelphia, PA. USA.
- Corless, R.; Gonnet, G.; Hare, D.; Jeffrey, D.; and Knuth, D. 1996. On the Lambert W Function. *Advances in Computational Mathematics*. 5(1):329–359.
- Cover, T. M., and Thomas, J. A. 2006. *Elements of Information Theory*. John Wiley & Sons, Inc., second edition. Hoboken, NJ. USA.
- Custódio, A.; Emmerich, M.; and Madeira, J. F. A. 2012. Recent Developments in Derivative-Free Multiobjective Optimisation. *Computational Technology Reviews*. 5(1):1–30.
- Dantzig, G. B. 1955. Linear Programming under Uncertainty. *Management Science*. 1(3-4):197–206.
- Davenport, T. H., and Harris, J. G. 2007. *Competing on Analytics: The New Science of Winning*. Harvard Business School Press. Boston, MA. USA.
- Davison, A. C., and Hinkley, D. V. 1997. *Bootstrap Methods and Their Application*. Cambridge Series on Statistical and Probabilistic Mathematics. Cambridge University Press. Cambridge, UK.
- Deniz, E., and Luxhøj, J. T. 2011. A Scenario Generation Method with Heteroskedasticity and Moment Matching. *The Engineering Economist*. 56(3):231–253.
- Duran, M. A., and Grossmann, I. E. 1986. An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Mathematical Programming*. 36(3):307–339.
- Efron, B. 1979. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*. 7(1):1–26.
- Eisner, M. J.; Kaplan, R. S.; and Soden, J. V. 1971. Admissible Decision Rules for the E-Model of Chance-Constrained Programming. *Management Science*. 17(5):337–353.

- Erdirik-Dogan, M., and Grossmann, I. E. 2007. Planning Models for Parallel Batch Reactors with Sequence-Dependent Changeovers. *AIChE Journal*. 53(9):2284–2300.
- Erdirik-Dogan, M., and Grossmann, I. E. 2008. Slot-Based Formulation for the Short-Term Scheduling of Multistage, Multiproduct Batch Plants with Sequence-Dependent Changeovers. *Industrial & Engineering Chemistry Research*. 47(4):1159–1183.
- Evans, M.; Hastings, N.; and Peacock, B. 2000. *Statistical Distributions*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., third edition. New York, NY. USA.
- Everitt, B. S.; Landau, S.; Leese, M.; and Stahl, D. 2011. *Cluster Analysis*. Probability and Statistics. John Wiley & Sons, Inc., fifth edition. West Sussex, UK.
- Fan, J., and Yao, Q. 2002a. *Nonlinear Time Series: Nonparametric and Parametric Methods*. Springer Series in Statistics. New York, NY. USA.
- Fan, J., and Yao, Q. 2002b. *Nonlinear Time Series: Nonparametric and Parametric Methods*. Springer Series in Statistics.
- Feng, Y., and Ryan, S. M. 2013. Scenario Construction and Reduction Applied to Stochastic Power Generation Expansion Planning. *Computers & Operations Research* 40(1):9–23.
- Fiorio, C. V. 2004. Confidence Intervals for Kernel Density Estimation. *The Stata Journal*. 4(2):168–179.
- Floudas, C. A., and Lin, X. 2004. Continuous-Time versus Discrete-Time Approaches for Scheduling of Chemical Processes: A Review. *Computers & Chemical Engineering*. 28(11):2109–2129.
- Forbes, C.; Evans, M.; Hastings, N.; and Peacock, B. 2011. *Statistical Distributions*. John Wiley & Sons, Inc., fourth edition. Hoboken, NJ. USA.
- Georghiou, A.; Wiesemann, W.; and Kuhn, D. 2011. The Decision Rule Approach to Optimisation under Uncertainty: Methodology and Applications in Operations Management. *Optimization Online*. Retrieved from [http://www.optimization-online.org/DB\\_FILE/2011/12/3290.pdf](http://www.optimization-online.org/DB_FILE/2011/12/3290.pdf) on Jan 17, 2015.
- Gjerdrum, J.; Shah, N.; and Papageorgiou, L. G. 2001. Transfer Prices for Multienterprise Supply Chain Optimization. *Industrial & Engineering Chemistry Research*. 40(7):1650–1660.

- Gochet, W. F., and Padberg, M. W. 1974. The Triangular E-Model of Chance-Constrained Programming with Stochastic A-Matrix. *Management Science*. 20(9):1284–1291.
- Goffin, J. L.; Haurie, A.; and Vial, J. P. 1992. Decomposition and Nondifferentiable Optimization with the Projective Algorithm. *Management Science*. 38(2):284–302.
- Goodstein, R. L. 2007. *Boolean Algebra*. Dover Books on Mathematics. Dover Publications, Inc. New York, NY. USA.
- Grimson, J. A., and Pyke, D. F. 2007. Sales and Operations Planning: An Exploratory Study and Framework. *The International Journal of Logistics Management*. 18(3):322–346.
- Grossmann, I. 2005. Enterprise-wide Optimization: A New Frontier in Process Systems Engineering. *AIChE Journal*. 51(7):1846–1857.
- Guignard, M. 2003. Lagrangean Relaxation. *Top*. 11(2):151–228.
- Guillén, G.; Espuña, A.; and Puigjaner, L. 2006. Addressing the Scheduling of Chemical Supply Chains under Demand Uncertainty. *AIChE Journal*. 52(11):3864–3881.
- Guillén-Gosálbez, G., and Grossmann, I. 2010. A Global Optimization Strategy for the Environmentally Conscious Design of Chemical Supply Chains Under Uncertainty in the Damage Assessment Model. *Computers & Chemical Engineering*. 34(1):42–58.
- Gülpınar, N.; Rustem, B.; and Settergren, R. 2004. Simulation and Optimization Approaches to Scenario Tree Generation. *Journal of Economic Dynamics and Control*. 28(7):1291–1315.
- Gupta, V., and Grossmann, I. E. 2012. Modeling and Computational Strategies for Optimal Development Planning of Offshore Oilfields under Complex Fiscal Rules. *Industrial & Engineering Chemistry Research*. 51(44):14438–14460.
- Haimes, Y. Y.; Lasdon, L. S.; and Wismer, D. A. 1971. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems, Man, and Cybernetics*. 1(3):296–297.
- Hall, P., and Horowitz, J. 2013. A Simple Bootstrap Method for Constructing Nonparametric Confidence Bands for Functions. *The Annals of Statistics*. 41(4):1693–2262.
- Harrell, F. E., and Davis, C. E. 1982. A New Distribution-Free Quantile Estimator. *Biometrika*. 69(3):635–640.



- Harrell, Jr, F. E.; with contributions from; Dupont, C.; and many others. 2014. *Hmisc: Harrell Miscellaneous*. R package version 3.14-3.
- Hayfield, T., and Racine, J. S. 2008. Nonparametric Econometrics: The np Package. *Journal of Statistical Software* 27(5).
- Held, M.; Wolfe, P.; and Crowder, H. P. 1974. Validation of Subgradient Optimization. *Mathematical Programming*. 6(1):62–88.
- Hogg, R. V.; McKean, J.; and Craig, A. T. 2012. *Introduction to Mathematical Statistics*. Pearson, seventh edition. London, UK.
- Höhn, M. I. 2010. *Relational Supply Contracts: Optimal Concessions in Return Policies for Continuous Quality Improvements*, volume 629 of *Lecture Notes in Economics and Mathematical Systems*. Springer Berlin Heidelberg. chapter 2. Literature Review on Supply Chain Contracts, 19–34.
- Høyland, K., and Wallace, S. W. 2001. Generating Scenario Trees for Multistage Decision Problems. *Management Science*. 46(2):295–307.
- Høyland, K.; Kaut, M.; and Wallace, S. W. 2003. A Heuristic for Moment-Matching Scenario Generation. *Computational Optimization and Applications*. 24(2-3):169–185.
- Hunting, M. 2012. Solving convex MINLP problems with AIMMS. An AIMMS White Paper. Technical report, Paragon Decision Technology BV. URL: <http://images.aimms.com/aimms/download/white-papers/coa-aimms-whitepaper-2012.pdf>. Accessed on February 11, 2015.
- Hwang, C.-L., and Masud, A. S. M. 1979. *Multiple Objective Decision Making, Methods and Applications: A State-of-the-Art Survey*, volume 164 of *Lecture Notes in Economics and Mathematical Systems*. Springer-Verlag. New York, NY. USA.
- Hyndman, R. J.; with contributions from; Athanasopoulos, G.; Razbash, S.; Schmidt, D.; Zhou, Z.; Khan, Y.; Bergmeir, C.; and Wang, E. 2014. *forecast: Forecasting functions for time series and linear models*. R package version 5.6.
- IHS Inc. 2013. *EViews 8.0*. IHS Inc., Irvine, California, United States. <http://www.eviews.com/>.
- Iyer, R. R., and Grossmann, I. E. 1998. A Bilevel Decomposition Algorithm for Long-Range Planning of Process Networks. *Industrial & Engineering Chemistry Research*. 37(2):474–481.
- Jairaj, P. G., and Vedula, S. 2003. Modeling Reservoir Irrigation in Uncertain Hydrologic Environment. *Journal of Irrigation and Drainage Engineering*. 129(3):164–172.

- Ji, X.; Zhu, S.; Wang, S.; and Zhang, S. 2005. A Stochastic Linear Goal Programming Approach to Multistage Portfolio Management Based on Scenario Generation via Linear Programming. *IIE Transactions*. 37(10):957–969.
- Jiang, R., and Guan, Y. 2013. Data-driven Chance Constrained Stochastic Program. *Optimization Online*. Retrieved from [http://www.optimization-online.org/DB\\_FILE/2013/09/4044.pdf](http://www.optimization-online.org/DB_FILE/2013/09/4044.pdf) on Feb 24, 2014.
- Kallrath, J. 2002. Planning and Scheduling in the Process Industry. *OR Spectrum*. 24(3):219–250.
- Kaplan, U.; Türkay, M.; Karasözen, B.; and Biegler, L. T. 2011. Optimization of Supply Chain Systems with Price Elasticity of Demand. *INFORMS Journal on Computing*. 23(4):557–568.
- Kaut, M. 2003. *Scenario Tree Generation for Stochastic Programming: Cases from Finance*. Ph.D. Dissertation, Department of Mathematical Sciences. Norwegian University of Science and Technology.
- Kaut, M. 2013. A Copula-Based Heuristic for Scenario Generation. *Computational Management Science*. 25. DOI: 10.1007/s10287-013-0184-4.
- Kelly, J. E. 1960. The Cutting-Plane Method for Solving Convex Programs. *Journal of the Society for Industrial and Applied Mathematics*. 8(4):703–712.
- Khalilpour, R., and Karimi, I. A. 2011. Modeling of Purchase and Sales Contracts in Supply Chain Optimization. *Industrial & Engineering Chemistry Research*. 50(17):10298–10312.
- Kim, J.; Realff, M. J.; and Lee, J. H. 2011. Optimal Design and Global Sensitivity Analysis of Biomass Supply Chain Networks for Biofuels under Uncertainty. *Computers & Chemical Engineering*. 35(9):1738–1751.
- King, A. J., and Wallace, S. W. 2012. *Modeling with Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer Science+Business Media New York. New York, NY. USA.
- Klemens, B. 2009. *Modeling with Data: Tools and Techniques for Scientific Computing*. Princeton University Press. Online Appendix M, pages 8-9. Princeton, NJ. USA.
- Kleywegt, A. J.; Shapiro, A.; and Homem-de-Mello, T. 2001. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal of Optimization*. 12(2):479–502.
- Koenker, R. 2005. *Quantile Regression*. Cambridge University Press. New York, NY. USA.

- Kopanos, G. M.; Laínez, M. J.; and Puigjaner, L. 2009. An Efficient Mixed-Integer Linear Programming Scheduling Framework for Addressing Sequence-Dependent Setup Issues in Batch Plants. *Industrial & Engineering Chemistry Research*. 48(13):6346–6357.
- Kopanos, G. M.; Puigjaner, L.; and Maravelias, C. T. 2011. Production Planning and Scheduling of Parallel Continuous Processes with Product Families. *Industrial & Engineering Chemistry Research*. 50(3):1369–1378.
- Koza, J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Computer Science Science and Intelligent System. A Bradford Book – The MIT Press. Cambridge, MA. USA.
- Labbé, M., and Violin, A. 2013. Bilevel Programming and Price Setting Problems. *4OR*. 11(1):1–30.
- Lane, M. 1973. Conditional Chance-Constrained Model for Reservoir Control. *Water Resources Research*. 9(4):937–948.
- Lemaréchal, C. 1974. An Algorithm for Minimizing Convex Functions. In *Proceedings IFIP '74 Congress*, 552–556.
- Li, Z., and Ierapetritou, M. G. 2011. Capacity Expansion Planning Through Augmented Lagrangian Optimization and Scenario Decomposition. *AIChE Journal*. 58(3):871–883.
- Li, Q., and Racine, J. S. 2007. *Nonparametric Econometrics: Theory and Practice*. Themes in Modern Econometrics. Princeton University Press. Princeton, NJ. USA.
- Li, J.; González, M.; and Zhu, Y. 2009. A Hybrid Simulation Optimization Method for Production Planning of Dedicated Remanufacturing. *International Journal of Production Economics*. 117(2):286–301.
- Li, Y.; Liu, Y.; and Zhu, J. 2007. Quantile Regression in Reproducing Kernel Hilbert Spaces. *Journal of the American Statistical Association*. 102(477):255–268.
- Lim, L. L.; Alpan, G.; and Penz, B. 2014. A Simulation-Optimization Approach for Managing the Sales and Operations Planning in the Automotive Industry. Technical Report 212, Laboratoire G-SCOP. ISSN: 1298-020X.
- Lima, R. M.; Grossmann, I. E.; and Jiao, Y. 2011. Long-Term Scheduling of a Single-Unit Multi-Product Continuous Process to Manufacture High Performance Glass. *Computers & Chemical Engineering* 35(3):554–574.
- Lin, C.-C., and Wu, Y.-C. 2014. Combined Pricing and Supply Chain Operations under Price-Dependent Stochastic Demand. *Applied Mathematical Modelling*. 38(5-6):1823–1837.

- Ling, R. C., and Goddard, W. E. 1988. *Orchestrating Success: Improve Control of the Business with Sales & Operations Planning*. John Wiley & Sons, Inc. New York, NY. USA.
- Liu, S.; Pinto, J. M.; and Papageorgiou, L. G. 2008. A TSP-based MILP Model for Medium-Term Planning of Single-Stage Continuous Multiproduct Plants. *Industrial & Engineering Chemistry Research*. 47(20):7733–7743.
- Lobo, M. S., and Boyd, S. 2003. Pricing and Learning with Uncertain Demand. Technical report, Duke University and Stanford University. URL: [https://web.stanford.edu/~boyd/papers/pdf/pric\\_learn\\_unc\\_dem.pdf](https://web.stanford.edu/~boyd/papers/pdf/pric_learn_unc_dem.pdf). Accessed on October 19, 2014.
- Luedtke, J., and Ahmed, S. 2008. A Sample Approximation Approach for Optimization with Probabilistic Constraints. *SIAM Journal on Optimization*. 19(2):674–699.
- Maplesoft. 2014. *Maple 18.01*. Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario. <http://www.maplesoft.com/index.aspx>.
- Maravelias, C. T., and Sung, C. 2009. Integration of Production Planning and Scheduling: Overview, Challenges and Opportunities. *Computers & Chemical Engineering* 33(12):1919–1930.
- Maravelias, C. T. 2012. General Framework and Modeling Approach Classification for Chemical Production Scheduling. *AIChE Journal* 58(6):1812–1828.
- Marsten, R. E.; Hogan, W. W.; and Blankenship, J. W. 1975. The Boxstep Method for Large-Scale Optimization. *Operations Research* 23(3):389–405.
- Mazadi, M.; Rosehart, W. D.; Zareipour, H.; Malik, O. P.; and Oloomi, M. 2013. Impact of Wind Integration on Electricity Markets: A Chance-Constrained Nash Cournot Model. *International Transactions on Electrical Energy Systems*. 23(1):83–96.
- Méndez, C. A.; Cerdá, J.; Grossmann, I. E.; Harjunkoski, I.; and Fahl, M. 2006. State-of-the-Art Review of Optimization Methods for Short-Term Scheduling of Batch Processes. *Computers & Chemical Engineering*. 30(6-7):913–946.
- Méndez, C. A.; Henning, G. P.; and Cerdá, J. 2000. Optimal Scheduling of Batch Plants Satisfying Multiple Product Orders with Different Due-Dates. *Computers & Chemical Engineering* 24(9-10):2223–2245.
- Méndez, C. A.; Henning, G. P.; and Cerdá, J. 2001. An MILP Continuous-Time Approach to Short-Term Scheduling of Resource-Constrained Multistage Flowshop Batch Facilities. *Computers & Chemical Engineering*. 25(4-6):701–711.

- Meyer, D.; Dimitriadou, E.; Hornik, K.; Weingessel, A.; and Leisch, F. 2012. *e1071: Misc Functions of the Department of Statistics (e1071)*, TU Wien. R package version 1.6-1.
- Miettinen, K. 1999. *Multiple Objective Decision Making, Methods and Applications: A State-of-the-Art Survey*, volume 12 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers. Boston, MA. USA.
- Montgomery, D. C., and Runger, G. C. 2003. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., third edition. New York, NY. USA.
- Mula, J.; Poler, R.; García-Sabater, J. P.; and Lario, F. C. 2006. Models for Production Planning under Uncertainty: A Review. *The International Journal of Production Economics*. 103(1):271–285.
- Nemirovski, A., and Shapiro, A. 2006. Convex Approximations of Chance Constrained Programs. *SIAM Journal on Optimization*. 17(4):969–996.
- Orçun, S.; Altinel, I. K.; and Hortaçsu, O. 1996. Scheduling of Batch Processes with Operational Uncertainties. *Computers & Chemical Engineering*. 20(2):S1191–S1196.
- Pagan, A., and Ullah, A. 1999. *Nonparametric Econometrics*. Themes in Modern Econometrics. Cambridge University Press. Cambridge, UK.
- Papoulis, A. 1991. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, Inc., third edition. New York, NY. USA.
- Pardo, L. 2006. *Statistical Inference Based on Divergence Measures*. Statistics: A Series of Textbooks and Monographs. Chapman & Hall/CRC. Boca Raton, FL. USA.
- Park, M.; Park, S.; Mele, F. D.; and Grossmann, I. E. 2006. Modeling of Purchase and Sales Contracts in Supply Chain Optimization. *Industrial & Engineering Chemistry Research*. 45(14):5013–5026.
- Parzen, E. 2004. Quantile Probability and Statistical Data Modeling. *Statistical Science*. 19(4):652–662.
- Pasche, M. 1998. Markup Pricing and Demand Uncertainty. Technical report, Friedrich-Schiller-Universität Jena, Wirtschaftswissenschaftliche Fakultät. URL: <http://EconPapers.repec.org/RePEc:jen:jenavo:1997-08>. Accessed on October 19, 2014.
- Paulk, M. C.; Weber, C. V.; and Chrissis, M. B. 1993. *The Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute. Carnegie Mellon University. Pittsburgh, PA. USA.

- Phillips, R. L. 2005. *Pricing and Revenue Optimization*. Stanford Business Books – Stanford University Press. Palo Alto, CA. USA.
- Pinedo, M. L. 2009. *Planning and Scheduling in Manufacturing and Services*. Springer Science+Business Media, LLC., second edition. New York, NY. USA.
- Pinto-Varela, T.; Barbosa-Povoa, A. P. F. D.; and Novais, A. Q. 2009. Design and Scheduling of Periodic Multipurpose Batch Plants under Uncertainty. *Industrial & Engineering Chemistry Research*. 48(21):9655–9670.
- Pochet, Y., and Wolsey, L. A. 2006. *Production Planning by Mixed Integer Programming*. Springer Science+Business Media, LLC. New ork, NY. USA.
- Powell, W. B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., second edition. Hoboken, NJ. USA.
- Prékopa, A. 1970. On Probabilistic Constrained Programming. *Proceedings of the Princeton Symposium on Mathematical Programming*. Princeton University Press, Princeton, NJ. 113–138.
- Prékopa, A. 1995. *Stochastic Programming*. Mathematics and Its Applications. Kluwer Academic Publishers. Dordrecht, The Netherlands.
- Quesada, I., and Grossmann, I. E. 1992. An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems. *Computers & Chemical Engineering*. 16(10-11):937–947.
- R Core Team. 2015. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org/>.
- Racine, J. S. 2008. *Nonparametric Econometrics: A Primer*, volume 3(1). Foundations and Trends® in Econometrics. Hoboken, NJ. USA.
- Rey, T.; Kordon, A.; and Wells, C. 2012. *Applied Data Mining for Forecasting Using SAS®*. SAS Institute, Inc. Cary, NC. USA.
- Richards, F. J. 1959. A Flexible Growth Function for Empirical Use. *Journal of Experimental Botany*. 10(2):290–301.
- Rios, L. M., and Sahinidis, N. V. 2013. Derivative-Free Optimization: A Review of Algorithms and Comparison of Software Implementations. *Journal of Global Optimization*. 56(3):1247–1293.
- Rockafellar, R. T., and Uryasev, S. 2000. Optimization of Conditional Value-at-Risk. *Journal of Risk*. 2:21–41.

- Rodríguez, M. A., and Vecchietti, A. 2009. Logical and Generalized Disjunctive Programming for Supplier and Contract Selection under Provision Uncertainty. *Industrial & Engineering Chemistry Research*. 48(11):5506–5521.
- Rodríguez, M. A., and Vecchietti, A. 2012. Mid-Term Planning Optimization Model with Sales Contracts under Demand Uncertainty. *Computers & Chemical Engineering*. 47(1):227–236.
- Roelofs, M., and Bisschop, J. 2015. *Advanced Interactive Multidimensional Modeling System (AIMMS)*. <http://www.aimms.com/>.
- Rosa, C. H., and Ruszczyński, A. 1996. On Augmented Lagrangian Decomposition Methods for Multistage Stochastic Programs. *Annals of Operations Research*. 64(1):289–309.
- Ruszczyński, A. 1997. On Augmented Lagrangian Decomposition Methods for Multistage Stochastic Programs. *Mathematical Programming*. 79(1-3):333–353.
- Sahinidis, N. V.; Grossmann, I. E.; Fornari, R. E.; and Chathrathi, M. 1989. Optimization Model for Long Range Planning in the Chemical Industry. *Computers & Chemical Engineering*. 13(9):1049–1063.
- Sahinidis, N. V. 2004. Optimization Under Uncertainty: State-of-the-Art and Opportunities. *Computers & Chemical Engineering*. 28(6-7):971–983.
- Sarykalin, S.; Serraino, G.; and Uryasev, S. 2008. Value-at-Risk vs. Conditional Value-at-Risk in Risk Management and Optimization. In Chen, Z.-L., and Raghavan, S., eds., *Tutorials in Operations Research*. INFORMS. 270–294.
- SAS Institute, I. 2015. *SAS® 9.3*. SAS Institute, Inc., Cary, North Carolina, United States. <http://www.sas.com/>.
- Sawaya, N. 2006. *Reformulations, Relaxations and Cutting Planes for Generalized Disjunctive Programming*. Ph.D. Dissertation, Carnegie Mellon University.
- Scott, D. W. 1992. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Probability and Statistics. John Wiley & Sons, Inc. New York, NY. USA.
- Shapiro, A. 2006. On Complexity of Multistage Stochastic Programs. *Operations Research Letters*. 34(1):1–8.
- Shimizu, K.; Ishizuka, Y.; and Bard, J. F. 1997. *Nondifferentiable and Two-Level Mathematical Programming*. Kluwer Academic Publishers. Dordrecht, The Netherlands.



- Silverman, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics & Applied Probability. Chapman & Hall/CRC. London, UK.
- Simonovic, S. P., and Srinivasan, R. 1993. Explicit Stochastic Approach for Planning the Operation of Reservoirs for Hydropower Production. In *Extreme Hydrological Events: Precipitation, Floods and Droughts (Proceedings of the Yokohama Symposium)*, number 213, 349–359. Yokohama, Japan: International Association of Hydrological Sciences (IAHS).
- StataCorp LP. 2015. *Stata 13*. StataCorp LP., College Station, Texas, United States. <http://www.stata.com/>.
- Stubbs, E. 2011. *The Value of Business Analytics: Identifying the Path to Profitability*. Wiley and SAS Business Series. John Wiley & Sons, Inc. Hoboken, NJ. USA.
- Sundaramoorthy, A.; Evans, J. M. B.; and Barton, P. I. 2012. Capacity Planning under Clinical Trials Uncertainty in Continuous Pharmaceutical Manufacturing, 1: Mathematical Framework. *Industrial & Engineering Chemistry Research*. 51(42):13692–13702.
- Sung, C., and Maravelias, C. T. 2007. Production Planning in Process Systems Engineering. In Pistikopoulos, E. N.; Georgiadis, M. C.; Dua, V.; and Papageorgiou, L. G., eds., *Process Systems Engineering: Supply Chain Optimization*, volume 4. Weinheim, Germany: Wiley-VCH Verlag GmbH & Co. KGaA. chapter 9.
- Sutiene, K., and Pranevicius, H. 2007. Scenario Generation Employing Copulas. *Proceedings of the World Congress on Engineering. Volume II*. 777–784.
- Taha, H. A. 2010. *Operations Research: An Introduction*. Pearson Education, Inc., tenth edition. Upper Saddle River, NJ. USA.
- Talluri, K. T., and van Ryzin, G. J. 2005. *The Theory and Practice of Revenue Management*. Springer Science+Business Media, Inc. New York, NY. USA.
- Tawarmalani, M., and Sahinidis, N. 2005. A Polyhedral Branch-and-Cut Approach to Global Optimization. *Mathematical Programming* 103:225–249.
- Terrazas-Moreno, S.; Trotter, P. A.; and Grossmann, I. E. 2011. Temporal and Spatial Lagrangean Decompositions in Multi-Site, Multi-Period Production Planning Problems with Sequence-Dependent Changeovers. *Computers & Chemical Engineering*. <http://dx.doi.org/10.1016/j.compchemeng.2011.01.004>.
- The MathWorks Inc. 2015. *MATLAB® R2013a*. The MathWorks Inc., Natick, Massachusetts, United States. <http://www.mathworks.com/>.



- Thiele, A. 2005. Multiperiod Pricing via Robust Optimization. Technical Report 05T-006, Department of Industrial and Systems Engineering, Lehigh University.
- Thomé, A. M. T.; Scavarda, L. F.; Fernandez, N. S.; and Scavarda, A. J. 2012. Sales and Operations Planning: A Research Synthesis. *International Journal of Production Economics*. 138(1):1–13.
- Tong, H. 1993. *Non-Linear Time Series: A Dynamical System Approach*. Oxford Statistical Science Series. Oxford, UK.
- Tsay, A. A.; Nahmias, S.; and Agrawal, N. 1999. Modeling Supply Chain Contracts: A Review. In Tayur, S.; Ganeshan, R.; and Magazine, M., eds., *Quantitative Models for Supply Chain Management*, volume 17 of *International Series in Operations Research & Management Science*. Springer US. 299–336.
- van der Vaart, A. W. 1998. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press. Cambridge, UK.
- Varma, V. A.; Reklaitis, G. V.; Blau, G. E.; and Pekny, J. F. 2007. Enterprise-wide Modeling & Optimization – An Overview of Emerging Research Challenges and Opportunities. *Computers & Chemical Engineering*. 31(5-6):692–711.
- Verderame, P. M.; Elia, J. A.; Li, J.; and Floudas, C. A. 2010. Planning and Scheduling under Uncertainty: A Review Across Multiple Sectors. *Industrial & Engineering Chemistry Research*. 49(9):3993–4017.
- Wassick, J. M. 2009. Enterprise-wide Optimization in an Integrated Chemical Complex. *Computers & Chemical Engineering*. 33(12):1950–1963.
- Wied, D., and Weißbach, R. 2012. Consistency of the Kernel Density Estimator: A Survey. *Statistical Papers*. 53(1):1–21.
- Xu, D.; Chen, Z.; and Yang, L. 2012. Scenario Tree Generation Approaches using K-means and LP Moment Matching Methods. *Journal of Computational and Applied Mathematics*. 236(17):4561–4579.
- You, F.; Grossmann, I. E.; and Wassick, J. M. 2011. Multisite Capacity, Production, and Distribution Planning with Reactor Modifications: MILP Model, Bilevel Decomposition Algorithm versus Lagrangean Decomposition Scheme. *Industrial & Engineering Chemistry Research*. 50(9):4831–4849.
- You, F.; Wassick, J. M.; and Grossmann, I. E. 2009. Risk Management for a Global Supply Chain Planning under Uncertainty: Models and Algorithms. *AIChE Journal*. 55(4):931–946.

Zhou, A.; Qu, B.-Y.; Li, H.; Zhao, S.-H.; Suganthan, P. N.; and Zhang, Q.  
2011. Multiobjective Evolutionary Algorithms: A Survey of the State of the  
Art. *Swarm and Evolutionary Computation*. 1(1):32–49.

# Appendices

## Appendix A Hybrid BD-TLD

### A.1 Data for Example 1

The following tables contain all the data for Example 1.

Table A.1: Product-Group mapping.

Product	Group
A	G1
B	G1
C	G2
D	G2
E	G2

Table A.2: Processing data where the acronyms have the following meanings: batch size (BS), batches per hour (BPH), and scale-up cost (SUC).

Product	Unit	Plant	BS [MT]	BPH [ $\text{hr}^{-1}$ ]	SUC [\$]	Amt. [MT]
A	U11	P1	3	0.20	50,000	100,000
B	U11	P1	4	0.20	50,000	-
C	U11	P1	2	0.20	50,000	100,000
D	U11	P1	2	0.20	50,000	100,000
E	U11	P1	2	0.21	50,000	100,000
A	U12	P1	2	0.20	50,000	-
B	U12	P1	3	0.20	50,000	100,000
C	U12	P1	3	0.20	50,000	-
D	U12	P1	4	0.20	50,000	100,000
E	U12	P1	3	0.20	50,000	100,000
A	U21	P2	4	0.20	50,000	100,000
B	U21	P2	2	0.20	-	100,000
C	U21	P2	2	0.20	50,000	100,000
D	U21	P2	2	0.25	50,000	-
E	U21	P2	4	0.25	50,000	100,000

Table A.3: Product demands.

Customer	Product	Demand [MT]
C2	A	25
C3	A	15
C1	A	20
C1	B	125
C3	B	25
C2	B	60
C2	C	25
C1	C	45
C3	C	10
C1	D	90
C2	D	65
C3	D	0
C1	E	40
C3	E	20
C2	E	75
C2	G	50
C1	G	60
C3	G	75
C2	H	10
C1	H	50
C3	H	75
C3	I	100
C2	I	75
C1	I	50

Table A.4: Outbound transportation costs. If product-plant-customer tuple is not listed, then the shipment is not allowed.

Product	From Plant	Customer	Cost [\$ / MT]
A	P2	C2	65
A	P1	C3	94
A	P1	C2	95
B	P1	C3	60
B	P2	C1	68
B	P1	C2	85
B	P2	C2	0
B	P1	C1	78
C	P1	C3	28
C	P1	C2	0
C	P1	C1	80
D	P1	C2	0
D	P1	C1	35
D	P1	C3	33
E	P2	C1	45
E	P2	C2	170
E	P2	C3	145
E	P1	C1	30
E	P1	C2	45
G	P1	C1	0
G	P1	C2	255
G	P2	C2	70
G	P2	C1	75
G	P1	C3	210
H	P1	C3	200
H	P1	C2	140
H	P2	C1	20
H	P2	C2	65
H	P2	C3	70
I	P2	C3	128
I	P1	C2	190
I	P2	C1	0
I	P1	C1	0

Table A.5: Inbound transportation costs.

<b>Plant</b>	<b>Raw Material</b>	<b>Cost [\$/MT]</b>
P1	RA	1,500
P1	RB	800
P1	RC	1,300
P2	RA	500
P2	RB	1,100
P2	RC	2,000

Table A.6: Plant-to-Plant transportation costs.

<b>From Plant</b>	<b>To Plant</b>	<b>Cost [\$/MT]</b>
P1	P2	55
P2	P1	50

Table A.7: Operating costs.

<b>Unit</b>	<b>Plant</b>	<b>Cost [\$/MT]</b>
U11	P1	35
U12	P1	30
U21	P2	32

Table A.8: Plants' capacities.

<b>Plant</b>	<b>Max Capacity [MT]</b>
P1	14,500
P2	8,500

Table A.9: Bill of materials for finished products. The column Unit Ratio represents the parameter  $CRR_{ril}$ . If plant-raw material-product-unit ratio tuple is not listed, then the unit ratio is zero.

Plant	Raw Material	Product	Unit Ratio
P1	RB	A	0.02
P1	RC	A	0.30
P2	RB	A	0.01
P2	RC	A	0.25
P1	RA	B	0.04
P2	RA	B	0.01
P2	RB	B	0.19
P2	RC	B	0.01
P1	RA	C	0.15
P1	RB	C	0.09
P1	RC	C	0.01
P2	RA	C	0.15
P2	RB	C	0.09
P2	RC	C	0.01
P1	RA	D	0.19
P1	RB	D	0.01
P1	RC	D	0.50
P2	RA	D	0.19
P2	RB	D	0.01
P2	RC	D	0.50
P1	RA	E	0.02
P1	RB	E	0.07
P1	RC	E	0.30
P2	RA	E	0.21
P2	RB	E	0.01
P2	RC	E	0.08

Table A.10: Bill of materials for blended products. The column Unit Ratio represents the parameter  $BR_{ii'l}$ . If plant-product-blend tuple is not listed, then the unit ratio is zero.

Plant	Raw Material	Product	Unit Ratio
P1	A	G	0.67
P1	B	G	0.33
P1	C	H	0.4
P1	D	H	0.6
P1	B	I	0.5
P1	D	I	0.5
P2	A	G	0.2
P2	E	G	0.8
P2	B	I	0.65
P2	C	I	0.25
P2	B	H	0.53
P2	E	H	0.47

Table A.11: Changeover times [hr] (costs [\$]) between groups of products.

Unit: U11	G1	G2
G1	0.75 (75)	3 (300)
G2	2 (200)	0.75 (75)

Unit: U12	G1	G2
G1	0.75 (75)	3 (300)
G2	2 (200)	0.75 (75)

Unit: U21	G1	G2
G1	0.72 (72)	3.3 (330)
G2	2.2 (220)	0.72 (72)

The remainder of the data is as follows: the maximum number of scale-up assignments per plant is 500, the maximum number of scale-up assignments per product is 5, the upper bounds on finished and blended products inventories are 10,000 MT and all inventory costs are \$1,000.

## Appendix B Data-Driven Scenario Generation

### B.1 Data for Example 1

The production yield data were randomly generated using the R programming language version 3.0.1 ([R Core Team, 2015](#)). In particular, the library



PearsonDS (Becker & Klößner, 2013) was used to generate 120 random numbers sampled from a Pearson distribution with given mean, variance, skewness, and kurtosis. The four moments of the generated data were calculated using functions of the library `e1071` (Meyer *et al.*, 2012). The code in Listing B.1 can be used to reproduce the yield data in Example 1.

```
library(PearsonDS)
library(e1071)
moments <- c(mean=0.7, variance=0.02, skewness=-1, kurtosis=4)
set.seed(1234)
v <- rpearson(10*12, moments=moments)
data.min <- min(v)
data.max <- max(v)
data.mean <- mean(v)
data.var <- var(v)
data.dskew <- skewness(v)*data.var^1.5 # Denormalized skewness
data.dkurt <- (kurtosis(v) + 3)*data.var^2 # Denormalized kurtosis
```

Listing B.1: R code to generate production yield data and their moments for Example 1.

Only the first two moments and the ECDF information were considered in the DMPs. The weights for the moments were chosen such that the weight of the mean is the same as the weight of the variance, that is  $w_{i,k} = 1.0/M_{i,k}^2$  and  $w_{i,k} = 1.0/|M_{i,k}|$  for  $L^2$  DMP and  $L^1$  DMP, respectively. The weights for the deviations from the approximation to the ECDF are  $\omega_{i,j} = 0.1$ . The ECDF data were estimated in MATLAB (The MathWorks Inc., 2015) with the function `ecdf` and a simplified GLF ( $\beta_0 = 0$  and  $\beta_1 = 1$ ) was fit to the ECDF data by using the function `lsqcurvefit` with the following options: 10,000 maximum function evaluations (`MaxFunEvals`), 10,000 maximum iterations (`MaxIter`),  $10^{-12}$  function tolerance (`TolFun`), exact Jacobian. Initial guesses for the parameters  $\beta_2$ ,  $\beta_3$ , and  $\beta_4$  were 100, 10, and 1, respectively.

The parameters for the production planning LP model are given in the following tables.

Table B.1: First-stage production yield ( $\theta_f$  at  $t = 1$ ) [-]. For facilities  $P2$  and  $P3$ , the yields remain the same at the second stage.

Facility	Group
$P1$	0.85
$P2$	0.7
$P3$	0.9

Table B.2: Product demand ( $\xi_{m,t}$ ) [t].

Product	Time Period			
	1	2	3	4
<i>C</i>	20	10	15	17
<i>D</i>	15	7	12	10

Table B.3: Maximum inventory ( $w_{m,t}^{\text{inv,max}}$ ) [t].

Product	Time Period			
	1	2	3	4
<i>D</i>	6	6	6	6

Table B.4: Maximum (minimum) capacity ( $w_{f,t}^{\text{min}}$  and  $w_{f,t}^{\text{max}}$ ) [t].

Facility	Time Period			
	1	2	3	4
<i>P1</i>	20 (5)	20 (5)	20 (5)	20 (5)
<i>P2</i>	18 (0)	18 (0)	18 (0)	18 (0)
<i>P3</i>	19 (2)	19 (2)	19 (2)	19 (2)

Table B.5: Selling price ( $\text{SP}_{m,t}$ ) [\$/t].

Product	Time Period			
	1	2	3	4
<i>C</i>	1.75	1.75	1.71	1.71
<i>D</i>	1.10	1.10	1.18	1.18

Table B.6: Operating cost ( $\text{OPC}_{f,t}$ ) [\$/t].

Facility	Time Period			
	1	2	3	4
<i>P1</i>	0.26	0.26	0.14	0.14
<i>P2</i>	0.16	1.16	0.13	0.13
<i>P3</i>	0.32	0.32	0.32	0.32

Table B.7: Purchase cost ( $\text{PC}_{m,t}$ ) [\$/t].

Product	Time Period			
	1	2	3	4
<i>A</i>	0.05	0.05	0.05	0.05
<i>C</i>	1.00	1.00	1.00	1.00

Table B.8: Inventory cost ( $IC_{m,t}$ ) [\$/t].

Product	Time Period			
	1	2	3	4
<i>D</i>	0.03	0.03	0.03	0.03

Table B.9: Raw material availability ( $x_{A,t}^{\text{purch,max}}$ ) [t].

Raw Material	Time Period			
	1	2	3	4
<i>A</i>	30	25	27	23

The penalties for unmet demand ( $PEN_{m,t}$ ), and maximum capacity and minimum capacity violations ( $PEN_{f,t}$ ) are 10 times the selling prices, and proportional to the operating costs, respectively.

## B.2 Data for Example 2

Time series modeling and forecasting were implemented in the R programming language (R Core Team, 2015). For demonstration purposes, only ARIMA models (see Appendix B.3) were fit to the time series data. The library used for fitting ARIMA models is called `forecast` (Hyndman *et al.*, 2014). In particular, the function `auto.arima` was used to automatically determine the best ARIMA model that fits the data according to the default information criterion (Akaike Information Criterion, AIC). Seasonality effects were allowed and no hold out sample was considered in the fitting process. The time series demand data for both products *C* and *D* were then modeled as ARIMA(1,0,0) processes with non-zero means. Future values were predicted with 95% level of confidence using the `predict` function in the `stats` library, which is part of R.

The constant variance of ARIMA models was estimated by the `sigma2` attribute of the ARIMA object. For the *LP Approach* in Subsection 3.4.2, the standard error of a forecast was estimated by setting the value of the argument `se.fit` of the `predict` function to `TRUE`. Covariance information was estimated through R’s builtin function `ccf`, which estimates the cross-covariance function of two time series data at different lags. Since the *NLP Approach* involves alternating one-step-ahead forecasting with optimization, the estimated covariance corresponds to the lag 1 outcome of function `ccf`.

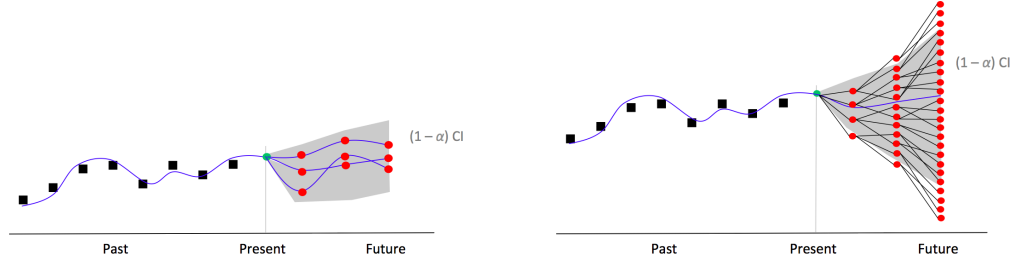
The parameters for the production planning LP model are the same as the data given in tables Tables B.3 to B.9. In addition, the production yield for all time periods are the same as those given in Table B.1. The product demand for every stage in each scenario tree is generated as described in Subsection 3.4.3.

## B.3 Time Series Analysis and Scenario Generation

An overview of time series analysis is given in Subsection 1.2.2. In this appendix, we illustrate the use of time series modeling in the context of scenario

tree generation.

Figure B.1 shows a schematic of how time series forecasting can be used when generating a scenario tree for a stochastic process. Figure B.1(a) shows time series data, represented by the black squares in the “past” region, a model fit to the data in the blue line across the markers, and point forecasts as the red circles in the “future” region. Possibly, additional nodes above and below the base node are added to each stage and they represent different outcomes of the stochastic process in a real-world problem. By connecting each parent node to its descendants, a scenario tree is obtained as shown in Figure B.1(b). Note that nodes pertaining to the third and beyond stages can overlap, thus resembling a mesh.



(a) Time series model fit to historical data (b) Scenario tree generated from forecasts. and used for forecasting.

Figure B.1: Using time series forecasting to generate a scenario tree for a stochastic process. CI denotes the confidence interval estimated at each forecast, respectively.

## B.4 Assessing Solution Quality in Stochastic Programming

In this section, we review two approaches based on Monte Carlo simulation and the Sample Average Approximation (SAA) to assess the quality of the solution in two- and multi-stage stochastic programming. These probabilistic- and statistics-based approaches can be used to quantify the effectiveness of the scenario generation scheme since the quality of the scenario tree affects the quality of the solution.

### B.4.1 Two-Stage Stochastic Programs

The Multiple Replications Procedure (MRP) as proposed by Bayraksan & Morton (2006) is reviewed here.

First, we define the two-stage stochastic programming (TSSP) model as follows:

$$z^* = \min_{x \in X} \mathbb{E}f(x, \tilde{\xi}) \quad (\text{B.1})$$

where  $f(\cdot, \cdot)$  is a real-valued function that determines the cost of operating with decision  $x$  under a realization of the random vector  $\tilde{\xi}$ .  $X \subseteq \mathbb{R}^d$  denotes the set

of constraints that the decision vector  $x$  must obey and  $\mathbb{E}$  is the expectation operator. For linear and continuous SP problems with recourse, we have:

$$\begin{aligned} f(x, \tilde{\xi}) &= cx + \min_{y \geq 0} \tilde{q}y \\ \text{s.t. } &\tilde{W}y = \tilde{r} - \tilde{T}x \end{aligned}$$

where  $X = \{x : Ax = b, x \geq 0\}$  and  $\tilde{\xi} = (\tilde{W}, \tilde{q}, \tilde{r}, \tilde{T})$  is a random vector.

Let  $\tilde{\xi}^1, \tilde{\xi}^2, \dots, \tilde{\xi}^n$  be independent and identically distributed (i.i.d.) outcomes generated from the distribution of  $\tilde{\xi}$  or some data-generating mechanism. Given a feasible, candidate solution  $\hat{x}$  and a sample size  $n$ , we bound the optimal value of problem (B.1) in the following manner: the upper bound is given by the evaluation of the candidate solution at each realization  $\tilde{\xi}^j$  generated, and the lower bound is obtained by solving the SAA problem that approximates the original SP problem, i.e., an approximate “true” solution. The difference between the upper and lower bounds is denoted the *optimality gap*, which can be written mathematically as follows:

$$G_n(\hat{x}) = \frac{1}{n} \sum_{j=1}^n f(\hat{x}, \tilde{\xi}^j) - \min_{x \in X} \frac{1}{n} \sum_{j=1}^n f(x, \tilde{\xi}^j) \quad (\text{B.2})$$

The first and second terms on the right-hand side of equation (B.2) are the upper and lower bound estimates on  $z^*$ , respectively. The procedure of calculating the gap  $G_n(\hat{x})$  with different samples of  $\tilde{\xi}^j$  is *replicated*  $n_g$  times in order to compute an average gap,  $\bar{G}(n_g)$ , and its one-sided confidence interval. This procedure, called Multiple Replications Procedure (MRP), is formalized below and illustrated in Figure B.2.

*Input:* Significance level  $\alpha$  (e.g.,  $\alpha = 0.05$  for 95% confidence), sample size  $n$ , number of replications  $n_g$ , and a candidate solution  $\hat{x} \in X$

*Output:*  $(1 - \alpha)$ -level confidence interval on  $\mu_{\hat{x}}$

1. For  $i = 1, 2, \dots, n_g$

1.1 Sample i.i.d. observations  $\tilde{\xi}^{i1}, \tilde{\xi}^{i2}, \dots, \tilde{\xi}^{in}$  from the distribution of  $\tilde{\xi}$

1.2 Solve  $(\text{SP}_n^i)$  using  $\tilde{\xi}^{i1}, \tilde{\xi}^{i2}, \dots, \tilde{\xi}^{in}$  to obtain  $x_n^{i*}$

1.3 Calculate  $G_n^i(\hat{x}) = \frac{1}{n} \sum_{j=1}^n (f(\hat{x}, \tilde{\xi}^{ij}) - f(x_n^{i*}, \tilde{\xi}^{ij}))$

2. Calculate gap estimate and sample variance by

$$\bar{G}(n_g) = \frac{1}{n_g} \sum_{i=1}^{n_g} G_n^i(\hat{x}) \quad \text{and} \quad s_G^2(n_g) = \frac{1}{n_g - 1} \sum_{i=1}^{n_g} (G_n^i(\hat{x}) - \bar{G}(n_g))^2$$

3. Output one-sided CI on  $\mu_{\hat{x}}$

$$\left[ 0, \bar{G}(n_g) + t_{n_g-1, \alpha} \frac{s_G(n_g)}{\sqrt{n_g}} \right]$$

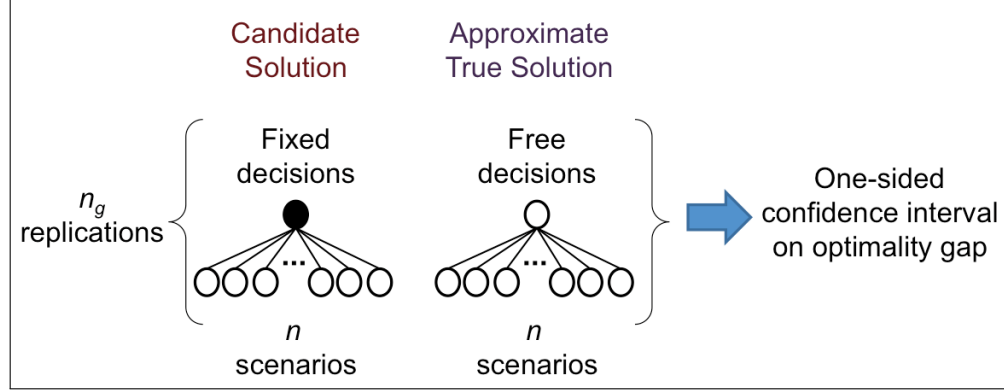


Figure B.2: Schematic of the Multiple Replications Procedure (MRP) to assess the quality of two-stage stochastic programming solutions.

#### B.4.2 Multi-Stage Stochastic Programs

The Procedure  $\mathcal{P}_2$  as proposed in [Chiralaksanakul & Morton \(2004\)](#) is reviewed here.

First, some notation for multi-stage stochastic programming (MSSP) problems. We consider a  $T$ -stage stochastic program in which a sequence of decisions,  $\{x_t\}_{t=1}^T$ , is made with respect to a stochastic process,  $\{\tilde{\xi}_t\}_{t=1}^T$ , as follows: at stage  $t$ , the decision  $x_t \in \mathbb{R}^{d_t}$  is made with only the knowledge of past decisions,  $x_1, \dots, x_{t-1}$ , and of realized random vectors,  $\tilde{\xi}_1, \dots, \tilde{\xi}_t$ , such that the conditional expected value of an objective function given the history,  $\tilde{\xi}_1, \dots, \tilde{\xi}_t$ , is minimized. A superscript  $t$  on an entity denotes its history through stage  $t$ , *e.g.*,  $\tilde{\xi}^t = (\tilde{\xi}_1, \dots, \tilde{\xi}_t)$  and  $x^t = (x_1, \dots, x_t)$ . Decision  $x_t$  is subject to constraints that may depend on  $x_1, \dots, x_{t-1}$  and  $\tilde{\xi}_1, \dots, \tilde{\xi}_t$ . The requirement that decision  $x_t$  not depend on future realizations of  $\tilde{\xi}_{t+1}, \dots, \tilde{\xi}_T$  is known in the stochastic programming literature as nonanticipativity.

A  $T$ -stage stochastic linear program can be expressed in the following form:

$$\begin{aligned} \min_{x_1} \quad & c_1 x_1 + \mathbb{E} \left[ h_1(x_1, \tilde{\xi}^2) | \tilde{\xi}^1 \right] \\ \text{s.t.} \quad & A_1 x_1 = b_1 \\ & x_1 \geq 0 \end{aligned} \tag{B.3}$$

where, for  $t = 2, \dots, T$ ,

$$\begin{aligned} h_{t-1}(x_{t-1}, \tilde{\xi}^t) = \min_{x_t} \quad & \tilde{c}_t x_t + \mathbb{E} \left[ h_t(x_t, \tilde{\xi}^{t+1}) | \tilde{\xi}^t \right] \\ \text{s.t.} \quad & \tilde{A}_t x_t = \tilde{b}_t - \tilde{B}_t x_{t-1} \\ & x_t \geq 0 \end{aligned} \tag{B.4}$$

and  $h_T(\cdot, \cdot) = 0$ . The random vector  $\tilde{\xi}_t$  consists of the random elements from  $(\tilde{A}_t, \tilde{B}_t, \tilde{b}_t, \tilde{c}_t)$ . Relatively complete recourse is assumed to hold for problems (B.3) and (B.4).

For problems where the stochastic process  $\{\tilde{\xi}_t\}_{t=1}^T$  is interstage dependent, the procedure to generate a feasible policy for the MSSP is as follows. For a given  $\tilde{\xi}^t$ , we obtain  $\hat{x}_t(\tilde{\xi}^t)$  by solving an approximating problem (from stage  $t$  to  $T$ ) based on an independently-generated sample subtree, denoted  $\Gamma_r(\tilde{\xi}^t)$  (the ‘ $r$ ’ subscript stands for “rolling”). Specifically, for a given  $\tilde{\xi}^t$  and  $x^{t-1}$ ,  $\Gamma_r(\tilde{\xi}^t)$  is constructed via a conditional sampling procedure. Then,  $\hat{x}_t(\tilde{\xi}^t)$  is defined as an optimal solution of the approximating problem (SAA) of (B.3) and (B.4).

The policy-generation Procedure  $\mathcal{P}_2$  is formally described as follows.

Given sample path  $\tilde{\xi}^T$ ,

Do  $t = 1$  to  $T$

Independently construct a sample subtree  $\Gamma_r(\tilde{\xi}^t)$ .

Solve approximating problem (SAA) with  $x^{t-1}$  equal to  $\hat{x}^{t-1}(\tilde{\xi}^{t-1})$ , and denote its optimal solution  $\hat{x}^t(\tilde{\xi}^t)$ .

Figure B.3 illustrates the procedure. A candidate first-stage solution is obtained by solving the MSSP problem using a given scenario tree, for example, from solving one of the Distribution Matching Problems proposed in this paper. Then, these first-stage decisions are fixed into the approximating problem (SAA) whose subtrees have been conditionally sampled (step (a)). The SAA problem is solved and its solution is stored. In step (b), the decisions and subtrees up until the second stage are fixed to the respective stored values, the subtrees of subsequent stages are conditionally sampled, and the SAA is resolved. In the last step to obtain a candidate solution (step (c)), all decisions and subtrees for all stages until  $T - 1$  are fixed, subtrees for the last stage  $T$  are conditionally sampled, and the SAA problem is solved. This candidate solution corresponds to an upper bound on the solution of the original MSSP problem. The lower bound on the objective function value of the original MSSP problem (step (d)) corresponds to the solution of the approximating problem with the same subtrees used in step (c) and with no fixed decisions. The difference between the two bounds is defined as the *optimality gap*. By replicating this procedure  $n_g$  times, an average gap and its one-sided confidence interval can be calculated.

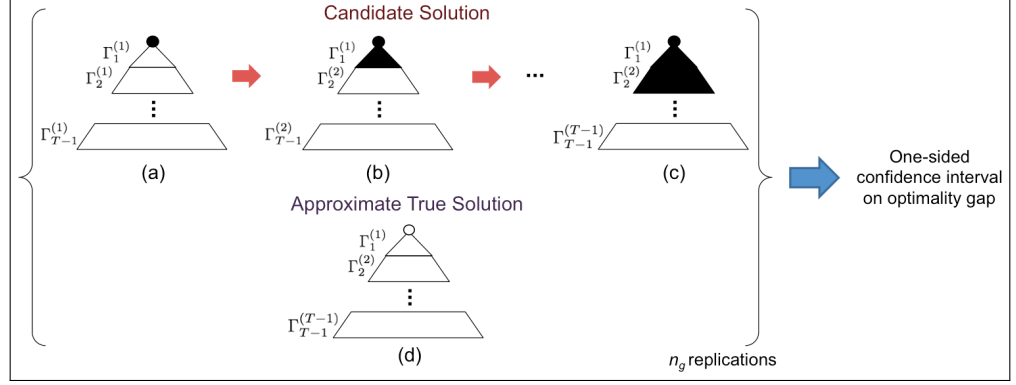


Figure B.3: Schematic of the policy-generation Procedure  $\mathcal{P}_2$  to assess the quality of multi-stage stochastic programming solutions. In this figure,  $\Gamma_t^{(r)}$  denotes the set of subtrees conditionally sampled from stage  $t$  at step  $r$ .

## Appendix C Data-Driven Chance Constraints

### C.1 Motivation of Chance-Constrained Optimization

We use the deterministic linear optimization model in equation (C.1) to illustrate the classical approach to optimization with individual and joint chance constraints (ICCs and JCCs) with right-hand side (RHS) uncertainty. The feasible region and optimal solution are shown in Figure C.1.

$$\max_{x_1, x_2} z = x_1 + x_2 \quad (\text{C.1a})$$

$$\text{s.t.} \quad \frac{1}{3}x_1 + x_2 \leq 4 \quad (\text{C.1b})$$

$$\frac{3}{2}x_1 + x_2 \leq \frac{15}{2} \quad (\text{C.1c})$$

$$x_1, x_2 \geq 0 \quad (\text{C.1d})$$



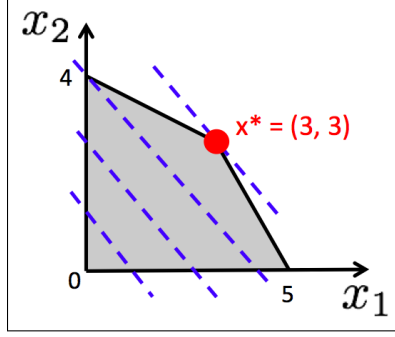


Figure C.1: Feasible region (grey area delimited by polytope in solid black lines) and optimal solution (red filled circle,  $x^* = (3, 3)$ ) for the deterministic linear optimization model in equation (C.1). Dashed blue lines are the contours of the objective function.

Suppose the right-hand sides of constraints in equations (C.1b) and (C.1c) are uncertain parameters. For demonstration purposes, let  $\tilde{\xi}_1 \sim N(4, 1)$  and  $\tilde{\xi}_2 \sim N\left(\frac{15}{2}, 2\right)$  be normally distributed RHS uncertain parameters, which are centered in the previously deterministic values. For specified individual and joint risk levels (e.g.,  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha$ ), the corresponding ICCs are given by,

$$\mathbb{P}\left\{\frac{1}{3}x_1 + x_2 \leq \tilde{\xi}_1\right\} \geq 1 - \alpha_1 \quad (\text{C.2})$$

$$\mathbb{P}\left\{\frac{3}{2}x_1 + x_2 \leq \tilde{\xi}_2\right\} \geq 1 - \alpha_2 \quad (\text{C.3})$$

while the JCC can be written as,

$$\mathbb{P}\left\{\begin{array}{l} \frac{1}{3}x_1 + x_2 \leq \tilde{\xi}_1 \\ \frac{3}{2}x_1 + x_2 \leq \tilde{\xi}_2 \end{array}\right\} \geq 1 - \alpha \quad (\text{C.4})$$

In order to reformulate the ICCs in equations (C.2) and (C.3) into algebraic constraints, notice that they model the following situation: “the probability of a random variable  $\tilde{\xi}_1$  ( $\tilde{\xi}_2$ ) to attain a value of *at least* the left-hand side  $\frac{1}{3}x_1 + x_2$  ( $\frac{3}{2}x_1 + x_2$ ) has to be greater than or equal to  $1 - \alpha_1$  ( $1 - \alpha_2$ )”. Mathematically, this is equivalent to requiring that the survival function (Evans, Hastings, & Peacock, 2000) of each random variable evaluated at the left-hand side has to be greater than or equal to the confidence level. In the univariate case, the survival function is the complement of the cumulative distribution function (CDF), which is denoted by  $F(\cdot)$ . In other words, the survival function is equivalent to  $1 - F(\cdot)$ . Thus, for the ICC in equation (C.2), we have that

$$\begin{aligned} 1 - F_{\tilde{\xi}_1}\left(\frac{1}{3}x_1 + x_2\right) &\geq 1 - \alpha_1 \\ F_{\tilde{\xi}_1}\left(\frac{1}{3}x_1 + x_2\right) &\leq \alpha_1 \end{aligned}$$

or

$$\frac{1}{3}x_1 + x_2 \leq F_{\tilde{\xi}_1}^{-1}(\alpha_1) \quad (\text{C.5})$$

and similarly for the ICC in equation (C.3), we have that

$$\frac{3}{2}x_1 + x_2 \leq F_{\tilde{\xi}_2}^{-1}(\alpha_2) \quad (\text{C.6})$$

where  $F_{\tilde{\xi}_1}^{-1}(\cdot)$  and  $F_{\tilde{\xi}_2}^{-1}(\cdot)$  are the inverse CDFs (quantile functions) of the random variables  $\tilde{\xi}_1$  and  $\tilde{\xi}_2$ , respectively. Note that the linearity of the original constraints is preserved.

The same inversion cannot be used in the multivariate case to reformulate JCCs into algebraic constraints. From the definition of multivariate CDFs, the JCC in equation (C.4) can be rewritten as follows

$$\int_{\frac{1}{3}x_1+x_2}^{\infty} \int_{\frac{3}{2}x_1+x_2}^{\infty} f_{\tilde{\xi}_1, \tilde{\xi}_2}(\xi_1, \xi_2) d\xi_2 d\xi_1 \geq 1 - \alpha \quad (\text{C.7})$$

where  $f_{\tilde{\xi}_1, \tilde{\xi}_2}(\cdot, \cdot)$  is the joint probability density function (PDF) of  $\tilde{\xi}_1$  and  $\tilde{\xi}_2$ . Since we assumed that the random variables are Gaussian, then the joint PDF in this case is a bivariate normal Gaussian PDF. Unlike in the case with ICCs, the linearity of the original constraints is not preserved in the JCC formulation.

Figure C.2 shows a schematic of the results of the linear optimization model with constraints in equations (C.1b) and (C.1c) replaced by equations (C.5) and (C.6) for decreasing values of the individual risk levels. Note that by decreasing them, the feasible region becomes smaller, and the solution becomes increasingly conservative. A similar conclusion can be drawn from Figure C.3 for the JCC case by replacing constraints in equations (C.1b) and (C.1c) by equation (C.7).

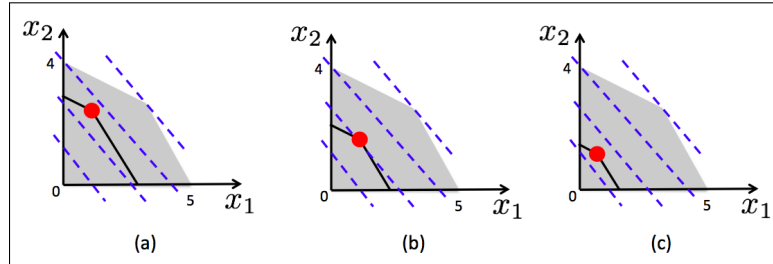


Figure C.2: Schematic of deterministic feasible region (grey area), optimal solutions (filled circles), and the new feasible regions (polytopes delimited by solid black lines) of the optimization model with ICCs for different values of individual risk levels: (a)  $\alpha_1 = \alpha_2 = 0.1$ , (b)  $\alpha_1 = \alpha_2 = 0.05$ , and (c)  $\alpha_1 = \alpha_2 = 0.01$ .

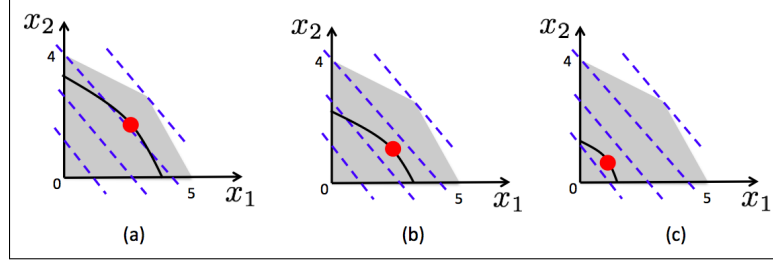


Figure C.3: Schematic of deterministic feasible region (grey area), optimal solutions (filled circles), and the new feasible regions (delimited by curved solid black lines) of the optimization model with JCC for different values of joint confidence levels: (a)  $1 - \alpha = 0.9$ , (b)  $1 - \alpha = 0.95$ , and (c)  $1 - \alpha = 0.99$ .

The accuracy of the reformulated ICCs and JCCs depends on the choice of probability distribution models. The classical approach may encounter limitations in cases when parametric models do not fit well available data for the uncertain parameters, or when distributions are more complex (e.g., multimodal). This motivates the use of data-driven approaches that weaken the modeling assumptions for the uncertainty.

## C.2 Derivation of Bound on ICC Risk Levels for Equivalent JCC

The motivation behind deriving inequality (4.17) is to approximate a joint chance constraint (JCC) by corresponding individual chance constraints (ICCs). Let  $A_j$  be a Boolean expression, an event, or a set that corresponds to a constraint in a chance-constrained formulation (e.g.,  $A_j \equiv g_j(x, \tilde{\xi}_j) \geq 0$ ), and  $A_j^c$  its complement (e.g.,  $A_j^c \equiv g_j(x, \tilde{\xi}_j) < 0$ ). In other words, we wish to approximate the JCC  $\mathbb{P}\{\cap_{j \in J} A_j\} \geq 1 - \alpha^{\text{JCC}}$  by ICCs  $\mathbb{P}\{A_j\} \geq 1 - \alpha_j^{\text{ICC}}, \forall j \in J$ . The derivation answers the following question: given  $\alpha^{\text{JCC}}$ , how should one set  $\alpha_j^{\text{ICC}}$  in order to obtain the aforementioned approximation?

We derive equation (4.17) by showing that the joint risk level of a JCC is a conservative upper bound on the summation of risk levels of corresponding individual ICCs. One can then write

$$\begin{aligned} \mathbb{P}\{A_j\} &\geq 1 - \alpha_j^{\text{ICC}} & j = 1, \dots, m \\ \mathbb{P}\{A_j^c\} &\leq \alpha_j^{\text{ICC}} & j = 1, \dots, m \end{aligned}$$

From Boole's Inequality (a special case of Bonferroni's Inequalities ([Bonferroni, 1936](#))), we have that

$$\mathbb{P}\left\{\bigcup_{j \in J} A_j^c\right\} \leq \sum_{j \in J} \mathbb{P}\{A_j^c\}$$

Note that the right-hand side of the Boole's Inequality corresponds to the summation of the probabilities of *individual* events, i.e., the summation of the

individual risk levels ( $\alpha_j^{\text{ICC}}$ ) of the ICCs. Next, applying one of De Morgan's Laws (Goodstein, 2007) to the left side of Boole's Inequality yields

$$\mathbb{P}\left\{\bigcup_{j \in J} A_j^c\right\} = \left(\mathbb{P}\left\{\bigcap_{j \in J} A_j\right\}\right)^c = 1 - \mathbb{P}\left\{\bigcap_{j \in J} A_j\right\}$$

Note that the term  $\mathbb{P}\{\bigcap_{j \in J} A_j\}$  corresponds to the probability of the *intersection* of events, i.e., the joint confidence level ( $1 - \alpha^{\text{JCC}}$ ) of the JCC. Combining this result with Boole's Inequality gives

$$\mathbb{P}\left\{\bigcap_{j \in J} A_j\right\} \geq 1 - \sum_{j \in J} \alpha_j^{\text{ICC}} \quad (\text{C.8})$$

For comparison purposes, we repeat the definition of the original JCC as follows

$$\mathbb{P}\left\{\bigcap_{j \in J} A_j\right\} \geq 1 - \alpha^{\text{JCC}} \quad (\text{C.9})$$

Note that inequalities (C.8) and (C.9) share the same left-hand side. If we *impose* that (inequality (4.17))

$$\alpha^{\text{JCC}} \geq \sum_{j \in J} \alpha_j^{\text{ICC}}$$

then we will guarantee that the above inequality is a conservative approximation.

To illustrate, suppose  $\alpha^{\text{JCC}} = 0.1$  such that  $\mathbb{P}\{\bigcap_{j \in J} A_j\} \geq 0.9$ . Therefore, the conservative approximation in inequality (4.17) *suggests* that the values of the individual risk levels  $\alpha_j^{\text{ICC}}$  are set such that  $\sum_{j \in J} \alpha_j^{\text{ICC}} \leq 0.1$ .

The authors acknowledge Dr. Shabbir Ahmed (School of Industrial & Systems Engineering at Georgia Institute of Technology) for the insights in the derivation of this conservative approximation.

### C.3 Data for Motivating Example

The kernel density and distribution estimators were obtained in the R programming language (R Core Team, 2015), package `np` (Hayfield & Racine, 2008). The Gaussian kernel was used in all estimation computational experiments. Tables C.1 and C.2 show the bandwidths and reduced risk levels used in the (*PPJCC*) model. In order to generate the data for production rates of plants *P2* and *P3*, we set the seed to 1234. Table C.3 shows the estimated quantiles used in the (*PPICC*) model.

Table C.1: Cross-validated bandwidths used in the (*PPJCC*) model.

Bandwidth	$n = 365$	$n = 730$
$h_{P2}$	1.6924	1.3172
$h_{P3}$	2.0628	1.7151

Table C.2: Reduced risk levels ( $\alpha'$ ) obtained with the K-L formula (Table 4.1) and used in the (*PPJCC*) model.

Reduced Risk Level	$\alpha = 0.10$		$\alpha = 0.05$		$\alpha = 0.01$	
	$n = 365$	$n = 730$	$n = 365$	$n = 730$	$n = 365$	$n = 730$
$\alpha'$	0.0894	0.0924	0.0424	0.0445	0.0068	0.0076

Table C.3: Estimated quantiles for nominal risk levels ( $\alpha$ ) used in the (*PPICC*) model.

Production Quantile	$\alpha = 0.10$		$\alpha = 0.05$		$\alpha = 0.01$	
	$n = 365$	$n = 730$	$n = 365$	$n = 730$	$n = 365$	$n = 730$
$\hat{F}_{\tilde{w}_{P2}^{\text{rate,max}}}^{-1}(\alpha'_{P2,+})$	18.30	15.45	8.07	17.88	14.90	7.82
$\hat{F}_{\tilde{w}_{P3}^{\text{rate,max}}}^{-1}(\alpha'_{P3,+})$	15.73	11.98	3.93	16.86	13.59	4.56

The parameters for the production planning LP model are given in the following tables.

Table C.4: Production yield ( $\theta_f$ ) [-].

Facility	Group
$P1$	0.85
$P2$	0.7
$P3$	0.9

Table C.5: Product demand ( $\gamma_{m,t}$ ) [t].

Product	Time Period			
	1	2	3	4
$C$	20	10	15	17
$D$	15	7	12	10

Table C.6: Maximum inventory ( $w_{m,t}^{\text{inv,max}}$ ) [t].

Product	Time Period			
	1	2	3	4
$D$	6	6	6	6

Table C.7: Maximum (minimum) capacity ( $w_f^{\text{min}}$  and  $w_f^{\text{max}}$ ) [t].

Facility	Max (Min) Capacity
$P1$	20 (5)
$P2$	18 (0)
$P3$	19 (2)

Table C.8: Selling price ( $SP_{m,t}$ ) [\$/t].

Product	Time Period			
	1	2	3	4
<i>C</i>	1.75	1.75	1.71	1.71
<i>D</i>	1.10	1.10	1.18	1.18

Table C.9: Operating cost ( $OPC_{f,t}$ ) [\$/t].

Facility	Time Period			
	1	2	3	4
<i>P1</i>	0.26	0.26	0.14	0.14
<i>P2</i>	0.16	0.16	0.13	0.13
<i>P3</i>	0.32	0.32	0.32	0.32

Table C.10: Purchase cost ( $PC_{m,t}$ ) [\$/t].

Product	Time Period			
	1	2	3	4
<i>A</i>	0.05	0.05	0.05	0.05
<i>C</i>	1.00	1.00	1.00	1.00

Table C.11: Inventory cost ( $IC_{m,t}$ ) [\$/t].

Product	Time Period			
	1	2	3	4
<i>D</i>	0.03	0.03	0.03	0.03

Table C.12: Raw material availability ( $x_{A,t}^{\text{purch,max}}$ ) [t].

Raw Material	Time Period			
	1	2	3	4
<i>A</i>	30	25	27	23

The penalties for unmet demand ( $PEN_{m,t}$ ), and maximum capacity and minimum capacity violations ( $PEN_{f,t}$ ) are 10 times the selling prices, and proportional to the operating costs, respectively.

# Appendix D Procurement Contracts and Pricing

## D.1 Purchase Contract Mixed-Integer Linear Models

The models for the three contract types (discount, bulk, and fixed duration) are proposed in [Park \*et al.\* \(2006\)](#). In this section, we present the mixed-integer linear constraints obtained by reformulating the disjunctions using the convex hull approach. At most one contract type can be selected for a given raw material  $j$ , which is expressed as follows:

$$\sum_{c \in C} y_{j,s,t}^c \leq 1 \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.1})$$

where  $C = \{d, b, l\}$ .

### D.1.1 Discount Contract

Contract type  $c = d$  and two price schemes,  $d_1$  and  $d_2$ .

$$\text{COST}_{j,s,t}^d = \varphi_{j,s,t}^{d_1} P_{j,s,t}^{d_1} + \varphi_{j,s,t}^{d_2} P_{j,s,t}^{d_2} \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.2})$$

$$P_{j,s,t}^d = P_{j,s,t}^{d_1} + P_{j,s,t}^{d_2} \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.3})$$

$$P_{j,s,t}^{d_1} = P_{j,s,t}^{d_{1,1}} + P_{j,s,t}^{d_{1,2}} \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.4})$$

$$0 \leq P_{j,s,t}^{d_{1,1}} \leq y_{j,s,t}^{d_1} \sigma_{j,s,t}^d \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.5})$$

$$P_{j,s,t}^{d_{1,2}} = y_{j,s,t}^{d_2} \sigma_{j,s,t}^d \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.6})$$

$$0 \leq P_{j,s,t}^{d_2} \leq y_{j,s,t}^{d_2} U_{j,s,t}^d \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.7})$$

$$y_{j,s,t}^{d_1} + y_{j,s,t}^{d_2} = y_{j,s,t}^d \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.8})$$

where  $y_{j,s,t}^{d_1}, y_{j,s,t}^{d_2} \in \{0, 1\}$ , and  $U_{j,s,t}^d$  is a number large enough (e.g., process capacity).

### D.1.2 Bulk Discount Contract

Contract type  $c = b$  and two price schemes,  $b_1$  and  $b_2$ .

$$\text{COST}_{j,s,t}^b = \varphi_{j,s,t}^{b_1} P_{j,s,t}^{b_1} + \varphi_{j,s,t}^{b_2} P_{j,s,t}^{b_2} \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.9})$$

$$P_{j,s,t}^b = P_{j,s,t}^{b_1} + P_{j,s,t}^{b_2} \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.10})$$

$$0 \leq P_{j,s,t}^{b_1} \leq y_{j,s,t}^{b_1} \sigma_{j,s,t}^b \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.11})$$

$$y_{j,s,t}^{b_2} \sigma_{j,s,t}^b \leq P_{j,s,t}^{b_2} \leq y_{j,s,t}^{b_2} U_{j,s,t}^b \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.12})$$

$$y_{j,s,t}^{b_1} + y_{j,s,t}^{b_2} = y_{j,s,t}^b \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.13})$$

where  $y_{j,s,t}^{b_1}, y_{j,s,t}^{b_2} \in \{0, 1\}$ , and  $U_{j,s,t}^b$  is a number large enough (e.g., process capacity).

### D.1.3 Fixed Duration Contract

Contract type  $c = l$  and three price-quantity schemes,  $l_p$  for  $p \in \text{LC} = T_{1:3}$ , where the notation  $T_{m:n}$  means a subset of the ordered set  $T$  containing all its elements in between and including the  $m$ -th and  $n$ -th elements. For example, if  $T = \{1, 2, 3, 4, 5, 6\} \subset \mathbb{Z}$ , then  $\text{LC} = T_{1:3} = \{1, 2, 3\}$ .

$$\text{COST}_{j,s,t}^l = \varphi_{j,s,t}^{l_1} P_{j,s,t,t}^{l_1} + \sum_{p \in \text{LC}_{2:|\text{LC}|}} \sum_{\substack{t' \in T \\ t' \leq t \\ t' \geq t-l_p+1}} \varphi_{j,s,t'}^{l_p} P_{j,s,t,t'}^{l_p} \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.14})$$

$$P_{j,s,t}^l = P_{j,s,t,t}^{l_1} + \sum_{p \in \text{LC}_{2:|\text{LC}|}} \sum_{\substack{t' \in T \\ t' \leq t \\ t' \geq t-l_p+1}} P_{j,s,t,t'}^{l_p} \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.15})$$

$$y_{j,s,t}^{l_p} \sigma_{j,s,t}^{l_p} \leq P_{j,s,t,t'}^{l_p} \leq y_{j,s,t}^{l_p} U_{j,s,t}^l \quad \forall j \in \text{JR}, s \in S, p \in \text{LC}, \quad (t, t') \in T, t' \leq t, \quad t' \geq t - l_p + 1 \quad (\text{D.16})$$

$$\sum_{p \in \text{LC}} y_{j,s,t}^{l_p} = y_{j,s,t}^l \quad \forall j \in \text{JR}, s \in S, t \in T \quad (\text{D.17})$$

$$y_{j,s,t}^{l_p} \leq 1 - y_{j,s,t'}^{l_{p'}} \quad \forall j \in \text{JR}, s \in S, \quad (p, p') \in \text{LC}, (t, t') \in T, \quad t' < t, t' \geq t - l_p + 1 \quad (\text{D.18})$$

where  $y_{j,s,t}^{l_p} \in \{0, 1\}$ , for  $p \in \text{LC}$ , and  $U_{j,s,t}^l$  is a number large enough (e.g., process capacity).

## D.2 Concavity of Sales Term when using Logit Demand-Response Model

In this appendix, we show the following result: the function

$$f(d) = \frac{1}{\beta_5} \left[ \ln \left( \frac{\beta_3 - d}{d} \right) - \beta_4 \right] d \quad (\text{D.19})$$

is concave if  $\beta_3 \cdot \beta_5 > 0$  and  $\beta_4 \in \mathbb{R}$ . This function arises in the sales term (selling price  $\times$  sales) when using the logit demand-response model (see [Table 6.2](#)). This is a univariate function, and its concavity can be verified by analyzing the sign of its second derivative with respect to  $d$  evaluated at critical points. Symbolic calculations were performed with Maple 18.01 ([Maplesoft, 2014](#)).

Setting the first derivative of  $f(d)$  with respect to  $d$  to zero yields

$$f'(d) = \frac{1}{\beta_5} \left[ \ln \left( \frac{\beta_3 - d}{d} \right) - \beta_4 \right] - \frac{\beta_3}{\beta_5(\beta_3 - d)} = 0$$



Solving the equation for  $d$  gives the single stationary point

$$d^* = \beta_3 \frac{W(e^{-1-\beta_4})}{1 + W(e^{-1-\beta_4})}$$

where  $W(\cdot)$  is the Lambert W function (Corless *et al.*, 1996) that satisfies the following equation

$$W(x)e^{W(x)} = x \quad (\text{D.20})$$

The second derivative of  $f(d)$  with respect to  $d$  is given by

$$f''(d) = -\frac{\beta_3^2}{\beta_5(\beta_3 - d)^2 d}$$

which evaluated at  $d^*$  yields

$$f''(d^*) = -\frac{[1 + W(e^{-1-\beta_4})]^3}{\beta_3 \beta_5 W(e^{-1-\beta_4})} \leq 0 \quad (\text{D.21})$$

If the relation in equation (D.21) is true, then the stationary point  $d^*$  is a maximizer of  $f(d)$ , i.e.,  $f(d)$  is concave. Let  $u = e^{-1-\beta_4}$  denote the argument of the Lambert W function in equation (D.21). Note that  $u$  is non-negative for  $\beta_4 \in \mathbb{R}$ . From the definition of  $W(\cdot)$  in equation (D.20), since the right-hand side ( $u$ ) and the exponential term in the left-hand side ( $e^{W(u)}$ ) are non-negative, the first-term in the left-hand side ( $W(u)$ ) must also be non-negative. Consequently, the numerator  $[1 + W(u)]^3$  is non-negative. Finally, the second derivative in equation (D.21) is negative if  $\beta_3 \cdot \beta_5 > 0$ .

### D.3 Data for Example 1

The parameters for the stochastic production planning model (see Subsection 6.4.4) are given in the following tables. Example 1 considers only a single site; therefore, in all tables below,  $s = \text{S1}$ . Inter-site transfer limits and costs ( $F_{j,s,s',t}^U$  and  $\eta_{j,s,s',t}$ ) are zero. All scenarios are assumed to have the same probability, i.e.,  $\pi_k = 0.1$ ,  $k = 1, \dots, 10$ .

Table D.1: Deterministic spot market price ( $\alpha_{j,s,t}^{\text{spot}}$ ) [\$/t].

Raw Material	Time Period					
	1	2	3	4	5	6
A	2.91	2.90	4.44	2.83	4.38	5.45

Table D.2: Operating cost ( $\delta_{i,s,t}$ ) [\$/t].

Process	Time Period					
	1	2	3	4	5	6
P1	0.26	0.26	0.26	0.14	0.14	0.14
P2	0.16	0.16	0.16	0.13	0.13	0.13
P3	0.32	0.32	0.32	0.32	0.32	0.32

Table D.3: Inventory cost ( $\xi_{j,s,t}$ ) [\$/t].

Product	Time Period
	1 – 6
D	0.03

Table D.4: Mass factor ( $\mu_{i,j,s}$ ) [-].

Process	Material			
	A	B	C	D
P1	0.83	-1.00		
P2		0.95	-1.00	
P3		1.11		-1.00

Table D.5: Production capacity ( $Q_{i,s,t}$ ) [t].

Process	Time Period
	1 – 6
P1	85
P2	35
P3	65

Table D.6: Lower (Upper) raw material availability ( $a_{j,s,t}^L$  and  $a_{j,s,t}^U$ ) [t].

Raw Material	Time Period
	1 – 6
A	0 (60)

Table D.7: Upper bound on inventory ( $V_{j,s,t}^U$ ) [t].

Product	Time Period
	1 – 6
D	6

Table D.8: Purchase price under discount ('d') contract type ( $\psi_{j,s,t}^{d,cs}$  for  $j = A$ ) [\$/t].

Time Period	Scheme (cs)	Price
1 – 6	1	3.15
1 – 6	2	2.47

Table D.9: Purchase threshold under discount ('d') contract type ( $\sigma_{j,s,t}^d$  for  $j = A$ ) [t].

Time Period	Threshold
1 – 6	20

Table D.10: Purchase price under bulk discount ('b') contract type ( $\psi_{j,s,t}^{b,cs}$  for  $j = A$ ) [\$ / t].

Time Period	Scheme (cs)	Price
1 – 6	1	3.06
1 – 6	2	2.38

Table D.11: Purchase threshold under bulk discount ('b') contract type ( $\sigma_{j,s,t}^b$  for  $j = A$ ) [t].

Time Period	Threshold
1 – 6	40

Table D.12: Purchase price under fixed duration ('l') contract type ( $\psi_{j,s,t}^{lp}$  for  $j = A$ ) [\$ / t].

Length (p)	Time Period
	1 – 6
1	3.40
2	2.04
3	1.70

Table D.13: Purchase threshold under fixed duration ('l') contract type ( $\sigma_{j,s,t}^{lp}$  for  $j = A$ ) [t].

Length (p)	Time Period	Threshold
1	1 – 6	5
2	1 – 6	25
3	1 – 6	30

Table D.14: Purchase spot price for the deterministic base case ( $\alpha_{j,s,t}^{\text{spot}}$  for  $j = A$ ) [\$ / t].

Time Period					
1	2	3	4	5	6
2.90	2.91	4.44	2.83	4.38	5.45

Table D.15: Sales spot price for the deterministic base case ( $\psi_{j,t}^{\text{spot}}$ ) [\$/t].

Product	Time Period					
	1	2	3	4	5	6
C	2.65	2.87	2.93	2.72	3.35	2.83
D	2.46	2.58	2.94	2.54	2.74	2.99

Table D.16: Stochastic spot market price ( $\alpha_{j,s,t,k}^{\text{spot}}$  for  $j = A$ ) [\$/t]. Simulated values from an ARIMA(0,1,1)(0,1,0)<sub>12</sub> time series model.

Time Period	Scenario									
	1	2	3	4	5	6	7	8	9	10
1	1.37	2.47	4.98	2.57	3.36	1.44	1.96	2.56	6.12	2.27
2	4.02	2.96	4.50	3.20	2.62	0.04	3.28	0.93	3.76	3.83
3	5.97	2.60	4.25	4.93	5.99	0.68	2.41	2.49	7.78	7.26
4	1.01	3.00	3.49	3.43	4.12	1.91	2.11	1.49	4.20	3.51
5	5.55	3.02	4.29	2.96	4.63	2.61	2.52	2.21	7.76	8.23
6	5.49	2.46	8.84	5.16	4.19	5.13	1.29	2.68	10.83	8.47

## Appendix E Future Work: Data-Driven Chance Constraints

In this appendix, we provide an initial sketch for extending the kernel-based reformulation method for data-driven chance constraints (CCs) proposed in Chapter 4.

We use the Method of Transformations (Casella & Berger, 2002) in order to model CCs where the uncertain parameters may not only appear in the right-hand side (RHS). The basic idea of the method can be summarized as follows. Suppose that we have a random variable (r.v.)  $X$ , taking values in a set  $S$ , and a function  $r$  from  $S$  into another set  $T$ . Then  $Y = r(X)$  is a new r.v. taking values in  $T$ . Considering that the distribution of  $X$  is known, we want to find the distribution of  $Y$  (see also <http://www.math.uah.edu/stat/>, Exploratory Material 2 and Basic Topic 7, for a very good summary of the topic).

### E.1 Linear Chance Constraints

#### E.1.1 Scalar Random Variables

Let the original CC be  $\mathbb{P}_{\tilde{a}} \{\tilde{a}x \leq c\} \geq 1 - \alpha$ , where  $\tilde{a}$  is a continuous r.v.,  $x$  is a continuous decision variable (usually, non-negative),  $c$  is a deterministic parameter, and  $\alpha$  is the risk level. Suppose that  $\tilde{a}$  has probability density function (PDF)  $f(a)$  and cumulative distribution function (CDF)  $F(a)$ . In this case, we can apply a *linear transformation* by treating  $x$  as a “coefficient” of the r.v.  $\tilde{a}$ . Introduce the r.v.  $\tilde{b} = \tilde{a}x$  with PDF  $g(b)$  and CDF  $G(b)$ . Therefore,

Table D.17: Stochastic part of demand-response models ( $\epsilon_k$ ) [\$/t].

Product	Scenario									
	1	2	3	4	5	6	7	8	9	10
C	-0.18	-0.17	-1.37	-0.17	0.85	0.70	0.55	-0.40	-0.20	-1.19
D	-0.11	0.51	3.41	2.00	-0.99	0.71	-2.27	1.76	1.95	4.24

the CC becomes  $\mathbb{P}_{\tilde{b}} \{\tilde{b} \leq c\} \geq 1 - \alpha$ , which seems more tractable than the original CC, since the new r.v. is not involved in nonlinear terms with decision variables. To reformulate this new CC, we need to ultimately invoke the CDF of  $\tilde{b}$ . By applying the Change of Variables formula, it can be shown that

$$g(b) = \frac{1}{|x|} f\left(\frac{b}{x}\right) \quad (\text{E.1})$$

$$G(b) = \frac{x}{|x|} F\left(\frac{b}{x}\right) = \frac{1}{|x|} xF\left(\frac{b}{x}\right) \quad (\text{E.2})$$

The term  $xF(b/x)$  corresponds to the *perspective* of the CDF of the original r.v.,  $\tilde{a}$ . The reformulation of the new CC is given as follows (assuming that the domain of all random variables (r.vs.) is the real line):

$$\begin{aligned} \mathbb{P}_{\tilde{b}} \{\tilde{b} \leq c\} &\geq 1 - \alpha \\ \int_{-\infty}^c g(b) db &\geq 1 - \alpha \\ \int_{-\infty}^c \frac{1}{|x|} f\left(\frac{b}{x}\right) db &\geq 1 - \alpha \\ \frac{1}{|x|} \int_{-\infty}^c f\left(\frac{b}{x}\right) db &\geq 1 - \alpha \\ \frac{1}{|x|} xF\left(\frac{c}{x}\right) &\geq 1 - \alpha \\ xF\left(\frac{c}{x}\right) &\geq |x|(1 - \alpha) \end{aligned}$$

The perspective of  $F(\cdot)$  poses some challenges. In addition to its nonlinearity, it is nondifferentiable at the point  $x = 0$ , which can cause numerical difficulties when solving the optimization problem. An approximation of the perspective of  $F(\cdot)$  that avoids this complication is given by ([Sawaya, 2006](#)):

$$xF\left(\frac{c}{x}\right) \approx [(1 - \epsilon)x + \epsilon] F\left(\frac{c}{(1 - \epsilon)x + \epsilon}\right) - \epsilon F(0)(1 - x)$$

Therefore, the reformulated CC is approximated as follows:

$$[(1 - \epsilon)x + \epsilon] F\left(\frac{c}{(1 - \epsilon)x + \epsilon}\right) - \epsilon F(0)(1 - x) \geq |x|(1 - \alpha) \quad (\text{E.3})$$

Note that, in situations where the original r.v.,  $\tilde{a}$ , is non-negative, we have that  $F(0) = 0$ , and the reformulation can be simplified as follows:

$$[(1 - \epsilon)x + \epsilon] F\left(\frac{c}{(1 - \epsilon)x + \epsilon}\right) \geq |x|(1 - \alpha) \quad (\text{E.4})$$

For data-driven CCs, the true distribution is replaced by its estimate, and the risk level is decreased as discussed in [Section 4.3](#). Thus, the reformulated data-driven versions of equations (E.3) and (E.4) are given by:

$$\begin{aligned} [(1 - \epsilon)x + \epsilon] \hat{F} \left( \frac{c}{(1 - \epsilon)x + \epsilon} \right) - \epsilon \hat{F}(0)(1 - x) &\geq |x|(1 - \alpha'_+) \\ [(1 - \epsilon)x + \epsilon] \hat{F} \left( \frac{c}{(1 - \epsilon)x + \epsilon} \right) &\geq |x|(1 - \alpha'_+) \end{aligned}$$

If the estimated distribution is obtained with kernel smoothing, the final forms become:

$$[(1 - \epsilon)x + \epsilon] \frac{1}{n} \sum_{i=1}^n \mathcal{K} \left( \frac{\psi - a_i}{h} \right) - \epsilon \frac{1}{n} \sum_{i=1}^n \mathcal{K} \left( \frac{-a_i}{h} \right) (1 - x) \geq |x|(1 - \alpha'_+) \quad (\text{E.5})$$

$$[(1 - \epsilon)x + \epsilon] \frac{1}{n} \sum_{i=1}^n \mathcal{K} \left( \frac{\psi - a_i}{h} \right) \geq |x|(1 - \alpha'_+) \quad (\text{E.6})$$

where  $n$  is the sample size, and  $\psi = \frac{c}{(1 - \epsilon)x + \epsilon}$ . Further modeling is necessary to avoid having the term  $|x|$  in the constraint (note:  $|x| = \sqrt{x^2}$ ).

### E.1.2 Random Vectors

Consider the following linear CC

$$\mathbb{P}_{\tilde{a}} \left\{ \tilde{a}^\top x \leq c \right\} \geq 1 - \alpha \quad (\text{E.7})$$

where  $\tilde{a}$  is a random vector,  $x$  is a vector of decision values, and  $c$  and  $\alpha$  are as defined before. Define a new random vector  $\tilde{b}$  as

$$\tilde{b} = \tilde{a}^\top x = \sum_{j=1}^m \tilde{a}_j x_j = \sum_{j=1}^m \tilde{b}_j \quad (\text{E.8})$$

where  $m$  is the number of elements in the respective vectors. Thus, the original CC becomes

$$\mathbb{P}_{\tilde{b}} \left\{ \tilde{b} \leq c \right\} \geq 1 - \alpha \quad (\text{E.9})$$

The main result of the Method of Transformations for this case (sum of r.vs.  $\tilde{b}_j$ , for  $j = 1, \dots, m$ ) is that the PDF of  $\tilde{b}$  is the convolution of the joint PDF for all  $\tilde{b}_j$ .

As an example, let us focus on the case where  $m = 2$ . Suppose the marginal PDF and CDF of  $\tilde{a}_j$  are  $f_j(\cdot)$  and  $F_j(\cdot)$ , and of  $\tilde{b}_j$  are  $g_j(\cdot)$  and  $G_j(\cdot)$ , for  $j = 1, 2$ , respectively. Consider the CC

$$\mathbb{P}_{\tilde{a}_1, \tilde{a}_2} \left\{ \tilde{a}_1 x_1 + \tilde{a}_2 x_2 \leq c \right\} \geq 1 - \alpha$$

which is rewritten as

$$\mathbb{P}_{\tilde{b}_1, \tilde{b}_2} \left\{ \tilde{b}_1 + \tilde{b}_2 \leq c \right\} \geq 1 - \alpha$$

The PDF of  $\tilde{b} = \tilde{b}_1 + \tilde{b}_2$ ,  $h(\cdot)$ , is given by the convolution of the joint PDF (change of variables:  $b_2 = b - b_1$ ),

$$h(b) = \int_{-\infty}^{\infty} g_{1,2}(b_1, b - b_1) db_1$$

and its CDF is given by

$$\begin{aligned} H(b) &= \int_{-\infty}^b h(b') db' \\ H(b) &= \int_{-\infty}^b \int_{-\infty}^{\infty} g_{1,2}(b_1, b' - b_1) db_1 db' \end{aligned}$$

If we assume that  $\tilde{b}_1$  and  $\tilde{b}_2$  are independent, then  $g_{1,2}(b_1, b_2) = g_1(b_1)g_2(b_2)$ , and from the previous subsection (equation (E.1), since  $\tilde{b}_j$  is the product of r.v.  $\tilde{a}$  and a “coefficient”  $x$ ),

$$g_j(b_j) = \frac{1}{|x_j|} f_j \left( \frac{b_j}{x_j} \right), \quad j = 1, 2$$

Therefore, the CDF of the random vector  $\tilde{b}$  with independent elements becomes

$$H(b) = \int_{-\infty}^b \int_{-\infty}^{\infty} \frac{1}{|x_1 x_2|} f_1 \left( \frac{b_1}{x_1} \right) f_2 \left( \frac{b' - b_1}{x_2} \right) db_1 db' \quad (\text{E.10})$$

The reformulation of the new CC (equation (E.9) with  $m = 2$  and assuming independent r.vs.) is given as follows:

$$\begin{aligned} \mathbb{P}_{\tilde{b}} \{ \tilde{b} \leq c \} &\geq 1 - \alpha \\ H(c) &\geq 1 - \alpha \\ \int_{-\infty}^c \int_{-\infty}^{\infty} \frac{1}{|x_1 x_2|} f_1 \left( \frac{b_1}{x_1} \right) f_2 \left( \frac{b - b_1}{x_2} \right) db_1 db &\geq 1 - \alpha \\ \frac{1}{|x_1 x_2|} \int_{-\infty}^{\infty} f_1 \left( \frac{b_1}{x_1} \right) \left[ \int_{-\infty}^c f_2 \left( \frac{b - b_1}{x_2} \right) db \right] db_1 &\geq 1 - \alpha \\ \frac{x_2}{|x_1 x_2|} \int_{-\infty}^{\infty} f_1 \left( \frac{b_1}{x_1} \right) F_2 \left( \frac{c - b_1}{x_2} \right) db_1 &\geq 1 - \alpha \end{aligned}$$

The integral represents the convolution between the PDF of  $\tilde{a}_1$  and the CDF of  $\tilde{a}_2$  with their respective arguments. For data-driven CCs, PDFs and CDFs are replaced by their estimates. At this point, it is not clear how tractable the final expression for the reformulated CC will be, even after including the formulas for kernel smoothers.