

A Lagrangian Relaxation Approach to Two-Stage  
Stochastic Facility Location Problem with  
Second-Stage Activation Cost

by

Ghazal Ghodsi

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Management Sciences

Waterloo, Ontario, Canada, 2012

© Ghazal Ghodsi 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

## Abstract

We study a two-stage stochastic facility location problem in the context of disaster response network design. The uncertainty inherent in disaster occurrence and impact is captured by defining scenarios to reflect a large spectrum of possible occurrences. In the first stage (pre-event response), planners should decide on locating a set of facilities in strategic regions. In the second stage (post-event response), some of these facilities are to be activated to respond to demand in the disaster affected region. The second-stage decisions depend on disaster occurrence and impact which are highly uncertain. To model this uncertainty, a large number of scenarios are defined to reflect a large spectrum of possible occurrences. In this case, facility activation and demand allocation decisions are made under each scenario. The aim is to minimize the total cost of locating facilities in the first stage plus the expected cost of facility activation and demand allocation under all scenarios in the second stage while satisfying demand subject to facility and arc capacities.

We propose a mixed integer programming model with binary facility location variables in the first stage and binary facility activation variables and fractional demand allocation variables in the second stage. We propose two Lagrangian relaxations and several valid cuts to improve the bounds. We experiment with aggregated, disaggregated and hybrid implementations in calculating the Lagrangian bound and develop several Lagrangian heuristics. We perform extensive numerical testing to investigate the effect of valid cuts and disaggre-

gation and to compare the relaxations. The second relaxation proved to provide a tight bound as well as high quality feasible solutions.

## **Acknowledgements**

I would like to express my sincere gratitude to my supervisor Professor Gzara for her continuous support on this study.

## **Dedication**

This is dedicated to my parents for their endless love and to my husband who supported me each step of the way.

# Table of Contents

List of Tables	x
List of Figures	xii
<b>1 Introduction</b>	<b>1</b>
<b>2 Literature Review</b>	<b>5</b>
<b>3 Problem Formulation and Relaxation I</b>	<b>10</b>
3.1 Problem Formulation . . . . .	10
3.2 Quality of LP Bound . . . . .	14
3.3 Lagrangian Relaxation I . . . . .	15
3.4 Solution of Subproblems . . . . .	20

3.5	Calculating the Lagrangian Bound . . . . .	22
3.6	Aggregated, Disaggregated and Hybrid Implementations . . . . .	23
3.7	Lagrangian Heuristic . . . . .	25
3.8	Complete Algorithm Based on Relaxation I . . . . .	27
<b>4</b>	<b>Relaxation II</b>	<b>29</b>
4.1	Lagrangian Relaxation II . . . . .	29
4.2	Solution of Subproblems . . . . .	33
4.3	Calculating the Lagrangian Bound . . . . .	35
4.4	Aggregated and Disaggregated Implementations . . . . .	35
4.5	Lagrangian Heuristics . . . . .	37
4.5.1	Heuristic from $[SP_1^2]$ . . . . .	37
4.5.2	Heuristic from $[SP_2^2]$ . . . . .	39
4.6	Complete Algorithm Based on Relaxation II . . . . .	40
<b>5</b>	<b>Implementation and Numerical Testing</b>	<b>42</b>
5.1	Generating Test Instances . . . . .	43
5.2	Checking Feasibility of Instances . . . . .	45

5.3	Effect of Adding Cuts to the Subproblems . . . . .	48
5.4	Comparison of Aggregated, Disaggregated and Hybrid Implementations . .	50
5.5	The Effect of Arc Density on the Quality of the Lagrangian Bound . . . .	55
5.6	Comparing Lagrangian Relaxations I and II . . . . .	58
5.7	The Effect of Number of Scenarios on GAP . . . . .	60
5.8	Comparison to Penuel et. al. (2010) . . . . .	61
5.9	Improvement Heuristic . . . . .	63
<b>6</b>	<b>Conclusion</b>	<b>65</b>
	<b>Bibliography</b>	<b>68</b>
<b>A</b>	<b>Appendix</b>	<b>72</b>

# List of Tables

5.1	Possible values for test instances. . . . .	44
5.2	Notation used in this study. . . . .	46
5.3	Rel I. Effect of adding cuts (15),(16),(17) to $[SP_1^1],[SP_2^1]$ . . . . .	49
5.4	Rel II. Effect of adding cuts to $[SP_1^2]$ . . . . .	51
5.5	Rel I. Comparing AGG,DISAGG,HYBRID implementations. . . . .	52
5.6	Rel II. Comparing AGG, DISAGG, DISAGG+Cuts implementations. . . . .	54
5.7	Rel I. Effect of arc density on Lagrangian bound. . . . .	56
5.8	Rel II. Effect of arc density on Lagrangian bound. . . . .	57
5.9	Summary of relaxations I and II. . . . .	59
5.10	Comparing Lagrangian Heuristics in Rel I and Rel II to optimal solutions. . . . .	59
5.11	Rel II. Instances (n=20,s=20) . . . . .	60

5.12 Rel II. Instances (n=20,s=30) . . . . .	61
5.13 Rel II. Comparing our results to Penuel et. al. (2010). . . . .	62
5.14 Rel II. Comparing our results (GAP(Opt)) to Penuel et. al. (2010). . . . .	62
5.15 Implementing improvement heuristic and comparing GAP(Inc) to Penuel et. al. (2010). . . . .	64

# List of Figures

3.1	Complete Algorithm - Relaxation I . . . . .	28
4.1	Complete Algorithm - Relaxation II . . . . .	41

# Chapter 1

## Introduction

World Health Organization defines a disaster as an unforeseen destructive phenomenon that causes damage to human life in large scale [21]. Once a region is hit by a disaster, outside resources might be needed to manage the catastrophic impact. This process, named as disaster operation, is a set of predictions and reactions planned ahead to decrease the destructive impact of a disaster and bring the system back to its normal state.

Disaster management also known as emergency management is divided in four stages: mitigation, preparedness, response, and recovery (Green, 2002; Waugh, 2000; Godschalk, 1991; Waugh and Hy, 1990). Mitigation is the set of measures and policies that will either prevent the disastrous event or reduce its harmful impact once it occurs. Preparedness involves activities that bring awareness to the community on how to respond to a disaster

should one occur. Once an emergency situation happens, a set of resources are employed to deal with its initial impact and to preserve life, property and the environment, providing emergency relief to victims and preventing further damage. This is called emergency response. Recovery includes the long term plans to stabilize the community and to return to normalcy after a disaster has passed.

This study relates to the preparedness and response steps. We decide on where to locate disaster response facilities before a disaster occurs as part of the preparedness process. After a disaster hits a region, we will activate some facilities to provide supply and respond to the needs in the system.

Operations Research scientists showed interest in the field of disaster management since 1980s. Since the beginning of the millennium, the number of papers published in mainstream OR on emergency management surpassed the number published in the 1990s due to the growing economic and social impact of disastrous events and the urge to disaster reduction (Altay, Green, 2006).

Tufekci and Wallace (1998) consider disaster response that consists of two stages; pre-event and post-event response. Considering the problem of allocating relief shelters to victims of a disaster, pre-event tasks include predicting the potential disastrous regions and developing necessary action plans for mitigation like locating some potential facilities. Post-event response begins while the disaster is still in progress. At this stage the challenge

is activating, coordinating, and managing available resources. To plan a disaster response effectively, Tufekci et. al. (1998) suggest that both these stages should be encountered within the objective. Otherwise solving pre-event and post-event responses separately may result in a suboptimal solution to the overall problem. This explains the two-stage nature of a disaster response network design.

Another challenge in the disaster response problems is the uncertain nature of design parameters. During the course of a disaster many of the design parameters of the problem may fluctuate due to the severity of the emergency situation. For instance in a facility location problem, uncertainty arises in demand, supply and route capacity perspectives. This explains why disaster response planning problem should be naturally treated as a stochastic problem.

This study focuses on disaster response network design in the context of two-stage facility location under uncertainty. In the first stage (pre-event response), planners should decide on locating a set of facilities in strategic regions. In the second stage (post-event response), some of these facilities are to be activated to satisfy demand in the network. The uncertainty in this problem is captured by defining scenarios where each scenario presents a possible data set in case of disaster.

A potential application of this problem would be locating hurricane relief shelters in hurricane-prone regions where some facilities are located in anticipation of an emergency

(Penuel et. al., 2010). Once the hurricane makes landfall, some of these facilities must be opened and staffed in the second stage to establish flows in the network to respond to the needs.

The thesis is organized as follows: Chapter 2 presents literature review on facility location under uncertainty. In Chapter 3, the mathematical formulation of our problem is presented and the first Lagrangian relaxation approach is derived. Chapter 4 describes the second Lagrangian relaxation model to solve the same problem. Chapter 5 describes the numerical results based on implementing the relaxations discussed in Chapters 3 and 4. Chapter 6 concludes the thesis.

# Chapter 2

## Literature Review

Facility location decisions are long term decisions that are expensive to alter. During the course of time that the decisions are implemented, there might be changes in design parameters like demand, costs, and capacities. Based on the level of information available about the uncertain parameters, different modeling approaches may apply. Rosenhead et. al. (1972) divide the decision-making environment to three categories of certainty, risk, and uncertainty. The decision environment for which the probability distribution of unknown parameters is defined is referred to as risk environment. Problems in risk decision-making environment are defined as stochastic optimization problems (SP). On the other hand when neither the values nor the probability distribution of parameters are known, robust optimization modeling may be used to formulate the problem in uncertainty situation. A

variety of optimization approaches have been proposed over the years on stochastic and robust location problems. A comprehensive review on these studies is provided by Snyder (2006). Stochastic optimization problems usually aim to minimize the expected cost or maximize the expected profit of the system. A Lagrangian relaxation algorithm for the stochastic P-Median Problem (PMP) is presented by Weaver and Church (1983). The stochasticity in their problem is defined by a set of scenarios with fixed probabilities and they show that some scenario-based stochastic problems may be solved as larger versions of the deterministic problem.

In the stochastic version of capacitated P-Median problems and capacitated fixed-charge location problem studied by Louveaux (1986), demands and production costs are random. The author suggests a special type of budget constraints under which these two stochastic models are equivalent. Louveaux (1993) reviews modeling approaches for these and related problems.

The stochastic facility location problems are often two-stage problems where the first-stage decision is to locate a set of facilities and the second-stage decision of assigning customers to facilities is made once the uncertainty is resolved. The objective of the stochastic problem is either to minimize the expected cost or to maximize the expected profit. A variety of modeling approaches have been proposed over the years for this type of two-stage stochastic facility location problems (Louveaux ,1993).

The problem studied in this thesis has a different setting than a typical two-stage stochastic facility location problem. Here a scenario-based stochastic facility location problem is considered. A set of scenarios with fixed probability of occurrence is provided to us and a realization of the demands, supplies, costs and capacities is specified for each scenario. Sheppard et. al. (1974) introduced a scenario approach to the facility location problem for the first time in 1974. Although identifying a set of scenarios covering a complete range of possible events is difficult, the scenario approach allows for the parameters to be statistically dependent and results in more manageable models.

The decision in the first stage is to locate a set of facilities that may or may not be activated in the future. However the located facilities in the first stage may not be ready to use and need to be further activated in the second stage. Also in the second stage the flows from active facilities must be established to satisfy demand in each future scenario. In the first stage, facility location decisions are modeled as binary variables whereas the second-stage model is a mixed-integer program because of binary facility activation variables and continuous flow variables.

A common approach to stochastic programming as well as integer programming is Benders decomposition (Benders, 1962). However, this method of decomposition relies on convexity of the value function of linear programming. The second stage subproblems may impose integer restrictions on some variables so the value function of such integer

subproblems is not convex, and new approaches must be designed.

Sen and Hige (2005) suggest the disjunctive decomposition ( $D^2$ ) method that uses disjunctive programming to convexify mixed-integer program subproblems. Sen and Sher-ali (2006) discuss alternative decomposition methods in which the second stage integer subproblems are solved using branch-and-cut methods. One of the main advantages of their decomposition scheme is that Stochastic Mixed-Integer Programming (SMIP) problems may be solved by dividing a large problem into smaller Mixed-Integer Programming (MIP) subproblems which can be solved in parallel.

Laporte et. al. (1993) present an optimal algorithm based on the L-shaped method of stochastic programming. For a general recourse structure, Laporte and Louveaux (1993) propose a decomposition-based approach for stochastic integer programs when the first-stage variables are pure binary. This restriction allows for the construction of optimality cuts that approximate the non-convex second-stage value function at only the binary first-stage solutions (but not necessarily at other points). The authors propose a branch-and-bound algorithm to search the space of the first-stage variables for the globally optimal solution, while using the optimality cuts to approximate the second-stage value function. Finite termination of the algorithm is obvious since the number of first-stage solutions is finite. The method has been successfully used in solving two-stage stochastic location-routing problems (Louveaux et. al., 1992,1994). However, the algorithm is not applicable

if any of the first-stage variables are continuous.

Penuel et. al. (2010) propose an integer decomposition method to solve stochastic location problems with some assumptions on the problem structure and they were able to solve most of the instances within reasonable computational time. We analyse the same setting as Penuel et. al. (2010) and propose a different formulation, then solve it using two Lagrangian relaxations.

# Chapter 3

## Problem Formulation and Relaxation

### I

In this Chapter, the mathematical model for the two-stage stochastic facility location problem is developed. Then Lagrangian relaxation is used to decompose the problem into smaller subproblems that are easier to solve.

#### 3.1 Problem Formulation

The disaster response network may be presented on a directed graph  $G = (V, A)$  with node set  $V$  and arc set  $A$ . In the context of the two-stage stochastic facility location problem

considered in this study, set  $V$  presents the set of potential facility locations connected through a set of transportation channels that form the arc set  $A$ . The stochastic nature of this problem is described by considering a set of possible scenarios  $S$ . Each scenario  $s \in S$  is assigned a fixed weight  $p^s$  that accounts for the probability of the scenario happening or the importance of attending to it so priority may be given to a scenario with a greater  $p^s$  value.

The facility location problem in the context of disaster response consists of two stages. In the first stage, a set of facilities are located but not activated yet. Locating a facility at location  $i \in V$  in the first stage incurs a fixed charge  $f_i$ . The second stage starts once a scenario  $s \in S$  occurs with probability  $p^s$ . The decision is then to activate some of the already located facilities to meet the demand in the system. Activation of a facility  $i \in V$  in a scenario  $s \in S$  has a cost  $g_{is}$ . Also in this stage, demand must be satisfied through the transportation channel  $A$ . Each node  $i \in V$  in scenario  $s \in S$  has a demand  $d_i^s \geq 0$  and a supply  $b_i^s \geq 0$ . There is a transportation cost  $c_{ij}^s$  associated with each arc  $(i, j) \in A$  in scenario  $s \in S$ . The transportation channels in each scenario have capacity  $u_{ij}^s$  for arc  $(i, j) \in A$ .

The aim is to minimize the total cost of locating facilities in the first stage plus the expected cost of facility activation and demand allocation under all scenarios in the second

stage. The problem can be modeled as a two-stage stochastic MIP. The location decision in the first stage is presented by the binary variable  $z_i$  that takes value 1 if a facility is located on a node  $i \in V$  and 0 otherwise. The second stage decision variables are  $y_i^s$  and  $x_{ij}^s$ .  $y_i^s$  is a binary variable that takes value 1 if facility  $i \in V$  is activated in scenario  $s \in S$  and 0 otherwise.  $x_{ij}^s$  is the flow variable which is defined as the fraction of demand at location  $j \in V$  satisfied by facility  $i \in V$  in scenario  $s \in S$ , hence takes values between 0 and 1.

The mathematical formulation of the problem is then presented as:

$$[\text{P}] \quad \min \quad \sum_{i \in V} f_i z_i + \sum_{s \in S} p^s \left( \sum_{i \in V} g_i^s y_i^s + \sum_{j \in V} \sum_{i \in V} c_{ij}^s d_j^s x_{ij}^s \right) \quad (1)$$

s.t.

$$y_i^s \leq z_i \quad \forall i \in V, \forall s \in S \quad (2)$$

$$d_j^s x_{ij}^s \leq u_{ij}^s \quad \forall (i, j) \in A, \forall s \in S \quad (3)$$

$$\sum_{i \in V} x_{ij}^s = 1 \quad \forall j \in V, \forall s \in S \quad (4)$$

$$\sum_{j \in V} x_{ij}^s d_j^s \leq y_i^s b_i^s \quad \forall i \in V, \forall s \in S \quad (5)$$

$$0 \leq x_{ij}^s \leq 1 \quad \forall (i, j) \in A, \forall s \in S \quad (6)$$

$$z_i \in \{0, 1\} \quad \forall i \in V \quad (7)$$

$$y_i^s \in \{0, 1\} \quad \forall i \in V, \forall s \in S \quad (8)$$

The objective function (1) minimizes the first-stage location cost and the expected cost of facility activation and demand allocation over all scenarios. Constraints (2) ensure that only a facility that is located in the first stage can be activated in the second stage. Constraints (3) ensure that under each scenario, demand allocation on an arc does not exceed arc capacity. Constraints (4) ensure that the demand at node  $j \in V$  is completely satisfied. Constraints (5) state that the flow out of a facility  $i \in V$  must be within its

supply capacity ( $b_i^s$ ). Constraints (6), (7) and (8) are sign and binary requirements on  $x_{ij}^s$ ,  $y_i^s$  and  $z_i$  respectively.

This formulation models the same problem presented by Penuel et. al. (2010) but uses a different set of variables ( $x_{ij}^s$ ). Constraints (2) link  $y_i^s$  and  $z_i$  variables and constraints (5) enforce capacity and link  $x_{ij}^s$  and  $y_i^s$  variables. This structure leads to the idea of implementing Lagrangian relaxation on either of these constraints which will decompose the problem into smaller subproblems. The rest of this section details relaxation based on constraint (5) while Chapter 4 presents the second relaxation based on constraints (2).

## 3.2 Quality of LP Bound

Before going further and using Lagrangian relaxation approach to solve the problem presented in Section 3.1, we evaluate the quality of the LP bound by using a simple example.

Consider a problem with two nodes (one supply and one demand node) and one scenario ( $n=2, s=1$ ). The costs and capacities are provided as follows :  $f_1 = 20, g_1^1 = 20, b_1^1 = 20, d_2^1 = 1, c_{12}^1 = 0, u_{12}^1 = \text{inf}$ .

The solution of [P] for this example considering the integrality conditions on location and activation variables is:  $z_1 = 1, y_1^1 = 1, x_{12}^1 = 1$ . This solution will result in the objective

value of  $(20 \times 1 + 20 \times 1 + 0 \times 1) = 40$ .

The LP relaxation bound is derived by dropping the integrality condition on  $z$  and  $y$ . So the solution in this case is :  $z_1 = \frac{1}{20}, y_1^1 = \frac{1}{20}, x_{12}^1 = 1$  with the LP bound of  $(20 \times \frac{1}{20} + 20 \times \frac{1}{20} + 0 \times 1) = 2$ .

This simple example proves that the LP bound in this type of problems can be really weak. Therefore we try to obtain a Lagrangian bound as far from the LP bound as possible. However this means solving more difficult subproblems in the Lagrangian relaxation approach as it is shown in Sections 3.4 and 4.2.

### 3.3 Lagrangian Relaxation I

[P] is a mixed-integer programming problem that exhibits a special structure suitable for Lagrangian relaxation: constraint (5) is the only constraint that links the activation variables and the flow variables. Relaxing constraint (5) with non-negative Lagrangian multipliers  $\lambda_i^s$  will result in the following Lagrangian problem:

$$L(\lambda_i^s) \quad \min \quad \sum_{i \in V} f_i z_i + \sum_{s \in S} p^s \left( \sum_{i \in V} g_i^s y_i^s + \sum_{j \in V} \sum_{i \in V} c_{ij}^s d_j^s x_{ij}^s \right) +$$

$$\sum_{s \in S} \sum_{i \in V} \lambda_i^s \left( \sum_{j \in V} x_{ij}^s d_j^s - y_i^s b_i^s \right)$$

s.t.

$$y_i^s \leq z_i \quad \forall i \in V, \forall s \in S \quad (9)$$

$$d_j^s x_{ij}^s \leq u_{ij}^s \quad \forall (i, j) \in A, \forall s \in S \quad (10)$$

$$\sum_{i \in V} x_{ij}^s = 1 \quad \forall j \in V, \forall s \in S \quad (11)$$

$$0 \leq x_{ij}^s \leq 1 \quad \forall (i, j) \in A, \forall s \in S \quad (12)$$

$$z_i \in \{0, 1\} \quad \forall i \in V \quad (13)$$

$$y_i^s \in \{0, 1\} \quad \forall i \in V, \forall s \in S \quad (14)$$

Constraints (9), (13) and (14) concern the first and second stage location variables  $z_i$  and  $y_i^s$ , while constraints (10), (11) and (12) concern assignment variables  $x_{ij}^s$ . Based on this observation, the Lagrangian problem decomposes into two subproblems  $[SP_1^1]$  and  $[SP_2^1]$ .

$$\begin{aligned}
[SP_1^1] \quad & \min \quad \sum_{i \in V} f_i z_i + \sum_{s \in S} \sum_{i \in V} (p^s g_i^s - \lambda_i^s b_i^s) y_i^s \\
& \text{s.t.} \quad y_i^s \leq z_i && \forall i \in V, \forall s \in S \\
& && z_i \in \{0, 1\} && \forall i \in V \\
& && y_i^s \in \{0, 1\} && \forall i \in V, \forall s \in S
\end{aligned}$$

and

$$\begin{aligned}
[SP_2^1] \quad & \min \quad \sum_{s \in S} \sum_{i \in V} \sum_{j \in V} (p^s c_{ij}^s + \lambda_i^s) d_j^s x_{ij}^s \\
& \text{s.t.} \quad x_{ij}^s d_j^s \leq u_{ij}^s && \forall (i, j) \in A, \forall s \in S \\
& && \sum_{i \in V} x_{ij}^s = 1 && \forall j \in V, \forall s \in S \\
& && 0 \leq x_{ij}^s \leq 1 && \forall (i, j) \in A, \forall s \in S
\end{aligned}$$

$[SP_1^1]$  is an integer program (IP) that determines which facilities to locate in the first stage of decision making and which facilities to activate in the second stage. It has the property of repeating over index  $i \in V$  that enables us to decompose it into  $|V|$  similar subproblems and solve them individually.  $[SP_2^1]$  is a linear program (LP) that also exhibits

a structure to be separated by scenarios.

Let  $\theta^1$  be the objective value of  $[SP_1^1]$  and  $\delta^1$  be that of  $[SP_2^1]$ . For any given value of  $\lambda_i^s \geq 0$ ,  $\theta^1 + \delta^1$  is a lower bound on the objective value of [P]. The Lagrangian bound is the maximum of  $\theta^1 + \delta^1$  over  $\lambda_i^s \geq 0$ . To calculate the Lagrangian bound, a cutting plane approach is used. Define  $H$  and  $K$  as the index sets of the integer points in feasible regions of  $[SP_1^1]$  and  $[SP_2^1]$ , respectively.

$$H = \{ \text{index set of the feasible region of } [SP_1^1] \}$$

$$K = \{ \text{index set of the feasible region of } [SP_2^1] \}$$

The Lagrangian Master problem [MP] is then defined as the problem of finding the maximum of  $\theta^1 + \delta^1$  over all the integer solutions in the feasible region of  $[SP_1^1]$  and  $[SP_2^1]$ .

$$\begin{aligned}
[\text{MP}] \quad & \max \quad \theta^1 + \delta^1 \\
\text{s.t.} \quad & \theta^1 + \sum_{s \in S} \sum_{i \in V} \lambda_i^s b_i^s y_i^{s^h} \leq \sum_{i \in V} f_i z_i^h + \sum_{s \in S} p^s \sum_{i \in V} g_i^s y_i^{s^h} \quad \forall h \in H \\
& \delta^1 - \sum_{s \in S} \sum_{i \in V} \sum_{j \in V} \lambda_i^s d_j^s x_{ij}^{s^k} \leq \sum_{s \in S} p^s \sum_{i \in V} \sum_{j \in V} c_{ij}^s d_j^s x_{ij}^{s^k} \quad \forall k \in K \\
& \lambda_i^s \geq 0 \quad \forall i \in A, \forall s \in S \\
& \theta^1, \delta^1 \quad \text{unrestricted in sign}
\end{aligned}$$

Since  $H$  and  $K$  are not known beforehand, we start by solving a relaxed master problem defined on small subsets of  $H$  and  $K$ ,  $\bar{H}$  and  $\bar{K}$ :

$$\begin{aligned}
[\text{RMP}] \quad & \max \quad \theta^1 + \delta^1 \\
\text{s.t.} \quad & \theta^1 + \sum_{s \in S} \sum_{i \in V} \lambda_i^s b_i^s y_i^{s^h} \leq \sum_{i \in V} f_i z_i^h + \sum_{s \in S} p^s \sum_{i \in V} g_i^s y_i^{s^h} \quad \forall h \in \bar{H} \\
& \delta^1 - \sum_{s \in S} \sum_{i \in V} \sum_{j \in V} \lambda_i^s d_j^s x_{ij}^{s^k} \leq \sum_{s \in S} p^s \sum_{i \in V} \sum_{j \in V} c_{ij}^s d_j^s x_{ij}^{s^k} \quad \forall k \in \bar{K} \\
& \lambda_i^s \geq 0 \quad \forall i \in A, \forall s \in S \\
& \theta^1, \delta^1 \quad \text{unrestricted in sign}
\end{aligned}$$

Solving [RMP] determines the Lagrangian multipliers  $\lambda_i^s$ . The latter are used to modify and solve the subproblems  $[SP_1^1]$  and  $[SP_2^1]$ . The detailed algorithm to find the Lagrangian bound is described in Section 3.7.2.

### 3.4 Solution of Subproblems

Subproblem  $[SP_1^1]$  is a 0 – 1 integer program that is easy to solve. It determines which facilities to locate in the first stage and among those located, which ones to activate in the second stage. For each location  $i \in V$ , if  $f_i < \sum_{s \in S} (p^s g_i^s - \lambda_i^s b_i^s)$ , a facility will open (i.e.  $z_i = 1$ ). In this case  $y_i^s = 1$  if  $p^s g_i^s - \lambda_i^s b_i^s < 0$ , and 0 otherwise. Furthermore, the decision to locate a facility at location  $i \in V$  does not depend on other locations. So  $[SP_1^1]$  can be further disaggregated into  $|V|$  subproblems, one for each  $i \in V$ . We note that these subproblems are not necessarily identical as each location  $i \in V$  has a different set of costs and capacities assigned to it, however the same solution procedure applies to all of them.

Subproblem  $[SP_2^1]$  is an LP that has the structure of a linear semi-assignment with upper bounds (Volgenant, 1996).  $[SP_2^1]$  can also be solved by inspection. For each demand point  $j \in V$ , order the objective coefficients in increasing order and set  $x_{ij}^s = \min \{u_{ij}^s / d_j^s\}$ .

$[SP_2^1]$  also may be decomposed by scenario  $s \in S$ , the resulting subproblems are not necessarily identical, but the same solution procedure applies.

It is well known that when the Lagrangian subproblems have the integrality property, the Lagrangian bound is equal to the LP relaxation bound (Winston, 2003). In order to strengthen the subproblems and improve the Lagrangian bound, valid inequalities are derived and added to both subproblems.

$$\sum_{i \in V} b_i^s y_i^s \geq \sum_{j \in V} d_j^s \quad \forall s \in S \quad (15)$$

$$\sum_{i \in V} u_{ij}^s z_i \geq d_j^s \quad \forall j \in V, \forall s \in S \quad (16)$$

$$\sum_{j \in V} x_{ij}^s d_j^s \leq b_i^s \quad \forall s \in S, \forall i \in V \quad (17)$$

The set of valid inequalities (15) are added to  $[SP_1^1]$  to ensure that enough facilities are activated in the second stage to satisfy the total demand. Adding constraints (16) to  $[SP_1^1]$  ensure that the transportation channels into a demand facility have enough capacity to satisfy the demand. The set of valid inequalities (17) are added to  $[SP_2^1]$  and state that for each facility location  $i \in V$ , the total flow out of  $i$  must not exceed its supply capacity. These constraints are redundant in the original formulation, but become active after applying Lagrangian relaxation. The downside is that the subproblems are no more trivial to solve, and the algorithms described above are not used after adding these cuts. In

fact, after adding constraints (15),  $[SP_1^1]$  does not decompose by facility any more.  $[SP_2^1]$  however remains separable by scenario after adding constraints (17).

### 3.5 Calculating the Lagrangian Bound

The Lagrangian bound is found using Kelly's cutting plane algorithm (Kelly, 1960). At each iteration, the method solves subproblems  $[SP_1^1]$  and  $[SP_2^1]$  and uses their solution to generate cuts and add them to the relaxed master problem [RMP]. Subproblems' solutions  $(x_{ij}^{s*}, y_i^{s*}, z_i^*)$  are used to update the sets  $\bar{H}$  and  $\bar{K}$ . At each iteration,  $\theta^1 + \delta^1$  gives a lower bound on the Lagrangian bound while [RMP] produces an upper bound. Kelly's cutting plane method (1960) is shown in the following algorithm.

---

#### Cutting plane algorithm

**step 1** Initialize  $\lambda_i^s$ , set  $t = 0$  and  $LB(Lag) = 0$  and  $UB(Lag) = \infty$ , define  $Gap(Rel) = \frac{UB(Lag) - LB(Lag)}{UB(Lag)}$ .

While  $Gap(Rel) > 0.001$

**step 2** Solve  $[SP_1^1]$  and  $[SP_2^1]$  at  $\lambda_i^s$ , use the optimal solution  $x_{ij}^{s*}, y_i^{s*}, z_i^*$  to find  $\theta^1, \delta^1$ .

**step 3** Update the Lagrangian lower bound  $LB(Lag) = \max(LB(Lag), z_{SP_1^1} + z_{SP_2^1})$ , update  $\bar{H}$  and  $\bar{K}$  with the new integer points  $x_{ij}^{s*}, y_i^{s*}, z_i^*$ .

**step 4** Solve [RMP], find Lagrangian multipliers  $\lambda_i^s$  and update  $UB(Lag) = \theta^1 + \delta^1$ .

**step 5** Update  $t = t + 1$  and  $\lambda_i^s$ .

---

With the Lagrangian relaxation described in this section, we may develop three implementations to calculate the Lagrangian bound due to the possibility of solving the subproblems in their aggregated or disaggregated format. The implementation issues are described in the following section.

## 3.6 Aggregated, Disaggregated and Hybrid Implementations

The aggregated implementation is when the aggregated format of  $[SP_1^1]$  and  $[SP_2^1]$  (described in Section 3.3) is used for solving the subproblems. Regarding this problem formulation, this method results in adding 2 cuts to [RMP] at each iteration of the cutting plane algorithm.

The disaggregated case benefits from the fact that  $[SP_1^1]$  is separable by facility  $i \in V$  and  $[SP_2^1]$  is separable by scenario  $s \in S$ . We define  $\theta_i^1$  as the objective value of  $[SP_1^1]$

for facility  $i \in V$  and  $\delta_s^1$  as that of  $[SP_2^1]$  in scenario  $s \in S$ .  $H_i$  and  $K_s$  are the index sets of integer points in feasible regions of  $[SP_1^1]$  for facility  $i \in V$  and  $[SP_2^1]$  in scenario  $s \in S$  respectively. We solve the relaxed master problem over the subsets  $\overline{H_i}$  and  $\overline{K_s}$ . The relaxed master problem in disaggregated format is:

$$\begin{aligned}
\max \quad & \sum_{i \in V} \theta_i^1 + \sum_{s \in S} \delta_s^1 \\
\text{s.t.} \quad & \theta_i^1 \leq f_i z_i^h + \sum_{s \in S} (p^s g_i^s - \lambda_i^{s^h} b_i^s) g_i^{s^h} \quad \forall i \in V, \forall h \in \overline{H_i} \\
& \delta_s^1 \leq \sum_{i \in V} \sum_{j \in V} (p^s c_{ij}^s + \lambda_i^{s^h}) d_j^s x_{ij}^{s^h} \quad \forall s \in S, \forall h \in \overline{K_s} \\
& \lambda_i^s \geq 0 \quad \forall i \in V, \forall s \in S \\
& \theta_i^1, \delta_s^1 \quad \text{unrestricted in sign}
\end{aligned}$$

In this case at each iteration of the cutting plane algorithm  $|V| + |S|$  cuts are added to [RMP], so it will increase in size significantly and takes more time to solve at each iteration. But at the same time it is expected that adding more cuts to [RMP] will cause the problem to converge to Lagrangian bound in fewer iterations.

Another possible implementation is the hybrid approach where  $[SP_1^1]$  is aggregated and  $[SP_2^1]$  is disaggregated by scenario  $s \in S$ . The resulting [RMP] for hybrid implementation is:

$$\begin{aligned}
& \max \quad \theta^1 + \sum_{s \in S} \delta_s^1 \\
& \text{s.t.} \\
& \theta^1 \leq \sum_{i \in V} f_i z_i^h + \sum_{s \in S} \sum_{i \in V} (p^s g_i^s - \lambda_i^{s^h} b_i^s) y_i^{s^h} \quad h \in \overline{H} \\
& \delta_s^1 \leq \sum_{i \in V} \sum_{j \in V} (p^s c_{ij}^s + \lambda_i^{s^h}) d_j^s x_{ij}^{s^h} \quad \forall s \in S, \forall h \in \overline{K}_s \\
& \lambda_i^s \geq 0 \quad \forall i \in V, \forall s \in S \\
& \theta^1, \delta_s^1 \quad \text{unrestricted in sign}
\end{aligned}$$

In this case  $|S| + 1$  cuts are added to RMP in each iteration.

### 3.7 Lagrangian Heuristic

Lagrangian relaxation determines a lower bound on the solution of [P]. At the same time, useful information is generated when solving the subproblems and relaxed master problem repeatedly, which can be used to build feasible solutions and an upper bound on [P]. We propose here a heuristic based on the second subproblem  $[SP_2^1]$ .

The heuristic makes facility location decisions  $(z_i, y_i^s)$  based on the flow variables  $x_{ij}^s$

after solving  $[SP_2^1]$ . Non-zero values of flow on arc  $(i, j) \in A$  in scenario  $s \in S$  require that facility  $i$  ( $i \in V$ ) should be located in the first stage ( $z_i = 1$ ) and activated in the second stage ( $y_i^s = 1$ ). This heuristic will always lead to a feasible solution to [P].

The feasible solution generated from the heuristic may be further improved to lead us to a tighter upper bound. After applying the heuristic and activating facility  $i \in V$  in scenario  $s \in S$ , we need to make sure that this facility is properly used. In other words, if a located facility is working with a small percentage of its capacity in scenario  $s \in S$ , we would close it ( $z_i = 0, y_i^s = 0$ ) and transfer the flow out of it to an open facility with minimum shipping cost.

This heuristic improvement may be further expanded by transferring the flow out of already activated facilities to the cheapest transportation channels as long as this process improves the objective function value. However it should be noted that there is a limit on the capacity of each transportation channel  $(i, j) \in A$  that may not be violated, so the heuristic improvement keeps transferring flow until either the capacity ( $u_{ij}^s$ ) is reached or this process no longer improves the objective function value.

## 3.8 Complete Algorithm Based on Relaxation I

The complete algorithm to solve our problem using relaxation I is presented in this section. The solution starts by solving the subproblems and finding lower bound on the Lagrangian bound ( $LB(Lag)$ ). The Lagrangian multipliers are found by solving [RMP] and are fed back into the subproblems to start a new iteration. [RMP] sets an upper bound on the Lagrangian bound  $UB(Lag)$ . This process will continue until the relative gap between the upper and lower bound on Lagrangian bound ( $Gap(Rel)$ ) is small enough. Since [P] is minimization problem, the Lagrangian bound found in this process will be a lower bound on the optimal solution to [P] ( $LB[P]$ ). An incumbent and a feasible solution of [P] may be found from the improvement heuristic. This process is presented in detail below and in Figure 3.1.

---

### Complete algorithm - Relaxation I

**step 1** Initialize  $\lambda_i^s$ , set  $count = 0, LB(Lag) = 0$  and  $UB(Lag) = \infty$ .

While  $Gap(Rel) > 0.001$

**step 2** Solve  $[SP_1^1]$  and  $[SP_2^1]$  at  $\lambda_i^s$ , use the optimal solution  $x_{ij}^{s*}, y_i^{s*}, z_i^*$  to find  $\theta^1, \delta^1$ .

**step 3** Update the Lagrangian lower bound  $LB(Lag) = \max(LB(Lag), z_{SP_1^1} + z_{SP_2^1})$ ,

update  $\bar{K}$  and  $\bar{H}$  with the new integer points  $x_{ij}^{s*}, y_i^{s*}, z_i^*$ .

**step 4** Solve improvement heuristic from  $[SP_2^1]$ , get  $z_i^{Heu}$  and  $y_i^{sHeu}$ , update incumbent.

**step 5** Solve [RMP], find Lagrangian multipliers  $\lambda_i^s$  and update  $UB(Lag)$ .

**step 6** Update  $count = count + 1$  and  $Gap(Rel)$

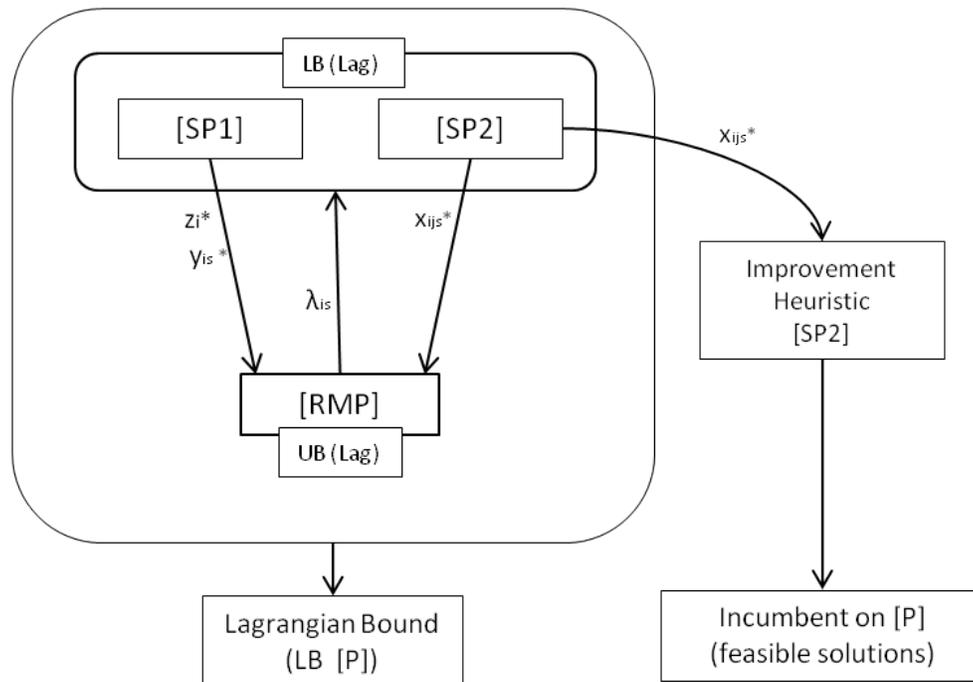


Figure 3.1: Complete Algorithm - Relaxation I

# Chapter 4

## Relaxation II

In this Chapter another Lagrangian relaxation is applied to the two-stage stochastic facility location problem [P] presented in Section 3.1. The Lagrangian relaxation formulation is derived and the methodology to solve the subproblems and generate feasible solutions is detailed.

### 4.1 Lagrangian Relaxation II

As discussed in Section 3.1, another relaxation approach to [P] would be to relax constraints (2). These constraints are the only ones that link the first stage location decision variables  $z_i$  to the second stage activation variables  $y_i^s$ . Relaxing constraint (2) with non-

negative Lagrangian multipliers  $\omega_i^s$  will result in the following Lagrangian problem:

$$L(\omega_i^s) \quad \min \quad \sum_{i \in V} f_i z_i + \sum_{s \in S} p^s \left( \sum_{i \in V} g_i^s y_i^s + \sum_{j \in V} \sum_{i \in V} c_{ij}^s d_j^s x_{ij}^s \right) + \sum_{s \in S} \sum_{i \in V} \omega_i^s (y_i^s - z_i)$$

s.t.

$$d_j^s x_{ij}^s \leq u_{ij}^s \quad \forall (i, j) \in A, \forall s \in S \quad (18)$$

$$\sum_{i \in V} x_{ij}^s = 1 \quad \forall j \in V, \forall s \in S \quad (19)$$

$$\sum_{j \in V} x_{ij}^s d_j^s \leq b_i^s y_i^s \quad \forall i \in V, \forall s \in S \quad (20)$$

$$0 \leq x_{ij}^s \leq 1 \quad \forall (i, j) \in A, \forall s \in S \quad (21)$$

$$z_i \in \{0, 1\} \quad \forall i \in V \quad (22)$$

$$y_i^s \in \{0, 1\} \quad \forall i \in V, \forall s \in S \quad (23)$$

Constraints (18-21) and (23) concern the second-stage activation variables  $y_i^s$  and demand allocation variables  $x_{ij}^s$  while only the binary constraints (21) concern the first-stage location variables  $z_i$ . Based on this observation, the Lagrangian problem decomposes into two subproblems  $[SP_1^2]$  and  $[SP_2^2]$ .

$$\begin{aligned}
[SP_1^2] \quad & \min \quad \sum_{i \in V} (f_i - \sum_{s \in S} \omega_i^s) z_i \\
& \text{s.t.} \\
& z_i \in \{0, 1\} \quad \forall i \in V
\end{aligned}$$

and

$$\begin{aligned}
[SP_2^2] \quad & \min \quad \sum_{s \in S} p^s \sum_{i \in V} g_i^s y_i^s + \sum_{s \in S} \sum_{i \in V} \omega_i^s y_i^s + \sum_{s \in S} p^s \sum_{j \in V} \sum_{i \in V} c_{ij}^s d_j^s x_{ij}^s \\
& \text{s.t.} \quad d_j^s x_{ij}^s \leq u_{ij}^s \quad \forall (i, j) \in A, \forall s \in S \\
& \quad \sum_{i \in V} x_{ij}^s = 1 \quad \forall j \in V, \forall s \in S \\
& \quad \sum_{j \in V} x_{ij}^s d_j^s \leq b_i^s y_i^s \quad \forall i \in V, \forall s \in S \\
& \quad 0 \leq x_{ij}^s \leq 1 \quad \forall (i, j) \in A, \forall s \in S \\
& \quad y_i^s \in \{0, 1\} \quad \forall i \in V, \forall s \in S
\end{aligned}$$

$[SP_1^2]$  is an integer program that determines which facilities to locate in the first stage of decision making and is straight forward to solve.  $[SP_2^2]$  is a mixed-integer program (MIP) that is decomposable by scenario  $s \in S$ .

Let  $\theta^2$  be the objective value of  $[SP_1^1]$  and  $\delta^2$  be that of  $[SP_2^2]$ . For any given value of  $\omega_i^s \geq 0$ ,  $\theta^2 + \delta^2$  is a lower bound on the objective value of [P]. The Lagrangian bound is the maximum of  $\theta^2 + \delta^2$  over  $\omega_i^s \geq 0$ . To calculate the Lagrangian bound Kelly's cutting plane algorithm is used as described in Section 3.3.

The Lagrangian Master problem (MP) is defined as the problem of finding the maximum of  $\theta^2 + \delta^2$  over all the integer solutions in the feasible region of  $[SP_1^1]$  and  $[SP_2^2]$ , and has the following format for this relaxation:

$$\begin{aligned}
[\text{MP}] \quad & \max \quad \theta^2 + \delta^2 \\
& \text{s.t.} \quad \theta^2 + \sum_{i \in V} \sum_{s \in S} \omega_i^s z_i^l \leq \sum_{i \in V} f_i z_i^l && \forall l \in L \\
& \quad \delta^2 - \sum_{s \in S} \sum_{i \in V} y_i^{s^l} \omega_i^s \leq \sum_{s \in S} \sum_{i \in V} p^s g_i^s y_i^{s^m} + \sum_{s \in S} \sum_{i \in V} \sum_{j \in V} p^s c_{ij}^s d_j^s x_{ij}^{s^m} && \forall m \in M \\
& \quad \omega_i^s \geq 0 && \forall i \in A, \forall s \in S \\
& \quad \theta^2, \delta^2 \quad \text{unrestricted in sign}
\end{aligned}$$

Where  $L$  and  $M$  are the sets of integer points in the feasible regions of  $[SP_1^2]$  and  $[SP_2^2]$  respectively. Since  $L$  and  $M$  are not known beforehand, we start by solving a relaxed

master problem defined on small subsets of  $L$  and  $M$ ,  $\bar{L}$  and  $\bar{M}$ .

$$\begin{aligned}
[\text{RMP}] \quad & \max \quad \theta^2 + \delta^2 \\
& \text{s.t.} \quad \theta^2 + \sum_{i \in V} \sum_{s \in S} \omega_i^s z_i^l \leq \sum_{i \in V} f_i z_i^l && \forall l \in \bar{L} \\
& \quad \delta^2 - \sum_{s \in S} \sum_{i \in V} y_i^{s^m} \omega_i^s \leq \sum_{s \in S} \sum_{i \in V} p^s g_i^s y_i^{s^m} + \sum_{s \in S} \sum_{i \in V} \sum_{j \in V} p^s c_{ij}^s d_j^s x_{ij}^{s^m} && \forall m \in \bar{M} \\
& \quad \omega_i^s \geq 0 && \forall i \in A, \forall s \in S \\
& \quad \theta^2, \delta^2 \quad \text{unrestricted in sign}
\end{aligned}$$

Solving [RMP] determines the Lagrangian multipliers  $\omega_i^s$ . The latter are used to modify and solve the subproblems  $[SP_1^2]$  and  $[SP_2^2]$ .

## 4.2 Solution of Subproblems

Subproblem  $[SP_1^2]$  is a 0 – 1 integer program that is easy to solve. It determines which facilities to locate in the first stage. For each location  $i \in V$ , a facility will open ( $z_i = 1$ ) if  $f_i < \sum_{s \in S} \omega_i^s$  and 0 otherwise. So  $[SP_1^2]$  can be further disaggregated into  $|V|$  subproblems,

one for each  $i \in V$ . We note that these subproblems are not necessarily identical but the same solution procedure applies.

Subproblem  $[SP_2^2]$  is a MIP that has the structure of a fixed-charge capacitated facility location problem for each scenario  $s \in S$  (Nozick, 2001).  $[SP_2^2]$  also may be decomposed by scenario  $s \in S$ , the resulting subproblems are not necessarily identical.

In this formulation, unlike the first formulation in Section 3.4,  $[SP_2^2]$  is not easy to solve so we expect a stronger Lagrangian bound that is closer to the IP solution than the LP relaxation. In order to further improve the Lagrangian bound however, valid inequalities are derived to add to the first subproblem  $[SP_1^2]$ .

$$\sum_{i \in V} b_i^s z_i \geq \sum_{j \in V} d_j^s \quad \forall s \in S \quad (24)$$

$$\sum_{i \in V} u_{ij}^s z_i \geq d_j^s \quad \forall j \in V, \forall s \in S \quad (25)$$

The set of valid inequalities (24) are added to  $[SP_1^2]$  to ensure that enough facilities are located in the first stage to satisfy the total demand in each scenario  $s \in S$ . After adding constraints (24),  $[SP_1^2]$  does not decompose by facility and is not easy to solve any more. Inequalities (25) state that in each scenario, we need to locate enough facilities so that the sum of arc capacities into a demand location  $j \in V$  will be at least as large as its demand.

### 4.3 Calculating the Lagrangian Bound

The Lagrangian bound is found using Kelly's (1960) cutting plane. The same procedure described in Section 3.5 applies here. At each iteration  $t$ , subproblems  $[SP_1^2]$  and  $[SP_2^2]$  are solved and their solution is used to generate cuts and add them to the relaxed master problem [RMP]. Subproblems' solutions  $(x_{ij}^{s*}, y_i^{s*}, z_i^*)$  are used to update the sets  $\bar{L}$  and  $\bar{M}$ . At each iteration,  $\theta^2 + \delta^2$  gives a lower bound on the Lagrangian bound while the relaxed master problem solution produces an upper bound. The algorithm stops when  $Gap(Rel)$  is small enough.

With the problem formulation described in this section, we can develop two implementations due to the possibility of solving  $[SP_2^2]$  in its aggregated or disaggregated format. These implementations are described in the following section.

### 4.4 Aggregated and Disaggregated Implementations

The aggregated implementation is when the aggregated format of  $[SP_1^2]$  and  $[SP_2^2]$ , described in Section 4.2 is used which results in adding 2 cuts to [RMP] at each iteration.

The disaggregated case benefits from the fact that  $[SP_2^2]$  is separable by scenario  $s \in S$ .

The relaxed master problem in disaggregated implementation is:

$$\begin{aligned}
\max \quad & \theta^2 + \sum_{s \in S} \delta_s^2 \\
\text{s.t.} \quad & \theta^2 \leq \sum_{i \in V} (f_i - \sum_{s \in S} \omega_i^s) z_i^l \quad \forall i \in V, \forall l \in \bar{L} \quad (26) \\
& \delta_s^2 \leq \sum_{i \in V} (p^s g_i^s + \omega_i^{s^m}) y_i^{s^m} + \sum_{i \in V} \sum_{j \in V} p^s c_{ij}^2 d_j^s x_{ij}^{s^m} \quad \forall s \in S, \forall m \in \overline{M}_s \quad (27) \\
& \omega_i^s \geq 0 \quad \forall i \in V, \forall s \in S \\
& \theta^2, \delta_s^2 \quad \text{unrestricted in sign}
\end{aligned}$$

In this formulation, at each iteration of the cutting plane algorithm  $|S| + 1$  cuts are added to [RMP]. So [RMP] increases in size that results in more time spent on it at each iteration of the algorithm but at the same time we expect it to converge to the Lagrangian bound in fewer iterations.

## 4.5 Lagrangian Heuristics

Lagrangian relaxation produces a lower bound on the solution to the original MIP ([P]) that is presented by  $LB[p]$ . However, the information generated at each iteration of the cutting plane algorithm may be used to build feasible solutions and an incumbent on the solution of [P]. We propose two heuristics based on the solution of subproblems  $[SP_1^2]$  and  $[SP_2^2]$  here.

### 4.5.1 Heuristic from $[SP_1^2]$

Solving  $[SP_1^2]$  provides a set of located facilities on  $i \in V$  in the first stage. In attempt to get the second-stage activation ( $y_i^s$ ) and demand allocation ( $x_{ij}^s$ ) values from the solution of  $[SP_1^2]$  ( $z_i = 1$ ), a heuristic is formed. The heuristic solves a problem similar to  $[SP_2^2]$ , with an extra constraint that limits the activated facilities to those already located in  $[SP_1^2]$ .

The model to be solved is described as follows.

$$\begin{aligned}
\min \quad & \sum_{s \in S} p^s \sum_{i \in V} g_i^s y_i^s + \sum_{s \in S} p^s \sum_{j \in V} \sum_{i \in V} c_{ij}^s d_j^s x_{ij}^s \\
\text{s.t.} \quad & d_j^s x_{ij}^s \leq u_{ij}^s && \forall (i, j) \in A, \forall s \in S \\
& \sum_{i \in V} x_{ij}^s = 1 && \forall j \in V, \forall s \in S \\
& \sum_{j \in V} x_{ij}^s d_j^s \leq b_i^s y_i^s && \forall i \in V, \forall s \in S \\
& 0 \leq x_{ij}^s \leq 1 && \forall (i, j) \in A, \forall s \in S \\
& y_i^s \in \{0, 1\} && \forall i \in V, \forall s \in S
\end{aligned}$$

With a set of extra constraints:

$$y_i^s \leq \bar{z}_i \quad \forall i \in (\text{facilities located in } [SP_1^2]) \quad (28)$$

It should be noted that adding constraints (28) to  $[SP_2^2]$  may make it infeasible to solve in some scenarios but once feasible, it generates  $y_i^{sHeu}$  and  $x_{ij}^{sHeu}$ . These values may be added to [RMP] as a set of extra cuts in the following format.

$$\delta_s^2 \leq \sum_{i \in V} (p^s g_i^s + \omega_i^s) y_i^{sHeu} + \sum_{i \in V} \sum_{j \in V} p^s c_{ij}^s d_j^s x_{ij}^{sHeu} \quad (29)$$

$y_i^{sHeu}$  and  $x_{ij}^{sHeu}$  together with  $\bar{z}_i = 1$  from  $[SP_1^2]$  form a feasible solution and an incumbent on [P].

The effect of adding these cuts to [RMP] is studied in Section 5.4.

#### 4.5.2 Heuristic from $[SP_2^2]$

Solution of  $[SP_2^2]$  at each iteration finds activated facilities as well as the demand allocation variables in each scenario. We propose to make first-stage facility location decision based on this information. Non-zero values of  $y_i^s$  for location  $i \in V$  in scenario  $s \in S$  require that facility  $i \in V$  be located in the first stage ( $z_i^{Heu} = 1$ ). This heuristic will always lead to a feasible solution to [P] and an incumbent on [P]. This process will also generate an extra set of cuts that is added to [RMP] at each iteration of the cutting plane algorithm.

$$\theta^2 \leq \sum_{i \in V} (f_i - \sum_{s \in S} \omega_i^s) z_i^{Heu} \quad (30)$$

The effect of adding this set of cuts to [RMP] is studied in Section 5.4.

## 4.6 Complete Algorithm Based on Relaxation II

We describe the complete algorithm to solve our problem using relaxation II in this section. The process starts by solving the subproblems which finds a lower bound on the Lagrangian bound ( $LB(Lag)$ ). Using solution of the subproblems, we generate cuts by heuristics from  $[SP_1^2]$  and  $[SP_2^2]$  to add to [RMP]. The solution of [RMP] is fed back into the subproblems to start a new iteration. Also it sets an upper bound on the Lagrangian bound  $UB(Lag)$ . This process will continue until a relative gap between the upper and lower bound on Lagrangian bound is reached. The final Lagrangian bound found by this method is itself a lower bound on the original problem [P] and we show it by  $LB[P]$ . At the same time, the heuristics produce an incumbent. This method is illustrated in the following algorithm as well as Figure 4.1.

---

### Complete algorithm - Relaxation II

**step 1** Initialize  $\omega_i^s$ , set  $count = 0$ ,  $LB(Lag) = 0$  and  $UB(Lag) = \infty$ .

While  $Gap(Rel) > 0.001$

**step 2** Solve  $[SP_1^2]$  and  $[SP_2^2]$  at  $\omega_i^s$ , use the optimal solution  $x_{ij}^{s*}, y_i^{s*}, z_i^*$  to find  $\theta^2, \delta^2$ .

**step 3** Update the Lagrangian lower bound  $LB(Lag) = \max(LB(Lag), z_{SP_1^2} + z_{SP_2^2})$ , update  $\bar{L}$  and  $\bar{M}$  with the new integer points  $x_{ij}^{s*}, y_i^{s*}, z_i^*$ .

**step 4** Solve heuristic from  $[SP_1^2]$ , get  $x_{ij}^{sHeu}, y_i^{sHeu}$ , update incumbent, add cuts (29) to [RMP].

**step 5** Solve heuristic from  $[SP_2^2]$ , get  $z_i^{Heu}$ , update incumbent, add cuts (30) to [RMP].

**step 6** Solve [RMP], find Lagrangian multipliers  $\omega_i^s$  and update  $UB(Lag)$ .

**step 7** Update  $count = count + 1$  and  $Gap(Rel)$

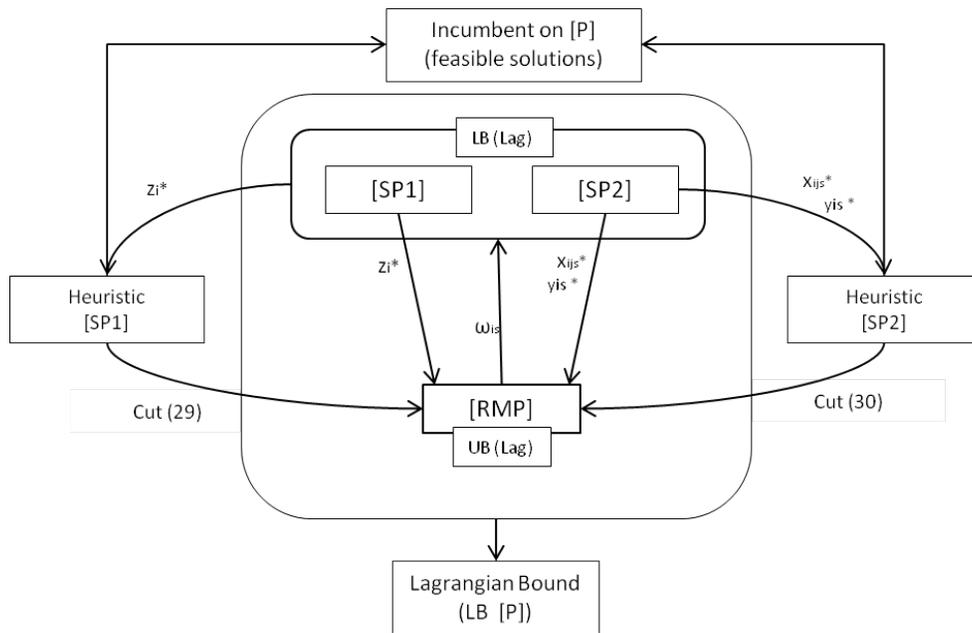


Figure 4.1: Complete Algorithm - Relaxation II

# Chapter 5

## Implementation and Numerical

## Testing

In this Chapter, the data generation process is discussed and a method is introduced to check their feasibility. The Lagrangian relaxations described in Chapters 3 and 4 are implemented on the data sets to verify the effectiveness of these methods. Sections 3-6 of this Chapter show the process of experimenting different variations of the methods on small instances. Once the best approach is found, further tests are done on larger instances in Section 5.7. All computations are performed on a MAC computer with Intel (R) Core i5 CPU @ 2.30 GHz, 4.00 GB RAM and MATLAB 7.9 is the programming language used. The linear problems were solved using LINPROG [25], a built-in MATLAB LP solver and

the integer problems were solved using GUROBI 4.6 [26], a mixed-integer programming solver.

## 5.1 Generating Test Instances

In order to test our models and compare the results to previous work (Penuel et. al., 2010), three data profiles were generated (Table 1). Profile A has expensive location/activation costs while profile B has large shipping costs therefore more is spent on shipping than locating. Instances generated in profile C have equal range of cost for total location/activation and shipping. Table 1 presents the range of possible values for the costs, demands and supplies and capacities in each profile. Note that  $0_n$  for demand values shows a set of  $n$  zeros. For instance, the set of demand in profile B shows that each location has a 75% chance of having no demand and a 25% chance of receiving a unit of demand. Also the priorities are assigned to scenarios such that each  $p^s$  is first assigned a weight of 1 or 2, then they are all normalized to one ( $\sum_{s \in S} p^s = 1$ ). The instances in each profile are generated randomly. We generate two sets of instances with different sizes of locations and scenarios. The “Small” instances have 20 locations and 10 scenarios while the “Large” instances in each profile have 40 locations and 50 scenarios. Small instances are used to evaluate our methods and determine the best approach to tackle the Large ones. To verify our results,

Profiles	A	B	C
Location Cost ( $f_i$ )	$\{100, \dots, 250\}$	$\{50, 60, 70\}$	$\{4, 5\}$
Activation Cost ( $g_i^s$ )	$\{100, \dots, 250\}$	$\{40, 50\}$	$\{4, 5\}$
Supply ( $b_i^s$ )	$\{100, \dots, 200\}$	$\{20, 30\}$	$\{200, \dots, 400\}$
Demand ( $d_j^s$ )	$0_{75} \cup \{1, \dots, 10\}$	$0_3 \cup \{1\}$	$0_{150} \cup \{100, \dots, 150\}$
Arc Capacity ( $u_{ij}^s$ )	$\{25, \dots, 75\}$	$\{2, 3, 4, 5\}$	$0_{50} \{150, \dots, 250\}$
Arc Cost ( $c_{ij}^s$ )	$\{1, \dots, 10\}$	$\{10, 15, 20\}$	$\{0.01, 0.02, 0.03\}$

Table 5.1: Possible values for test instances.

we generate 10 instances for each profile and report the average value. For large instances, we use the same instances as Penuel et. al. (2010).

Also in each profile two different arc densities are considered: 10% and 80%. Arc density is defined as the proportion of arcs with non-zero capacity and is introduced to the program the same as demand in Table 1. For example, when forming an instance of 40 facilities and 50 scenarios, there are 1560 ( $40 \times 39$ ) possible arcs in the system connecting all the locations together. Now with arc density of 10%, only 156 arcs of this set will have non-zero capacities, hence carry flow in the system (with arc density of 80% this value will reach 1248 arcs).

The instances are designed so that the supply and demand nodes are separated in the sense that a single node may be assigned either a demand or supply value. This assumption was used by Penuel et. al. (2010) and we kept it for a fair comparison.

In the following sections as we present our results, the notation in Table 2 is used.

## 5.2 Checking Feasibility of Instances

Once the instances are generated, their feasibility is checked by the following methods. An initial feasibility check would be to make sure that there is enough supply in the system to satisfy the total demand.

$$\sum_{i \in V} b_i^s \geq \sum_{j \in V} d_j^s \quad \forall s \in S$$

Another method is to check if the sum on capacities of all arcs incoming to a demand facility is actually greater than the demand value itself.

$$\sum_{i \in V} u_{ij}^s \geq d_j^s \quad \forall j \in V, \forall s \in S$$

However an instance passing the previous tests is not guaranteed to be feasible unless the following process is done.

Assuming that all the facilities are located in the first stage and activated in the second one, we should be able to allocate demand through transportation channels so that the

CPUSP11	Average CPU seconds to solve $[SP_1^1]$
CPUSP12	Average CPU seconds to solve $[SP_2^1]$
CPUSP21	Average CPU seconds to solve $[SP_1^2]$
CPUSP22	Average CPU seconds to solve $[SP_2^2]$
CPUT	Total CPU seconds to solve an instance
CPUITER	Average CPU seconds per iteration
ITER	Average number of iterations of algorithm to solve an instance
GAP	Average gap : $\frac{\text{Lagrangian bound} - \text{optimal solution}}{\text{Lagrangian bound}} \times 100$
AGG	Aggregated implementation (Sections 3.6 and 4.4)
DISAG	Disaggregated implementation (Sections 3.6 and 4.4)
Small Instance	20 locations, 10 scenarios
Large Instance	40 locations, 50 scenarios
Rel I	Relaxation I (Chapter 3)
Rel II	Relaxation II (Chapter 4)
A1,B1,C1	Data profiles with 10% arc density
A2,B2,C2	Data profiles with 80% arc density
IncRelI	Incumbent derived from $[SP_2^1]$ in Rel I
IncRelII-1	Incumbent derived from $[SP_1^2]$ in Rel II
IncRelII-2	Incumbent derived from $[SP_2^2]$ in Rel II
GAP(Inc)	Average gap : $\frac{\text{Incumbent} - \text{Lagrangian Bound}}{\text{Incumbent}} \times 100$

Table 5.2: Notation used in this study.

demand is satisfied in each scenario. In terms of the mathematical model, we should solve the original MIP in Section 3.1 ([P]) with additional assumption of  $z_i = 1(\forall i \in V)$  and  $y_i^s = 1(\forall i \in V, \forall s \in S)$ . As a result the following linear programming problem should be solved to assure feasibility of instances:

$$\begin{aligned}
\min \quad & \sum_{s \in S} \sum_{j \in V} \sum_{i \in V} p^s c_{ij}^s d_j^s x_{ij}^s \\
\text{s.t.} \quad & \\
& d_j^s x_{ij}^s \leq u_{ij}^s & \forall (i, j) \in A, \forall s \in S \\
& \sum_{i \in V} x_{ij}^s = 1 & \forall j \in V, \forall s \in S \\
& \sum_{j \in V} x_{ij}^s d_j^s \leq b_i^s & \forall i \in V, \forall s \in S \\
& 0 \leq x_{ij}^s \leq 1 & \forall (i, j) \in A, \forall s \in S
\end{aligned}$$

This is a linear problem that may be solved for each scenario  $s \in S$  separately.

### 5.3 Effect of Adding Cuts to the Subproblems

We start our computational testing by Lagrangian relaxation I discussed in Section 3.3. In Section 3.4, we proposed to add extra cuts to the subproblems to improve the Lagrangian bound. In this section, we evaluate the effect of adding cuts (15), (16), (17) to  $[SP_1^1]$  and  $[SP_2^1]$  in relaxation I.

We examine our model on small instances from all data profiles and arc densities available. A comparison is made in the time spent in each subproblem before and after adding extra cuts and the quality of bounds in each case is compared to the optimal solution derived from solving [P] directly as a MIP. Table 3 gives the results.

Although adding the extra cuts (15), (16), (17) to subproblems increases the computational time spent in each subproblem, it improves the total computational time and number of iterations by 95% and 90% respectively for profiles A and B. Also it decreases the gap between the Lagrangian bound and the optimal solution by 33%, so we choose to add cuts when finding Lagrangian bound in relaxation I.

The second Lagrangian relaxation discussed in Chapter 4 is to be studied here. We study the effect of adding cuts (24) and (25) (Section 4.2) to  $[SP_1^2]$  on computational time and quality of the Lagrangian bound. The method is experimented on small instances

		CPUSP11	CPUSP12	CPUT	ITER	GAP (%)
A1	No cuts	0.0099	0.1233	1918	2045	87.4
	Cuts added	0.0631	0.1313	10.3	53	57.5
A2	No cuts	0.0056	0.0735	1442	2602	81.7
	Cuts added	0.0640	0.1552	57.9	263.2	29.7
B1	No cuts	0.0094	0.1105	335.9	1079	80.7
	Cuts added	0.0570	0.1345	27	144	63
B2	No cuts	0.0111	0.1315	762.9	1568	67.9
	Cuts added	0.0643	0.1192	46.9	233	35.7
C1	No cuts	0.0120	0.1438	3.19	18.75	46.9
	Cuts added	0.0175	0.1441	4.16	23	41.5
C2	No cuts	0.0135	0.1641	8.29	42.4	58.5
	Cuts added	0.0192	0.1437	7.02	38.9	47.1

Table 5.3: Rel I. Effect of adding cuts (15),(16),(17) to  $[SP_1^1],[SP_2^1]$  .

of profile A2 where a facility is assigned either a supply or demand value. Table 4 is a presentation of this experiment.

Table 4 states that adding cuts (24) to  $[SP_1^2]$  in relaxation II only decreases the computational time by 5%. However, cuts (25) improve the gap between the Lagrangian bound and the optimal solution by 93% and reduce the total computational time by approximately 42%. So we choose to add cuts (24) and (25) to  $[SP_1^2]$  when implementing relaxation II in this study.

## 5.4 Comparison of Aggregated, Disaggregated and Hybrid Implementations

Another variation in relaxation I is the the three approaches in finding the Lagrangian bound as discussed in Section 3.6. In order to evaluate the aggregated, disaggregated and hybrid approaches, we test them on our Small instances of each data profile and the results are reported in Table 5.

These results confirm our discussion in Section 3.6. The number of cuts added to [RMP]

Profile	Implementation	CPUSP21	CPUT	ITER	GAP (%)
A1	No Cuts	8E-05	141.68	394.6	5.25
	Cuts (24)	0.01	132.70	429.4	6.31
	Cuts (24),(25)	0.03	111.69	367.8	0.38
A2	No Cuts	1E-04	829.6	1368	2.85
	Cuts (24)	0.0108	789.5	1372	2.85
	Cuts (24),(25)	0.032	374	948	0.21
B1	No Cuts	7E-05	99.29	245.6	5.34
	Cuts (24)	0.14	114.76	258.8	5.37
	Cuts (24),(25)	0.04	63.20	186.4	0.40
B2	No Cuts	9E-05	432.97	784.6	9.87
	Cuts (24)	0.01	425.86	759.75	7.02
	Cuts (24),(25)	0.03	333.28	733.2	0.56
C1	No Cuts	1E-04	2.42	11.5	0
	Cuts (24)	0.01	2.75	12.75	0
	Cuts (24),(25)	0.01	1.02	4.75	0
C2	No Cuts	6E-05	8.75	39.6	0
	Cuts (24)	0.01	7.91	34.6	0
	Cuts (24),(25)	0.01	3.41	16.8	0

Table 5.4: Rel II. Effect of adding cuts to  $[SP_1^2]$

Profile	Implementation	CPUITER	CPUT	ITER	GAP (%)
A1	AGG	0.19	10.32	53	57.5
	DISAGG	0.23	10.03	42.6	57.8
	HYBRID	0.21	9.47	44	58.6
A2	AGG	0.22	57.89	263.2	29.72
	DISAGG	0.36	21.4	58.7	29.69
	HYBRID	0.28	16.54	63	29.84
B1	AGG	0.19	27.02	144	63.04
	DISAGG	0.26	17.83	69.63	62.96
	HYBRID	0.18	13.63	73.8	63.10
B2	AGG	0.20	46.89	233.4	35.67
	DISAGG	0.32	26.93	84.2	35.60
	HYBRID	0.20	17.84	86.6	35.82
C1	AGG	0.18	4.16	23	41.50
	DISAGG	0.21	4.46	21.5	41.49
	HYBRID	0.19	4.29	22.5	41.52
C2	AGG	0.18	8.28	45.2	45.18
	DISAGG	0.25	7.53	30.2	45.07
	HYBRID	0.19	6.26	32.8	45.24

Table 5.5: Rel I. Comparing AGG,DISAGG,HYBRID implementations.

is the highest in the disaggregated implementation hence more time is spent in each iteration than in the hybrid and aggregated approaches. However, the Lagrangian bound is reached in fewer iterations in the disaggregated implementation. Adding more cuts to [RMP] in the cutting plane algorithm is expected to accelerate reaching the Lagrangian bound but not to improve the gap. Comparing GAP values for these implementations in Table 5 confirms this theory. The disaggregated approach shows 64% decrease in CPU time and 78% improvement in number of iterations while these values reach 72% and 76% for the hybrid approach.

We decide that the hybrid implementation is suitable for our study because it benefits from the decrease in the number of iterations compared to the aggregated approach and at the same time the total computational time is improved over both other approaches.

In Section 4.4, we discussed that the [RMP] in relaxation II may be solved either by aggregated or disaggregated implementation. Also we further explained in Sections 4.5.1 and 4.5.2 that at each iteration of Kelly’s cutting plane algorithm (1960), cuts (29) and (30) may be constructed from the solutions of  $[SP_1^2]$  and  $[SP_2^2]$  to be added to [RMP]. We test these implementations on Small instances of all data profiles and compare the computational time and gap obtained by each implementation in Table 6.

We can see that the disaggregated implementation in relaxation II (Section 4.4) im-

Profile	Implementation	CPUITER	CPUT	ITER	GAP (%)
A1	AGG	0.30	111.69	367.8	0.38
	DISAGG	0.36	17.88	49.2	0
	DISAGG + Cuts	0.52	15.88	30.4	0
A2	AGG	0.375	374	947.6	0.21
	DISAGG	0.387	24.6	61.8	0
	DISAGG + Cuts	0.571	22.3	37.4	0
B1	AGG	0.34	63.20	186.4	0.40
	DISAGG	0.40	14.96	37.4	0.11
	DISAGG + Cuts	0.60	12.95	21.4	0.16
B2	AGG	0.45	333.28	733.2	0.56
	DISAGG	0.53	25.11	47.6	0
	DISAGG + Cuts	0.70	22.68	32.2	0
C1	AGG	0.21	1.02	4.75	0
	DISAGG	0.22	0.61	2.75	0
	DISAGG + Cuts	0.43	0.64	1.5	0
C2	AGG	0.20	4.12	20.25	0
	DISAGG	0.26	2.35	9	0
	DISAGG + Cuts	0.47	1.54	3.25	0

Table 5.6: Rel II. Comparing AGG, DISAGG, DISAGG+Cuts implementations.

proves the computational time by 93% and enables us to close the gap for our test instances. Adding cuts (29) and (30) from the heuristics (Sections 4.5.1 and 4.5.2) will further improve the advantages of disaggregation. Although more time is spent in each iteration (34%) as more cuts are added to [RMP] in disaggregation+cuts approach, the 96% decrease in number of iterations compensate for that and total computational time improves by 94%. So it is best if we choose the disaggregation+cuts approach once implementing relaxation II.

## 5.5 The Effect of Arc Density on the Quality of the Lagrangian Bound

As we generate the instances in each profile with various arc densities, we may experiment both relaxations on them to verify the computational time and the quality of the Lagrangian bound in each case. Based on the results so far, we choose to add extra cuts to subproblems in both relaxations (Sections 3.4 and 4.2). The hybrid approach in relaxation I (Section 3.6) and the disaggregated implementation in relaxation II with the extra cuts from heuristics (Sections 4.4-5) are picked to solve [RMP] as they proved to be our best approaches so far. Tables 7 and 8 summarize the results.

The results in Table 7 show that when implementing relaxation I, increasing arc den-

Arc Density	CPUT	ITER	GAP (%)
A10%	9.34	44	56.2
A80%	16.54	63	29.8
A100%	17.86	86.4	9.5
B10%	15.13	70.6	56.2
B80%	23.49	106.2	29.8
B100%	26	111.6	10.1
C10%	9.34	62.6	36.9
C80%	16.54	63	29.8
C100%	8.15	36.1	41.9

Table 5.7: Rel I. Effect of arc density on Lagrangian bound.

Arc Density	CPUT	ITER	GAP (%)
A10%	20.8	38.1	0
A80%	22.3	37.4	0
A100%	22.2	40.6	0
B10%	19.97	28.2	0
B80%	13.41	16	0
B100%	16.7	28.3	0
C10%	0.75	1.5	0
C80%	2.00	4.2	0.05
C100%	2.53	6.9	0

Table 5.8: Rel II. Effect of arc density on Lagrangian bound.

sity will increase the computational time and number of iterations needed to reach the Lagrangian bound. However, the gap improves by 83% as the network gets more densely packed. Table 8 shows that the GAP is closed for all the instances by implementing relaxation II.

## 5.6 Comparing Lagrangian Relaxations I and II

After experimenting with different variations of each Lagrangian relaxation, we may now compare the effectiveness of relaxations I and II by testing them on a variety of small data instances from different data profiles as discussed in Section 5.1. Table 9 is a summary of the more comprehensive results shown in the appendix.

The results in Table 9 show that Lagrangian relaxation II proves to be more effective in terms of closing the gap in all data profiles and for all arc densities.

As we continue our experiments on larger data instances, the optimal solution to [P] may not be available to measure GAP. So it is important to verify the quality of bounds derived from Lagrangian heuristics in Sections 3.7, 4.5, and 4.6. In Table 10, Lagrangian heuristic derived from  $[SP_2^1]$  in relaxation I ( $H(I)$ ) and from  $[SP_1^2]$  and  $[SP_2^2]$  in relaxation II ( $H(II)^1$  and  $H(II)^2$ ) are compared to the optimal solution. The same data instances from  $A2$  are tested in each relaxation to form a fair comparison.

		<i>A1</i>	<i>A2</i>	<i>B1</i>	<i>B2</i>	<i>C1</i>	<i>C2</i>
CPU	Rel I	9.34	16.54	15.13	23.49	4.55	6.16
	Rel II	20.8	38.1	19.97	13.41	0.75	2.00
ITER	Rel I	44	63	70.6	106.2	22.5	30.6
	Rel II	22.3	37.4	28.2	16	1.5	4.2
GAP (%)	Rel I	56.3	29.8	62.6	36.9	41.5	49.6
	Rel II	0	0	0	0	0	0.05

Table 5.9: Summary of relaxations I and II.

	$H(I)$	$H(II)^1$	$H(II)^2$
GAP(Inc)%	84.13	0	62.72

Table 5.10: Comparing Lagrangian Heuristics in Rel I and Rel II to optimal solutions.

So we choose relaxation II to find Lagrangian bound for the large instances of Penuel et. al (2010) and use the incumbent from  $H(II)^1$  to find GAP(Inc), and the results are presented in Section 5.8.

Profile	ITER	CPUT	GAP(Inc)%
A1	42.4	49.15	0.17
A2	47.8	55.41	0.29
B1	15.8	21.92	0
B2	46.6	66.08	0
C1	1.4	1.15	0
C2	4.4	3.54	0

Table 5.11: Rel II. Instances (n=20,s=20)

## 5.7 The Effect of Number of Scenarios on GAP

Before implementing relaxation II on Large instances with 40 locations and 50 scenarios as in Penuel et. al. (2010), we experiment our method on other instances. For this purpose, instances from all data profiles are generated with the same number of facilities as Small instances (n=20) but different number of scenarios (s=20,30,40). Then our best approach found so far is implemented on them and the results are presented in Tables 11 and 12.

We note that our method is still able to close the GAP for instances with larger number of scenarios. In the following section, the results are presented for Large instances.

Profile	ITER	CPUT	GAP(Inc)%
A1	32.8	49.37	0
A2	53.4	96.73	0
B1	19.2	36.44	0
B2	31.8	71.24	0
C1	35	41.08	0
C2	4.8	4.81	0

Table 5.12: Rel II. Instances (n=20,s=30)

## 5.8 Comparison to Penuel et. al. (2010)

In this section we conclude our study by implementing Lagrangian relaxation II on the same instances as used by Penuel (2010). Large instances are generated from different data profiles. When experimenting with these instances, the final gap is the relative difference between the Lagrangian bound and the incumbent derived from the Lagrangian heuristic described in Section (4.5). Table 13 presents the computational effort in solving these instances.

The GAP(Inc) in Table 13 is derived by the relative difference between the incumbent from Lagrangian heuristic and the Lagrangian bound. However we are able to solve the instances directly and find the GAP(opt) between the Lagrangian bound and the optimal solution. Table 14 presents the results and compares them to Penuel (2010).

	Relaxation II						Penuel (2010)					
	<i>A1</i>	<i>A2</i>	<i>B1</i>	<i>B2</i>	<i>C1</i>	<i>C2</i>	<i>A1</i>	<i>A2</i>	<i>B1</i>	<i>B2</i>	<i>C1</i>	<i>C2</i>
CPUT	978	847	654	1162	1121	1432	185	203	1698	183	2575	3023
ITER	113.9	78.7	57.1	71	79.3	88	17.3	14.6	35.3	14.2	42.8	26.7
GAP(Inc) (%)	1.05	5.83	0.51	4.37	6.21	10.15	0	0	0.72	0	1.93	1.18

Table 5.13: Rel II. Comparing our results to Penuel et. al. (2010).

	Relaxation II						Penuel (2010)					
	<i>A1</i>	<i>A2</i>	<i>B1</i>	<i>B2</i>	<i>C1</i>	<i>C2</i>	<i>A1</i>	<i>A2</i>	<i>B1</i>	<i>B2</i>	<i>C1</i>	<i>C2</i>
GAP(Opt) (%)	1.01	2.81	0.51	1.96	2.43	2.74	0	0	0.72	0	1.93	1.18

Table 5.14: Rel II. Comparing our results (GAP(Opt)) to Penuel et. al. (2010).

Comparing Tables 13 and 14 shows the quality of Lagrangian heuristic derived in Section 4.5. There is a great improvement in the GAP for profiles  $A2$ ,  $B2$ ,  $C1$  and  $C2$  which means the Lagrangian heuristic is not close to the optimal solution for some data profiles and needs to be improved. For the rest of data profile where there is not much of a difference between  $GAP(Inc)$  and  $GAP(Opt)$ , we may conclude that the incumbent is equal to the optimal solution.

## 5.9 Improvement Heuristic

In this section we propose an improvement heuristic based on partial branching. We expect this heuristic to improve the incumbent in Section 5.8 and bring it closer to the optimal solution. In this method, we branch on the number of facilities located in the first stage. Once the problem is solved and an incumbent is reached, the improvement heuristic includes solving the original problem [P] with an additional constraint on the number of location facilities ( $z$ ). We suspect that the optimal solution and the incumbent only differ in the number of facilities located in the first stage ( $z(opt)$  and  $z(inc)$ ). So that we consider three cases:

	Relaxation II						Penuel (2010)					
	A1	A2	B1	B2	C1	C2	A1	A2	B1	B2	C1	C2
GAP(Inc) (%)	1.01	2.81	0.51	1.96	2.43	2.74	0	0	0.72	0	1.93	1.18
CPUT	1436	1347	1103	2274	2158	3071	185	203	1698	183	2575	3023

Table 5.15: Implementing improvement heuristic and comparing GAP(Inc) to Penuel et. al. (2010).

$$\sum_{i \in V} z_i = \sum_{i \in V} z(inc)_i - 1$$

$$\sum_{i \in V} z_i = \sum_{i \in V} z(inc)_i$$

$$\sum_{i \in V} z_i = \sum_{i \in V} z(inc)_i + 1$$

And solve the original problem [P] directly adding one of these constraint at a time. The feasible solution with best objective value is then presented as the incumbent. Table 15 gives the results of implementing this improvement heuristic on Penuel (2010) data instances.

Comparing Tables 14 and 15 proves that although the improvement heuristic increases the computational time by approximately 40%, it generates the optimal solution in all data profiles.

# Chapter 6

## Conclusion

In this study, the purpose was to build exact algorithms based on Lagrangian relaxation to solve the two-stage facility location problem with second-stage activation cost. This problem was studied by Penuel et. al. (2010) and they introduced the Residual Path (RP) cutting-plane approach to estimate the second-stage improvement based on subproblem activation and arc flow data. We modeled the same problem differently by defining a continuous flow decision variable  $x_{ij}^s$  that is a fraction between 0 and 1 and is defined as the fraction of demand in location  $j$  satisfied from facility  $i$  through arc  $(i,j)$  connecting them. This definition led to a different model that was suitable for Lagrangian relaxation. We studied two different relaxations. In the first approach, the constraints that links the second-stage activation and the flow decisions are relaxed. This results in decomposition

of the original problem in two subproblems that are easy to solve. In order to strengthen the Lagrangian bound, sets of cuts are introduced to be added to the subproblems. Also an improvement on Lagrangian heuristic is suggested to get feasible solutions as well as an incumbent.

In the second relaxation, the constraints linking the first-stage location and the second-stage activation variables are relaxed. The problem decomposes into two subproblems where the second one is a fixed-cost capacitated facility location problem. We propose to add extra sets of cuts to the subproblem and it proves to strengthen the Lagrangian bound. A Lagrangian heuristic is derived from each subproblem to generate feasible solutions as well as an incumbent. Moreover, we generate cuts based on Lagrangian heuristics and add them to the relaxed master problem that improve the results significantly.

The experiments are done on small instances using both relaxations and the results are compared. We find that our second approach (relaxation II) is more effective as we are able to solve all the small instances within acceptable computational time. So we continue with this implementation and try it on Penuel et. al. (2010) data instances to verify and compare our method to theirs.

Our method is more promising in solving instances of 10% arc density than the 80% ones. We improve over Penuel et. al. (2010) results in profile B1 in terms of computational time and the gap. However there is room for improvement.

For future work, we suggest to evaluate the performance of our approach with more instances. Also adding branch-and-bound method would be worth exploring to close the gap and reach the exact solution.

# Bibliography

- [1] J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numerical Mathematics*, 4:238-252, 1962.
- [2] D.R. Godschalk. *Disaster mitigation and hazard management*. International City, 1991.
- [3] W.G. Green III. *Four phases of emergency management*. Electronic Encyclopaedia of Civil Defence and Emergency Management.
- [4] J.E. Kelly. The Cutting Plane Method for Solving Convex Problems. *Journal of SIAM*, 1960.
- [5] G. Laporte, F.V. Louveaux, and H. Mercure. The vehicle routing problem with stochastic travel times. *Transportation Science*, 26:161-170, 1992.

- [6] G. Laporte and F.V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, 13:133-142, 1993.
- [7] G. Laporte, F.V. Louveaux and L. Van Hamme. Exact solution to a location problem with stochastic demands. *Transportation Science*, 28(2):95-103, 1994.
- [8] F.Y. Louveaux. Discrete stochastic location models. *Annals of Operations Research*, 6:23-34, 1986.
- [9] F.Y. Louveaux. Stochastic location analysis. *Location Science*, 1(2):127-154, 1993.
- [10] L.K. Nozick. The fixed charge facility location problem with coverage restrictions. *Transportation Research Part E: Logistics and Transportation Review*, 37(4):281-296, 2001.
- [11] J. Penuel, J.C. Smith, and Y. Yuan. An integer decomposition algorithm for solving a two-stage facility location problem with second-stage activation costs. *Naval Research Logistics (NRL)*, 57:391402, 2010.
- [12] J. Rosenhead, M. Elton, and S.K. Gupta. Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly*, 23(4):413-431, 1972.

- [13] S. Sen and J.L. Hige. The  $C^3$  theorem and a  $D^2$  algorithm for large scale stochastic mixed-integer programming: Set convexification. *Mathematical Programming*, 104:1-20, 2005.
- [14] S. Sen and h.D.Sheralli. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106:203-223, 2006.
- [15] E.S. Sheppard. A conceptual framework for dynamic location allocation analysis. *Environment and Planning*, 6:547-564, 1974.
- [16] L.V. Snyder. Facility location under uncertainty: a review. *IIE Transactions*, 38:537-554, 2006.
- [17] S. Thompson, N. Altay, W.G. Green, and J. Lapetina. Improving disaster response efforts with decision support systems. *International Journal of Emergency Management*, 3(4):250-263, 2006.
- [18] S. Tufekci and W.A. Wallace. The emerging area of emergency management and engineering. *IEEE Transactions on Engineering Management*, 45(2):103105, 1998.

- [19] W.L. Waugh and R.J. Hy. *Handbook of Emergency Management: Programs and Policies Dealing with Major Hazards and Disasters*. Greenwood Press, New York, NY, 1990.
- [20] W.L. Waugh. *Living with Hazards, dealing with Disasters: An Introduction to Emergency Management*. Armonk , NY, 2000.
- [21] J.R. Weaver, and R.L. Church. Computational procedures for location problems on stochastic networks. *Transportation Science*, 17(2):168-180, 1983.
- [22] A. Volgenant. Linear and semi-assignment problems: A core oriented approach. *Computers and Operations Research*, 23(10):917-932, 1996.
- [23] W.L. Winston, M. Venkataramanan. *Operations Research: Introduction to Mathematical Programming*. Thompson, 4th Edition, 2003.
- [24] <http://www.who.int>
- [25] <http://www.mathworks.com/help/toolbox/optim/ug/linprog.html>
- [26] <http://www.gurobi.com>

# Appendix A

## Appendix

The results presented in Chapter 5 are average values of the following experiments.

Rel II, Pennel (2010) instances

	nb	GAP(Inc)	CPUT	ITER		nb	GAP(Inc)	CPUT	ITER		nb	GAP(Inc)	CPUT	ITER
A1	1	0	1333	147		1	1.285	433.370	36		1	4.658	1342.49	82
	2	0.940	1673.4	178		2	1.2969	560.344	46		2	5.083	819.649	85
	3	0.874	965.245	116		3	0.5233	535.563	43	C1	3	5.283	1058.50	81
	4	2.731	527.361	78		4	0	1176.306	106		4	8.258	1326.34	76
	5	0	1853.43	203		5	1.0341	490.486	45		5	7.814	1063.57	73
	6	0.790	490.386	71	B1	6	0.3279	514.654	41					
	7	0.893	662.610	77		7	0.5407	593.047	55					
	8	1.146	721.518	88		8	0.1675	859.627	71					
	9	1.226	803.788	94		9	1.2316	421.061	38					
	10	1.870	749.989	87		10	0.2701	644.657	58					

Rel II, Pemuel (2010) instances

	nb	GAP(Inc)	CPUT	ITER		nb	GAP(Inc)	CPUT	ITER		nb	GAP(Inc)	CPUT	ITER
A2	1	13.1	760	64		1	0.9928	995	67		1	9.8755	2410	93
	2	2.73	858	76		2	9.3167	1290	84		2	10.7116	1485	88
	3	0.83	733	72		3	4.4357	1218	69		3	7.2838	1289	91
	4	1.5456	758	77		4	4.009	1331	67		4	9.5732	1471	87
	5	2.7605	750	71	B2	5	5.37	1209	79		5	11.537	1392	109
	6	3.2864	929	87		6	1.7724	1060	67		6	2.9802	1090	86
	7	3.2227	970	91		7	8.4847	1061	68		7	6.8611	996	81
	8	7.6709	1172	113		8	4.0812	1271	71		8	32.0198	1344	82
	9	8.2351	787	74		9	2.3497	1095	64		9	4.1283	1234	79
	10	14.9452	752	62		10	2.9425	1089	74		10	6.5308	1608	84

Rel I. Small instances

	ITER	CPUT	GAP		ITER	CPUT	GAP		ITER	CPUT	GAP
A1	41	8.82	56.7	B1	107	24.11	63.12	C1	28	5.62	43.39
	28	5.24	58.34		48	10.3	59.89		19	3.65	39.85
	42	9.2	53.82		70	13.46	66.22		11	2.16	42.22
	62	12.66	65.76		61	11.98	63.6		32	6.06	40.61
	47	9.56	58.26		67	14.57	60.23		26	4.71	43.83
	57	12.88	51.44		57	11.88	58.36		16	5.76	41.78
	46	9.38	60.74		73	15.82	61.29		21	6.28	40.27
	37	8.34	61.36		63	12.76	65.29		34	7.24	39.53
	29	5.95	54.84		66	13.27	67.28		28	5.19	45.27
	53	11.67	55.39		59	11.92	59.28		29	6.03	37.27
A2	46	10.01	25.67	B2	87	18.87	39.66	C2	25	4.67	47.82
	107	21.85	29.05		145	33.21	40.16		36	6.49	53.39
	54	10.35	33.85		120	25.49	31.89		34	6.39	49.35
	67	12.4	36.6		99	21.96	35.02		36	6.56	48.82
	41	7.44	24		80	16.77	37.92		22	4.01	48.64
	61	11.56	28.56		101	22.93	38.51		33	6.49	49.73
	58	10.06	25.72		95	20.18	37.09		31	5.9	47.93
	43	9.96	30.82		112	22.39	40.83		27	5.19	46.94
	52	10.7	27.93		139	31.08	41.03		35	6.78	45.11
	60	11.49	22.08		89	20.83	33.18		30	5.39	50.39

Rel II. Small instances

	ITER	CPUT	GAP		ITER	CPUT	GAP		ITER	CPUT	GAP
A1	24	24.13	0	B1	25	15.74	0.12	C1	2	0.5	0
	25	21.98	0		25	18.24	0		2	0.61	0
	20	18.54	0		27	18.76	0		1	0.02	0
	19	17.37	0		11	6.38	0		1	0.02	0
	23	23.93	0		53	37.38	0		2	0.64	0
	25	21.38	0		32	20.18	0		2	0.48	0
	22	23.45	0		34	25.92	0		1	0.07	0
	21	19.35	0		27	19.78	0		1	0.06	0
	26	22.09	0		26	19.73	0.15		2	0.59	0
	25	22.1	0		28	21.06	0		1	0.03	0
A2	36	21.81	0	B2	23	19.45	0	C2	8	3.07	0
	67	45.01	0		8	5.67	0		1	0.02	0.24
	30	16.68	0		10	6.48	0		4	1.54	0
	39	18.48	0		22	18.51	0		7	2.84	0
	15	6.97	0		17	13.14	0		1	0.03	0
	32	17.83	0		20	16.37	0		6	2.96	0
	37	19.36	0		16	14.25	0		4	1.72	0.32
	39	21.79	0		15	15.13	0		5	1.98	0
	41	18.68	0		7	6.38	0		1	0.03	0
	26	15.96	0		21	17.32	0		1	0.05	0