

Empirical Comparison of Robust, Data Driven and Stochastic Optimization

by

Wang, Yanbo

Bachelor of Science in Computational Finance (2007)
Bachelor of Business Administration (2007)
National University of Singapore

Submitted to the Computation for Design and Optimization
in partial fulfillment of the requirements for the degree of
Master of Science in Computation for Design and Optimization

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

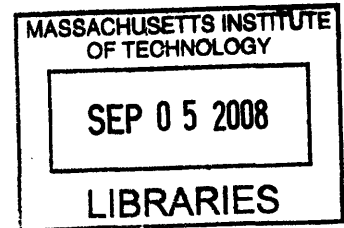
September 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author
Computation for Design and Optimization
August 11, 2008

Certified by
Bertsimas, Dimitris J
Boeing Professor of Operations Research
Co-director, Operations Research Center
Thesis Supervisor

Accepted by
Freund, Robert Michael
Theresa Seley Professor of Management Science
Co-director, Computation for Design and Optimization



Empirical Comparison of Robust, Data Driven and Stochastic Optimization

by

Wang, Yanbo

Submitted to the Computation for Design and Optimization
on August 11, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Computation for Design and Optimization

Abstract

In this thesis, we compare computationally four methods for solving optimization problems under uncertainty:

- Robust Optimization (RO)
- Adaptive Robust Optimization (ARO)
- Data Driven Optimization (DDO)
- Stochastic Programming (SP)

We have implemented several computation experiments to demonstrate the different performance of these methods. We conclude that ARO outperform RO, which has a comparable performance with DDO. SP has a comparable performance with RO when the assumed distribution is the same as the true underlying distribution, but underperforms RO when the assumed distribution is different from the true distribution.

Thesis Supervisor: Bertsimas, Dimitris J
Title: Boeing Professor of Operations Research
Co-director, Operations Research Center

Acknowledgments

I would like to thank my advisor Prof. Dimitris Bertsimas for guiding me through my master thesis process. He provided the potential topics that I can work on and helped me to overcome the difficulties when I stuck to some problems. I also want to thank Prof. Georgia Perakis, who taught me the linear and nonlinear optimization courses that provide the building blocks of my master research. I also would like to thank Prof. Jaime Peraire, Prof. Robert Freund and Ms. Laura Koller for the administration assistance. They are very responsive and responsible when I encountered problems. I also like to thank Raghavendran Sivaraman, Nelson Uhan and Shashi Mittal for setting up the software I needed for my master thesis. I also want to thank my peer students from CDO and ORC for their support. Everyone has tried their best to help me finalize the thesis and any remaining errors are mine.

Contents

1	Introduction	11
2	Performance Comparison of Stochastic Programming and Robust Optimization	13
2.1	A Powerplant Planning Problem	13
2.2	A Stochastic Programming Formulation	15
2.3	A Robust Optimization Formulation	16
2.4	Numerical Results	16
2.5	Conclusions	20
3	Performance Comparison of Data Driven and Robust Optimization	23
3.1	Metal Production Problem	23
3.2	Data Driven Optimization	25
3.2.1	Formulation 1	25
3.2.2	Formulation 2	27
3.3	Robust Optimization Formulation	29
3.4	Numerical Results	30
3.5	Conclusions	33
4	Performance Comparison of Adaptive, Robust and Data Driven Optimization	35
4.1	Inventory Management Problem with Lost Sales	35
4.2	A Robust Optimization Formulation	39

4.3	An Adaptive Robust Optimization Formulation	41
4.4	A Data Driven Optimization Formulation	44
4.5	Numerical Results	45
4.6	Conclusions	48

List of Tables

2.1	Distributions Used for Simulation in Chapter 2	18
2.2	The cost of SP using the assumed discrete distribution	18
2.3	The cost of RO using the discrete distribution	19
2.4	The cost of SP using normal distribution	19
2.5	The cost of RO using normal distribution	20
2.6	The cost of SP using lognormal distribution	20
2.7	The cost of RO using lognormal distribution	21
3.1	Distributions Used for Simulation in Chapter 3	30
4.1	Performance Comparison ($\Delta w = 20\%w, V_{max} = 2000$)	47
4.2	Performance Comparison ($\Delta w = 20\%w, V_{max} = 500$, including infeasible cases)	47
4.3	Performance Comparison ($\Delta w = 20\%w, V_{max} = 500$, excluding infeasible cases)	47

Chapter 1

Introduction

In recent years, there have been several methods proposed to address optimization problem under uncertainty. They are:

- Robust Optimization (RO)
- Adaptive Robust Optimization (ARO)
- Data Driven Optimization (DDO)
- Stochastic Programming (SP)

The goal of the thesis is to compare the performance of these methods in various settings. We next briefly review these methods.

RO has been introduced by Soyster [1] and has shown significant research activity in recent years. We refer readers to the paper by Ben Tal et.al. [3], Bertsimas and Sim [2].

ARO was introduced in Ben Tal et.al. [4] to address multistage problems. We refer the reader to the PhD thesis of Caramanis [6] for a review of the method.

DDO was introduced in Thiele [7] to address optimization problem under uncertainty when only prior data is known.

SP has a long history that started with the work of Dantzig [9].

The structure of the thesis is as follows. In Chapter 2, we compare SP and RO. In Chapter 3, we compare DDO and RO and in Chapter 4, we compare ARO, RO

and DDO.

Chapter 2

Performance Comparison of Stochastic Programming and Robust Optimization

The goal of this chapter is to compare the performance of Stochastic Programming (SP) and Robust Optimization (RO) in a power plant planning problem setting. Section 2.1 introduces the powerplant planning problem. Sections 2.2 and 2.3 provide SP and RO formulations respectively. Section 2.4 summarizes our computational results and Section 2.5 draws our conclusions.

2.1 A Powerplant Planning Problem

An electric utility company is installing two generators (indexed by $j = 1, 2$) with different fixed and operating costs, in order to meet the demand within its service region. Each day is divided into three parts of equal duration, indexed by $i = 1, 2, 3$. These correspond to parts of the day during which demand takes a base, medium, or peak value, respectively. The fixed cost per unit capacity of generator j is c_j . The operating cost of generator j during the i th part of the day is f_{ij} . If the demand during the i th part of the day cannot be served due to lack of capacity, additional capacity must be purchased at a cost of g_i . Finally, the capacity of each generator j

is required to be at least b_j .

There are two sources of uncertainty, namely, the exact value of the demand d_i during each part of the day, and the availability a_j of generator j . The demand d_i can take one of four values $d_{i,1}, \dots, d_{i,4}$, with probability $p_{i,1}, \dots, p_{i,4}$, respectively. The availability of generator 1 is $a_{1,1}, \dots, a_{1,4}$ with probability $q_{1,1}, \dots, q_{1,4}$ respectively. Similarly, the availability of generator 2 is $a_{2,1}, \dots, a_{2,4}$ with probability $q_{2,1}, \dots, q_{2,4}$, respectively. In summary, the input data is:

- j : index of the electricity generator
- i : index of the electricity supply period
- f_{ij} : variable cost of the generator j in period i
- g_i : capacity in period i
- b_j : minimum initial capacity of generator j
- c_j : unit cost to build the capacity of j at the beginning

To formulate the problem, we introduce the decision variables:

- x_j : installed capacity of generator j
- y_{ij} : operating level of generator j in period i
- y_i : capacity needed to purchase in period i to meet the demand

If the demand and the available capacity are fixed, the following formulation can solve the problem.

$$\begin{aligned}
\min \quad & \sum_{j=1}^2 c_j x_j + \sum_{j=1}^2 \sum_{i=1}^2 f_{ij} y_{ij} + \sum_{i=1}^3 g_i y_i \\
s.t \quad & x_j \geq b_j \\
& y_{ij} \leq a_j x_j \\
& \sum_{j=1}^2 y_{ij} + y_i \geq d_i \\
& x_j, y_{ij}, y_i \geq 0.
\end{aligned} \tag{2.1}$$

2.2 A Stochastic Programming Formulation

Under the demand uncertainty, we define the following decision variables:

- x_j : the initial capacity to build at the beginning of the plan
- y_{ij}^ω : operating level of generator j in period i given that scenario ω happens
- y_i^ω : capacity needed to purchase in period i to meet the demand given that scenario ω happens

This is a multiple-stage stochastic programming problem. The first stage decision variables are x_j , and y_{ij}^ω and y_i^ω are the decisions in the later stages.

$$\begin{aligned}
\min \quad & \sum_{j=1}^2 c_j x_j + E \left[\sum_{j=1}^2 \sum_{i=1}^2 f_{ij} y_{ij}^\omega + \sum_{i=1}^3 g_i y_i^\omega \right] \\
s.t \quad & x_j \geq b_j, \forall j \\
& y_{ij}^\omega \leq a_j^\omega x_j, \forall i, j, \omega \\
& \sum_{j=1}^2 y_{ij}^\omega + y_i^\omega \geq d_i^\omega, \forall i, j, \omega \\
& x_j, y_{ij}^\omega, y_i^\omega \geq 0, \forall i, j, \omega.
\end{aligned} \tag{2.2}$$

2.3 A Robust Optimization Formulation

The following additional input data is known:

- κ : parameter to control robustness level
- σ_{a_j} : the standard deviation of capacity availability of generator j , a_j
- σ_{d_i} : the standard deviation of demand in period i , d_i

The variable κ controls the robustness level. As κ increases, the robustness level increases. When $\kappa = 0$, RO is equivalent to the basic linear programming formulation (2.1). When κ becomes extremely large, the formulation may become infeasible because the robustness level is too high to be satisfied.

The robust formulation is as following:

$$\begin{aligned}
 \min \quad & \sum_{j=1}^2 c_j x_j + \sum_{j=1}^2 \sum_{i=1}^2 f_{ij} y_{ij} + \sum_{i=1}^3 g_i y_i \\
 \text{s.t.} \quad & x_j \geq b_j \\
 & y_{ij} \leq a_j x_j - \kappa \sigma_{a_j} x_j \\
 & \sum_{j=1}^2 y_{ij} + y_i \geq d_i + \kappa \sigma_{d_i} \\
 & x_j, y_{ij}, y_i \geq 0.
 \end{aligned} \tag{2.3}$$

2.4 Numerical Results

For the computational result, the values of parameters are assigned as follows:

- $c_1 = 4, c_2 = 2.5$
- $f_{11} = 4.3, f_{21} = 2, f_{31} = 0.5$
 $f_{12} = 8.7, f_{22} = 4, f_{32} = 1$

- $g_1 = g_2 = g_3 = 10$
- $b_1 = b_2 = 1000$
- $d_{i,1} = 900, d_{i,2} = 1000, d_{i,3} = 1100, d_{i,4} = 1200$
- $p_{i,1} = 0.15, p_{i,2} = 0.45, p_{i,3} = 0.25, p_{i,t} = 0.15$
- $a_{1,1} = 1, a_{1,2} = 0.9, a_{1,3} = 0.3, a_{1,4} = 0.1$
- $q_{1,1} = 0.2, q_{1,2} = 0.3, q_{1,3} = 0.4, q_{1,t} = 0.1$
- $a_{2,1} = 1, a_{2,2} = 0.9, a_{2,3} = 0.7, a_{2,4} = 0.1, a_{2,5} = 0$
- $q_{2,1} = 0.1, q_{2,2} = 0.2, q_{2,3} = 0.5, q_{2,t} = 0.1, q_{2,5} = 0.1$

We next outline the simulation process:

- In order to solve (2.2), we generate scenarios according to the discrete distribution outlined in Table 2.1.
- We solve the RO formulation (2.3) for $\kappa = 0, \dots, 2$ with step size of 0.025 using only the mean and the standard deviation of the discrete distribution.
- In order to assess the performance of solutions obtained in (2.2) and (2.3), we generate random scenarios from three distinct distributions as in Table 2.1.
- We simulate the solutions from (2.2) and (2.3) assuming the various distributions in Table 2.1 and report the total cost. We use 1000 samples.

Table 2.2 reports on the distribution of cost for stochastic programming when the real distribution is the same as the distribution assumed: the average cost, standard deviation and 90, 80 and 50 percentiles. Table 2.3 reports on the distribution of cost for robust optimization given the assumed discrete distribution. As observed, the average costs of robust optimization fluctuate without clear trend when the κ value is very small. When the κ value becomes larger, the trend becomes stable: when κ is too small or too big, RO underperforms SP; when $\kappa = 1$, both methods obtain the

Table 2.1: Distributions Used for Simulation in Chapter 2

Distribution Name	Distribution Parameters
Discrete Distribution d_1 and d_2 a_1 a_2	$d_{i,1} = 900, d_{i,2} = 1000, d_{i,3} = 1100, d_{i,4} = 1200$ $p_{i,1} = 0.15, p_{i,2} = 0.45, p_{i,3} = 0.25, p_{i,t} = 0.15$ $a_{1,1} = 1, a_{1,2} = 0.9, a_{1,3} = 0.3, a_{1,4} = 0.1$ $q_{1,1} = 0.2, q_{1,2} = 0.3, q_{1,3} = 0.4, q_{1,t} = 0.1$ $a_{2,1} = 1, a_{2,2} = 0.9, a_{2,3} = 0.7, a_{2,4} = 0.1, a_{2,5} = 0$ $q_{2,1} = 0.1, q_{2,2} = 0.2, q_{1,3} = 0.5, q_{1,t} = 0.1, q_{2,5} = 0.1$
Normal Distribution d_1 and d_2 a_1 a_2	same mean and standard deviation as in Discrete Distribution same mean and standard deviation as in Discrete Distribution same mean and standard deviation as in Discrete Distribution
Lognormal Distribution d_1 and d_2 a_1 a_2	same mean and standard deviation as in Discrete Distribution same mean and standard deviation as in Discrete Distribution same mean and standard deviation as in Discrete Distribution

same average cost and SP obtains a lower standard deviation, RO yields lower 90, 80 and 50 percentiles.

Table 2.2: The cost of SP using the assumed discrete distribution

	Average Cost	Standard Deviation	90%	80%	50%
	23071	27819	20783	19413	18550

Tables 2.4 and 2.5 list the numerical performance of SP and RO when the real distribution is different from what has been assumed in the model. We use a normal distribution in the simulation process. The mean and deviation of the distribution used in the numerical simulation are the same as the discrete distribution in the model. When $\kappa = 1$, RO yields a better objective value than SP. The appropriate κ may be obtained by simulating with different κ values and choosing the best performing κ in the real application. Moreover, RO also obtains lower standard deviation and lower 90, 80, 50 percentiles. Therefore, RO outperforms SP across the board.

Table 2.3: The cost of RO using the discrete distribution

Kappa	Average Cost	Standard Deviation	90%	80%	50%
0	30026	39244	63365	18411	16778
0.125	30316	39903	69783	18468	17098
0.25	27890	32872	57868	18531	16966
0.375	35980	49730	102620	19490	16750
0.5	31626	46755	73683	18187	15686
0.625	33730	45205	99102	19657	16061
0.75	27291	31697	63818	18401	16706
0.875	23378	27333	23629	18535	17165
1	23118	28747	20063	18693	17469
1.125	25436	32451	20253	19527	18847
1.25	26748	33608	21891	21211	20531
1.375	27491	27401	25863	23312	22632
1.5	32739	34136	29466	26691	25331
1.625	31897	19086	30283	29603	28923
1.75	35970	12007	35298	34618	33938
1.875	42690	6873	42795	42115	41435
2	54183	632	55217	54537	53857

Table 2.4: The cost of SP using normal distribution

	Average Cost	Standard Deviation	90%	80%	50%
	22484	22251	21662	19622	18958

Tables 2.6 and 2.7 list the numerical performance of stochastic programming and robust optimization when the real distribution is different from what has been assumed in the model and, in particular, asymmetric. We assume now that the distribution used is lognormal. The mean and deviation of the distribution used in the numerical simulation are the same as the discrete distribution in the model. When $\kappa = 1$, RO yields a better objective value than SP. The appropriate κ may be obtained by simulating with different κ value and choosing the best performing κ in the real application. Although SP leads to a lower standard deviation, RO yields lower 90, 80 and 50 percentiles. Therefore, RO outperforms SP in terms of mean value, but the performance in terms of standard deviation is worse, which may be caused by the fact that the uncertainty set in RO is symmetric and cannot capture the asymmetry

Table 2.5: The cost of RO using normal distribution

Kappa	Average Cost	Standard Deviation	90%	80%	50%
0	24318	30551	27304	17761	16234
0.125	22607	27582	20825	17987	16540
0.25	22366	23954	20942	17812	16861
0.375	34181	46176	93051	19963	16128
0.5	30991	40208	81570	18902	15938
0.625	22751	25336	31151	17588	15711
0.75	23247	25061	27755	18010	16237
0.875	21050	22845	19739	17935	16898
1	21040	17437	20436	18489	17906
1.125	21701	15652	21535	19804	19239
1.25	22599	11970	22172	21408	20882
1.375	24811	14681	24093	23499	22971
1.5	27269	15679	26482	26067	25584
1.625	30115	11860	30042	29637	29202
1.75	34098	1421	35064	34714	34264
1.875	41890	6019	42629	42188	41719
2	54203	5971	54951	54547	54134

property of the distribution as well.

Table 2.6: The cost of SP using lognormal distribution

Average Cost	Standard Deviation	90%	80%	50%
22766	15613	22375	20233	18822

In summary, robust optimization has two advantages over stochastic programming: firstly, robust optimization is more computationally economical when the distribution of uncertain parameters are complicated; secondly, robust optimization can outperform stochastic programming given an appropriate robustness level.

2.5 Conclusions

- In the application given in this section, Robust Optimization can outperform stochastic programming by choosing an appropriate robustness level.

Table 2.7: The cost of RO using lognormal distribution

Kappa	Average Cost	Standard Deviation	90%	80%	50%
0	27567	26648	65302	19414	17077
0.125	27834	27389	64385	19306	17197
0.25	25990	23763	49738	19461	17403
0.375	47851	46511	121565	76078	17760
0.5	42278	42787	110613	56149	17827
0.625	32504	30932	86626	34004	17189
0.75	29239	31165	63038	19740	17056
0.875	27448	27952	62476	19471	17318
1	22497	18170	20962	19397	17741
1.125	22107	11912	21837	20450	19120
1.25	23668	12760	23023	20804	20803
1.375	23867	8298	23565	22905	22904
1.5	25962	5302	25604	25603	25603
1.625	29545	5190	29195	29195	29194
1.75	34216	81	34211	34210	34210
1.875	41707	17	41707	41707	41707
2	54129	1	54130	54129	54129

- When the assumed distribution is accurate, stochastic programming can reach the same level of optimality as robust optimization.
- When the assumed distribution is different from the real distribution, robust optimization can outperform the stochastic optimization by choosing an appropriate level of robustness.
- Even if the assumed discrete distribution is correct, the computational complexity is too high to afford for stochastic programming if there are too many scenarios. However, robust optimization has the same computational complexity as before; thus, in this case, robust optimization is a more economical method computationally.

Chapter 3

Performance Comparison of Data Driven and Robust Optimization

This chapter compares the performance of Data Driven Optimization (DDO) and Robust Optimization (RO) in a metal production problem setting.

In Section 3.1, we introduce the metal production problem. In Section 3.2, we formulate two DDO methods. Section 3.3 formulates the RO approach. Section 3.4 lists and discusses the numerical simulation results. In Section 3.5, we draw our conclusions.

3.1 Metal Production Problem

In this section, we introduce a metal production problem. Suppose there are three metals to be produced and sold. The raw materials are ores. There are three types of ores available in the market place to produce the metals. Different ores contain different amount of each type of metals per unit weight. The objective is to minimize the production cost (the total purchase cost of ores).

The decision variables are the purchase quantity x_j of ore j , $j = 1, \dots, 3$. Let $x = (x_1, x_2, x_3)'$.

The model inputs are:

- j : index of the ores, $j=1,2,3$

- i : index of output metals, $i=1,2,3$
- c_j : unit price of ore j , $j=1,2,3$. Let $c = (c_1, c_2, c_3)'$.
- b_i : the demand for output metal i . Let $b = (b_1, b_2, b_3)'$.
- $A_{i,j}$: The net content of metal i in ore j . For example, $A_{i,j}$ means one unit of ore j contains $A_{i,j}$ of metal i . The unit of ore is assumed to be a million ton and the unit of metal is a thousand tons. Let $A = \begin{pmatrix} A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix}$.
- d_j : the maximum purchase quantity of ore j . Let $d = (d_1, d_2, d_3)'$

When all uncertain data is known, we can formulate the problem as a linear programming problem:

$$\begin{aligned}
 \min \quad & c'x \\
 \text{s.t.} \quad & Ax \geq b \\
 & 0 \leq x \leq d.
 \end{aligned} \tag{3.1}$$

The prices of the ores change over time due to uncertain market conditions. For example, in 2008, the iron ore price quoted by BHP Billiton (the market leader in the industry) increased by 79.88%. Therefore, the price vector c is subject to uncertainty. The exact amount of materials used to produce each unit is subject to some deviation because the net content of each metal in each ore is not exactly fixed; therefore, A is uncertain. The demands, b , are also uncertain and they are effected by economic condtions. We model uncertainty by considering the range of the deviation. Specially, we assume that:

$$A \in [\bar{A} - \Delta A, \bar{A} + \Delta A]$$

$$b \in [\bar{b} - \Delta b, \bar{b} + \Delta b]$$

$$c \in [\bar{c} - \Delta c, \bar{c} + \Delta c],$$

where, $\bar{A}, \bar{b}, \bar{c}$ are the average value of parameters A, b, c .

3.2 Data Driven Optimization

3.2.1 Formulation 1

One way to tackle the uncertainty in the problem is to obtain a wide range of the scenarios of (A, b, c) . Let the sample scenarios be $\{(A_1, b_1, c_1), (A_2, b_2, c_2), \dots, (A_N, b_N, c_N)\}$, where $A_k \geq 0, k = 1, \dots, N$. The scenario (A_i, b_i, c_i) is generated uniformly from the boxes:

$$A \in [\bar{A} - \Delta A, \bar{A} + \Delta A]$$

$$b \in [\bar{b} - \Delta b, \bar{b} + \Delta b]$$

$$c \in [\bar{c} - \Delta c, \bar{c} + \Delta c].$$

To ensure robustness, a large portion of the scenarios has to be feasible. In particular, we require a portion p of the constraints $A_k x \geq b_k, k = 1, \dots, N$ to be satisfied. We introduce binary variables $z_k, k = 1, \dots, N$:

$$z_k = \begin{cases} 1, & k \text{ scenario is feasible,} \\ 0, & \text{otherwise,} \end{cases} \quad (3.2)$$

and model the requirement that a portion p of the scenarios are feasible:

$$A_k x \geq b \cdot z_k$$

$$\sum_{k=1}^N z_k \geq \lfloor p \cdot N \rfloor.$$

In order to model the uncertainty in the objective function coefficients, we argue as follows: A given solution x gives rise to N costs $c'_1 x, \dots, c'_N x$. Let us order these

costs as $c'_{(1)}x \geq \dots \geq c'_{(N)}x$. We want to obtain a solution x that minimizes the average cost over the $\lfloor \alpha N \rfloor$ largest cost $c'_i x$, $i = 1, \dots, N$, i.e.,

$$\min_x \frac{1}{\lfloor \alpha N \rfloor} \sum_{i=1}^{\lfloor \alpha N \rfloor} c'_{(i)} x$$

where $c'_{(i)} x$, $i = 1, \dots, \lfloor \alpha N \rfloor$ is the i -th largest cost among $c'_i x$, $i = 1, \dots, N$.

We then have:

$$\begin{aligned} \min_x \quad & \frac{1}{\lfloor \alpha N \rfloor} \sum_{i=1}^{\lfloor \alpha N \rfloor} c'_{(i)} x \\ \text{s.t} \quad & A_k x \geq b_k z_k, k = 1, \dots, N \\ & \sum_{k=1}^N z_k \geq \lfloor p \cdot N \rfloor \\ & 0 \leq x \leq d \\ & z_k \in \{0, 1\}. \end{aligned} \tag{3.3}$$

The sum $\sum_{i=1}^{\lfloor \alpha N \rfloor} c'_{(i)} x$ can be written as the linear programming problem:

$$\begin{aligned} \max \quad & \sum_{k=1}^{\lfloor \alpha N \rfloor} (c'_k x) \cdot y_k \\ \text{s.t} \quad & \sum_{k=1}^N y_k \leq \lfloor \alpha N \rfloor \\ & 0 \leq y_k \leq 1. \end{aligned} \tag{3.4}$$

The dual problem of formulation (3.4) is:

$$\begin{aligned}
\min_{\phi, \theta} \quad & \left\{ \lfloor \alpha N \rfloor \cdot \phi + \sum_{k=1}^N \theta_k \right\} \\
s.t. \quad & \phi + \theta_k \geq c'_k x, k = 1, \dots, N \\
& \phi \geq 0 \\
& \theta_k \geq 0.
\end{aligned} \tag{3.5}$$

Based on strong duality, formulation (3.4) and formulation (3.5) have the same optimal objective value. Substituting (3.5) into formulation (3.3), we obtain that formulation (3.3) is equivalent to

$$\begin{aligned}
\min_{x, \phi, \theta} \quad & \frac{1}{\lfloor \alpha N \rfloor} \left\{ \lfloor \alpha N \rfloor \cdot \phi + \sum_{k=1}^N \theta_k \right\} \\
s.t. \quad & \lfloor \alpha N \rfloor \cdot \phi + \theta_k \geq c'_k x, k = 1, \dots, N \\
& A_k x \geq z_k \cdot b_k \\
& \sum_{k=1}^N z_k \geq \lfloor p \cdot N \rfloor \\
& 0 \leq x \leq d \\
& \phi \geq 0 \\
& \theta_k \geq 0.
\end{aligned} \tag{3.6}$$

3.2.2 Formulation 2

We consider the same linear programming problem and we generate the random scenarios in the same way as in Section 3.2.1. Instead of requiring a proportion p of the constraints $A_k x \geq b_k$ to be satisfied, we penalize the violation of these constraints as follows:

Let $y_k = \max(b_k - A_k x, 0)$ where the max operation is considered componentwise. If $y_{kj} \geq 0$, it means that the j -th constraint of the scenario k is violated by y_{kj} . Let w be a vector of penalties associated with violations y_k . Then $w'y_k$ is the violation

penalty given a solution x on scenario k .

Let $w'y_{(1)} \geq \dots \geq w'y_{(N)}$. We impose a penalty $\frac{1}{\lfloor Np \rfloor} \sum_{i=1}^{\lfloor Np \rfloor} w'y_{(i)}$. Our objective in this formulation is to

$$\begin{aligned}
\min_x \quad & \frac{1}{\lfloor N \cdot \alpha \rfloor} \sum_{i=1}^{\lfloor N \cdot \alpha \rfloor} c(i)'x + \frac{1}{\lfloor N \cdot p \rfloor} \sum_{i=1}^{\lfloor N \cdot p \rfloor} w'y_{(i)} \\
s.t \quad & y = \max(b_i - A_i \cdot x, 0) \\
& 0 \leq x \leq d.
\end{aligned} \tag{3.7}$$

Arguing similarly as in the previous section, we arrive at the equivalent reformulation (3.8):

$$\begin{aligned}
\min_{x,y,\phi,\theta} \quad & \frac{1}{\lfloor \alpha N \rfloor} \left\{ \lfloor \alpha N \rfloor \cdot \phi_1 + \sum_{k=1}^N \theta_{1,k} \right\} + \frac{1}{\lfloor p \cdot N \rfloor} \left\{ \lfloor p \cdot N \rfloor \cdot \phi_2 + \sum_{k=1}^N \theta_{2,k} \right\} \\
s.t. \quad & \lfloor \alpha N \rfloor \cdot \phi_1 + \theta_{1,k} \geq c'_k x, k = 1, \dots, N \\
& \lfloor p \cdot N \rfloor \cdot \phi_2 + \theta_{2,k} \geq w'y_k, k = 1, \dots, N \\
& A_k x \geq z_k \cdot b_k \\
& \phi_1 \geq 0 \\
& \theta_{1,k} \geq 0 \\
& y_k \geq 0 \\
& y_k \geq b_k - A_k \cdot x \\
& \phi_2 \geq 0 \\
& \theta_{2,k} \geq 0 \\
& 0 \leq x \leq d.
\end{aligned} \tag{3.8}$$

3.3 Robust Optimization Formulation

In this section, we formulate the problem as RO with ellipsoidal uncertainty set. The problem can be rewritten as follows:

$$\begin{aligned}
 \min \quad & y \\
 \text{s.t.} \quad & 0 \leq x \leq d \\
 & \begin{pmatrix} -A & 0 & b \\ c' & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leq 0 \\
 & z = 1,
 \end{aligned} \tag{3.9}$$

where, y is the original objective value and z is a dummy variable which is forced to be 1.

In this formulation, all the uncertainty arises from the matrix $\begin{pmatrix} -A & 0 & b \\ c & -1 & 0 \end{pmatrix}$. We define a control parameter Γ to control the robustness level. Let the standard deviation of A_{ij} be $\sigma_{A_{ij}}$, the standard deviation of $b_i, i = 1, 2, 3$ be σ_{b_i} and the standard deviation of $c_j, j = 1, 2, 3$ be σ_{c_j} . We compute the standard deviations from the uniform distribution assumed in Section 3.1. For example, $\sigma_{A_{i,j}} = \frac{\Delta A_{i,j}}{\sqrt{3}}$. Following [3] the robust formulation is:

$$\begin{aligned}
 \min \quad & y \\
 \text{s.t.} \quad & -\sum_{j=1}^3 A_{ij}x_j + b_jz + \Gamma \sqrt{\sum_{j=1}^3 \sigma_{A_{ij}}^2 x_j^2 + \sigma_{b_j}^2 z^2} \leq 0, i = 1, \dots, 3 \\
 & \sum_{j=1}^3 c_j x_j - y + \Gamma \sqrt{\sum_{j=1}^3 \sigma_{c_j}^2 x_j^2} \leq 0 \\
 & z = 1 \\
 & 0 \leq x \leq d
 \end{aligned} \tag{3.10}$$

3.4 Numerical Results

Let $\bar{A} = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 1 & 2 \\ 1 & 3 & 3 \end{pmatrix}$, $\bar{b} = \begin{pmatrix} 2 \\ 2 \\ 2 \end{pmatrix}$, and $\bar{c} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$, and penalty vector $w = \begin{pmatrix} 0.8333 \\ 0.8333 \\ 0.8333 \end{pmatrix}$.

Let $\Delta A = 0.25\bar{A}$, $\Delta b = 0.25\bar{b}$ and $\Delta C = 0.25\bar{C}$. The procedure for the numerical experiment is as follows.

- Step 1: 100 random scenarios are used as the data. We find the optimal solutions for Formulation 1, Formulation 2 and RO formulation.
- Step 2: We generate 1000 random scenarios according to distributions listed in Table 3.1.
- Step 3: We apply the optimal solutions from Step 1 to the random sample generated in Step 2. We report the total production costs.

Table 3.1: Distributions Used for Simulation in Chapter 3

Distribution Name	Distribution Parameters
Uniform Distribution	
A	$\mu_{A_{i,j}} = \bar{A}_{i,j}, \sigma_{A_{i,j}} = \frac{\Delta A_{i,j}}{\sqrt{3}}$
b	$\mu_{b_i} = \bar{b}_i, \sigma_{b_i} = \frac{\Delta b_i}{\sqrt{3}}$
c	$\mu_{c_j} = \bar{c}_j, \sigma_{c_j} = \frac{\Delta c_j}{\sqrt{3}}$
Normal Distribution	
A	$\mu_{A_{i,j}} = \bar{A}_{i,j}, \sigma_{A_{i,j}} = \frac{\Delta A_{i,j}}{\sqrt{3}}$
b	$\mu_{b_i} = \bar{b}_i, \sigma_{b_i} = \frac{\Delta b_i}{\sqrt{3}}$
c	$\mu_{c_j} = \bar{c}_j, \sigma_{c_j} = \frac{\Delta c_j}{\sqrt{3}}$
Lognormal Distribution	
A	$\mu_{A_{i,j}} = \bar{A}_{i,j}, \sigma_{A_{i,j}} = \frac{\Delta A_{i,j}}{\sqrt{3}}$
b	$\mu_{b_i} = \bar{b}_i, \sigma_{b_i} = \frac{\Delta b_i}{\sqrt{3}}$
c	$\mu_{c_j} = \bar{c}_j, \sigma_{c_j} = \frac{\Delta c_j}{\sqrt{3}}$

For Formulations 1 and 2, we apply the optimal solution with various p and α to the random scenarios. We test $0 \leq p \leq 1$ and $0 \leq \alpha \leq 1$ with step size 0.1. For RO, we apply the optimal solution with various Γ to the random scenarios. We test

$0 \leq \Gamma \leq 5$ with step size 0.05. The output for each solution contains two vectors: the average cost and the feasibility ratio. Feasibility is defined as the percentage of the scenarios which have feasible optimal solutions. The average cost is defined as the average cost given the optimization model is feasible. The trade off between these two values are plotted for comparison.

Figure 3-1 presents the performance when the assumed distribution is the same as the true underlying distribution. The figure shows that DDO and RO have similar performance in terms of feasibility and optimality trade off. Different methods seem to outperform others in some region but the scale of improvement is too small and it should be considered as random events caused by random scenarios.

Figure 3-1: Performance comparison of DDP and RO with uniform distribution as the true distribution

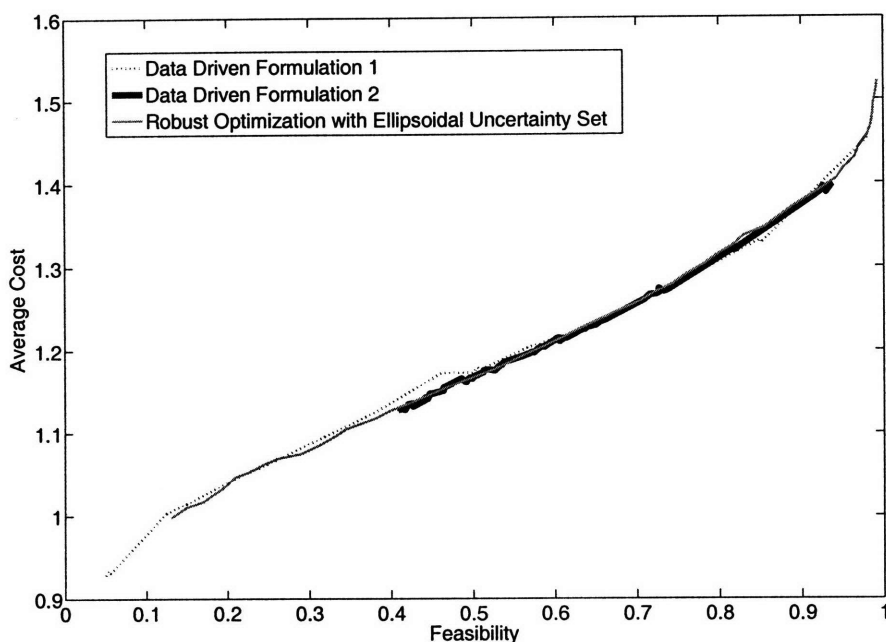


Figure 3-2 presents the performance when the assumed distribution is different from the true underlying distribution. The figure shows that DDO and RO have a similar performance in terms of feasibility and optimality trade off. However, one interesting observation is that RO can reach to a very high feasibility level even when the assumed distribution is wrong. DDO Formulation 1 can only reach about 75%

feasibility level even when p is 1 but RO can reach to above 80%. This is considered as an advantage of RO. The performance of RO is less dependent on the correctness of the assumed distribution while DDO is very dependent on the correctness of the assumed distribution. This advantage becomes more obvious in Figure 3-3 which presents the performance when the true distribution is lognormal instead of uniform distribution. DDO Formulation 1 can only ensure the feasibility of 60% and DDO Formulation 2 can only ensure the feasibility level of 55%. RO can realize the feasibility level of more than 90%. We provide one possible intuition of this observation. DDO can only incorporate the scenarios within the assumed distribution. If the assumed distribution is different from the true underlying distribution, DDO fails to capture the missing scenarios. In contrast, RO can capture the uncertainty even out of the assumed distribution by making the uncertainty set large enough.

Figure 3-2: Performance comparison of DDP and RO with normal distribution as the true Distribution

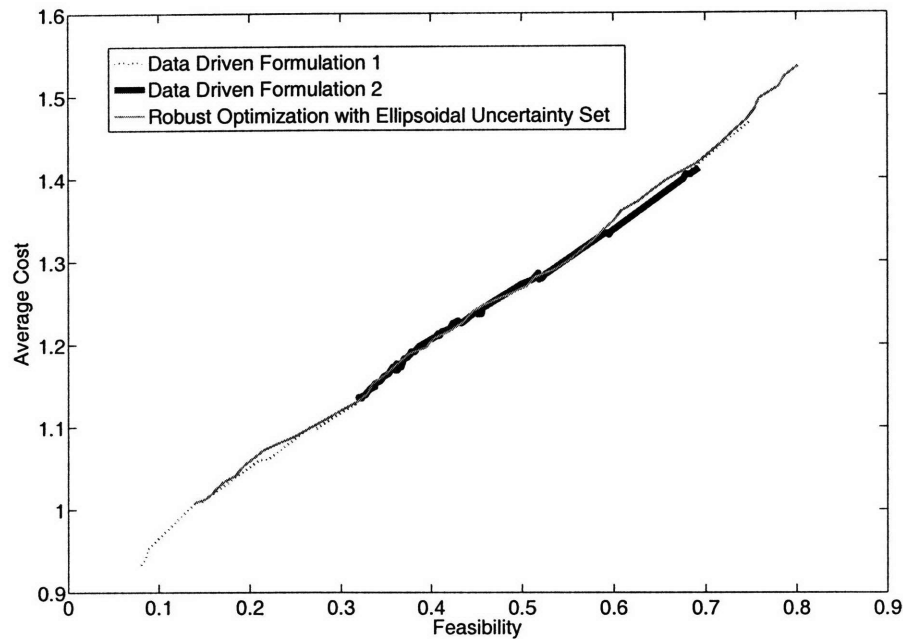
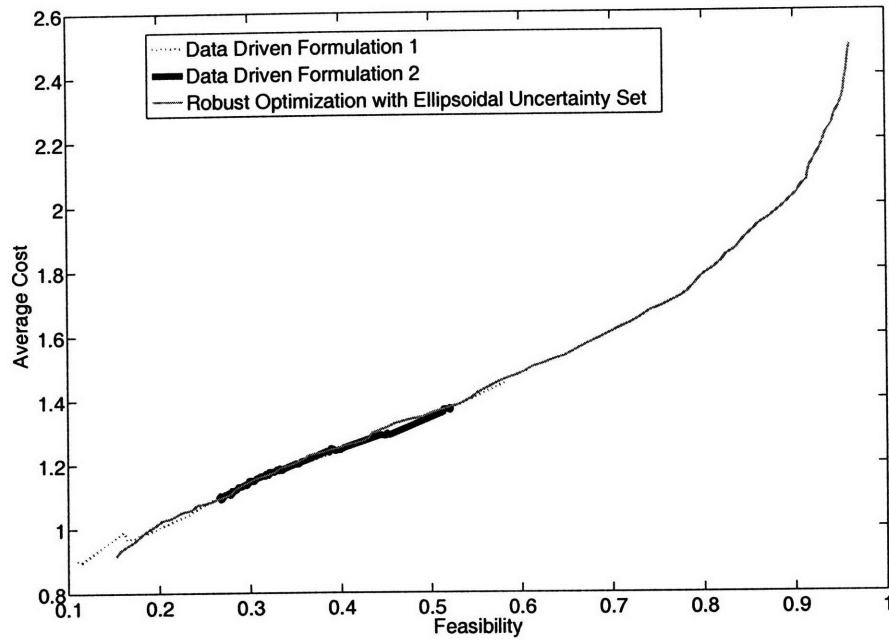


Figure 3-3: Performance comparison of DDP and RO with lognormal distribution as the true distribution



3.5 Conclusions

- The accuracy of the sampling points are crucial in the performance of DDO when we require a high feasibility level.
- RO can reach higher feasibility than DDO at a larger cost.
- DDO and RO have comparable performance in terms feasibility and cost trade off, when DDO can reach the required feasibility level.

Chapter 4

Performance Comparison of Adaptive, Robust and Data Driven Optimization

This chapter compares the performance of ARO, RO and DDO in an inventory management setting. In Section 4.1, we formulate an inventory problem with lost sales. Section 4.2 provides the RO formulation of the problem. In Section 4.3, we derive the ARO formulation. Section 4.4 formulates the DDO approach. We present the numerical results in Section 4.5 and summarize our conclusions in Section 4.6.

4.1 Inventory Management Problem with Lost Sales

Suppose that there is a product that is produced in three factories. The completed products from the three factories will be stored in a single warehouse until the products are sold out. The production costs for the factories differ. Some factories incur higher production costs than others. The production cost is also seasonal, meaning that the production cost is lower in some months and higher in other months. The differentiation of the production cost among different factories and across different seasons leads to the necessity to consider production capacity allocation. If the holding cost is zero and the demand is completely predictable, the optimal plan

should allocate all the production to the lower-cost season. However, we assume that a non-zero holding cost is incurred and the demand is uncertain. If the inventory in the warehouse cannot meet the real demand, the customer will go to other suppliers to purchase similar products instead of waiting for the production of the firm; in other words, the potential sales would be lost forever and can not be recovered. The objective of the problem is to minimize the inventory management cost. The cost includes three components: production cost, holding cost and lost sale cost. The three components are interrelated. The production cost is seasonal but the holding cost prevents the plan from producing the goods too far ahead of the demand due to the heavy holding cost. The company cannot produce too many goods because of the zero residual value of the extra production after the sales season. The company cannot produce too little because the lost sales would be considered an opportunity cost.

This section presents the basic linear programming formulation first

The model input parameters are defined as below:

- T : number of time periods
- I : number of factories
- $C_{i,k}$: production cost for time k and factory i
- H_k : holding cost per item at period k , when $k = 1, \dots, T$; when $k = T + 1$, H_{T+1} denotes the residual cost, the cost of the inventory left for disposal after the whole supply season.
- B_k : lost sale cost per item at period k
- \bar{w}_k : demand in period k
- Δw_k : the deviation of demand in period k

$$w_k \in [\bar{w}_k - \Delta w_k, \bar{w}_k + \Delta w_k]$$

- Q : total production capacity of each factory during the whole time horizon
- P : maximum production capacity of each factory
- V_{max} : maximum warehouse capacity

The decision variables in the model are as follows:

- $u_{i,k}$: production quantity for time k and factory i
- y_k : inventory level at the warehouse at the beginning of period k , where $y_1 = 0$ and

$$y_k = \max \left(0, y_{k-1} + \sum_{i=1}^I u_{i,k-1} - w_{k-1} \right), k = 2, \dots, T + 1$$

- z_k : lost sale quantity at the end of period k

$$z_k = \max \left(0, w_k - y_k - \sum_{i=1}^I u_{i,k} \right), k = 1, \dots, T$$

As introduced at the beginning of the section, the objective of the problem is to minimize the total cost, where the total cost is the sum of the production, holding and lost sales cost. If the demands are fixed, the problem can be formulated as follows:

$$\begin{aligned}
min \quad & \sum_{k=1}^T \sum_{i=1}^I C_{i,k} u_{i,k} + \sum_{k=1}^T H_{k+1} y_{k+1} + \sum_{k=1}^T B_k z_k \\
s.t \quad & 0 \leq u_{i,k} \leq P, i = 1, \dots, I; k = 1, \dots, T \\
& \sum_{k=1}^T u_{i,k} \leq Q, i = 1, \dots, I \\
& y_1 = 0 \\
& y_k + \sum_{i=1}^I u_{i,k} - w_k \leq V_{max}, k = 1, \dots, T \\
& -z_k + y_{k+1} = y_k + \sum_{i=1}^I u_{i,k} - w_k, k = 1, \dots, T \\
& z_k \geq 0, k = 1, \dots, T \\
& y_k \geq 0, k = 2, \dots, T + 1.
\end{aligned} \tag{4.1}$$

We name Formulation (4.1) as the linear optimization formulation (LO). Constraint $0 \leq u_{i,k} \leq P$ ensures the production quantity of each factory does not exceed the production capacity in each period. Constraint $\sum_{k=1}^T u_{i,k} \leq Q$ states that the aggregated production capacity in the whole sale season is bounded by the capacity limit Q . A possible reason for this constraint is that the sales manager has predicted the aggregated demand of the season and has placed the raw materials order according to this fixed forecast. Once the raw materials order is made, the total production capacity is bounded by the total amount of raw materials available for the whole season. Although it is economical to produce more in the periods with lower production cost given the holding cost is not very high, the capacity of the warehouse is limited by the total capacity of V_{max} , which is captured by constraints $y_k + \sum_{i=1}^I u_{i,k} - w_k \leq V_{max}, k = 1, \dots, T$. and $y_k \geq 0, k = 2, \dots, T + 1$. These two constraints ensure that $y_k = \max\left(0, y_{k-1} + \sum_{i=1}^I u_{i,k-1} - w_{k-1}\right), k = 2, \dots, T + 1$ and it is assumed that $y_1 = 0$. Here, y_{k+1} actually denotes the residual inventory. It is the inventory still left in the warehouse at the end of the sale season and it is going to be disposed at a cost, which is denoted by H_{k+1} . z_k is the lost

sale quantity in period k . It is equal to $\max\left(0, w_k - y_k - \sum_{i=1}^I u_{i,k}\right)$. Constraints $-z_k + y_{k+1} = y_k + \sum_{i=1}^I u_{i,k} - w_k$ and $z_k \geq 0$ capture this relationship. When $y_k + \sum_{i=1}^I u_{i,k} - w_k$ is positive $y_{k+1} = y_k + \sum_{i=1}^I u_{i,k} - w_k$; when $y_k + \sum_{i=1}^I u_{i,k} - w_k$ is negative $z_k = w_k - y_k - \sum_{i=1}^I u_{i,k}$. This claim is the natural consequence of the fact that z_k and y_{k+1} cannot be nonzero at the same time at the optimal solution. Suppose this happens, then the objective function value can be improved by decreasing both z_k and y_{k+1} at the same time to make the constraints still valid and making the objective value smaller.

4.2 A Robust Optimization Formulation

The robust formulation is based on the linear programming formulation and it incorporates the uncertainty set into the formulation. In this robust formulation, the polytope uncertainty set is used. Each demand parameter w_k falls into an interval $[\bar{w}_k - \Delta w_k, \bar{w}_k + \Delta w_k]$, where \bar{w}_k is the average value for the demand at period k and Δw_k is the vector of standard deviation of the demand forecast. These values can be calibrated from real data. The larger the values of Δw_k 's, the more robust the formulation against uncertainty. However, if the values of Δw_k 's are too large, the optimal policy becomes too conservative and the optimality of the solution would be lost; even worse, the optimal problem might become infeasible.

Since $z_k + y_{k+1} = \sum_{i=1}^I u_{i,k} - w_k$, relationship $y_{k+1} = \sum_{i=1}^I u_{i,k} - w_k - z_k$ can be substituted into Formulation (4.1) to replace all the y_k 's. The decision variables y_k 's are eliminated from the model. The formulation becomes Formulation (4.2).

$$\begin{aligned}
\min \quad & \sum_{k=1}^T \sum_{i=1}^I C_{i,k} u_{i,k} + \sum_{k=1}^T H_{k+1} \left\{ \sum_{t=1}^k z_t + \sum_{t=1}^k \sum_{i=1}^I u_{i,t} - \sum_{t=1}^k w_t \right\} + \sum_{k=1}^T B_k z_k \\
\text{s.t.} \quad & 0 \leq u_{i,k} \leq P, i = 1, \dots, I; k = 1, \dots, T \\
& \sum_{k=1}^T u_{i,k} \leq Q, i = 1, \dots, I \\
& \sum_{i=1}^I u_{i,1} - w_1 \leq V_{max} \\
& \sum_{t=1}^k z_t + \sum_{t=1}^k \sum_{i=1}^I u_{i,t} - \sum_{t=1}^k w_t + \sum_{i=1}^I u_{i,k+1} - w_{k+1} \leq V_{max}, k = 1, \dots, T \\
& z_k \geq 0, k = 1, \dots, T \\
& \sum_{t=1}^k z_t + \sum_{t=1}^k \sum_{i=1}^I u_{i,t} - \sum_{t=1}^k w_t \geq 0, k = 1, \dots, T
\end{aligned} \tag{4.2}$$

All the uncertainty in Formulation (4.2) arises from the terms $\sum_{t=1}^k w_t$ so the uncertainty structure imposed is $-\kappa \leq \frac{\sum_{t=1}^k w_t - \mu_k}{\sigma_k} \leq \kappa, \forall k \in 1, \dots, T$, where,

$$\mu_k = E\left[\sum_{t=1}^k w_t\right] = \sum_{i=1}^k E[w_t] = \sum_{t=1}^k \bar{w}_t$$

and since w_k is uniformly distributed, the variance of w_k is $\frac{\Delta w_k^2}{3}$. This is based on the fact that the variance of a uniform distribution in the interval $[a, b]$ is $\frac{(b-a)^2}{12}$ with $a = w_k - \Delta w_k$ and $b = w_k + \Delta w_k$. Since w'_k s are independently distributed,

$$\sigma_k = \sqrt{\text{Var}\left[\sum_{t=1}^k w_t\right]} = \sqrt{\sum_{t=1}^k \text{Var}[w_t]} = \sqrt{\sum_{t=1}^k \frac{\Delta w_k^2}{3}},$$

κ is the control parameter to control the robustness level. For a given level of κ , in order to impose robustness, $\sum_{t=1}^k w_t$ is replaced by $\mu_{k+1} + \kappa \cdot \sigma_{k+1}$ if $\sum_{t=1}^k w_t$ appears on the left hand side of “ \leq ”; $\sum_{t=1}^k w_t$ is replaced by $\mu_{k+1} - \kappa \cdot \sigma_{k+1}$ if $\sum_{t=1}^k w_t$ appears on the right hand side of “ \leq ”. After the substitution, Formulation (4.2) is

transformed to Formulation (4.3). This is a linear programming problem and can be solved very efficiently.

$$\begin{aligned}
\min \quad & \sum_{k=1}^T \sum_{i=1}^I C_{i,k} u_{i,k} + \sum_{k=1}^T H_{k+1} \left\{ \sum_{t=1}^k z_t + \sum_{t=1}^k \sum_{i=1}^I u_{i,t} - \sum_{t=1}^k \bar{w}_t \right\} + \sum_{k=1}^T B_k z_k \\
\text{s.t.} \quad & 0 \leq u_{i,k} \leq P, i = 1, \dots, I; k = 1, \dots, T \\
& \sum_{k=1}^T u_{i,k} \leq Q, i = 1, \dots, I \\
& \sum_{i=1}^I u_{i,1} - \mu_1 + \kappa \cdot \sigma_1 \leq V_{max} \\
& \sum_{t=1}^k z_t + \sum_{t=1}^k \sum_{i=1}^I u_{i,t} + \sum_{i=1}^I u_{i,k+1} - \mu_{k+1} + \kappa \cdot \sigma_{k+1} \leq V_{max}, k = 1, \dots, T \\
& z_k \geq 0, k = 1, \dots, T \\
& \sum_{t=1}^k z_t + \sum_{t=1}^k \sum_{i=1}^I u_{i,t} - \mu_k - \kappa \cdot \sigma_k \geq 0, k = 1, \dots, T.
\end{aligned} \tag{4.3}$$

This robust counterpart of the basic linear programming problem makes all the decisions (both current stage and future stage) at the current stages. There is a potential improvement if the robust formulation adapt to the future information that will be known at later stages. This idea leads to the next improved formulation: the adaptive robust counterpart.

4.3 An Adaptive Robust Optimization Formulation

To modify the robust counterpart in the previous section, an affine function is appropriate due to its simplicity. In the affine approximation, the future decision variables are expressed as the affine function of the uncertain parameters. The new decision

variable would be:

$$u_{i,k} = uf_{i,k,0} + \sum_{t=1}^k uf_{i,k,t} \cdot w_t, i = 1, \dots, I; k = 1, \dots, T$$

and

$$y_k = yf_{k,0} + \sum_{t=1}^k yf_{k,t} \cdot w_t, k = 1, \dots, T+1.$$

We use $z_k = \sum_{i=1}^I u_{i,k} - w_k - y_{k+1}$ to replace z_k in (4.4). After substituting the affine functions into Formulation (4.1), we obtain the adaptive counterpart (4.4).

$$\begin{aligned} \min \quad & \sum_{k=1}^T \sum_{i=1}^I c_{i,k} uf_{i,k,0} + \sum_{k=1}^{T+1} [H_k - B_k] yf_{k,0} \\ & + \sum_{k=1}^T B_k \left[yf_{k+1,0} - \sum_{i=1}^I uf_{i,k,0} \right] \\ & + \left\{ \sum_{k=1}^T \sum_{i=1}^I c_{i,k} uf_{i,k,1} + \sum_{k=1}^{T+1} [H_k - B_k] yf_{k,1} \right\} w_1 \\ & + \left\{ \sum_{k=1}^T B_k \left[yf_{k+1,1} - \sum_{k=1}^T \sum_{i=1}^I uf_{i,k,1} + B_1 \right] \right\} w_1 \\ & + \sum_{t=2}^T \left\{ \sum_{k=1}^T \sum_{i=1}^I c_{i,k} uf_{i,k,t} + \sum_{k=1}^{T+1} [H_k - B_k] yf_{k,t} \right\} w_t \\ & + \sum_{t=2}^T \left\{ \sum_{k=1}^T B_k \left[yf_{k+1,t} - \sum_{k=1}^T \sum_{i=1}^I uf_{i,k,t} + B_t \right] \right\} w_t \\ \text{s.t.} \quad & 0 \leq uf_{i,k,0} + \sum_{t=1}^k uf_{i,k,t} w_t \leq P, i = 1, \dots, I; k = 1, \dots, T \\ & \sum_{k=1}^T uf_{i,k,0} + \sum_{t=1}^T \left(\sum_{k=1}^T uf_{i,k,t} \right) w_t \leq Q, i = 1, \dots, I \\ & y_1 = 0 \\ & \left[yf_{k,0} + \sum_{i=1}^I uf_{i,k,0} \right] + \sum_{t=1}^{k-1} \left[yf_{k,t} + \sum_{i=1}^I uf_{i,k,t} \right] w_t \\ & + \left[yf_{k,k} + \sum_{i=1}^I uf_{i,k,k} \right] w_k \leq V_{max}, k = 1, \dots, T \end{aligned}$$

$$\begin{aligned}
& \left\{ yf_{k+1,0} - yf_{k,0} - \sum_{i=1}^I uf_{i,k,0} \right\} \\
& + \sum_{t=1}^{k-1} \left\{ yf_{k+1,t} - yf_{k,t} - \sum_{i=1}^I uf_{i,k,t} \right\} w_t \\
& + \left\{ yf_{k+1,k} - yf_{k,k} - \sum_{i=1}^I uf_{i,k,k} + 1 \right\} w_k \\
& + yf_{k+1,k+1} \geq 0, k = 1, \dots, T \\
& yf_{k,0} + \sum_{t=1}^k yf_{k,t} w_t \geq 0, k = 1, \dots, T
\end{aligned} \tag{4.4}$$

The terms in Formulation (4.4) have been rearranged such that the terms are grouped by the demand w_t and the terms can be simply denoted by the form in Formulation (4.5), where m is the number of constraints.

$$\begin{aligned}
\min \quad & g_0 + \sum_{k=1}^T g_k \cdot w_k \\
s.t. \quad & f_{r,0} + \sum_{k=1}^T f_{r,k} \cdot w_k \leq 0, r = 1, \dots, m
\end{aligned} \tag{4.5}$$

The box uncertainty set can be easily imposed to formulation (4.5). The result is Formulation (4.6). Formulation (4.6) is a simple linear programming problem.

$$\begin{aligned}
\min \quad & g_0 + \sum_{k=1}^T g_k \cdot w_k + \sum_{k=1}^T |g_k| \cdot \Delta w_k \\
s.t. \quad & f_{r,0} + \sum_{k=1}^T f_{r,k} \cdot w_k + \sum_{k=1}^T |f_{r,k}| \cdot \Delta w_k \leq 0, r = 1, \dots, m
\end{aligned} \tag{4.6}$$

An incremental computational cost arises from the adaptive structure imposed.

The number of decision variables are in the order of $O(T^2)$: each decision variable becomes T new decision variables in the adaptive structure. Therefore, the computational complexity increases, compared with the basic linear programming formulation and robust optimization without the adaptive structure. By paying this cost, the expected gain is the improved performance of the optimal solution when it is exposed to the uncertainty environment. We demonstrate the performance comparison in Section 4.5.

4.4 A Data Driven Optimization Formulation

The decision variables are defined as follows:

- $u_{i,1}$ is the current decision
- $u_{s,i,k}$ is the future decision given scenario s happens, $k = 2, \dots, T$
- $y_{s,k}$ is the inventory at the beginning of period k under demand scenario s
- $z_{s,k}$ is the lost sale quantity at the end of period k under demand scenario s

$w_{s,k}$ is s^{th} scenarios for demand in period k , $s = 1, \dots, M, k = 1, \dots, T$. The DDO formulation is as in (4.7).

$$\begin{aligned}
\min \quad & \frac{1}{M} \sum_{s=1}^M \left\{ \left(\sum_{i=1}^I C_{i,k} u_{i,1} + H_{k+1} y_2 + \sum_{k=1}^T B_k z_1 \right) \right\} \\
& + \frac{1}{M} \sum_{s=1}^M \left\{ \left(\sum_{k=2}^T \sum_{i=1}^I C_{i,k} u_{s,i,k} + \sum_{k=2}^T H_{k+1} y_{s,k+1} + \sum_{k=2}^T B_k z_{s,k} \right) \right\} \\
s.t \quad & 0 \leq u_{s,i,k} \leq P, i = 1, \dots, I; k = 1, \dots, T; s = 1, \dots, M \\
& \sum_{k=1}^T u_{s,i,k} \leq Q \\
& y_1 = 0 \\
& y_{s,k} + \sum_{i=1}^I u_{s,i,k} - w_{s,k} \leq V_{max}, \\
& -z_{s,k} + y_{s,k+1} = y_{s,k} + \sum_{i=1}^I u_{s,i,k} - w_{s,k} \\
& z_{s,k} \geq 0 \\
& y_{s,k} \geq 0
\end{aligned} \tag{4.7}$$

4.5 Numerical Results

The model parameters are assigned as follows:

- $T = 24$: number of time periods
- $I = 3$: number of factories
- $C_{i,k} = \alpha_i \cdot (1 - 0.5 \sin(\frac{\pi(k-1)}{12}))$: production cost for time k and factory i
- $H_k = 0.2 \cdot \sum_{i=1}^I \sum_{t=1}^T C_{i,t}, \forall k$: holding cost per item at period k
- $B_k = 1.2 \cdot \frac{\sum_{i=1}^I \sum_{t=1}^T C_{i,t} \cdot w_k}{\sum_{t=1}^T w_t}$: lost sale cost per item at period k
- $\bar{w}_k = 1000 \cdot (1 + 0.5 \sin(\frac{\pi(k-1)}{12}))$: demand in period k
- $\Delta w_k = 20\% w_k$

- $Q = 13600$: total production capacity of each factory during the whole time horizon
- $P = 567$: maximum production capacity of each factory

The detailed procedure for the numerical experiment is shown below:

- Step 1: Four optimal solutions are obtained: LO, RO, ARO and DDO.
- Step 2: Generate random demands and simulate the solutions from Step 1 and report the inventory cost.
- Step 3:

We use 100 samples for $V_{max} = 2000$. We also try $V_{max} = 500$ with 450 samples. The numerical simulation assumes the true demand is uniformly distributed within a box with $\Delta w = 20\%w$. All the simulations use rollover procedures to refresh the decision every time new information is available. The total inventory cost for each simulation path is recorded. At the end, the average inventory cost and the standard deviation of the inventory cost are recorded and compared.

For the RO formulation, we find the parameter κ by testing κ from 0 to 1 with step size 0.1 and compare the average cost of 20 sample paths. We found that 0.2 is best choice among the κ we tried in this given numerical setting.

The result with $\Delta w = 20\%w$ is listed in Table 4.1. Table 4.1 lists the simulation result of 100 simulation path. It shows that the ARO has the lowest average inventory cost. RO has lower inventory cost than LO but the inventory cost is higher than that of ARO. Therefore, ARO outperforms. The statistics shows that in 83% of the time, ARO has a lower inventory cost than RO so we conclude that ARO obtains a lower inventory cost most of the time. If the basic linear programming formulation is the benchmark, RO improves the inventory cost by 1.4 %, DDO improves the inventory cost by 1.73% and ARO improves the inventory cost by 3.05%. 1.4% and 1.7% improvement is close to each other compared with 3% improvement. Therefore, we conclude that DDO and RO reach a similar average inventory cost and they have a comparable performance.

Table 4.1: Performance Comparison ($\Delta w = 20\%w, V_{max} = 2000$)

	LO	DDO	RO	ARO
Average Cost	27516	27049	27136	26702
Standard Deviation	1452	1275	1294	985

When $\Delta w = 20\%w$ and $V_{max} = 500$, an interesting phenomenon is that, the RO becomes infeasible while ARO is still feasible. Specially, based on 450 samples, 26.67 % of the samples become infeasible during the rollover iteration of RO. If an infeasible instance is encountered, we apply the LO approach instead of RO in order to overcome infeasibility.

Table 4.2 includes all cases (including the cases when we used to LO when RO is infeasible). Table 4.3 includes only the cases for which RO solution is feasible. Both tables suggest that RO generates lower average inventory cost and lower standard deviation than LO; ARO generates lower average inventory cost and lower standard deviation of inventory cost than RO.

Table 4.2: Performance Comparison ($\Delta w = 20\%w, V_{max} = 500$, including infeasible cases)

	Linear Programming	Robust Optimization	Adaptive Approach
Average Cost	27,824.70	27,456.00	27,247.54
Standard Deviation	1,181.42	1,058.64	971.27

Table 4.3: Performance Comparison ($\Delta w = 20\%w, V_{max} = 500$, excluding infeasible cases)

	Linear Programming	Robust Optimization	Adaptive Approach
Average Cost	28,007.14	27,564.53	27,410.27
Standard Deviation	1,158.79	1,040.00	953.29

4.6 Conclusions

- Robust Optimization outperforms the Linear Programming formulation.
- Adaptive Robust Optimization outperforms Robust Optimization.
- Data Driven Optimizaiton and Robust Optimization have a comparable performance.
- The computational cost for ARO and DDO are higher than that of RO and LP.

Bibliography

- [1] A.L.Soyster, Convex Programming with Set-Inclusive Constraints and Application to Inexact Linear Programming, *Operations Research*, Vol.21, No. 5 (Sep.,1973), pp.1154-1157.
- [2] D. Bertsimas and M. Sim, The price of robustness, *Operations Research*, Vol.52, No. 1 (May., 2004), pp.35-53.
- [3] Ben-Tal,A., Nemirovski,A., Robust solutions of Linear Programming problems contaminated with uncertainty data, *Mathematical Programming*, 88, pp.411-424, 2000.
- [4] Ben-Tal,A., Goryashko,A.,Guslitzer.E,and Nemirovski,A., Adjustable robust solutions of uncertain linear programs, *Mathematical Programming*, 99,351-367, 2004.
- [5] D. A. Pachamanova, A robust optimization approach to finance, *MIT Ph.d. Thesis*, 2002.
- [6] Constantine Caramanis, Adaptable Optimization: Theory and Algorithms, *MIT Ph.d. Thesis*, 2006.
- [7] Aurlie Thiele, A Robust Optimization Approach to Supply Chains and Revenue Management, *MIT Ph.d. Thesis*, 2004.
- [8] D. Bertsimas, J.Tsitsiklis, Introduction to Linear Optimization, *Athena Scientific Series in Optimization and Neural Computation*, 1997.
- [9] George B. Dantzig, Linear Programming under Uncertainty, *Management Science*, Vol. 1, No. 3, 1955