

University of Alberta

**Advanced Integer Linear Programming Techniques for Large
Scale Grid-Based Location Problems**

by

Md. Noor-E-Alam

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Engineering Management

Mechanical Engineering

©Md. Noor-E-Alam

Spring 2013

Edmonton, Alberta

Permission is hereby granted to the University of Alberta Libraries to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only. Where the thesis is converted to, or otherwise made available in digital form, the University of Alberta will advise potential users of the thesis of these terms.

The author reserves all other publication and other rights in association with the copyright in the thesis and, except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatsoever without the author's prior written permission.

To my Family

Abstract

Many real-world facility location problems can be approximated by a grid-based system of small-sized cells, which can then be used to model a heterogeneous demand distribution. We can also relate an amount of supply for each cell from its supply distribution relationships with the various potential facility locations in other cells. Based on these demand distributions and supply relationships, we can determine the optimal capacities and locations to place facilities while fulfilling certain objectives. Here, these types of location problems are referred to as *grid-based location problems* (GBLPs). In the GBLPs we address herein, we seek the optimum number, location(s), and size(s) of facilities to place. The applications of GBLPs are wide ranging, and include problems in business, engineering, defense, resource exploitation, and medical science.

To make such complex decisions, we need to develop mathematical models in the form of *integer linear programming* (ILP) problems, and associated procedures to solve them. To model a real-world GBLP, we must generally consider a large number of discrete variables, complex demand and supply distributions, and fixed costs. Combinations of these considerations conspire to produce large-scale ILP problems, which are often not scalable and often become intractable even for small instances.

In this research, we propose a number of GBLP ILP models for two real-world applications: a *light post location problem* and a *wireless transmitter location*

problem. Our experimental results show that our ILP models are efficient in solving moderately-sized problems but are computationally difficult to solve for large-scale instances. As a result, we develop two decomposition techniques to solve these large-scale instances. To reduce the solution time further, we also propose integration of logical restrictions and valid inequalities. Our experiments demonstrate that the proposed approaches outperform the exact solution method, significantly reducing solution runtimes while not severely impacting optimality. The results of this work is expected to have a significant impact in solving large-scale GBLP ILP models that result from real-world business, engineering, and science problems.

Acknowledgements

First, I want to thank my supervisor and mentor Dr. John Doucette, for his excellent guidance, direction and support. I have been privileged to be a part of his research group, where I have enjoyed developing myself as an independent researcher in the area of decision science.

I am grateful to Dr. Peter Flynn, Dr. Mike Lipsett, Dr. Ming Zuo, Dr. Marc Secanell Gallart and Dr. Hooman Askari-Nasab for their valuable advice and help.

I would like to acknowledge four scholarships: FS Chia Ph. D. Scholarship, Izzak Walton Killam Memorial Scholarship from the Killam Trusts, Queen Elizabeth II Graduate Scholarship-Doctoral Level and Profiling Alberta's Graduate Students Award, all administered by the Faculty of Graduate Studies and Research (FGSR), University of Alberta for their generous financial support in this research. I am also grateful to the Natural Science and Engineering Research Council of Canada (NSERC) for their support.

I would like to extend my gratitude to all of my lab mates and other friends in the Dept. of Mechanical Engineering for their continuous encouragement and help throughout my program. Special thanks go to Ahmed Zaky Kasem, Brody Todd, Roberto Gallardo, Mark Ruhl, Mohammed Dome, Ashique Khan, Md. Moin Uddin, Mayank Pandey and Mohammad Hoseini. I would also like to thank Ms. Gail Dowler, Senior Graduate Program Assistant, for her continuous support for

the last five years. Also the support and encouragement from all of my other friends and relatives are greatly acknowledged.

I would like to thank Dr. Shabbir Ahmed, Professor, School of Industrial & Systems Engineering, Georgia Institute of Technology for his time and advice in IEOM 2010. Also various suggestions and helps from Dr. Paul Rubin, Professor Emeritus at Michigan State University and Dr. Robert Fourer, Professor, Dept. of Industrial Engineering & Management Sciences, Northwestern University, through Google AMPL group, are greatly appreciated. I also thank Andrew Ma for his help to generate demand distributions.

I am indebted to my parents for their inspiration, unconditional love and support throughout my life. I am also grateful to my sisters and brother for their constant encouragement and cares. I am truly grateful to my wife Tahmina for her encouragement, care, love and constant support. Without her endless sacrifice, it would be almost impossible to achieve this goal. I am thankful to my wife's family (her parents and two sisters) for their inspiration and valuable support. My special thanks go to my son Areeb for his invaluable love and affection that encouraged me greatly to finish this research.

Finally, I am grateful to Almighty God for giving me the strength, guidance and determination to achieve this success.

Table of Contents

Chapter 1: Introduction	1
1.1. Location Problem	1
1.2. Grid-Based Location Problem (GBLP).....	4
1.3. Motivation and Objectives	6
1.4. Proposed Methodology.....	7
1.5. Organization and Relationships Between our Manuscripts	8
References	12
Chapter 2: Large-Scale Integer Linear Programming	14
2.1. Integer Linear Programming	14
2.2. Advanced Solution Methods	16
2.2.1. Reformulation Techniques.....	17
2.2.2. Classical Decomposition Techniques	18
2.2.3. Relaxation-Based Decomposition Techniques	19
2.2.4. Meta-Heuristic Techniques.....	21
2.2.5. Other Heuristic Techniques	22
References	22
Chapter 3: ILP Models for Light Post Location Problem	28
3.1. Introduction	28
3.2. Problem Description.....	34
3.2.1 Supply Calculations	35
3.3. Basic Model.....	38
3.3.1. Simplified Supply Calculation.....	38
3.3.2. Basic Optimization Model	40
3.3.3. Equivalent Basic ILP Model.....	43
3.3.3.1 Linearization of Objective Function.....	44
3.3.3.2. Linearization of Constraint Equations.....	45
3.3.4. Preliminary Results.....	47
3.4. Enhanced Models	50
3.4.1. Model 2.....	52

3.4.2. Model 3	55
3.5. Result Analysis	56
3.6. Conclusion	76
References	77
Chapter 4: Solving Large Scale GBLPs	81
4.1. Introduction	81
4.2. Optimization Model Description	87
4.3. Runtime Complexity	93
4.4. Relaxation-Based Decomposition Techniques	98
4.4.1. Relax-and-Fix-Based Decomposition (RFBD)	99
4.4.2. Logical Restrictions	100
4.5. Result Analysis	103
4.6. Concluding Discussion	118
References	120
Chapter 5: Solving Large Scale Fixed Cost GBLPs	123
5.1. Introduction	123
5.1.1. Proposed Work	131
5.2. Fixed Cost GBLP Model	132
5.3. Preliminary Results	139
5.4. Problem-Specific Decomposition Technique	151
5.4.1. Partition-and-Fix-Based Decomposition (PFBD)	152
5.4.2. Relax-and-Fix-Based Decomposition (RFBD)	159
5.5. Result Analysis	161
5.6. Concluding Discussion	170
References	171
Chapter 6: ILP Model for Wireless Transmitter Location Problem	176
6.1. Introduction	176
6.2. Problem Description	182
6.2.1. Signal Strength Variation	183
6.2.1.1. Log-distance Path Loss Model	183
6.2.1.2. Log-normal Shadowing Model	184
6.2.2. Attenuation calculation process	185

6.3. ILP Model.....	188
6.4. Preliminary Results	192
6.5. Relax-and-Fix-Based Decomposition	199
6.6. Results and Analysis	204
6.7. Conclusion.....	211
References	212
Chapter 7: Concluding Discussion	216
7.1. Brief Summary of Thesis	216
7.1.1. Main Contributions	220
7.2. Other Contributions of Ph.D. Work	221
7.2.1. Journals Papers.....	222
7.2.2. Refereed Conference Publication.	223
7.2.3. Other Peer Reviewed Publications and Presentations	223
7.3. Future Research Avenues	224
Appendices.....	227

List of Tables

Table 3.1: Runtime statistics for selected Model 3 test cases.	76
Table 4.1: Comparison of Objective function value and CPU Time (10x20 grid).	106
Table 4.2: Comparison of Objective function value and CPU Time (15x15 grid).	107
Table 4.3: Comparison of Objective function value and CPU Time (20x30 grid).	108
Table 4.4: Details of CPU Time for RFBD with LR to solve large scale instances.	111
Table 4.5: Branch-and-bound statistics for 10x20 grid solutions.	118
Table 5.1: Run time statistics for different test-case grids solved by the exact method.	150
Table 5.2: Comparison of runtime statistics.	166
Table 5.3: Details of PFBD runtime statistics for 40x60 grid.	168
Table 5.4: PFBD runtime statistics for 40x100 grid.	169
Table 6.1: Translation of interference ratings to attenuation factors.	188
Table 6.2: Run time statistics for different test-case grids solved by the exact method.	199
Table 6.3: Run time statistics for different test-case grids solved by the exact method.	211

List of Figures

Figure 3.1: Basic relationship between light brightness and distance, adapted from NASA (2006).	37
Figure 3.2: Geometry of a point source of light above a horizontal plane.	37
Figure 3.3: Distribution of light supply for the basic model.	39
Figure 3.4: Demand distribution for 7x7 grid.	49
Figure 3.5: Objective function values for 7x7 grid test cases solved with the basic ILP model.	50
Figure 3.6: Exact distribution of light supply.	52
Figure 3.7: Demand distribution for 10x10 grid.	57
Figure 3.8: Demand distribution for 10x12 grid.	58
Figure 3.9: Demand distribution for 12x12 grid.	59
Figure 3.10: Demand distribution for 10x15 grid.	60
Figure 3.11: Demand distribution for 10x17 grid.	61
Figure 3.12: Demand distribution for 10x20 grid.	61
Figure 3.13: Demand distribution for 15x15 grid.	62
Figure 3.14: Variation of objective function value and CPU time for 10x10 grid.	65
Figure 3.15: Variation of objective function value and CPU time for 10x12 grid.	66
Figure 3.16: Variation of objective function value and CPU time for 12x12 grid.	67
Figure 3.17: Variation of objective function value and CPU time for 10x15 grid.	68
Figure 3.18: Variation of objective function value and CPU time for 10x17 grid.	69
Figure 3.19: Variation of objective function value and CPU time for 10x20 grid.	70

Figure 3.20: Variation of objective function value and CPU time for 15x15 grid.	71
Figure 3.21: Optimum locations and sizes of light posts for 10x10 grid.	72
Figure 3.22: Optimum locations and sizes of light posts for 10x12 grid.	73
Figure 3.23: Optimum locations and sizes of light posts for 12x12 grid.	74
Figure 4.1: Light distribution model.	90
Figure 4.2: Demand distribution for 10x20 grid.	94
Figure 4.3: Demand distribution for 15x15 grid.	95
Figure 4.4: Objective function value and solution runtime for the GBLP on the 10x20 grid.	97
Figure 4.5: Objective function value and solution runtime for the GBLP on the 15x15 grid.	98
Figure 4.6: Illustration of the logical restrictions in equation (3.12).	102
Figure 4.7: Illustration of the relax-and-fix-based decomposition approach.	103
Figure 4.8: Demand distribution for 20x30 grid.	104
Figure 4.9: Objective function value and solution runtime of the GBLP using RFBD with LR on the 10x20 grid.	112
Figure 4.10: Objective function value and solution runtime of the GBLP using RFBD with LR on the 15x15 grid.	113
Figure 4.11: Objective function value and solution runtime of the GBLP using RFBD with LR on the 20x30 grid.	114
Figure 4.12: Objective function value of the GBLP using RFBD with LR and GA on the 10x20 grid.	115
Figure 4.13: Objective function value of the GBLP using RFBD with LR and GA on the 15x15 grid.	116
Figure 4.14: Objective function value of the GBLP using RFBD with LR and GA on the 20x30 grid.	117
Figure 5.1: Light distribution model.	135
Figure 5.2: Demand distribution for 10x10 grid.	140

Figure 5.3: Demand distribution for 10x10a grid.	141
Figure 5.4: Demand distribution for 10x12 grid.	141
Figure 5.5: Demand distribution for 12x12 grid.	142
Figure 5.6: Demand distribution for 10x15 grid.	143
Figure 5.7: Demand distribution for 10x17 grid.	143
Figure 5.8: Demand distribution for 10x17a grid.	144
Figure 5.9: Demand distribution for 10x20 grid.	144
Figure 5.10: Demand distribution for 15x15 grid.	145
Figure 5.11: Demand distribution for 20x30 grid.	146
Figure 5.12: Demand distribution for 20x30a grid.	147
Figure 5.13: Illustration of the decomposition procedure.	155
Figure 5.14: Illustration of the PFBD approach for fixed cost GBLP.	158
Figure 5.15: Illustration of the RFBD approach for fixed cost GBLP.	161
Figure 5.16: Demand distribution for 30x40 grid.	164
Figure 5.17: Demand distribution for 30x60 grid.	164
Figure 5.18: Demand distribution for 40x60 grid.	165
Figure 6.1: A sample 4x4 grid highlighting one transmitter (T), an obstacle (O), and three served regions (S1-S3).	187
Figure 6.2: Obstructions' map for the 5x5 grid (left) and 5x5a grid (right).	193
Figure 6.3: Obstructions' map for the 7x7 grid (left) and 7x7a grid (right).	193
Figure 6.4: Obstructions' map for the 8x8 grid (left) and 8x8a grid (right).	194
Figure 6.5: Obstructions' map for 10x10 grid.	195
Figure 6.6: Obstructions' map for 10x10a grid.	196
Figure 6.7: Illustration of valid inequalities.	202

Figure 6.8: Illustration of the RFBD approach.	204
Figure 6.9: Obstructions' map for the 10x15 grid.	205
Figure 6.10: Obstructions' map for the 10x20 grid.	206
Figure 6.11: Obstructions' map for the 15x15 grid.	206
Figure 6.12: Obstructions' map for the 15x20 grid.	207
Figure 6.13: Obstructions' map for the 20x20 grid.	208

List of Abbreviations

BCP	Branch-and-cut-and-price
CFL	Capacitated facility location
CMFWP	Capacitated multi-facility Weber problem
FDG	Finite difference gradient
FEM	Finite element method
GA	Genetic algorithm
GBLPs	Grid-based location problems
ILP	Integer linear programming
LD	Lagrangian decomposition
LRs	Logical restrictions
MCDM	Multi-criteria decision making
MCMS	Maximum coverage with minimum sensors
MKP	Multiple knapsack problems
OFV	Objective function value
OOFV	Optimum objective function values
PFBD	Partition-and-fix-based decomposition
RFBD	Relax-and-fix-based decomposition
SPSA	Simultaneous perturbation stochastic approximation
TLS	Total least squares
UFL	Uncapacitated facility location
VFSA	Very fast simulated annealing

List of Nomenclature

Q is the set of all x-coordinates in the grid, indexed by i

R is the set of all y-coordinates in the grid, indexed by j

$D_{i,j}$ is the demand for the grid point whose coordinates are at (i,j)

i_{max} is the maximum value of i

i_{min} is the minimum value of i

j_{max} is the maximum value of j

j_{min} is the minimum value of j

β is the system boundary constant

N is the set of all light sources, indexed by n

n_{opt} is the optimum number of light sources

S_{ij} is the total supply at location (i,j)

S_{ijn} is the supply at location (i,j) from the n^{th} light post

S_{in} is the supply at location (i) in x direction from the light source n

S_{jn} is the supply at location (j) in y direction from the light source n

x_n is the x coordinate of the optimal location of the n^{th} light post

y_n is the y coordinate of the optimal location of the n^{th} light post

P_n is the optimum size of the n^{th} light source (required to be integer)

UB is the upper bound on decision variable P_n

ED_{ij} is the excess demand at location (i,j)

ES_{ij} is the excess supply at location (i,j)

X is the set of all x-coordinates of the light source, indexed by x

Y is the set of all y-coordinates of the light source, indexed by y

P_{xy} is the size of the source at location (required to be integer)

UB_{xy} is the upper bound on decision variable P_{xy}

T_{xy} is a binary variable at location (x,y) : T_{xy} is 1 if $P_{xy}>0$; T_{xy} is 0 if $P_{xy}=0$

C_f is the fixed cost to install a light source

C_v is the per unit variable cost

ξ_{xy}^{ij} is the attenuation factor between source (x, y) to destination (i, j)

Chapter 1

Introduction

1.1. Location Problem

Determining optimal location of facilities is a very common and complex problem in business, engineering, defense, resource exploitation, and even in medical sciences (Noor-E-Alam et al., 2012). Locating facilities can be a critical decision for a manager, as location plays a vital role in the success of an organization. Over the last several decades, numerous methods have been developed in the area of location theory, resulting in a number of notable approaches that seek to find optimum locations. These problem-specific methods are particularly designed for the various types of location problems. Among them, the most significant facility location problems are *Weber problems* (Cooper, 1963), *coverage problems*

(Church and ReVelle, 1974), *uncapacitated facility location* (UFL) problems (Wolsey, 1998) and *capacitated facility location* (CFL) problems (Wolsey, 1998). While there are other types of problems, to be discussed shortly these cover a majority of the cases.

The Weber problem was first proposed by Cooper (1963), and is also well known as the *multisource Weber problem*. This problem has a known number of facilities with equal fixed costs. Since this problem was first proposed, a lot of research has been done on various aspects and versions of the problem. For example, a *two-dimensional facility problem* is modeled as a Weber problem to locate multiple new facilities with respect to existing facilities (Francis, 1964). Later, Wesolowsky (1972) proposed a model for the solution of the Weber problem using rectilinear distances. A *probabilistic multi-facility Weber problem* was proposed by Katz and Cooper (1974), which was later revisited by Altinel et al. (2009). Sherali and Noradi (1988) and Manzour-al-Ajdad et al. (2012) proposed models for a *capacitated multi-facility Weber problem* (CMFWP).

Church and ReVelle (1974) introduced another special type of location problem with the objective of coverage. This problem is known as a coverage problem, where it ensures an appropriate set of facilities for each customer. This model is widely used to find optimum locations of emergency services, retail facilities, cell-phone towers and sensor networks. For example, Drezner and Wesolowsky

(1997) described a method of placing signal detectors to cover a certain area such that the probability that an event is not detected is minimized. More information on coverage location models is available in Berman et al. (2010).

Furthermore, depending on the capacity restriction of the sources, location problems can be classified as a UFL problem or a CFL problem. In UFL problems, variable transportation costs are considered without any capacity restrictions (Wolsey, 1998). A UFL problem becomes a CFL problem when there is an upper limit on the amount of supply available (Ghiani et al., 2002; Chen, 2010). Among other significant applications of location models, Marín (2011) described a new discrete location model, where the number of customers allocated to every plant has to be balanced. Ingolfsson et al. (2007) proposed an ambulance location optimization model that minimizes the number of ambulances needed to provide a target service level. It measured service level as the fraction of calls reached within a given time standard and considered response time as determined by a random delay (prior to travel to the scene) plus a random travel time.

Of all the models discussed above, it is important to recognize that models to solve location problems can be classified into two distinct groups: *discrete location analysis* and *continuous location analysis*. Discrete location analysis is the most common form of modeling for a location problem, where it typically refers to the use of a node-and-network (transportation) approach. Here, facilities

and supply points are modeled as the vertices and nodes (Domschke and Krispin, 1996). On the other hand, continuous location analysis involves the modeling of the location problem on a continuous plane. Here, all customer demands are coordinate points and the feasible solution for the optimal placement of the facilities can be any coordinate point in the considered plane. Daskin (1995) proposed to use discrete location analysis to solve location problems as continuous location analysis can produce difficult ILP instances.

1.2. Grid-Based Location Problem (GBLP)

In the work herein, we consider a special type of location problem called *grid-based location problems* (GBLPs) that can be used to solve some single/multisource location problems (Noor-E-Alam et al., 2012). Here, facility location problems can be approximated by a grid-based system of small-sized cells. In real-world situations, demand is not a singular point, but rather, many individual points located adjacent to each other, often forming a heterogeneous distribution. To model a location problem as a GBLP, the heterogeneous demand distribution and a linear/non-linear supply function could be developed with respect to those cells. Based on the demand distributions and supply relationships, we can then determine the optimal capacities and locations to place our facilities while fulfilling certain objectives. In the GBLPs we have addressed herein, we seek the optimum number, location(s), and size(s) of facilities to place.

The wider applications of GBLPs range from business and engineering to medical sciences. Some of them are as follows:

- In real world business, it is important to determine the optimum location of a facility such as a retail store, a service centre or a bank etc., to maximize customer satisfaction in an area, while minimizing the total cost.
- In engineering, there are many applications of GBLPs such as where to install a machine in a plant, where to build a warehouse, where to put sensors in a chemical refinery, and where to target mining operations.
- In healthcare, the optimum location of radiation therapy could be another application of GBLPs. Determining appropriate location and dose of radiation therapy is very important since the objective of the treatment is to apply the dose in such a way that it will affect only the cancerous cells, not the normal healthy cells (Lim, 2002). In this case, the objective of GBLPs would be minimizing the dose to the healthy cell while applying a sufficient amount of dose to the affected cell.

To make such complex decisions, we need to develop mathematical models, and more specifically, *integer linear programming* (ILP) formulations, and related procedures to solve them. To model a real-world GBLP, we need to consider a

large number of discrete variables, a heterogeneous demand distribution, a non-linear supply distribution, and potentially fixed costs for placing a facility at the specified location. Furthermore, to get an optimum decision, the ILP models need to be designed in such a way that they will simultaneously determine the location(s), size(s) and, number of facilities to achieve given objectives. Combinations of these considerations make GBLPs large-scale ILP problems, which are not scalable and often become intractable even for small cases. Therefore, our target is to develop advanced ILP techniques to solve such large-scale instances.

1.3. Motivation and Objectives

Considering the wider applications of GBLPs, it is important and challenging to decide how to formulate mathematical models for these problems and which solution techniques can be used to solve large-scale instances. The key contribution of this thesis is to provide greater understanding of the concept of GBLPs and develop techniques to solve them. To fulfill these objectives, our goal is to develop ILP models for solving specific GBLPs. Our research also aims to reduce their solution time by developing problem-specific decomposition techniques for solving large-scale GBLP ILP instances. The results of this work are expected to represent a significant step towards solving large-scale GBLPs associated with important real-world applications.

1.4. Proposed Methodology

In this research, our intent is to develop ILP models and the associated effective and efficient solution procedures for solving grid-based location problems. More specifically following methodologies are adopted in this work:

- GBLP ILP models are developed for a simple real-world application (the light post location problem).
- A relaxation-based decomposition technique is developed to solve large-scale instances of the above problem.
- A fixed cost ILP model is developed to incorporate costs associated with light post establishment, and a partition-and-fix-based decomposition technique is proposed to solve large-scale instances of that modified ILP.
- The wider applicability of our proposed methods is demonstrated by applying them to another GBLP (the wireless transmitter location problem).

This thesis follows the paper-based format as described in the University of Alberta Faculty of Graduate Studies and Research's "Thesis Format specifications" document. The following sub-section will briefly describe the

organization of this thesis by outlining the relationships between our various manuscripts.

1.5. Organization and Relationships Between our Manuscripts

We follow with a discussion of ILP solution techniques in Chapter 2. Chapter 3 through Chapter 6 are adopted from four journal papers, where each chapter is prepared to be read independently. We wrap up with a concluding discussion in Chapter 7. The following brief summary will provide an overview of this thesis:

We have started this thesis with a background discussion on different types of location problems and their solution strategies in Chapter 1. We have introduced a special type of location problem referred to as grid-based location problems (GBLPs) and described several potential applications. We have also described our goals and objectives in this chapter.

In Chapter 2, advanced integer linear programming solution techniques are described.

In Chapter 3, we focus on developing ILP models for solving GBLPs targeting to solve a real-world problem of placing lights in a city park to minimize the amount of darkness and excess supply. In a city park, demand is not a singular point, but rather, can be thought of as many individual points adjacent to each other to form a complex heterogeneous distribution. We approximate this location problem as a

GBLP, where the entire area is divided into small cells. To start, we develop a basic optimization model with a simplified distribution of light supply, but it has a non-linear objective function and non-linear supply relationship. To better fulfill our objectives, we then develop an equivalent ILP model. Our preliminary results show that this model becomes intractable for even small instances, likely a result of the many binary variables and associated constraints.

To overcome this computational difficulty and to represent a more precise light distribution model, we propose two enhanced models. In these two new models, we use the same objective function, but adopt a simplified approach for defining the feasible region. In fact, we replace the original sets of constraints that arose from the non-linear supply function with a single set of constraints. In the first enhanced model, we do not have any capability to control the number of light sources. Therefore, we develop the second enhanced model to permit that. These ILP models are designed to provide the optimal solution for the light post problem: the total number of light posts, the location of each light post, and their capacities (i.e., brightness). Finally, the ILP models are implemented within a standard modeling language and solved with the CPLEX solver. Our experimental results show that the ILP models are efficient in solving moderately-sized problems with a small optimality gap.

In Chapter 3, we show that for large-scale instances, solution generally takes days and even weeks to solve. So in Chapter 4, we evaluate the computational complexity of the GBLP model with several large scale test-case grids. We also investigate the structure of the mathematical model to identify the causes for the exponential behaviour of runtimes. Based on the findings from this investigation, we propose a problem-specific relaxation-based decomposition approach we call relax-and-fix-based decomposition (RFBD) to solve large-scale GBLPs. To reduce the solution time further, we also propose problem-specific logical restrictions that reduce the feasible region and the resulting branch-and-bound tree. Finally, the decomposition technique is implemented within a standard modeling language and tested on a number of large test-case grids. Our experiments demonstrate that the RFBD approach outperforms the exact method (conventional ILP techniques) and significantly reduces solution runtimes while not severely impacting optimality.

In Chapter 3 and Chapter 4, the implementation costs (fixed costs) are not considered. Therefore, the optimal decisions found from these models are not optimal on the basis of overall cost criteria. To solve a fixed cost GBLP, we develop another integer linear programming (ILP) model in Chapter 5. Our preliminary results reveal that the ILP model is efficient in solving small to moderately-sized problems. However, it becomes very difficult in solving large-

scale GBLP instances. To overcome this difficulty, we evaluate the computational complexity of the GBLP model with several large-scale grids. We also investigate the structure of the mathematical model to identify the cause of the exponential behaviour of runtimes for the fixed cost GBLP. Based on the findings from this investigation, we propose a problem-specific decomposition approach, called partition-and-fix-based decomposition (PFBD), to solve large-scale GBLP instances. We find that the proposed PFBD approach significantly reduces solution runtimes and outperforms the exact method. Furthermore, to solve very large instances faster, we propose an integration of the RFBD approach with the PFBD approach.

In Chapter 6, we solve a wireless transmitter location problem to demonstrate another application of our proposed methods, discussed in the previous chapters. We describe this location problem in the context of a GBLP and develop an ILP model for optimal placement of transmitters to ensure effective and reliable wireless communication. The ILP model is designed by considering variation in signal strength due to distance and propagation environments (i.e., different degrees of obstruction). The ILP model becomes computationally difficult due to the consideration of the above factors. Therefore, we propose a problem-specific RFBD approach to solve large-scale instances. To reduce the runtimes further, we develop valid inequalities and logical restrictions. Our experimental results

demonstrate that the proposed RFBD approach significantly reduces solution runtimes while not impacting optimality.

Finally, in Chapter 7, a brief summary of this thesis is given. We list all contributions of the Ph.D. work, and conclude with a brief description of future research opportunities.

References

- Altinel, I. K., Durmaz, E., Aras, N., Özkisacik, K. C. (2009), “A location-allocation heuristic for the capacitated multi-facility Weber problem with probabilistic customer locations”, *European Journal of Operational Research*, 198, 790-799.
- Berman, O., Drezner, Z., Krass, D. (2010), “Generalized coverage: New developments in covering location models”, *Computers & Operations Research*, 37, 1675–1687.
- Chen, D.-S., Baton, R. G., Dang, Y. (2010), *Applied Integer Programming*, A John Willey & Sons, Inc., Publication, (Chapter 2).
- Church, R. L. and ReVelle, C. S. (1974), “The maximal covering location problem”, *Papers of the Regional Science Association*, 32, 101–118.
- Cooper, L. (1963), “Location-Allocation Problems”, *Operations Research*, 11, 331-343.
- Daskin, M. S. (1995), *Network and Discrete Location: Models, Algorithms, and Applications*, John Wiley, New York.
- Domschke, W., Krispin, G. (1997), “Location and layout planning-a survey”, *OR Spektrum*, 19, 181-194.
- Drezner, Z., Wesolowsky, G.O. (1997), “On the best location of signal detectors”, *IIE Transactions*, 29, 1007-1015.

- Francis, R. L. (1964), “On the Location of Multiple New Facilities with Respect to Existing Facilities”, *Journal of Industrial Engineering*, 15, 106-107.
- Ghiani, G., Grandinetti, L., Guerriero, F., Musmanno, R. (2002), “A Lagrangean Heuristic for the Plant Location Problem with Multiple Facilities in the Same Site”, *Optimization Methods and Software*, 17, 1059–1076.
- Ingolfsson, A., Budge, S., Erkut, E. (2008), “Optimal ambulance location with random delays and travel times”, *Health Care Management Science*, 11, 262-274.
- Katz, I. N., Cooper, L. (1974), “An always convergent numerical scheme for a random locational equilibrium problem”, *SIAM Journal of Numerical Analysis*, 11, 683-692.
- Lim, J. (2002), “Optimization in radiation treatment planning”, Ph. D Thesis, University of Wisconsin-Madison.
- Manzour-al-Ajdad, S. M. H., Torabi, S. A., Eshghi, K. (2012), “Single-Source Capacitated Multi-Facility Weber Problem - An iterative two phase heuristic algorithm”, *Computers & Operations Research*, 39, 1465-1476.
- Marín, A. (2011), “The discrete facility location problem with balance allocation of customers”, *European Journal of Operational Research*, 210, 27-38.
- Noor-E-Alam, M., Ma, A., Doucette, J. (2012), “Integer Linear Programming Models for Grid-Based Light Post Location Problem”, *European Journal of Operational Research*, 222, 17-30.
- Sherali, H. D., Nordai, F.L. (1988), “NP-Hard, Capacitated Balanced p-Median Problems on a Chain Graph with a Continuum of Link Demands”, *Mathematics of Operations Research*, 13, 32-49.
- Wesolowsky, G. O. (1972), “Rectangular Distance Location under the Minimax Optimality Criterion”, *Transportation Science*, 6, 103-113.
- Wolsey, L. A. (1998), *Integer Programming*, Wiley-Interscience, Series in Discrete Mathematics and Optimization, Toronto, (Chapter 1).

Chapter 2

Large-Scale Integer Linear Programming

2.1. Integer Linear Programming

A *linear programming* (LP) is a special type of *mathematical programming* problem, in which we determine a set of values for continuous decision variables (y_1, y_2, \dots, y_n) that minimizes or maximizes a linear objective function, while a set of linear constraints is satisfied (Chen et al., 2010). Mathematical expression of an LP is as follows:

$$\text{Minimize } \sum_{\forall i} c_i y_i \tag{2.1}$$

Subject to:

$$\sum_i a_{ij}y_i \leq b_j \quad (j = 1, 2, \dots, m) \quad (2.2)$$

$$y_i \geq 0 \quad (i = 1, 2, \dots, n) \quad (2.3)$$

In the above model, if at least one of the variables is restricted to integer values then the resulting model is called an *integer linear programming* (ILP) problem (Chen et al., 2010).

Although there is no known polynomial-time algorithm for solving ILP problems, they can be practically solved using a variety of techniques (Wolsey, 1998). *LP relaxation* is one of the most popular techniques, where the integer decision variables are permitted to take continuous values that result in an upper (lower) bound on the optimal solution for maximization (minimization) problems (Chen et al., 2010). An optimal solution to the LP relaxation of an ILP is often quite a weak bound. The *branch-and-bound* algorithm is the most widely used technique to solve ILPs utilizing weak bounds (Chen et al., 2010). This algorithm uses LP relaxation to solve an instance of the problem, where (for a maximization problem) the upper bounds obtained from LP relaxations and lower bounds obtained from semi-relaxed problems are used to fathom the branch-and-bound search tree (Chen et al., 2010; Wolsey, 1998). Another widely used technique is the *cutting plane algorithm*, where cuts are generated to create tight LP relaxation (Chen et al., 2010). The cutting plane algorithm can often be integrated with

branch-and-bound techniques to solve ILP instances more efficiently (Chen et al., 2010). In fact, combination of these two techniques is now widely used in commercial ILP solvers such as CPLEX, Gurobi Optimization and MINTO (Chen et al., 2010). However, due to the combinatorial nature of the hard ILP instances, it is often intractable to solve them with current methods; many real-world GBLP ILP instances require weeks or months of solution time to solve (Noor-E-Alam et al., 2012).

2.2. Advanced Solution Methods

As we have mentioned earlier, the ILP models for real-world location problems are complex and combinatorial in nature. The complexity of an ILP model depends on the objective(s) to be optimized and the considered constraints to make an optimal decision. These criteria are determined by the decision makers based upon the nature of the problem (Teixeira and Antunes, 2008). More specifically, the ILP models are required to consider a large number of discrete variables, complex demand and supply distributions, and fixed costs. In addition to that, the ILP models need to be designed in such a way that they will simultaneously determine the location(s), size(s) and number of facilities. It is therefore imperative that solution techniques are specially engineered for solving highly intractable instances. There are many advanced ILP techniques developed by many researchers for solving large-scale location problems. It is crucial that

decision makers select an efficient method from the variety of available methods, depending upon the structure and the properties of an ILP model. These techniques are briefly discussed in the following sub-sections:

2.2.1. Reformulation Techniques

To overcome the above mentioned difficulties, several other advanced ILP techniques can also be applied. One of the most useful techniques is *problem reformulation*, where the mathematical model is reformulated such that the LP relaxation produces very tight bounds, which results in the solver requiring less time to reach optimality (Wolsey, 1998). Careful reformulation also helps solvers to improve the efficiency of the branch-and-bound algorithm (Wolsey, 1998). A number of solvers generate certain classes of inequalities for simple structures, such as the knapsack, single-node flow, and path poly-topes for efficient relaxation of many combinatorial optimization problems (Roy and Wolsey, 1987; Savelsbergh et al., 1995). Therefore, to help solvers create better automatic reformulations, manual reformulation is often needed to redefine variables (Trick, 2005). For example, in Aardal (1998), an alternative way of modeling CFL problems is discussed. In that work, redundant constraints with a set of new decision variables were combined such that the solver was able to generate stronger linear inequalities and take less time to reach optimality. To reduce computational time, ILP models with extra *logical restrictions* are discussed in

Williams (1978) and Aardal et al. (1995). Leung and Magnanti (1989) proposed a polyhedral structure of CFL to identify *valid inequalities* to get better formulation, whereas, related analysis is carried out for *p-median facility location* problem in Avella and Sassano (2001).

2.2.2. Classical Decomposition Techniques

Other useful advanced techniques for solving large-scale ILP problems are decomposition techniques that have been developed and evaluated in recent decades. For example, Barnhart et al. (1998) discussed the *Dantzig-Wolfe* decomposition technique for solving large ILP problems, where the *column generation* (CG) method has been invoked for implicit pricing of non-basic variables. A *branch-and-price-and-cut* algorithm is proposed in Barnhart et al. (2000) that allows CG and a cutting plane algorithm to be applied throughout the branch-and-bound search tree to reduce computational complexity. Furthermore, CG and cutting plane algorithms are integrated with the branch-and-bound algorithm to improve the relaxation of the problem and achieve price out efficiency in the *branch-cut-price* (BCP) algorithm (Belov and Scheithauer, 2006). However, for some cases, CG shows longer convergence due to large fluctuations in the simplex multipliers (Valero, 2005). To improve this convergence rate, a heuristic technique is often required for stabilizing this procedure (Amor et al, 2006).

The above decomposition techniques are widely used in solving location problems. Klose and Görtz (2007) used the *branch-and-price* algorithm to solve CFL problems, where a stabilized CG method was used for solving corresponding master problem to optimality. The *capacitated p-median problem* was also solved with the branch-and-price algorithm in Lorena and Senne (2004) and Senne et al. (2005). Sonmez and Lim (2012) proposed a decomposition algorithm to solve an ILP model for facility location problems. Their computational results showed that the decomposition algorithm produces near optimal solutions very quickly. *Uncapacitated multiple allocation p-hub median problems* were solved by a branch-and-cut algorithm in García et al. (2012). Furthermore, *Benders decomposition* technique was used to solve large-scale ILP instances for uncapacitated hub location problems with multiple assignments in Contreras et al. (2011). This technique has also been proposed for solving CFL in Magnanti and Wong (1981) and Wentges (1996). While the decomposition methods we have discussed herein often represent significant reductions in solution time, they are often appropriate only for the specific types of ILP models having specific mathematical structure.

2.2.3. Relaxation-Based Decomposition Techniques

Another commonly used decomposition technique is a *relaxation-based* decomposition technique, where the original problem is decomposed into easier

sub-problems by relaxing the complicating constraints and/or integrality restrictions (Chen, 2010). Later, the relaxed problem is solved and its partial solutions are fixed in the original problem to generate an easier sub-problem, often called the *core problem*. A near-optimal solution of the original ILP problem is then obtained by solving this core problem (Wolsey, 1998). Such relax-and-fix strategies have been used in the literature to solve large-scale ILP instances (Beraldi et al., 2006; Ferreira and Morabito, 2010; Kelly and Mann, 2004; Mohammadi et al., 2010). The most widely used relaxation-based decomposition technique is *Lagrangian decomposition* (LD), which creates an easier sub-problem by relaxing certain complicating constraints (Mauri et al., 2010). The solution of this sub-problem is then fixed in the original problem to form the core problem. This core problem is then solved to get a near optimal solution (Rajagopalan et al., 2004). An LD based heuristic technique was proposed in Lee and Lee (2012) to make facility location decisions with customer preference. This heuristic successfully solved the associated ILP model within a reasonable time. Ghiani et al. (2002) used *dynamic programming* to calculate lower bounds in the LD approach to solve a CFL problem. Giortzis et al. (2000) proposed a decomposition of the original problem into a number of smaller sub-problems, and a final solution is obtained sequentially by solving each of these sub-problems.

2.2.4. Meta-Heuristic Techniques

Meta-heuristic techniques are also widely used to solve complex location problems. For example, a *genetic algorithm* (GA) was used to solve large-scale maximum expected covering location problems (Aytug and Saydam, 2002), and it has been shown that GA can be very effective in obtaining high quality solutions. Genetic search algorithms have been used to find solutions for location problems in Abdinnour-Helm and Venkataramanan (1998) and Taniguchi et al. (1999). Anderson and Ferris (1994) also showed the effectiveness of GA for some combinatorial optimization problems.

In some cases, meta-heuristic techniques have been combined with other decomposition techniques to obtain better results. GA and LD have been combined for solving combinatorial optimization problems, such as *unit commitment problems* (Yamin and Shahidehpour, 2004; Cheng et al., 2000). In their work, GA was used to update Lagrangian multipliers and improve the performance of the LD. A new branch-and-bound algorithm was developed in You and Yamada (2011) to solve *multiple knapsack problems* (MKP), where LD was used to obtain an upper bound, and a greedy heuristic was used to obtain a lower bound. Gendron and Potvin (2003) and Sun (2012) proposed a *tabu search* heuristic in solving location problems as it has been found that this heuristic is successful in solving intractable instances.

2.2.5. Other Heuristic Techniques

Besides the meta-heuristic techniques discussed above, several other heuristic techniques have also been used to solve location problems. Lee and Chang (2007) proposed a dual-based heuristic technique to solve large-scale instances for an unreliable discrete location problem, where it was used to minimize the sum of the fixed costs and expected operating costs. A local search heuristic approach with a probabilistic line barrier method was proposed for the the Weber problem in Canbolat and Wesolowsky (2010). Bangerth et al. (2006) compared and analyzed the efficiency, effectiveness, and reliability of *simultaneous perturbation stochastic approximation* (SPSA), *finite difference gradient* (FDG), and *very fast simulated annealing* (VFSA) algorithms for solving location problems. It was found that that none of these algorithms guarantees the optimal solution; however, SPSA and VFSA are very efficient in finding near optimal solutions with high probability. Other notable methods for solving location problems proposed in the past few years have included methods such as a *gravity model* (Kubis and Hartmann, 2007) and the usage of a *greedy algorithm* (Zhang, 2006).

References

Aardal, K., Pochet, Y., Wolsey L. (1995), “Capacitated facility location: valid inequalities and facets”, *Mathematics of Operations Research*, 20, 552–582.

- Aardal, K. (1998), “Reformulation of capacitated facility location problems: How redundant information can help”, *Annals of Operations Research*, 82, 289-308.
- Abdinnour-Helm, S., Venkataramanan, M. A. (1998), “Solution approaches to hub location problems”, *Annals of Operations Research*, 78, 31-50.
- Amor, B., Desrosiers, Valero de Carvalho, J. M. (2006), “Dual-optimal inequalities for stabilized column generation”, *Operations Research*, 54 (3), 454–463.
- Anderson, E. J., Ferris, M.C. (1994), “Genetic algorithms for combinatorial optimization: The assembly line balancing problem”, *ORSA Journal on Computing*, 6 (2),161–173.
- Avella, P. and Sassano, A. (2001), “On the p-median polytope”, *Mathematical Programming*, 89, 395-411.
- Aytug, H., Saydam, C. (2002), “Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study”, *European Journal of Operational Research*, 141, 480–494.
- Bangerth, W., Klie, H., Wheeler, M. F., Stoffa, P.L., Sen, M.K. (2006), “On optimization algorithms for the reservoir oil well placement problem”, *Computational Geosciences*, 10, 303-319.
- Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.F., Vance, P.H. (1998), “Branch-and-Price: Column Generation for Solving Huge Integer Programs”, *Operations Research*, 46 (3) , 316-329.
- Barnhart, C., Hane, C.A., Vance, P.H. (2000), “Using Branch-and-Price-and-Cut to Solve Origin-Destination Integer Multi commodity Flow Problems”, *Operations Research*, 48 (2), 318-326.
- Belov, G., Scheithauer, G. (2006), “A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting”, *European Journal of Operational Research*, 171, 85–106.
- Beraldi, P., Ghiani, G., Grieco, A. Guerriero, E. (2006), “Fix and Relax Heuristic for a Stochastic Lot-Sizing Problem”, *Computational Optimization and Applications*”, 33, 303–318.

- Canbolat, M. S., Wesolowsky, G. O. (2010), “The rectilinear distance Weber problem in the presence of a probabilistic line barrier”, *European Journal of Operational Research*, 202, 114-121.
- Chen, D.-S., Baton, R. G., Dang, Y. (2010), *Applied Integer Programming*, A John Willey & Sons, Inc., Publication, (Chapter 2).
- Cheng, C.-P. Liu, C.-W., Liu, C.-C. (2000), “Unit Commitment by Lagrangian Relaxation and Genetic Algorithms”, *IEEE Transactions on Power Systems*, 15 (2), 707-714.
- Contreras, I., Cordeau, J., Laporte, G. (2011), “Benders decomposition for large-scale uncapacitated hub location”, *Operations Research*, 59(6), 1477-1490.
- Ferreira, D., Morabito, R., Rangel, S. (2010), “Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants”, *Computers & Operations Research*, 37, 684 – 691.
- García, S., Landete, M., Marín, A. (2012), “New formulation and a branch-and-cut algorithm for the multiple allocation p-hub median problem”, *European Journal of Operational Research*, 220, 48–57.
- Gendron, B., Potvin, J.-Y., Soriano, P. (2003), “A Tabu Search with slope scaling for the multi commodity capacitated location problem with balancing requirements”, *Annals of Operations Research*, 122, 193-217.
- Ghiani, G., Grandinetti, L., Guerriero, F., Musmanno, R. (2002), “A Lagrangean Heuristic for the Plant Location Problem with Multiple Facilities in the Same Site”, *Optimization Methods and Software*, 17,1059–1076.
- Giortzis, A. I., Turner, L.F. and Barria, J. A. (2000), “Decomposition technique for fixed channel assignment problems in mobile radio networks”, *IEE Proc., Commun*, 147(3), 187–194 .
- Kelly, J. D., Mann, J. L. (2004), “Flow sheet decomposition heuristic for scheduling: a relax-and-fix method, *Computers and Chemical Engineering*”, 28, 2193–2200.

- Klose, A., Görtz, S. (2007), “A branch-and-price algorithm for the capacitated facility location problem”, *European Journal of Operational Research*, 179, 1109–1125.
- Kubis, A., Hartmann, M. (2007), “Analysis of Location of Large-area Shopping Centers – A Probabilistic Gravity Model for the Halle-Leipzig Area”, *Jahrbuch für Regionalwissenschaft*, 27, 43-57.
- Lee, J. M., Lee, Y. H. (2012), “Facility location and scale decision problem with customer preference”, *Computers and Industrial Engineering*, 63, 184–191.
- Lee, S.-D., Chang, W.-T. (2007), “On solving the discrete location problems when the facilities are prone to failure”, *Applied Mathematical Modeling*, 31, 817–831.
- Leung, J. M. Y., Magnanti, T. L. (1989), “Valid inequalities and facets of the capacitated plant location problem. Mathematical Programming”, 44, 271–291.
- Lorena, L.A.N., Senne, E.L.F. (2004), “A column generation approach to capacitated p-median”, *Computers & Operations Research*, 31, 863–876.
- Magnanti, T. L., Wong, R. T. (1981), “Accelerating Benders decomposition: algorithmic enhancement and model selection criteria”, *Operations Research*, 29, 464–484.
- Marín, A. (2011), “The discrete facility location problem with balance allocation of customers”, *European Journal of Operational Research*, 210, 27-38.
- Mauri, G. R., Ribeiro G. M., Lorena, L. A. N. (2010), “A new mathematical model and a Lagrangian decomposition for the point-feature cartographic label placement problem”, *Computers & Operations Research*, 37, 2164–2172.
- Mohammadi, M., Ghomi, S. M. T. F., Karimi, B, Torabi, S. A. (2010), “Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lot sizing problem with sequence-dependent setups”, *Journal of Intelligent Manufacturing*, 21, 501–510.
- Noor-E-Alam, M., Ma, A., Doucette, J. (2012), “Integer Linear Programming Models for Grid-Based Light Post Location Problem”, *European Journal of Operational Research*, 222, 17-30.

- Rajagopalan S., Heragu S. S., Taylor, G. D. (2004), “A Lagrangian relaxation approach to solving the integrated pick-up/drop-off point and AGV flow path design problem”, *Applied Mathematical Modelling*, 28, 735–750.
- Roy, T .J. V., Wolsey, L.A. (1987), “Solving mixed integer programming using automatic reformulation”, *Operations Research*, 35, 45 – 57.
- Savelsbergh, M.W.P., Sigismondi, G.C., Nemhauser, G.L. (1995), “Functional description of MINTO, a Mixed INTEger Optimizer”, *Operations Research Letters*, 15, 47 – 58.
- Senne, E.L.F., Lorena, L.A.N., Pereira, M.A. (2005), “A branch-and-price approach to p-median location problems”, *Computers & Operations Research*, 32, 1655–1664.
- Sonmez, A. D., Lim, G. J. (2012), “A decomposition approach for facility location and relocation problem with uncertain number of future facilities”, *European Journal of Operational Research*, 218, 327–338.
- Sun, M. (2012), “A tabu search heuristic procedure for the capacitated facility location problem”, *J. Heuristics*, 18, 91–118.
- Trick, M. A. (2005), “Formulations and Reformulations in Integer Programming”, Second International Conference on Integration of Artificial Intelligence and Operations Research, CPAIOR, Prague, Czech Republic, May 30 - June 1, 366-379.
- Taniguchi, E., Noritake, M., Yamada, T., Izumitani, T. (1999), “Optimal size and location planning of public logistics terminals”, *Transportation Research Part E: Logistics and Transportation Review*, 35, 207-222.
- Teixeira, J. C., Antunes, A.P. (2008), “A hierarchical location model for public facility planning”, *European Journal of Operational Research*, 185, 92-104.
- Valero de Carvalho, J. M. (2005), “Using extra dual cuts to accelerate convergence in column generation”, *INFORMS Journal of Computing*, 17 (2), 75–182.

- Wentges, P. (1996), “Accelerating Benders’ decomposition for the capacitated facility location problem”, *Mathematical Methods of Operations Research*, 44, 267–290.
- Williams, H. P. (1978), “The Reformulation of Two Mixed Integer Programming Problems”, *Mathematical Programming*, 14, 325-331.
- Wolsey, L. A. (1998), *Integer Programming*, Wiley-Interscience, Series in Discrete Mathematics and Optimization, Toronto, (Chapter 1).
- Yamin, H.Y., Shahidehpour, S. M. (2004), “Unit commitment using a hybrid model between Lagrangian relaxation and genetic algorithm in competitive electricity markets”, *Electric Power Systems Research*, 68, 83-92.
- You, B., Yamada, T. (2011), “An exact algorithm for the budget-constrained multiple knapsack problem”, *International Journal of Computer Mathematics*, 88(16), 3380-3393.
- Zhang, J. (2006), “Approximating the two-level facility location problem via a quasi-greedy approach”, *Mathematical Programming, Series A*, 108, 159-176.

Chapter 3¹

ILP Models for Light Post Location Problem

3.1. Introduction

Many real-world facility location problems can be approximated by a grid-based system of small-sized cells. These cells can then be used to model a heterogeneous demand distribution. We can also express the amount of supply in each cell from its supply distribution relationships with the various potential facility locations. Based on these demand and supply relationships, we can then

¹A version of this chapter has been published: Noor-E-Alam, M., Ma, A., Doucette, J. (2012), "Integer Linear Programming Models for Grid-Based Light Post Location Problem", *European Journal of Operational Research*, 222, 17-30.

determine the optimal capacities and locations to place our facilities while fulfilling certain objectives. In this work, these types of location problems are referred to as *grid-based location problems* (GBLPs). In the GBLPs, we will seek the optimum number, location(s), and size(s) of facilities to place. The applications of GBLPs are wide ranging, and include problems in business, every discipline of engineering, defense, resource exploitation, and even the medical sciences. To make such complex decisions, we need to develop mathematical models, and procedures to solve them.

Determining optimal location is a common, and often complex, problem in business and engineering. Over the last several decades and especially in recent years, several methods have been developed in the area of location theory resulting in a number of notable solving methods. These methods are problem specific and particularly designed for the various types of the location problem. One of the most significant facility location problems was first proposed by Cooper (1963), now well-known as the multisource Weber problem. The Weber problem has a known number of facilities and all the fixed costs for the facilities are equal. Since 1963, a lot of research has been done on the Weber problem. Wesolowsky (1972) proposed a model for the solution of the Weber problem using rectilinear distances. Sherali and Noradi (1988) focused on a *capacitated multi-facility Weber problem* (CMFWP) and demonstrated that the CMFWP is

NP-Hard. Manzour-al-Ajdad et al. (2012) proposed an algorithm for solving a single-source CMFWP. Katz and Cooper (1974) first proposed a probabilistic multi-facility Weber problem which was later revisited by Altinel et al. (2009). A two-dimensional facility model is discussed by Francis (1964) to locate multiple new facilities with respect to existing facilities.

The complexity of the location problem depends on the nature of the problem and the criterion to be considered to make the decision. These criteria are selected by the decision maker from the problem description (Teixeira and Antunes, 2008). Marín (2011) described a new discrete location problem where the number of customers allocated to every plant has to be balanced. Ingolfsson et al. (2007) described an ambulance location optimization model that minimizes the number of ambulances needed to provide a specified service level. The model measures service level as the fraction of calls reached within a given time standard and considers response time to be composed of a random delay (prior to travel to the scene) plus a random travel time. Drezner and Wesolowsky (1997) proposed a method of placing signal detectors to cover a certain area such that the probability that an event is not detected is minimized.

Another special type of location problem is a location problem with the objective of coverage, which is first introduced by Church and ReVelle (1974). It ensures a set of facilities for each customer. The key applications of this model are to find

optimum location of emergency services, retail facilities, cell-phone towers and sensor networks. The well-known *uncapacitated facility location* (UFL) problem is similar to these problems, except for the consideration of variable transportation costs (Wolsey, 1998). Moreover, the UFL becomes a *capacitated facility location* (CFL) problem when there is an upper limit for the amount of supply (Ghiani et al., 2002; Chen, 2010). For more information on the coverage location models, readers are referred to Berman et al. (2010). Of all the models developed, it is important to recognize that models to solve the location problems can be classified into two distinct groups: discrete location analysis and continuous location analysis. Discrete location analysis, the most common form of modeling a location problem, typically refers to the use of a node-and-network (transportation) approach where facilities and supply points are modeled as the vertices and nodes (Domschke and Krispin, 1996). Continuous location analysis involves the modeling of the location problem on a continuous plane. With the continuous location-allocation problem, all customer demands are coordinate points and furthermore, the feasible solution for the optimal placement of the facilities can be any coordinate point in the plane. Daskin (1995) points out that modeling the location as a grid can be NP-Complete, and as such, a transportation network (discrete location analysis) is typically employed.

To make location decisions, we need to develop mathematical models, more specifically, *integer linear programming* (ILP) problems. To model a real-world problem, we generally need to consider a large number of discrete variables, a heterogeneous demand distribution, non-linear supply distributions, and fixed costs associated with facility placement. Furthermore, to get an optimum decision, the ILP models need to be designed in such a way that they will simultaneously determine the locations, sizes and number of facilities to achieve certain objectives. Combinations of these considerations make the problem a large scale ILP problem, which are generally not scalable and often become intractable even with small problems. Therefore different types of heuristics are used to find the near optimal solution. Genetic search algorithm has been used to find solutions for location problems (Abdinnour-Helm and Venkataramanan, 1998). A genetic search algorithm is also used by Taniguchi et al. (1999) to obtain a near optimal solution for a logistics terminal location problem that also factors in traffic conditions by using queuing theory and nonlinear programming to trade-off between both transportation and facility costs at terminals to minimize total logistics costs. In Aytug and Saydam (2002), a *genetic algorithm* (GA) is used to solve large-scale maximum expected covering location problems. Methods such as the *simultaneous perturbation stochastic approximation* (SPSA), *finite difference gradient* (FDG), and *very fast simulated annealing* (VFSA) algorithms have also been used. Bangerth et al. (2006) compared and analyzed the efficiency,

effectiveness, and reliability of these optimization algorithms for solving location problems. They found that none of these algorithms guarantees the optimal solution, but demonstrated that both SPSA and VFSA are very efficient in finding nearly optimal solutions with a high probability. Other methods for solving location problems proposed in the past few years have included methods such as a gravity model (Kubis and Hartmann, 2007), ILP-based formulations (Chen et al., 2005), the use of a Tabu search (Gendron and Potvin, 2003), and the usage of a Greedy Algorithm (Zhang, 2006). Canbolat and Wesolowsky (2010) proposed an alternate local search heuristic approach to solving the Weber problem with a probabilistic line barrier method.

In this chapter, we propose new formulations for a multisource location problem with the goal of determining the optimal combination for a facility distribution problem: the number of facilities, the location of each facility, and their capacities. Furthermore, in real-world situations, demand is not a singular point, but rather, many individual points located adjacent to each other forming a heterogeneous distribution that is extremely complex in nature. Such facility location problems can be approximated by a GBLP, where the entire area of this location problem is divided into small cells. These cells are then used to locate the heterogeneous demand distribution. On the other hand, we can express the amount of supply in each cell associated with each individual facility located in a

specific cell from its supply distribution relationship. From this demand distribution and supply relationship, we have to place sufficiently-sized facilities in such a way that we can fulfill certain objectives. Our research herein focuses on the development of ILP models for GBLP, using the grid-based light post location problem to make optimal decisions in installation of lights in a city park.

The remainder of this chapter is organized as follows. Section 3.2 provides the description of the problem with demand and supply calculation process. In Section 3.3, we discuss the basic model with simplified supply distribution. In Section 3.4, we propose two enhanced models with enhanced supply distribution. Section 3.5 describes the result analysis. Section 3.6 ends with conclusions and future research opportunities.

3.2. Problem Description

Suppose we consider a city park, described as a 2-dimensional grid of known dimensions. Light posts must be installed throughout the park to provide adequate lighting conditions. We must determine the location and light intensity of each light post such that dark areas are lit and excess (waste) lighting is minimized. The brighter the light source, the more expensive this will be due to installation and electricity costs. As such, the objective is to satisfy the demand as much as possible while minimizing excess supply. Factors affecting the number of lights, their size, and their placement are many and varied. In a city park, there are

different areas used for various purposes. Trees in the park and its topography create demand variation throughout the park. Furthermore, installing lights in boundary regions would not be feasible due to various physical restrictions such as roadways and underground power cables for utility service. This city park can be represented by a grid-based area, where the heterogeneous demand distribution can be represented by each cell in the grid. The idea is that the light sources should be placed in such a way that the areas they illuminate don't overlap too much, but not so far apart that there are unlit cells.

3.2.1 Supply Calculations

Supply for a grid cell associated with each source can be calculated on the basis of the distribution of light brightness throughout the grid. It is well known that an inverse square relationship exists between the brightness of light and the distance from the light source (Simons and Bean, 2001). According to this relationship, brightness at distance r can be calculated with equation (3.1), where S is the supply at distance r and P is the luminosity of the point source. Here, units of P and S are candela and candela per square meter respectively.

$$S = \frac{P}{r^2} \tag{3.1}$$

Figure 3.1 shows the apparent brightness of a source with luminosity P at distances 1, 2, 3, 4 and 5 distance units with the above relationship. The surface brightness decreases as the distance increases because the light must spread out over a larger surface. However, this relationship is not entirely accurate; various locations on a horizontal plane will have slightly different degrees of brightness since they will all be somewhat different distances from the point source of light. A better representation of the geometry can be seen in Figure 3.2, where the brightness can be calculated with equations (3.2), (3.3), and (3.4) (Simons and Bean, 2001). In this model, the vertical distance between the point source, o , and point a on the horizontal plane is r , as we used it in the previous simplified model, above. The angle between vertical line oa and line ob is α_b , and the angle between vertical line oa and line oc is α_c . In this relationship, as we move along the horizontal plane away from point a , the distance between the point source and the various locations on the plane will increase. As a result, the amount of supply will decrease according to equations (3.2), (3.3), and (3.4), where S_a , S_b and S_c represent the total amount of available supply at location a , b and c .

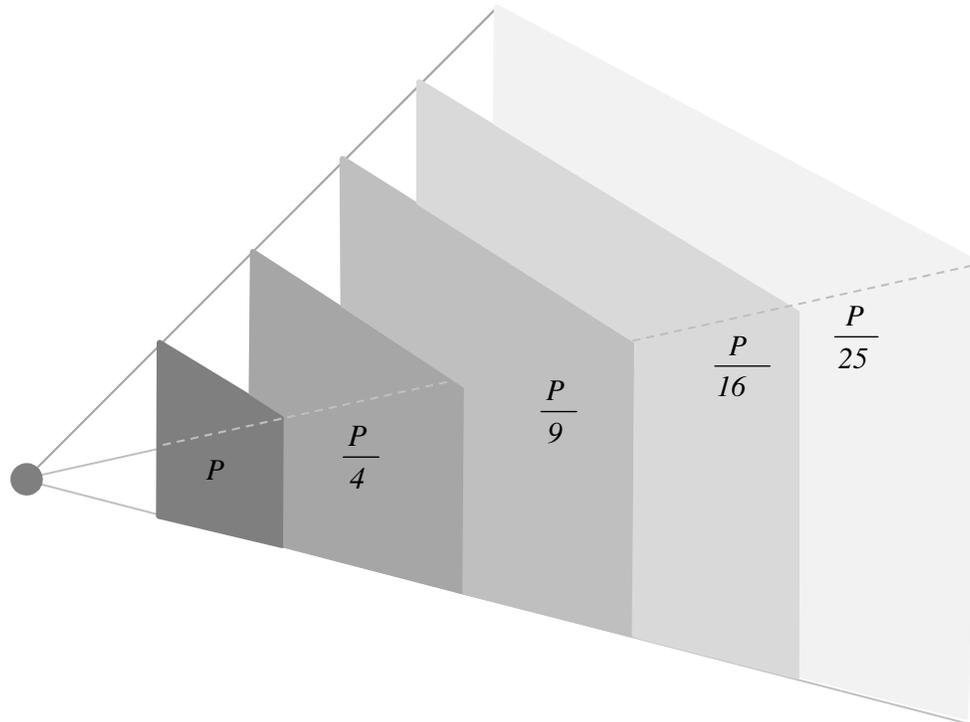


Figure 3.1: Basic relationship between light brightness and distance, adapted from NASA (2006).

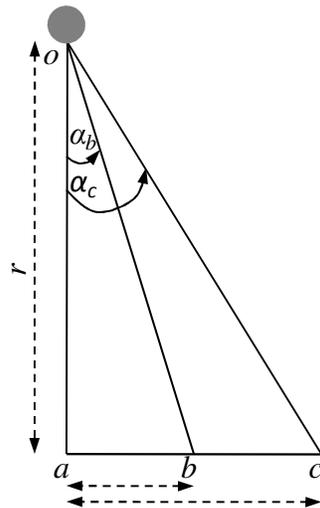


Figure 3.2: Geometry of a point source of light above a horizontal plane.

$$S_a = \frac{P}{r^2} \quad (3.2)$$

$$S_b = \frac{P}{r^2} \cos(\alpha_b) \quad (3.3)$$

$$S_c = \frac{P}{r^2} \cos(\alpha_c) \quad (3.4)$$

3.3. Basic Model

As mentioned earlier, our goal is to develop one or more optimization models that will help us to determine the location, size, and number of light posts to place in order that we can achieve an optimal distribution of light over a grid with varying demands for light intensity. As a first step, we develop a basic model, which we will expand on in later, more accurate models.

3.3.1. Simplified Supply Calculation

Before presenting the model itself, we first need to simplify the scale of the problem by assuming that each cell within the grid is uniform throughout the entire cell. In other words, the amount of light intensity at one point in the cell is the same as in all other points in the cell. Furthermore, we will assume that certain neighboring cells will have identical light intensities, as shown in Figure 3.3.

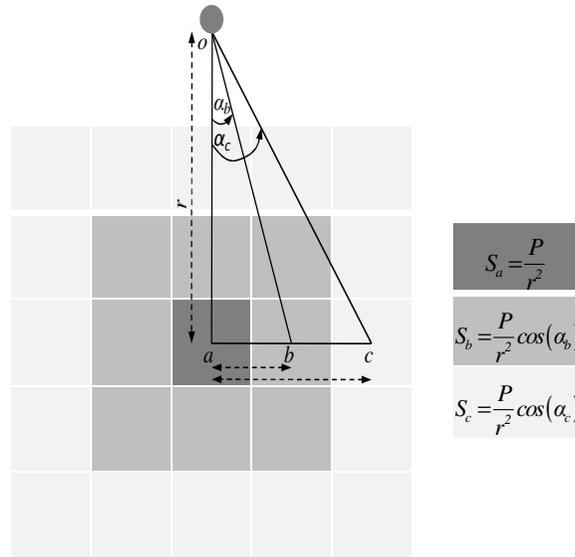


Figure 3.3: Distribution of light supply for the basic model.

Here, the individual cell directly below the light source will have the light intensity calculated by equation (3.2). Then if we expand out to a 3x3 grid surrounding that cell, all of those new cells will have the light intensity calculated by equation (3.3). Expanding again to a 5x5 grid centered on the light source, then those cells will have the light intensity calculated by equation (3.4), and so on. We can also note that angles α_b and α_c can be calculated by equations (3.5) and (3.6), where ab is the distance between points a and b , and ac is the distance between points a and c . This will become particularly important as we further develop our models.

$$\alpha_b = \tan^{-1}\left(\frac{ab}{r}\right) \quad (3.5)$$

$$\alpha_c = \tan^{-1}\left(\frac{ac}{r}\right) \quad (3.6)$$

3.3.2. Basic Optimization Model

In order to formulate our basic optimization model, we first need to define the notation we will use, as follows:

Q is the set of all x-coordinates in the grid, indexed by i

R is the set of all y-coordinates in the grid, indexed by j

$D_{i,j}$ is the demand for the grid point whose coordinates are at (i,j)

i_{max} is the maximum value of i

i_{min} is the minimum value of i

j_{max} is the maximum value of j

j_{min} is the minimum value of j

β is the system boundary constant

N is the set of all light sources, indexed by n

n_{opt} is the optimum number of light sources

S_{ij} is the total supply at location (i,j)

S_{ijn} is the supply at location (i,j) from the n^{th} light post

S_{in} is the supply at location (i) in x direction from the light source n

S_{jn} is the supply at location (j) in y direction from the light source n

x_n is the x coordinate of the optimal location of the n^{th} light post

y_n is the y coordinate of the optimal location of the n^{th} light post

P_n is the optimum size of the n^{th} light source (required to be integer)

UB is the upper bound on decision variable P_n

It is obvious from the nature of our light demand distribution that the optimal placement of the light post would satisfy as much demand as possible while also minimizing the light source surplus. Therefore, our objective function will be to minimize both the unmet demand and extra supply of light. To fulfill this goal, equation (3.7) will be used as the objective function for the basic model:

$$\text{Minimize } \sum_{i \in Q} \sum_{j \in R} |D_{ij} - S_{ij}| \quad (3.7)$$

In order to minimize this objective function, we now define the feasible region with the following constraint equations, being careful to appropriately represent the model illustrated in Figure 3.3. To do so, we first need to recognize that the amount of light, S_{ijn} , available at coordinates (i, j) is the minimum of the light available if we calculate along the x-axis only, or along the y-axis only. In order to arrive at the amount of available light for some arbitrary cell, we can calculate the available light the cell would have if we considered only the x-axis direction and then only the y-axis direction, and take the smaller of the two values. For instance, according to the discussion relating to Figure 3.3, the cell second from the top and furthest to the left has available light as defined by equation (3.4). If

we calculate the available light at that cell using only the x-axis direction, we would find the distance from the light source would be ac , and therefore we would use equation (3.4). However, if we calculate the available light using only the y-axis direction, we would find the distance from the light source would be ab , and so we would need to use equation (3.3) instead. The former provides the smaller value for available light at that cell, and so that is the equation we must use.

From the above illustration of light source distribution, equations (3.8) and (3.9) can be used to calculate the supply of the n^{th} light source in the x-axis and y-axis directions, with a single source located at (x_n, y_n) . These equations follow from equations (3.2), (3.3), and (3.4), above.

$$S_{in} = \frac{P_n}{r^2} \cos\left(\tan^{-1}\left(\frac{|i-x_n|}{r}\right)\right) \quad \forall i, n \quad (3.8)$$

$$S_{jn} = \frac{P_n}{r^2} \cos\left(\tan^{-1}\left(\frac{|j-y_n|}{r}\right)\right) \quad \forall i, n \quad (3.9)$$

Finally, the supply of light at location (i, j) is calculated with equation (3.10) for n^{th} light source. Total supply for all light sources is calculated with equation (3.11).

$$S_{ijn} = \min(S_{in}, S_{jn}) \quad \forall i, j, n \quad (3.10)$$

$$S_{ij} = \sum_{n \in N} S_{ijn} \quad \forall i, j \quad (3.11)$$

The following two bounding constraints (3.12) and (3.13) are incorporated so that our optimization problem will not consider the feasible region very near to the boundary area. Constraint equation (3.14) is used to put an upper bound on decision variable P_n .

$$i_{min} + \beta \leq x_n \leq i_{max} - \beta \quad \forall n \quad (3.12)$$

$$j_{min} + \beta \leq y_n \leq j_{max} - \beta \quad \forall n \quad (3.13)$$

$$0 \leq P_n \leq UB \quad \forall n \quad (3.14)$$

The system boundary constant, β , can be determined empirically on the basis of problem description, and the upper bound, UB , is determined depending on the maximum magnitude of the demand. Equations (3.7)-(3.14) constitute the basic optimization model.

3.3.3. Equivalent Basic ILP Model

In the above optimization model, the objective function and some of the constraint equations are not linear. Our objective of this research is to develop an equivalent *integer linear programming* (ILP) model to get the optimum number, locations, and sizes of light posts.

3.3.3.1 Linearization of Objective Function

In our basic model, the objective function has an absolute operator, which makes this a non-linear function. However, we can develop a set of equivalent linear equations to handle this nonlinearity in the objective function. We introduce the following new notation in addition to the notation we have already used:

ED_{ij} is the excess demand at location (i,j)

ES_{ij} is the excess supply at location (i,j)

They can now linearize the objective function as follows in equations (3.15)-(3.19):

$$\text{Minimize } \sum_{i \in Q} \sum_{j \in R} (ED_{ij} + ES_{ij}) \quad (3.15)$$

$$ED_{ij} \geq D_{ij} - S_{ij} \quad \forall i, j \quad (3.16)$$

$$ED_{ij} \geq 0 \quad \forall i, j \quad (3.17)$$

$$ES_{ij} \geq S_{ij} - D_{ij} \quad \forall i, j \quad (3.18)$$

$$ES_{ij} \geq 0 \quad \forall i, j \quad (3.19)$$

3.3.3.2. Linearization of Constraint Equations

From the nature of light distribution, we know that the further a cell is located from the source cell, the less the light supply there will be. For simplification, we ignore as negligible any supply more than 2 units distant from the source, as illustrated in Figure 3.3. To develop an equivalent linear equation for constraint equations (3.8) and (3.9), sets of piecewise if-then constraints in (3.20) and (3.21) are developed. From this it follows that we may have seven possible cases in the x-axis direction and seven possible cases in the y-axis direction.

$$S_{in} = \begin{cases} \frac{p_n}{r^2} & \text{if } i = x_n \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{1}{r}\right)\right) & \text{if } i = x_n + 1 \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{2}{r}\right)\right) & \text{if } i = x_n + 2 \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{1}{r}\right)\right) & \text{if } i = x_n - 1 \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{2}{r}\right)\right) & \text{if } i = x_n - 2 \\ 0 & \text{if } i > x_n + 2 \\ 0 & \text{if } i < x_n - 2 \end{cases} \quad \forall i, n \quad (3.20)$$

$$S_{jn} = \begin{cases} \frac{p_n}{r^2} & \text{if } j = y_n \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{1}{r}\right)\right) & \text{if } j = y_n + 1 \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{2}{r}\right)\right) & \text{if } j = y_n + 2 \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{1}{r}\right)\right) & \text{if } j = y_n - 1 \\ \frac{p_n}{r^2} \cos\left(\tan^{-1}\left(\frac{2}{r}\right)\right) & \text{if } j = y_n - 2 \\ 0 & \text{if } j > y_n + 2 \\ 0 & \text{if } j < y_n - 2 \end{cases} \quad \forall j, n \quad (3.21)$$

However, if-then constraints are themselves non-linear. Their equivalent linear equations can be developed using the method from Winston and Venkataramanan (2003). If an if-then constraint can be expressed in a form where: if some function $K(x_1, x_2, \dots, x_n) > 0$, then some other function $L(x_1, x_2, \dots, x_n) \geq 0$, then we can replace that if-then pair of equations with the following two linear equations where $y \in \{0,1\}$ and M is some large positive number:

$$-L(x_1, x_2, \dots, x_n) \leq My \quad (3.22)$$

$$K(x_1, x_2, \dots, x_n) \leq M(1 - y) \quad (3.23)$$

To use this technique, we can express the constraints in the first part of equation (3.20) with equations (3.24)-(3.27), where z_1 and z_2 are binary variables:

$$z_1 = 1 \quad \text{if } i+1 > x_n \quad \forall n \quad (3.24)$$

$$z_2 = 1 \quad \text{if } x_n + 1 > i \quad \forall n \quad (3.25)$$

$$S_{in} = \frac{P_n}{r^2} \quad \text{if } z_1 + z_2 > 1 \quad \forall n \quad (3.26)$$

$$S_{in} = 0 \quad \text{if } z_1 + z_2 < 2 \quad \forall n \quad (3.27)$$

We can do the same for the remaining constraints in equation (3.20) and those in equation (3.21) as well, where each of the 14 separate if-then constraints is expressed as an equivalent set of four new if-then constraint equations. Each of those is further converted into an equivalent pair of linear constraints as equations (3.22) and (3.23), for a total of 112 sets of constraints (“sets” because we have one of each of those constraints for each light source, n). And this is only for linearization of constraints (3.20) and (3.21). Equation (3.10) must also be linearized in a similar manner.

3.3.4. Preliminary Results

We solve our instance of the above problem on an 8 processor ACPI multiprocessor X64-based PC with Intel Xeon® CPU X5460 running at 3.16 GHz with 32 GB memory. We have implemented our models in AMPL (Fourer et. al., 2002), and used CPLEX 11.2 solver (ILOG, 2007) to solve them.

To obtain preliminary results, we chose a small problem with a 7×7 grid, shown in Figure 3.4. In this solution, other parameters are assumed as follows: $\beta = 2$, $UB = 10$, and $r = 2$. In addition, we solve five instances of the problem, with 1, 2, 3, 4, and 5 light sources. We plot the objective function values of the five instances of the problem in Figure 3.5. While we show data for all five instances, only those with 1 and 2 light sources solve to optimality in a five-day runtime window (that's five days for each instance). For the instances with 3, 4, and 5 light sources, after a five-day runtime for each, the solver was only able to obtain sub-optimal solutions with optimality gaps of 8%, 64%, and 79%, respectively. This suggests that this model is practically intractable, likely a result of the great many binary variables and associated constraints, which tend to make optimization problems computationally heavy. This drives our efforts for an enhanced model in the next section.

2.10	2.40	2.40	2.30	2.20	1.60	0.50
2.40	2.50	2.60	2.40	2.30	1.60	0.50
2.40	2.60	2.50	2.40	2.40	1.70	0.60
2.40	2.40	2.50	2.50	2.50	1.80	0.60
2.30	2.40	2.40	2.60	2.50	1.90	0.60
1.70	1.70	1.70	1.80	1.90	1.20	0.50
0.60	0.60	0.60	0.60	0.60	0.50	0.00

Figure 3.4: Demand distribution for 7x7 grid.

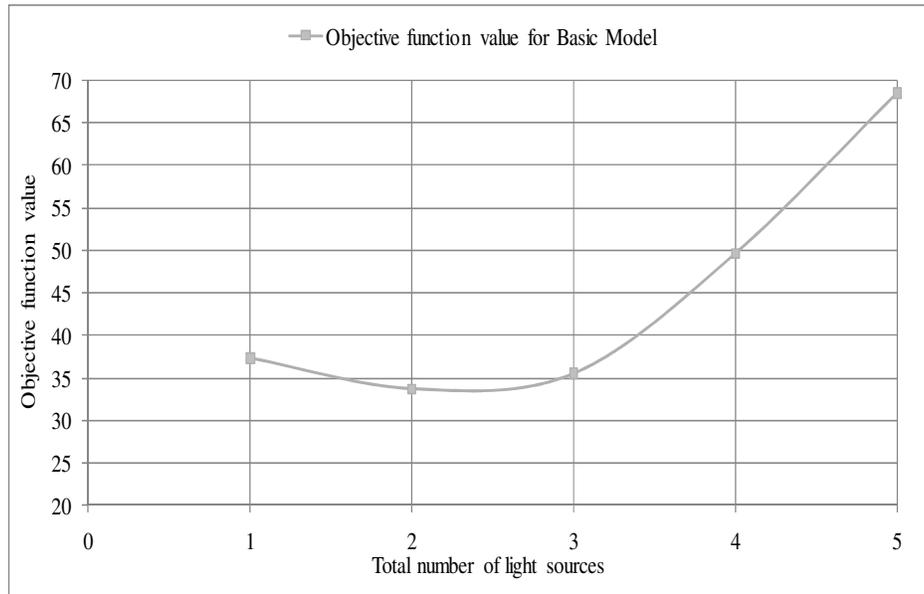


Figure 3.5: Objective function values for 7x7 grid test cases solved with the basic ILP model.

3.4. Enhanced Models

To reduce the above computational burden in the basic ILP model and to represent a more precise light distribution model, we can now propose enhanced models. The light distribution model used above does not accurately represent the real light distribution model. Instead, various locations on a horizontal plane will have different degrees of brightness since they will all be somewhat different distances from the point source of light. Because the rectilinear distances from the point source to the different cells are not the same, neighboring cells do not necessarily have identical light intensities. A better representation of the supply distribution can be seen in Figure 3.6, where the brightness can be calculated with

the relationships shown. These relationships are determined according to the brightness calculation process described in Figure 3.2 with equations (3.2), (3.3), and (3.4), except that in the present figure, distances are calculated as the 3-dimensional Euclidean distances between the light source and the centers of the various cells. In those equations, the vertical distance between the point source, o , and centre point of a grid, a , is r , as we used it in the previous model, above. The angle between vertical line oa and line ob or ob' is α_b , the angle between vertical line oa and line oc is α_c , the angle between vertical line oa and line od or od' is α_d , the angle between vertical line oa and line oe or oe' is α_e , and the angle between vertical line oa and line of is α_f . In the following two enhanced ILP models we will consider this exact calculation of light supply when optimizing light post locations.

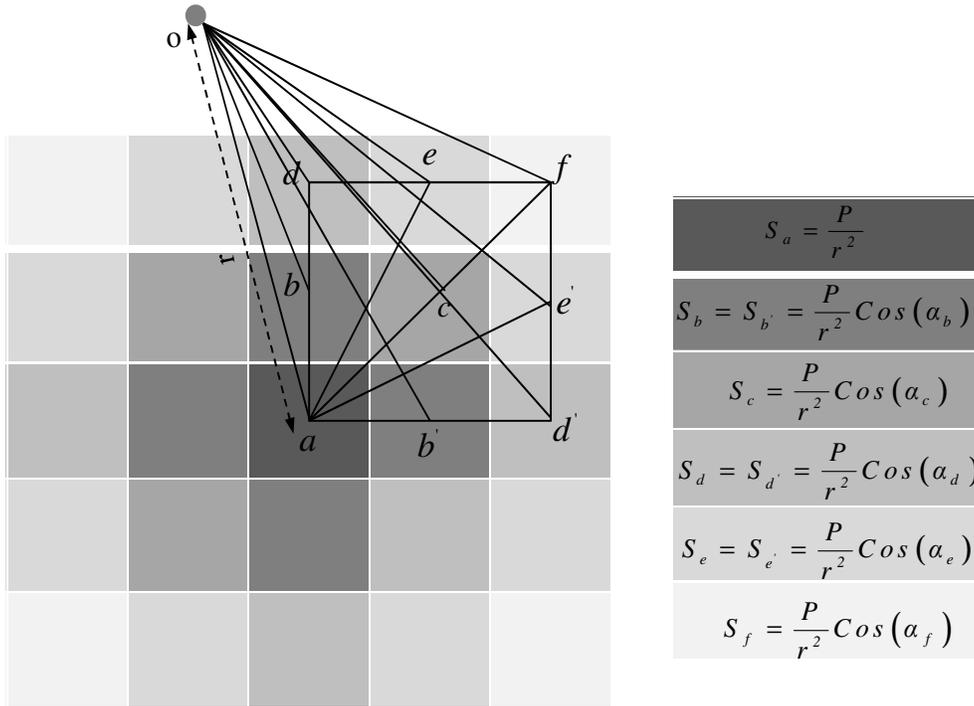


Figure 3.6: Exact distribution of light supply.

3.4.1. Model 2

In the basic model we have a large number of constraint equations and 0/1 decision variables, even for a small problem (say, the 7x7 grid solved earlier) and a simplified demand distribution. As a result, that ILP model is simply not scalable as described. To overcome this difficulty we have developed a second ILP model, in which we use the same objective function, but a highly simplified approach for defining the feasible region. In fact, we can replace the original sets of constraints with a single set by pre-processing supply data.

We see in the basic model that it is required to consider $112n$ linear constraint equations just for equations (3.8) and (3.9). As a result we need to consider a huge number of linear constraints and binary variables in order to calculate the simplified supply distribution. To overcome this difficulty, we use the following constraint equation (3.28), where the x and y coordinates of the location are considered as parameters, instead of variables. We introduce the following new notation in addition to the notation we have already used:

X is the set of all x -coordinates of the light source, indexed by x ,

$$i_{min} + \beta \leq x \leq i_{max} - \beta$$

Y is the set of all y -coordinates of the light source, indexed by y ,

$$j_{min} + \beta \leq y \leq j_{max} - \beta$$

P_{xy} is the size of the light source at location (required to be integer)

UB_{xy} is the upper bound on decision variable P_{xy}

From the illustration of light source distribution (in Figure 3.6), equation (3.28) can be used to calculate the supply in each cell (i,j) . This equation follows from the relationships provided in Figure 3.6, above. Suppose the location of a light post is at point a , and we want to calculate the supply at point e . In this case we need to find the value of angle α_e , to calculate the supply at point e . This angle can be calculated by the ratio of ae and r . The value of ae is the distance between the point, a and the point, e . On the other hand, the total supply in a particular cell

(i,j) is the summation of all the individual supplies coming from all light sources. Considering these facts, the following equation can be used to calculate the supply in each cell:

$$S_{ij} = \sum_{x \in X} \sum_{y \in Y} \frac{P_{xy}}{r^2} \cos \left(\tan^{-1} \left(\frac{\sqrt{(i-x)^2 + (j-y)^2}}{r} \right) \right) \quad \forall i, j \quad (3.28)$$

Furthermore, bounding constraint (3.29) is used to put an upper bound on decision variable P_{xy} .

$$0 \leq P_{xy} \leq UB_{xy} \quad \forall x, y \quad (3.29)$$

The objective function, (3.15), along with the constraint equations, (3.16)-(3.19), (3.28) and (3.29), constitute the whole of our first enhanced ILP model, which we will refer to as Model 2. To better handle this model, the $\cos()$ component in the equation (3.28) is calculated as a part of data pre-processing and fed into the model. We can note that in Model 2, we do not actually have any capability to control the number of light sources, and in fact, it is conceivable that a light source could be placed at each cell. While this can be dealt with by including a cost (in the objective function) for each light source we place, we can also more directly control that as we did in the basic model. We do that in our next enhanced model, which we will call Model 3.

3.4.2. Model 3

To control the number of sources in Model 2, we can incorporate the following sets of constraint equations, (3.30)-(3.32), where T_{xy} is a binary variable and n_a is the total number of allowable light sources:

$$T_{xy} = 1 \quad \text{if } P_{xy} > 0 \quad \forall x, y \quad (3.30)$$

$$T_{xy} = 0 \quad \text{if } P_{xy} = 0 \quad \forall x, y \quad (3.31)$$

$$\sum_{x \in X} \sum_{y \in Y} T_{xy} = n_a \quad (3.32)$$

Finally, the objective function, (3.15), along with the constraint equations, (3.16)-(3.19) and (3.28)-(3.32) constitute our second enhanced ILP model, which is referred to as Model 3. By controlling the number of light sources, we can optimize our problem for various instances of n_a in order to observe how this will impact the objective function, which provides additional insights we might not have otherwise.

As with many of the equations we've seen so far, equations (3.30) and (3.31) are non-linear, and so they too will need to be replaced with equivalent linear equations, as we did earlier. We can use the technique outlined in equations (3.22)-(3.23) to develop equivalent linear equations for these two non-linear equations.

3.5. Result Analysis

We use the same experimental set up described earlier to run test case solutions for the enhanced models, Model 2 and Model 3. However, because these enhanced models are much more scalable, we use the larger 10x10, 10x12, 12x12, 10x15, 10x17, 10x20 and 15x15 test-case grids with demands shown in Figure 3.7 to Figure 3.13. We used a CPLEX *mipgap* setting of 0.001, which means all test cases solved to full termination are provably within 0.1% of optimality.

Figures 3.14 through 3.20 show the respective solution data for these seven test-case grids. In each of those figures, the square data points represent the *optimum objective function values* (OOFV) of the optimally solved test case with the indicated number of light sources (i.e., input parameter, n_a) using Model 3. The triangular data points represent the OOFV of the Model 2 solutions, and we note that for these data points, the number of light sources indicated along the x-axis is not an input, but rather, is obtained from the solution itself along with the OOFV. The diamond data points represent the CPU time required to solve the test cases with the indicated number of light sources. Note that in all seven of these figures, the OOFV data points are to be read against the left-hand y-axes, while CPU time data points are to be read against the right-hand y-axes.

0.36	0.55	0.73	0.84	0.98	1.21	1.45	0.96	0.46	0.12
0.53	0.84	1.17	1.23	1.38	1.82	1.66	1.49	0.66	0.16
0.66	1.11	1.31	1.52	1.51	1.68	1.90	1.35	0.68	0.17
0.68	1.06	1.42	1.43	1.45	1.51	1.55	1.35	0.70	0.17
0.68	1.02	1.30	1.35	1.34	1.36	1.45	1.14	0.76	0.19
0.67	1.05	1.42	1.32	1.20	1.13	1.09	1.01	0.53	0.13
0.63	1.10	1.21	1.30	1.02	0.88	0.77	0.62	0.34	0.08
0.45	0.73	1.01	0.86	0.71	0.60	0.50	0.38	0.20	0.05
0.23	0.35	0.44	0.41	0.35	0.30	0.24	0.18	0.10	0.02
0.06	0.09	0.11	0.10	0.09	0.07	0.06	0.04	0.02	0.00

Figure 3.7: Demand distribution for 10x10 grid.

0.76	0.81	0.93	0.93	0.95	0.88	0.90	0.97	1.02	0.89	0.73	0.58
1.03	1.12	1.22	1.30	1.23	1.22	1.30	1.49	1.72	1.39	1.09	0.86
1.27	1.42	1.55	1.54	1.47	1.46	1.58	1.96	1.89	1.86	1.37	1.10
1.48	1.74	2.03	1.83	1.65	1.56	1.60	1.78	2.01	1.69	1.42	1.26
1.62	2.03	2.05	2.10	1.73	1.55	1.50	1.53	1.57	1.47	1.37	1.37
1.53	1.75	2.05	1.86	1.60	1.42	1.32	1.28	1.26	1.24	1.24	1.37
1.25	1.40	1.60	1.67	1.41	1.20	1.08	1.01	0.98	0.97	1.00	1.09
0.87	1.02	1.26	1.22	1.18	0.91	0.77	0.70	0.67	0.67	0.68	0.73
0.45	0.53	0.64	0.76	0.61	0.48	0.40	0.36	0.34	0.34	0.35	0.36
0.11	0.13	0.16	0.19	0.15	0.12	0.10	0.09	0.09	0.08	0.09	0.09

Figure 3.8: Demand distribution for 10x12 grid.

1.05	1.12	1.21	1.31	1.33	1.35	1.38	1.37	1.40	1.46	1.53	1.23
1.13	1.19	1.27	1.35	1.42	1.51	1.61	1.54	1.49	1.48	1.44	1.26
1.20	1.25	1.31	1.39	1.49	1.65	1.66	1.68	1.57	1.51	1.48	1.40
1.24	1.29	1.35	1.42	1.50	1.61	1.71	1.63	1.57	1.54	1.57	1.38
1.28	1.33	1.38	1.44	1.50	1.56	1.60	1.57	1.53	1.50	1.46	1.40
1.31	1.35	1.40	1.45	1.49	1.53	1.55	1.53	1.50	1.45	1.39	1.27
1.33	1.38	1.42	1.46	1.50	1.53	1.54	1.51	1.47	1.42	1.37	1.28
1.36	1.40	1.45	1.48	1.52	1.55	1.55	1.51	1.46	1.40	1.38	1.21
1.38	1.44	1.47	1.50	1.54	1.59	1.61	1.53	1.43	1.34	1.26	1.17
1.41	1.50	1.51	1.52	1.56	1.65	1.76	1.58	1.41	1.27	1.15	1.02
1.46	1.62	1.54	1.50	1.54	1.71	1.65	1.67	1.35	1.18	1.06	0.94
1.48	1.50	1.55	1.40	1.38	1.44	1.53	1.35	1.18	1.05	0.97	0.91

Figure 3.9: Demand distribution for 12x12 grid.

0.76	0.81	0.93	0.93	0.95	0.88	0.90	0.97	1.02	0.89	0.73	0.58	0.45	0.31	0.16
1.03	1.12	1.22	1.30	1.23	1.22	1.30	1.49	1.72	1.39	1.09	0.86	0.67	0.48	0.25
1.27	1.42	1.55	1.54	1.47	1.46	1.58	1.96	1.89	1.86	1.37	1.10	0.90	0.68	0.37
1.48	1.74	2.03	1.83	1.65	1.56	1.60	1.78	2.01	1.69	1.42	1.26	1.15	0.99	0.54
1.62	2.03	2.05	2.10	1.73	1.55	1.50	1.53	1.57	1.47	1.37	1.37	1.46	1.57	0.81
1.53	1.75	2.05	1.86	1.60	1.42	1.32	1.28	1.26	1.24	1.24	1.37	1.78	1.48	1.14
1.25	1.40	1.60	1.67	1.41	1.20	1.08	1.01	0.98	0.97	1.00	1.09	1.27	1.44	0.75
0.87	1.02	1.26	1.22	1.18	0.91	0.77	0.70	0.67	0.67	0.68	0.73	0.77	0.74	0.42
0.45	0.53	0.64	0.76	0.61	0.48	0.40	0.36	0.34	0.34	0.35	0.36	0.36	0.32	0.19
0.11	0.13	0.16	0.19	0.15	0.12	0.10	0.09	0.09	0.08	0.09	0.09	0.09	0.08	0.05

Figure 3.10: Demand distribution for 10x15 grid.

0.96	0.92	0.88	0.84	0.80	0.78	0.75	0.72	0.68	0.67	0.64	0.61	0.55	0.49	0.42	0.39	0.37
0.95	0.89	0.83	0.79	0.77	0.76	0.76	0.74	0.73	0.72	0.71	0.68	0.63	0.56	0.50	0.45	0.41
0.93	0.85	0.77	0.71	0.72	0.75	0.78	0.79	0.79	0.80	0.80	0.78	0.71	0.63	0.56	0.50	0.45
0.92	0.82	0.70	0.57	0.66	0.75	0.81	0.84	0.87	0.89	0.91	0.93	0.81	0.70	0.60	0.54	0.49
0.93	0.82	0.62	0.61	0.60	0.77	0.86	0.91	0.95	0.98	1.04	0.98	0.91	0.74	0.62	0.56	0.51
0.98	0.89	0.78	0.66	0.77	0.87	0.95	1.00	1.03	1.05	1.05	1.04	0.89	0.73	0.60	0.55	0.52
1.04	0.99	0.93	0.90	0.94	1.00	1.06	1.11	1.13	1.12	1.08	1.01	0.87	0.69	0.50	0.51	0.53
1.12	1.09	1.07	1.06	1.09	1.13	1.18	1.23	1.27	1.22	1.15	1.05	0.89	0.65	0.57	0.48	0.55
1.20	1.19	1.19	1.20	1.22	1.26	1.31	1.37	1.37	1.35	1.25	1.14	1.00	0.83	0.65	0.65	0.64
1.28	1.29	1.30	1.32	1.34	1.38	1.41	1.45	1.48	1.42	1.35	1.26	1.15	1.03	0.91	0.83	0.76

Figure 3.11: Demand distribution for 10x17 grid.

0.35	0.61	0.71	0.82	0.76	0.81	0.93	0.93	0.95	0.88	0.90	0.97	1.02	0.89	0.73	0.58	0.45	0.31	0.16	0.04
0.44	0.69	0.94	0.98	1.03	1.12	1.22	1.30	1.23	1.22	1.30	1.49	1.72	1.39	1.09	0.86	0.67	0.48	0.25	0.06
0.50	0.76	0.99	1.13	1.27	1.42	1.55	1.54	1.47	1.46	1.58	1.96	1.89	1.86	1.37	1.10	0.90	0.68	0.37	0.09
0.56	0.86	1.13	1.30	1.48	1.74	2.03	1.83	1.65	1.56	1.60	1.78	2.01	1.69	1.42	1.26	1.15	0.99	0.54	0.14
0.61	0.98	1.36	1.45	1.62	2.03	2.05	2.10	1.73	1.55	1.50	1.53	1.57	1.47	1.37	1.37	1.46	1.57	0.81	0.20
0.62	1.08	1.28	1.51	1.53	1.75	2.05	1.86	1.60	1.42	1.32	1.28	1.26	1.24	1.24	1.37	1.78	1.48	1.14	0.29
0.50	0.81	1.15	1.17	1.25	1.40	1.60	1.67	1.41	1.20	1.08	1.01	0.98	0.97	1.00	1.09	1.27	1.44	0.75	0.19
0.34	0.53	0.70	0.78	0.87	1.02	1.26	1.22	1.18	0.91	0.77	0.70	0.67	0.67	0.68	0.73	0.77	0.74	0.42	0.10
0.17	0.26	0.34	0.39	0.45	0.53	0.64	0.76	0.61	0.48	0.40	0.36	0.34	0.34	0.35	0.36	0.36	0.32	0.19	0.05
0.04	0.06	0.08	0.10	0.11	0.13	0.16	0.19	0.15	0.12	0.10	0.09	0.09	0.08	0.09	0.09	0.09	0.08	0.05	0.00

Figure 3.12: Demand distribution for 10x20 grid.

0.51	0.51	0.54	0.62	0.74	0.88	0.78	0.70	0.65	0.64	0.66	0.69	0.69	0.58	0.43
0.72	0.72	0.77	0.88	1.10	1.12	1.16	1.00	0.94	0.94	0.98	1.07	1.17	0.91	0.64
0.88	0.89	0.93	1.02	1.17	1.33	1.25	1.19	1.17	1.18	1.25	1.45	1.35	1.24	0.83
1.01	1.01	1.05	1.12	1.21	1.31	1.33	1.35	1.38	1.37	1.40	1.46	1.53	1.23	0.90
1.09	1.10	1.13	1.19	1.27	1.35	1.42	1.51	1.61	1.54	1.49	1.48	1.44	1.26	0.96
1.15	1.16	1.20	1.25	1.31	1.39	1.49	1.65	1.66	1.68	1.57	1.51	1.48	1.40	1.05
1.19	1.21	1.24	1.29	1.35	1.42	1.50	1.61	1.71	1.63	1.57	1.54	1.57	1.38	1.16
1.22	1.24	1.28	1.33	1.38	1.44	1.50	1.56	1.60	1.57	1.53	1.50	1.46	1.40	1.07
1.24	1.27	1.31	1.35	1.40	1.45	1.49	1.53	1.55	1.53	1.50	1.45	1.39	1.27	1.00
1.25	1.29	1.33	1.38	1.42	1.46	1.50	1.53	1.54	1.51	1.47	1.42	1.37	1.28	0.99
1.26	1.31	1.36	1.40	1.45	1.48	1.52	1.55	1.55	1.51	1.46	1.40	1.38	1.21	1.01
1.26	1.32	1.38	1.44	1.47	1.50	1.54	1.59	1.61	1.53	1.43	1.34	1.26	1.17	0.90
1.25	1.32	1.41	1.50	1.51	1.52	1.56	1.65	1.76	1.58	1.41	1.27	1.15	1.02	0.82
1.21	1.31	1.46	1.62	1.54	1.50	1.54	1.71	1.65	1.62	1.35	1.18	1.06	0.94	0.79
1.14	1.25	1.48	1.50	1.55	1.40	1.38	1.44	1.53	1.35	1.18	1.05	0.97	0.91	0.84

Figure 3.13: Demand distribution for 15x15 grid.

We can observe that in general, Model 3 is much more capable than Model 1 of solving larger test cases in a reasonable period of time, at least those of intermediate size (e.g., 10x17). For instance, all solutions for the 10x10, 10x12, 12x12, 10x15 and 10x17 test cases were solved to optimality (well, within the 0.1% optimality gap specified above). The highly irregular nature of CPU times for those test cases was unexpected, but we think the reason is because

peculiarities within the problem, although minor in the grand scheme of things, can add enough additional complexity to individual test cases that the underlying complexity of the problem is overwhelmed and CPU time can triple, say from approximately 12 seconds or so on the 10x17 test case with 20 light sources to a little under 40 seconds for the test case with 21 light sources. In other words, due to the heterogeneity of demand distribution, some instances of the problem might create much tighter LP relaxations than other instances when we add or take away a light source, and/or algorithms used by CPLEX's internal branch-and-bound procedures might be better suited to some of those specific cases. However, in the larger test cases on the 10x20 grid and 15x15 grid, the underlying complexity becomes larger and stable enough that those minor variations and peculiarities in the problem are not enough to significantly impact the CPU time, so we can observe a more well-defined increase in solution times as the problem becomes more complex (i.e., we add more light sources, increasing n_a , and therefore increasing the number of constraints in the problem).

In Model 3, we can observe that the optimum number of light sources (n_{opt}) is 13 for the 10x10 grid, 16 for the 10x12 grid, 19 for the 12x12 grid, 23 for the 10x15 grid and 15 for the 10x17 grid, and the OOFV of these solutions correspond to the optimal solutions using Model 2 on those same grids. We can also observe that, at least for the 10x10, 10x12, 12x12, 10x15 and 10x17 grids, objective function

values initially decrease as we increase the number of light sources. This continues until we reach the Model 2 optimal solution (i.e., an optimal number of light sources), after which objective function values increase with increasing values of n_a .

While the Model 3 solutions were generally obtainable for the 10x20 grid and 15x15 grid with smaller values of n_a (the number of light sources), we could not obtain solutions for Model 2 on the 10x20 and 15x15 grids with a reasonable time. Similarly, as we increase the number of light sources, Model 3 problems become increasingly difficult to solve in these two grids. With $n_a \leq 10$, solution runtimes are exceedingly fast, just seconds or minutes. However, when we set $n_a = 11$ for 10x20 grid and $n_a = 14$ for 15x15 grid, runtime increases to approximately one day, growing in an exponential-like fashion thereafter with increasing values of n_a , with $n_a = 13$ taking approximately 2 weeks to solve the 10x20 grid and with $n_a = 16$ taking 19 days to solve 15x15 grid. Solutions with $n_a \geq 14$ for 10x20 grid and $n_a \geq 16$ for 15x15 grid were not obtainable in reasonable time, even very sub-optimal solutions. In fact, we do not even reach an overall optimal number of light sources, as objective function values are still decreasing with increasing values of n_a . It is also worth to mention here that if we divided the area of a location problem into smaller sized cells, then the ILP problem becomes comparatively

difficult to solve due to the greater number of integer variables associated with the greater number of cells.

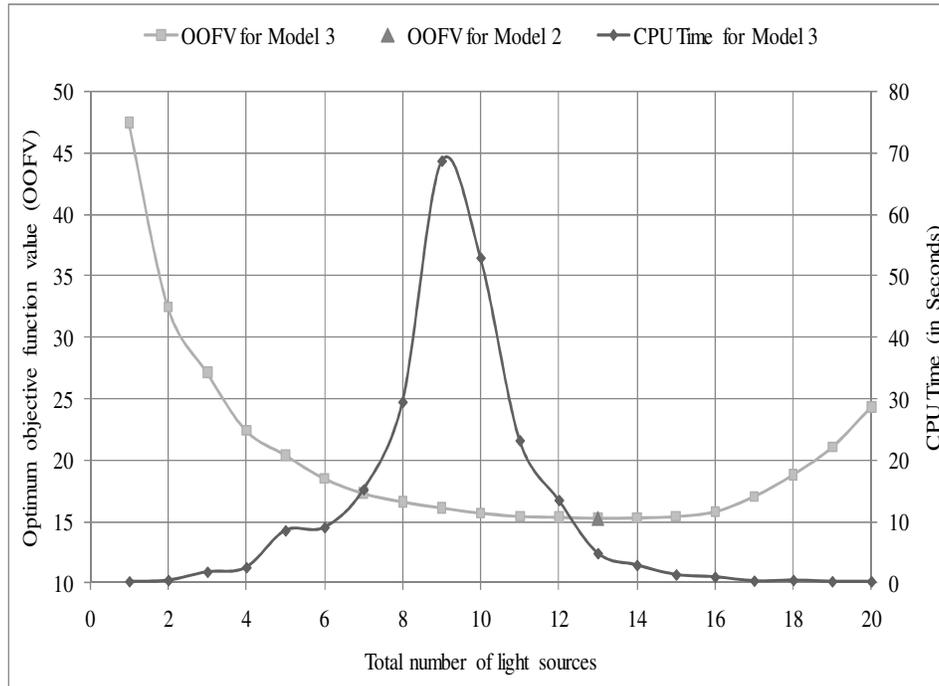


Figure 3.14: Variation of objective function value and CPU time for 10x10 grid.

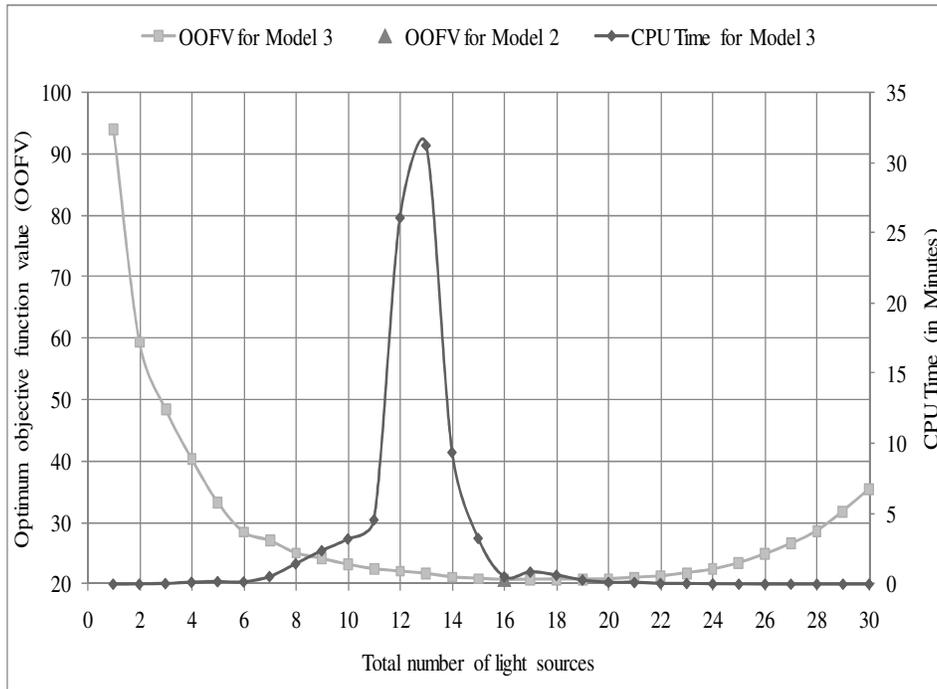


Figure 3.15: Variation of objective function value and CPU time for 10x12 grid.

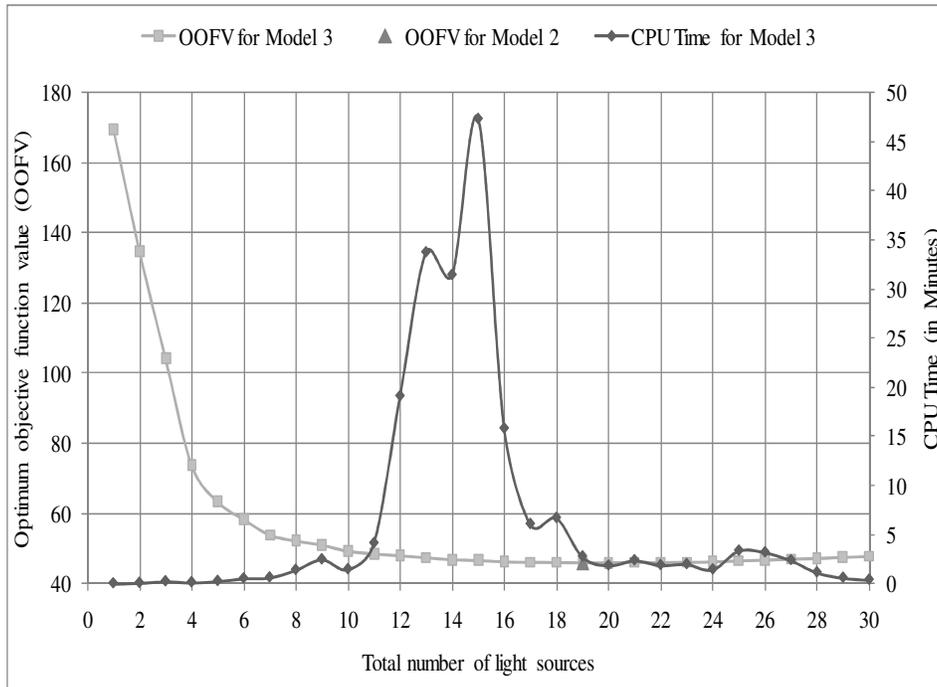


Figure 3.16: Variation of objective function value and CPU time for 12x12 grid.

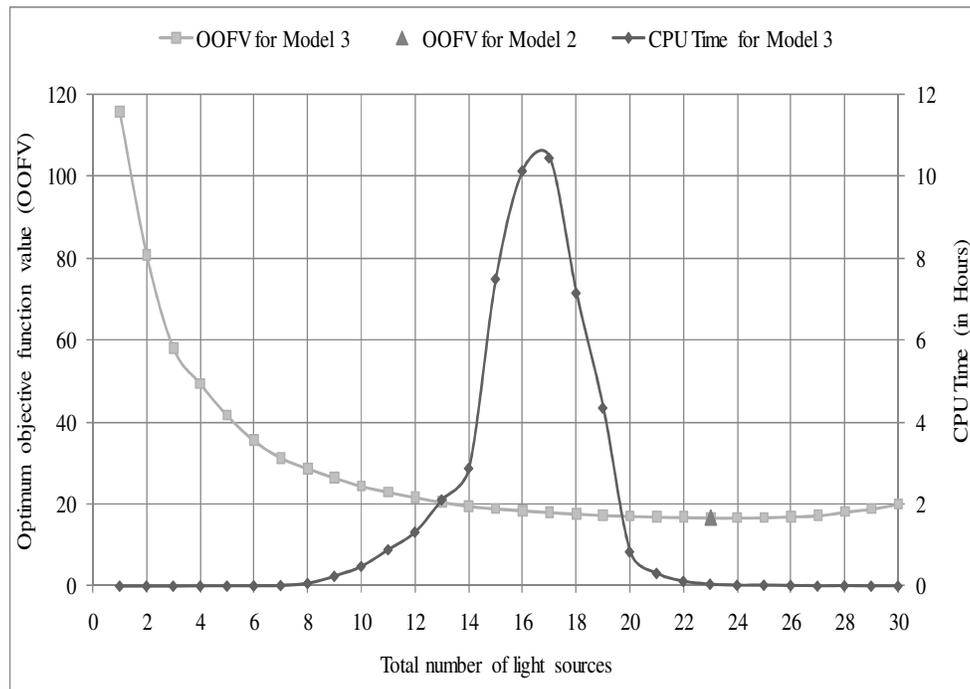


Figure 3.17: Variation of objective function value and CPU time for 10x15 grid.

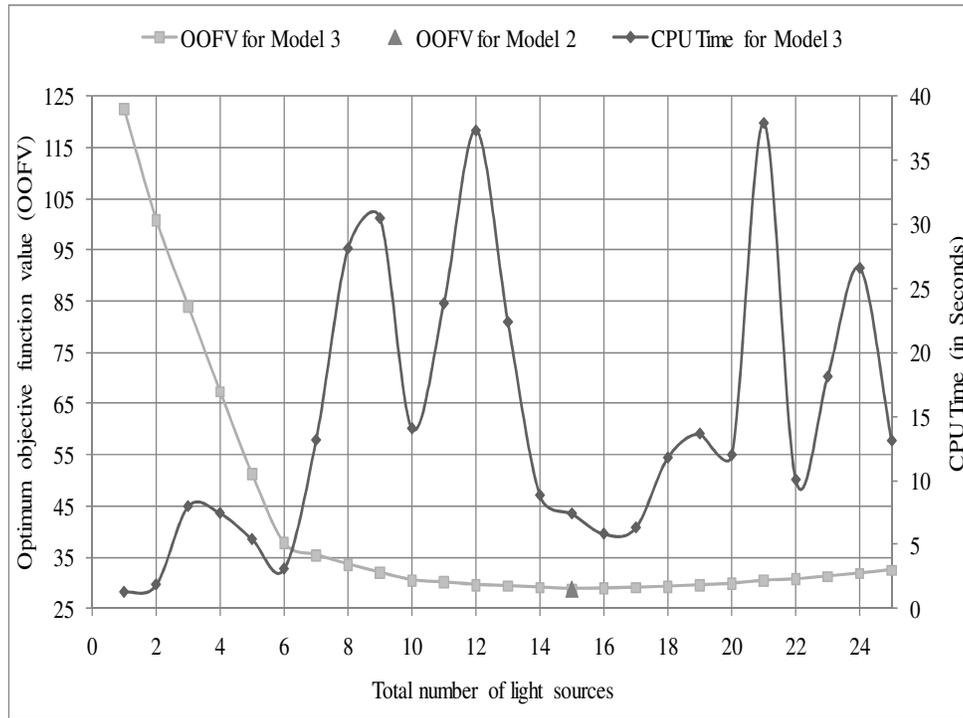


Figure 3.18: Variation of objective function value and CPU time for 10x17 grid.

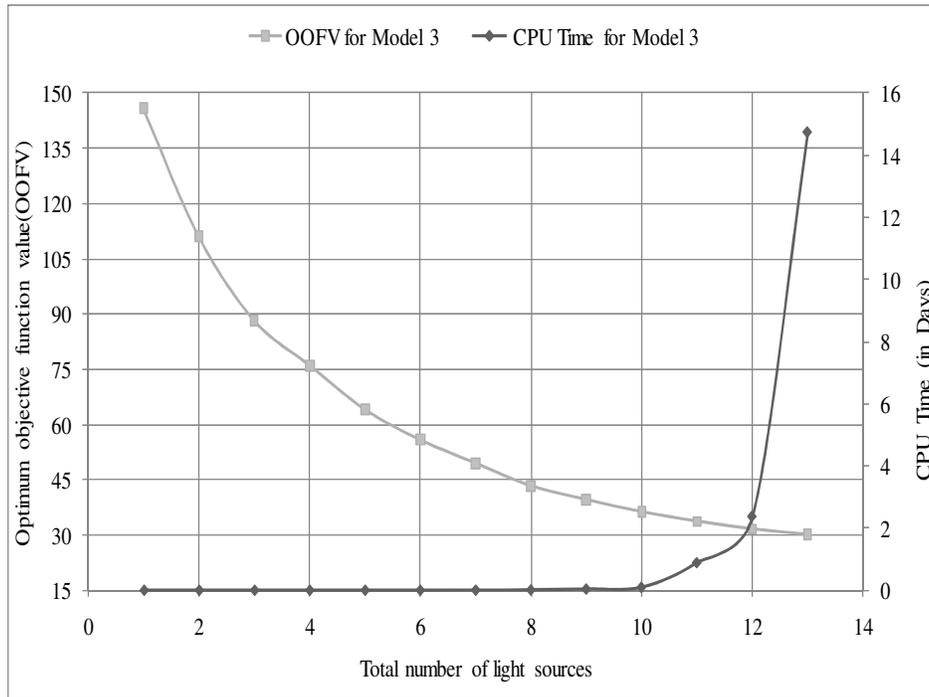


Figure 3.19: Variation of objective function value and CPU time for 10x20 grid.

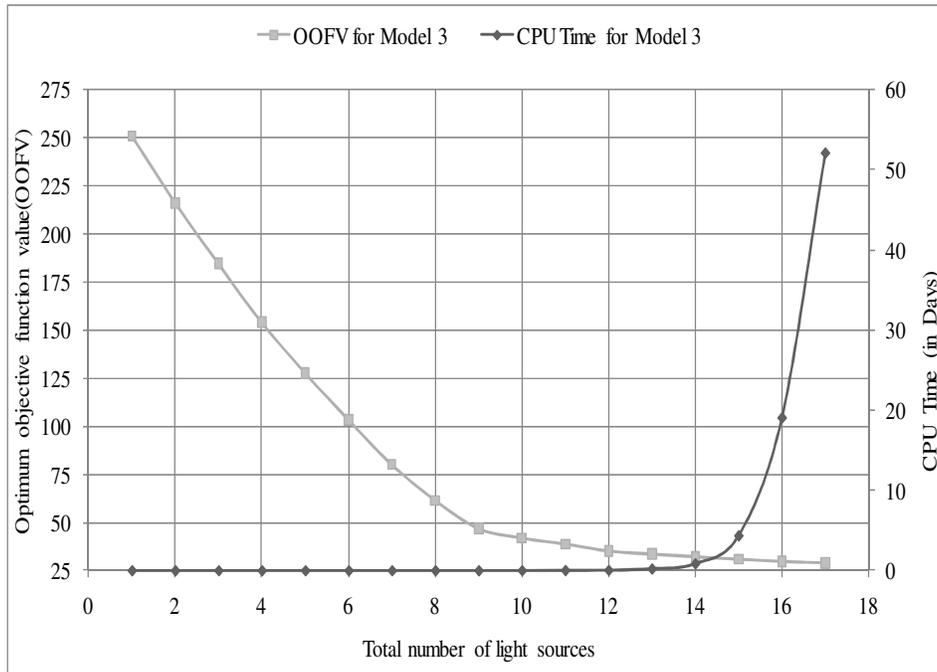


Figure 3.20: Variation of objective function value and CPU time for 15x15 grid.

Figures 3.21 through 3.23 show optimum locations and sizes of light posts for the three test-case grids: 10x10, 10x12 and 12x12. The number in each cell represents optimum size of light posts.

		2	1		2		1		
		1				1			
		1							
		1		1					
		3							
		1			1		1		

Figure 3.21: Optimum locations and sizes of light posts for 10x10 grid.

		4		1		1	2		3		
		1							2		
		1					2				
		2		1							
		2			1				2		
		1					1				

Figure 3.22: Optimum locations and sizes of light posts for 10x12 grid.

		5			1	2			5		
				1			2				
		2		1					2		
		1		1					2		
				1		1	1				
		6			2	1			5		

Figure 3.23: Optimum locations and sizes of light posts for 12x12 grid.

Table 3.1 summarizes the runtime statistics for our various test case grids using Model 3 (showing only the instances with n_a equivalent to the respective optimal number of light sources). As discussed above, CPLEX was able to solve most test cases within a reasonable amount of time (several minutes or less) and with a reasonable number of Simplex iterations and branch-and-bound nodes. In addition, our proposed model solves moderately sized instances with optimality gaps ranging from 0.06% to 0.1%. However, the model becomes computationally difficult for test case instances with $n_a \geq 11$ in the 10x20 grid and $n_a \geq 14$ for the 15x15 grid, due in part to the greater LP gaps. Note that the LP gaps given in the table represent the difference between the optimal (or best found) integer solution and the fully relaxed version of the problem (i.e., the root of the branch-and-bound tree). To solve for these larger instances of those grids, CPLEX had to explore quite a significant number of branch-and-bound nodes, involving a very large number of Simplex iterations, requiring weeks to reach optimality. In general for such instances, we find that the CPLEX solver's branch-and-bound procedure makes considerable progress early on, with rapid improvements in the objective function values of the best-to-date branch-and-bound nodes. In most of the problems tested, optimality gaps are reduced to 0.1% or less in just a few seconds or minutes. However, reductions in optimality gap are slow for large problems, and even after many days of runtime, higher optimality gaps remain or the solver runs out of memory. In future extensions of this work, our target is to

develop techniques to solve even these large test case instances efficiently. For now, however, we were able to obtain optimal solutions for $n_a = 13$ in the 10x20 grid and $n_a = 17$ in the 15x15 grid, and so we provide the data corresponding to those instances of the problem in the table.

Table 3.1: Runtime statistics for selected Model 3 test cases.

Grid size	OOFV	# of lights	Simplex Iterations	Branch and Bound Nodes	CPU Time (seconds)	# of integer variables	# of constraints	MIP gap	LP gap
10x10	15.28	13	87171	5941	2.71875	432	1113	0.000616	0.0903
10x12	20.63	16	1792730	92705	47.3438	576	1417	0.000989	0.0785
10x15	16.64	23	4182388	216631	166.594	792	1873	0.000996	0.1532
10x17	28.81	15	93506	4965	4.0625	936	2177	0.000810	0.0538
10x20	30.13	13*	2147483648	72440018	1033440	1152	2633	0.000999	0.4191
12x12	45.99	19	3893188	185741	107.922	768	1809	0.000999	0.0239
15x15	28.94	17*	2147483648	1369697304	4491820	1452	3183	0.001	0.2698

**We were unable to solve for larger numbers of lights in these grids.*

3.6. Conclusion

This research proposes three GBLP ILP models to optimally place light posts in a park. Our ILP models are designed to optimize the number of light posts, their locations, and their sizes. While this particular problem represents just one specific GBLP, in reality, many problems can be modeled as GBLPs, and thus, can be solved using these methods. For example, we could use these methods to determine where to optimally place retail outlets and/or warehouses; a retail outlet

will supply an area and meet the surrounding cell's demand. Retail demands can be established through market research or surveys in the surrounding neighborhoods. Other topics where our models can potentially be applied include health/biological sciences (e.g., optimal application of radiation), communications (e.g., transmitter locations), real estate, and emergency service dispatching, physics, and resource exploration/exploitation.

Our results demonstrate that our ILP models can be used to solve GBLPs, and that they are scalable at least up to intermediate sized problems. However, for larger problem, it takes days and even weeks to solve to optimality. In the future, we plan to extend this work to develop advanced optimization techniques to solve large-scale problems using relaxation-based decomposition along with the addition of logical restrictions. Furthermore, we have also developed an extension to the models where we include a fixed-charge component to the objective function (and the associated constraint equations) to include light source installation costs and other such features to our models.

References

- Abdinnour-Helm, S., Venkataramanan, M. A. (1998), "Solution approaches to hub location problems", *Annals of Operations Research*, 78, 31-50.
- Altinel, I. K., Durmaz, E., Aras, N., Özkisacik, K. C. (2009), "A location-allocation heuristic for the capacitated multi-facility Weber problem with

- probabilistic customer locations”, *European Journal of Operational Research*, 198, 790-799.
- Aytug, H., Saydam, C. (2002), “Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study”, *European Journal of Operational Research*, 141, 480–494.
- Bangerth, W., Klie, H., Wheeler, M. F., Stoffa, P.L., Sen, M.K. (2006), “On optimization algorithms for the reservoir oil well placement problem”, *Computational Geosciences*, 10, 303-319.
- Berman, O., Drezner, Z., Krass, D. (2010), “Generalized coverage: New developments in covering location models”, *Computers & Operations Research*, 37, 1675–1687.
- Canbolat, M. S., Wesolowsky, G. O. (2010), “The rectilinear distance Weber problem in the presence of a probabilistic line barrier”, *European Journal of Operational Research*, 202, 114-121.
- Chen, D.-S., Baton, R. G., Dang, Y. (2010), *Applied Integer Programming*, A John Willey & Sons, Inc., Publication, (Chapter 2).
- Chen, G., Ozturk, O., Kandemir, M. (2005), “An ILP-Based Approach to Locality Optimization”, *Lecture Notes in Computer Science*, 3602, 149-163.
- Church, R. L. and ReVelle, C. S. (1974), “The maximal covering location problem”, *Papers of the Regional Science Association*, 32, 101–118.
- Cooper, L. (1963), “Location-Allocation Problems”, *Operations Research*, 11, 331-343.
- Daskin, M. S. (1995), *Network and Discrete Location: Models, Algorithms, and Applications*, John Wiley, New York.
- Domschke, W., Krispin, G. (1997), “Location and layout planning-a survey”, *OR Spektrum*, 19, 181-194.
- Drezner, Z., Wesolowsky, G.O. (1997), “On the best location of signal detectors”, *IIE Transactions*, 29, 1007-1015.

- Fourer, R., Gay, D. M. and Kernighan, B. W. (2002), *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press, Cole Publishing Co.
- Francis, R. L. (1964), “On the Location of Multiple New Facilities with Respect to Existing Facilities”, *Journal of Industrial Engineering*, 15, 106-107.
- Gendron, B., Potvin, J.-Y., Soriano, P. (2003), “A Tabu Search with slope scaling for the multi commodity capacitated location problem with balancing requirements”, *Annals of Operations Research*, 122, 193-217.
- Ghiani, G., Grandinetti, L., Guerriero, F., Musmanno, R. (2002), “A Lagrangean Heuristic for the Plant Location Problem with Multiple Facilities in the Same Site”, *Optimization Methods and Software*, 17, 1059–1076.
- ILOG (2007), “ILOG CPLEX 11.0 User’s Manual”, ILOG, Inc.
- Ingolfsson, A., Budge, S., Erkut, E. (2008), “Optimal ambulance location with random delays and travel times”, *Health Care Management Science*, 11, 262-274.
- Katz, I. N., Cooper, L. (1974), “An always convergent numerical scheme for a random locational equilibrium problem”, *SIAM Journal of Numerical Analysis*, 11, 683-692.
- Kubis, A., Hartmann, M. (2007), “Analysis of Location of Large-area Shopping Centers—A Probabilistic Gravity Model for the Halle-Leipzig Area”, *Jahrbuch für Regionalwissenschaft*, 27, 43-57.
- Manzour-al-Ajdad, S. M. H., Torabi, S. A., Eshghi, K. (2012), “Single-Source Capacitated Multi-Facility Weber Problem - An iterative two phase heuristic algorithm”, *Computers & Operations Research*, 39, 1465-1476.
- Marín, A. (2011), “The discrete facility location problem with balanced allocation of customers”, *European Journal of Operational Research*, 210, 27-38.

- NASA (2006), “More on Brightness as a Function of Distance”, online, viewed 5 February 2010, <http://imagine.gsfc.nasa.gov/YBA/M31-velocity/1overR2-more.html>.
- Sherali, H. D., Nordai, F.L. (1988), “NP-Hard, Capacitated Balanced p-Median Problems on a Chain Graph with a Continuum of Link Demands”, *Mathematics of Operations Research*, 13, 32-49.
- Simons, R. H., Bean, A. R. (2001), *Lighting Engineering: Applied Calculations*, Architectural Press.
- Taniguchi, E., Noritake, M., Yamada, T., Izumitani, T. (1999), “Optimal size and location planning of public logistics terminals”, *Transportation Research Part E: Logistics and Transportation Review*, 35, 207-222.
- Teixeira, J. C., Antunes, A.P. (2008), “A hierarchical location model for public facility planning”, *European Journal of Operational Research*, 185, 92-104.
- Wesolowsky, G. O. (1972), “Rectangular Distance Location under the Minimax Optimality Criterion”, *Transportation Science*, 6, 103-113.
- Winston, W. L. Venkataramanan, M. (2003), *Introduction to Mathematical Programming*, (4th ed.), Thomson (Chapter 9).
- Wolsey, L. A. (1998), *Integer Programming*, Wiley-Interscience, Series in Discrete Mathematics and Optimization, Toronto, (Chapter 1).
- Zhang, J. (2006), “Approximating the two-level facility location problem via a quasi-greedy approach”, *Mathematical Programming, Series A*, 108, 159-176.

Chapter 4²

Solving Large Scale GBLPs

4.1. Introduction

Many problems in business, engineering, defence, resource exploitation, and even the medical sciences with location aspects can be expressed as *grid-based location problems* (GBLPs) (Noor-E-Alam et al., 2012). In such problems, a region is divided up into a number of small cells (e.g., facility locations) where a heterogeneous weight distribution (e.g., demands) is associated with the cells and

²A version of this chapter has been published: Noor-E-Alam, M., Doucette, J. (2012), “Relax-and-Fix-Based Decomposition Technique for Solving Large Scale GBLPs”, *Computers and Industrial Engineering*, 63, 1062-1073.

we must make decisions related to each cell (e.g., how much supply to provide to the various cells). Based on these demand distributions and supply relationships, we can then determine the optimum number, location(s), and size(s) of facilities while fulfilling certain objectives. To model a real-world GBLP, we generally need to consider a large number of discrete variables, a heterogeneous demand distribution and non-linear supply distributions. Furthermore, to get an optimum decision, the mathematical models need to be designed in such a way that they will simultaneously determine the locations, sizes and total number of facilities to achieve certain objectives. Collectively, these considerations contribute to producing large-scale and computationally difficult problems, which are generally not scalable and often become intractable even with small problems. As such, our goal in the present research is to develop effective and efficient methodologies for solving large GBLP instances.

To solve real-world GBLPs efficiently, mathematical models are generally designed as *integer linear programming* (ILP) problems. While there is no known polynomial-time algorithm for solving general ILPs (Wolsey, 1998), they are often easily solved using a variety of techniques in practice. *LP relaxation*, where the integer decisions variables are permitted to take non-integer values, results in a lower bound on the optimal solution to the ILP for minimization problems (and an upper bound for maximization problems). In general, an optimal solution to the

LP relaxation version of an ILP is often quite a weak bound on the optimal solution to the ILP. The most frequently used technique to solve ILPs is the *branch-and-bound* algorithm, which makes use of LP relaxation. In the branch-and-bound algorithm, the lower bounds obtained from LP relaxation and upper bounds obtained from increasingly more optimal feasible ILP solutions are used to fathom the branch-and-bound search tree (Chen, 2010; Wolsey, 1998). Widely used ILP solvers such as CPLEX, MINTO, etc., implement versions of the branch-and-bound algorithm. However, large scale ILPs are often very hard to solve with the current branch-and-bound methods due to a combinatorial explosion in the number of branch-and-bound nodes; many real-world ILP problems require weeks or months of solution time to solve on the most powerful systems.

A number of large-scale ILP problem solution techniques have been developed and evaluated in recent decades. Barnhart et al. (1998) discussed solution of large ILP problems with the *Dantzig-Wolfe* decomposition technique. In this work, *Column generation* has been employed for implicit pricing of non-basic variables. Barnhart et al. (2000) developed a *branch-and-price-and-cut* algorithm that permits column generation and a *cutting plane* algorithm to be applied throughout the branch-and-bound search tree to reduce the computational time for ILPs. In the *branch-and-cut-and-price* (BCP) algorithm (Belov and Scheithauer, 2006),

cutting plane and column generation algorithms are integrated with the branch-and-bound algorithm to improve the relaxation of the problem and achieve price out efficiency. Klose and Görtz (2007) used a column generation procedure within a *branch-and-price* algorithm for computing optimal solutions to the capacitated facility location problem. In their proposed method, demand constraints are relaxed with *Lagrangean relaxation* and a stabilized column generation is used for solving the corresponding master problem to optimality. Lorena and Senne (2004) and Senne et al. (2005) proposed a branch-and-price algorithm geared with column generation to solve the *capacitated p-median* problem. While the above-mentioned methods often represent significant improvements in solution runtimes, they are often suitable only for specific types of problems and implementation can be quite cumbersome.

Relaxation-based decomposition methods are often used to decompose the original problem into easier sub-problems, where complicating constraints or integrality restrictions are relaxed to obtain an easier problem. This relaxed problem is solved and its partial solutions are fixed into the original problem to generate another easier sub-problem, which is then solved to obtain a near-optimal solution for the original ILP (Wolsey, 1998). Such relax-and-fix strategies have been discussed in many works in the literature to solve scheduling and lot-sizing decision problems (Beraldi et al., 2006; Ferreira and Morabito, 2010; Kelly

and Mann, 2004; Mohammadi et al., 2010). Mauri et al. (2010) proposed a binary integer programming model for point-feature cartographic label placement. A non-trivial valid inequality is presented to strengthen their proposed method. They also proposed a *Lagrangian decomposition* technique to solve this problem within a reasonable time. Ghiani et al. (2002) developed a Lagrangian heuristic to solve the capacitated facility location problem with multiple facilities; their computational results indicate that a Lagrangian heuristic is able to find good lower and upper bounds in a reasonable amount of time. Rajagopalan et al. (2004) used Lagrangian relaxation to fix some variables in the original problem that can be solved easily with the CPLEX solver. In Aytug and Saydam (2002), a *genetic algorithm* (GA) is used to solve large-scale maximum expected covering location problems. They find that GA outperforms other heuristic techniques for this location problem and a near-optimal solution is obtained within a reasonable amount of time. GA is also found to be very effective for solving combinatorial optimization problems (Anderson and Ferris, 1994). GA and Lagrangian relaxation have also been combined together for solving combinatorial optimization problems such as unit commitment problems, where GA is used to update the Lagrangian multipliers and improve the performance of the Lagrangian relaxation method (Yamin and Shahidehpour, 2004; Cheng et al., 2000).

A sophisticated solver like CPLEX (ILOG, 2007) performs better automatic tightening on some problems by including only the relevant constraints with respect to linear relaxation. In this case standard tightening methods often require longer solution times than manual problem-specific approaches. However, introducing creative constraints or logical restrictions, we can reduce the solution time significantly. To help solver for better automatic reformulation, manual reformulation is often needed to redefine variables (Trick, 2005). On the other hand if a user implements his/her own problem-specific algorithm, then the solver will recognize the various relaxations, and the inequalities. For efficient relaxation of many combinatorial optimization problems, some branch-and-bound packages contain algorithms for generating certain classes of inequalities for simple structures, such as the knapsack, single-node flow and the path polytopes (Roy and Wolsey, 1987; Savelsbergh et al., 1995). In work by Aardal (1998), an alternative way of modeling the capacitated facility location problem is discussed. This model involves addition of new decision variables and redundant constraints such that the relaxations can be better utilized by the software; the solver is able to generate stronger linear inequalities and takes less time to reach optimality. Williams (1978) also described the formulation of ILP models with extra logical restrictions that reduced the computational time significantly. To generate strong inequalities, the polyhedral structure of *capacitated facility location* (CFL) has been studied by Aardal et al. (1995).

Much less effort has been made to study the scope of relax-and-fix strategies to solve large scale GBLP-type problems. Therefore, in this research our objective is to develop a relaxation-based decomposition technique to solve large scale GBLPs. Computational complexity of the GBLP model is evaluated with several large scale test-case grids and the structure of the mathematical model is investigated to identify the cause of the exponential behavior of the CPU time. Based on findings from this investigation, we propose a relaxation-based decomposition technique to efficiently solve large instances of the problem. To reduce the solution time further, we also propose additional problem-specific logical restrictions. Finally, the ILP model and the decomposition technique are implemented within a standard modeling language and tested on a number of large test-case grids to compare the performance of the proposed technique with the benchmark.

4.2. Optimization Model Description

In the present work, we use the light post location problem described in Noor-E-Alam et al. (2012) as a sample GBLP to test and evaluate the effectiveness of our proposed decomposition algorithm as a means to more efficiently solve large scale GBLPs. Detailed description and analysis of this problem is available in that prior work, however, a brief discussion is nonetheless provided herein. This GBLP involves making optimum decisions on where to install lights in a city

park. Factors affecting the number of light sources, their sizes, and their locations are many and varied. In a city park, there are different areas used for various purposes, requiring differing amounts of light, and trees and topography can also vary quite widely throughout the park, further affecting the degree of lighting an area may require. Moreover, in boundary regions there may be various physical restrictions such as roadways and underground power cables for utility service. Therefore, installing lights in boundary regions might not even be physically feasible. This city park can be modeled as a GBLP, where a heterogeneous demand distribution (i.e., the amount of light required at various locations) can be represented by cells. The idea is that the light sources should be placed in such a way that the areas they illuminate they don't excessively overlap, but not so far apart that there are unlit or under-lit areas. The intent is to find the optimal placement of light posts that would best satisfy the heterogeneous demand distribution while also minimizing the number and sizes of the light sources.

In the light distribution model (Noor-E-Alam et al., 2012) the amount of light delivered to neighbouring cells depends on the rectilinear distances from the point source to the respective cells, as shown in Figure 4.1. These relationships are determined according to the brightness calculation process, except that distances are calculated as the 3-dimensional Euclidean distances between the light source and the centers of the various cells. The amount of light delivered to various cells

can be calculated by the set of equations shown in this figure, where S_a, S_b, S_c, S_d, S_e and S_f represent the total amount of available supply at locations a, b or b', c, d or d', e or e' and f , respectively. Here, the vertical distance between the point source, o , and centre point of the grid below it, a , is r . P is the luminosity of the point source. The angle between vertical line oa and line ob or ob' is α_b , the angle between vertical line oa and line oc is α_c , the angle between vertical line oa and line od or od' is α_d , the angle between vertical line oa and line oe or oe' is α_e , and the angle between vertical line oa and line of is α_f .

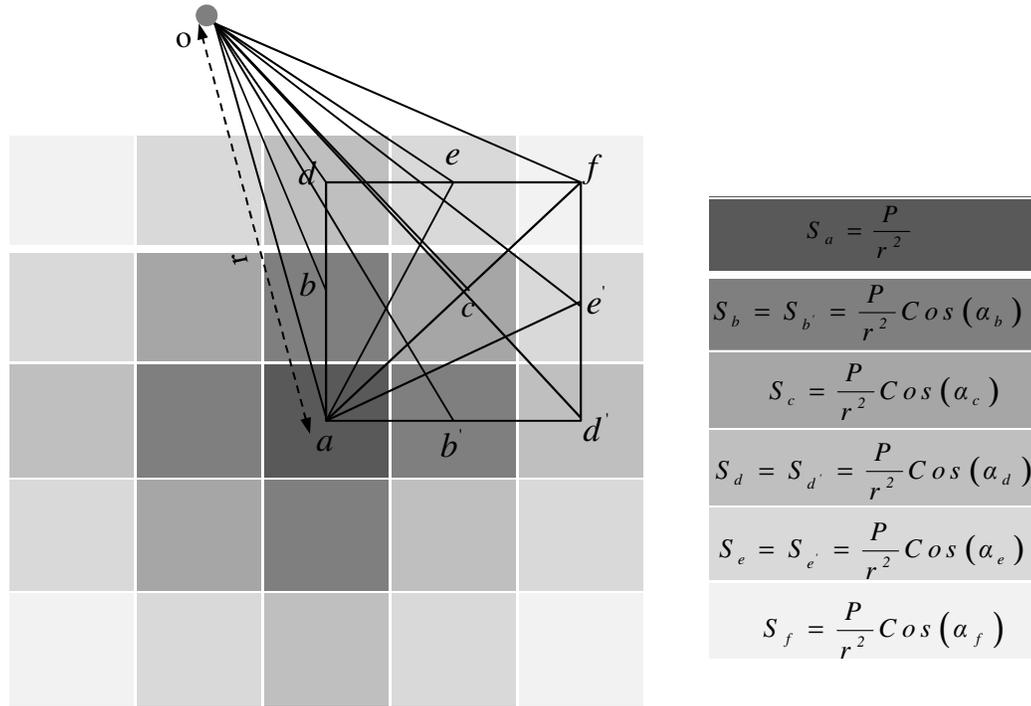


Figure 4.1: Light distribution model.

To solve this above decision problem effectively, it is required to develop ILP model. In order to formulate the ILP model of the light post problem, we first need to define the notation we will use, as follows:

Input Parameters:

β is the system boundary constant

D_{ij} is the demand at location (i,j)

n is the total number of light source

UB_{xy} is the upper bound on decision variable P_{xy}

Q is the set of all x-coordinates in the grid, indexed by i

R is the set of all y -coordinates in the grid, indexed by j

X is the set of all x -coordinates of the light source, indexed by x ,

$$i_{min} + \beta \leq x \leq i_{max} - \beta$$

Y is the set of all y -coordinates of the light source, indexed by y ,

$$j_{min} + \beta \leq y \leq j_{max} - \beta$$

Decision Variables:

S_{ij} is the total supply at location (i,j)

P_{xy} is the integer size of the light source at location (x,y)

T_{xy} is the binary decision variables

UD_{ij} is the unmet demand at location (i,j)

ES_{ij} is the excess supply at location (i,j)

n_{opt} is the optimum number of light sources

Finally, the discussed GBLP ILP model can be formulated as follows:

$$\text{Minimize } \sum_{i \in Q} \sum_{j \in R} (UD_{ij} + ES_{ij}) \quad (4.1)$$

Subject to:

$$UD_{ij} \geq D_{ij} - S_{ij} \quad \forall i, j \quad (4.2)$$

$$ES_{ij} \geq S_{ij} - D_{ij} \quad \forall i, j \quad (4.3)$$

$$S_{ij} = \sum_{x \in X} \sum_{y \in Y} \frac{P_{xy}}{r^2} \cos \left(\tan^{-1} \left(\frac{\sqrt{(i-x)^2 + (j-y)^2}}{r} \right) \right), \quad \forall i, j \quad (4.4)$$

$$T_{xy} = 1 \quad \text{if } P_{xy} > 0 \quad \forall x, y \quad (4.5)$$

$$T_{xy} = 0 \quad \text{if } P_{xy} = 0 \quad \forall x, y \quad (4.6)$$

$$\sum_{x \in X} \sum_{y \in Y} T_{xy} = n \quad (4.7)$$

$$T_{xy} \in \{0, 1\}, \quad \forall x, y \quad (4.8)$$

$$UD_{ij} \geq 0, \quad \forall i, j \quad (4.9)$$

$$ES_{ij} \geq 0, \quad \forall i, j \quad (4.10)$$

$$0 \leq P_{xy} \leq UB_{xy} \quad \forall x, y \quad (4.11)$$

The objective function in equation (4.1) seeks to minimize the sum of the total unmet demand and excess supply, while equations (4.2) and (4.3) calculate unmet demand and excess supply, respectively, for each cell. Based on our illustration of light distribution in Figure 4.1, equation (4.4) will calculate the total supply available in each cell (i, j) . Suppose a light post is located at point a , and we want to calculate the supply at point e . In this case we need to find the value of angle α_e , to calculate the supply at point e . This angle can be calculated by the ratio of ae and r . The value of ae is the distance between the point a and the point e . On the other hand, the total supply in a particular cell (i, j) is the summation of all the individual supplies coming from all light sources.

To control the number of light sources, we incorporate equations (4.5) through (4.7). By controlling the number of light sources, we can optimize our problem for specified values of n , say, to observe how the solution changes with an increasing number of light sources. Equation (4.8) is used to define variable T_{xy} properly. Equations (4.9) and (4.10) are used as non negativity constraints for the variables UD_{ij} and ES_{ij} . Furthermore, bounding constraint (4.11) places an upper bound on decision variable P_{xy} . Equations (4.5) and (4.6) are not in linear form. To develop complete ILP model, we replace them with their equivalent set linear equations using the same procedure as in Noor-E-Alam et al. (2012).

4.3. Runtime Complexity

To demonstrate the runtime complexity of the above GBLP model, we have used two large-scale instances: 10x20 and 15x15 test-case grids with demand distributions shown in Figure 4.2 and Figure 4.3 respectively, taken from Noor-E-Alam et al. (2012). We solve our instances of the ILP problem on an 8 processor ACPI multiprocessor X64-based PC with Intel Xeon® CPU X5460 running at 3.16GHz with 32 GB memory. We have implemented our model in AMPL (Fourer et. al., 2002), and solved using the CPLEX 11.2 solver (ILOG, 2007). In these experiments, other parameters are assumed as follows: $UB = 10$, and $\beta=2$, $r = 2$. We used a CPLEX *mipgap* setting of 0.001, which means all test cases solved to full termination are provably within 0.1% of optimality.

0.35	0.61	0.71	0.82	0.76	0.81	0.93	0.93	0.95	0.88	0.90	0.97	1.02	0.89	0.73	0.58	0.45	0.31	0.16	0.04
0.44	0.69	0.94	0.98	1.03	1.12	1.22	1.30	1.23	1.22	1.30	1.49	1.72	1.39	1.09	0.86	0.67	0.48	0.25	0.06
0.50	0.76	0.99	1.13	1.27	1.42	1.55	1.54	1.47	1.46	1.58	1.96	1.89	1.86	1.37	1.10	0.90	0.68	0.37	0.09
0.56	0.86	1.13	1.30	1.48	1.74	2.03	1.83	1.65	1.56	1.60	1.78	2.01	1.69	1.42	1.26	1.15	0.99	0.54	0.14
0.61	0.98	1.36	1.45	1.62	2.03	2.05	2.10	1.73	1.55	1.50	1.53	1.57	1.47	1.37	1.37	1.46	1.57	0.81	0.20
0.62	1.08	1.28	1.51	1.53	1.75	2.05	1.86	1.60	1.42	1.32	1.28	1.26	1.24	1.24	1.37	1.78	1.48	1.14	0.29
0.50	0.81	1.15	1.17	1.25	1.40	1.60	1.67	1.41	1.20	1.08	1.01	0.98	0.97	1.00	1.09	1.27	1.44	0.75	0.19
0.34	0.53	0.70	0.78	0.87	1.02	1.26	1.22	1.18	0.91	0.77	0.70	0.67	0.67	0.68	0.73	0.77	0.74	0.42	0.10
0.17	0.26	0.34	0.39	0.45	0.53	0.64	0.76	0.61	0.48	0.40	0.36	0.34	0.34	0.35	0.36	0.36	0.32	0.19	0.05
0.04	0.06	0.08	0.10	0.11	0.13	0.16	0.19	0.15	0.12	0.10	0.09	0.09	0.08	0.09	0.09	0.09	0.08	0.05	0.00

Figure 4.2: Demand distribution for 10x20 grid.

0.51	0.51	0.54	0.62	0.74	0.88	0.78	0.70	0.65	0.64	0.66	0.69	0.69	0.58	0.43
0.72	0.72	0.77	0.88	1.10	1.12	1.16	1.00	0.94	0.94	0.98	1.07	1.17	0.91	0.64
0.88	0.89	0.93	1.02	1.17	1.33	1.25	1.19	1.17	1.18	1.25	1.45	1.35	1.24	0.83
1.01	1.01	1.05	1.12	1.21	1.31	1.33	1.35	1.38	1.37	1.40	1.46	1.53	1.23	0.90
1.09	1.10	1.13	1.19	1.27	1.35	1.42	1.51	1.61	1.54	1.49	1.48	1.44	1.26	0.96
1.15	1.16	1.20	1.25	1.31	1.39	1.49	1.65	1.66	1.68	1.57	1.51	1.48	1.40	1.05
1.19	1.21	1.24	1.29	1.35	1.42	1.50	1.61	1.71	1.63	1.57	1.54	1.57	1.38	1.16
1.22	1.24	1.28	1.33	1.38	1.44	1.50	1.56	1.60	1.57	1.53	1.50	1.46	1.40	1.07
1.24	1.27	1.31	1.35	1.40	1.45	1.49	1.53	1.55	1.53	1.50	1.45	1.39	1.27	1.00
1.25	1.29	1.33	1.38	1.42	1.46	1.50	1.53	1.54	1.51	1.47	1.42	1.37	1.28	0.99
1.26	1.31	1.36	1.40	1.45	1.48	1.52	1.55	1.55	1.51	1.46	1.40	1.38	1.21	1.01
1.26	1.32	1.38	1.44	1.47	1.50	1.54	1.59	1.61	1.53	1.43	1.34	1.26	1.17	0.90
1.25	1.32	1.41	1.50	1.51	1.52	1.56	1.65	1.76	1.58	1.41	1.27	1.15	1.02	0.82
1.21	1.31	1.46	1.62	1.54	1.50	1.54	1.71	1.65	1.62	1.35	1.18	1.06	0.94	0.79
1.14	1.25	1.48	1.50	1.55	1.40	1.38	1.44	1.53	1.35	1.18	1.05	0.97	0.91	0.84

Figure 4.3: Demand distribution for 15x15 grid.

Figure 4.4 and Figure 4.5 show the respective solution data for the above ILP on these two test-case grids. In those figures, each square data point represents the *optimal objective function value* (OOFV) of the specified test case with the indicated number of light sources (i.e., input parameter, n), while the diamond data points represent the CPU time required for those test cases. Note that in both of these figures, the OOFV data points are to be read against the left-hand y-axes, while CPU time data points are to be read against the right-hand y-axes. We see from these figures that solution time increases exponentially with problem size (i.e., number of light sources). For example, to solve the 10x20 test-grid for 13

light sources, it takes 14.7 days, and to solve the 15x15 test-grid for 17 light sources, it takes 52 days. We were not able to obtain optimal solutions for larger test case instances, as we exceeded memory limits of our system before reaching optimality in these instances.

In general for large problems, we find that our CPLEX solver's branch-and-bound procedure makes considerable progress early on, with rapid improvements in the objective function values of the best-to-date branch-and-bound nodes. However, reductions in optimality gap slow quickly, and even after many days of runtime, large optimality gaps remain or solver runs out of memory (e.g., after several days of runtime, the 10x20 grid with 14 light sources ran out of memory). While the 15x15 and 10x20 grids are the largest test cases from Noor-E-Alam et al. (2012), these may not be particularly large in terms of real-world problems; this problem is intractable for such cases. This drives our efforts to develop our decomposition algorithm to efficiently solve large scale instances of this problem.

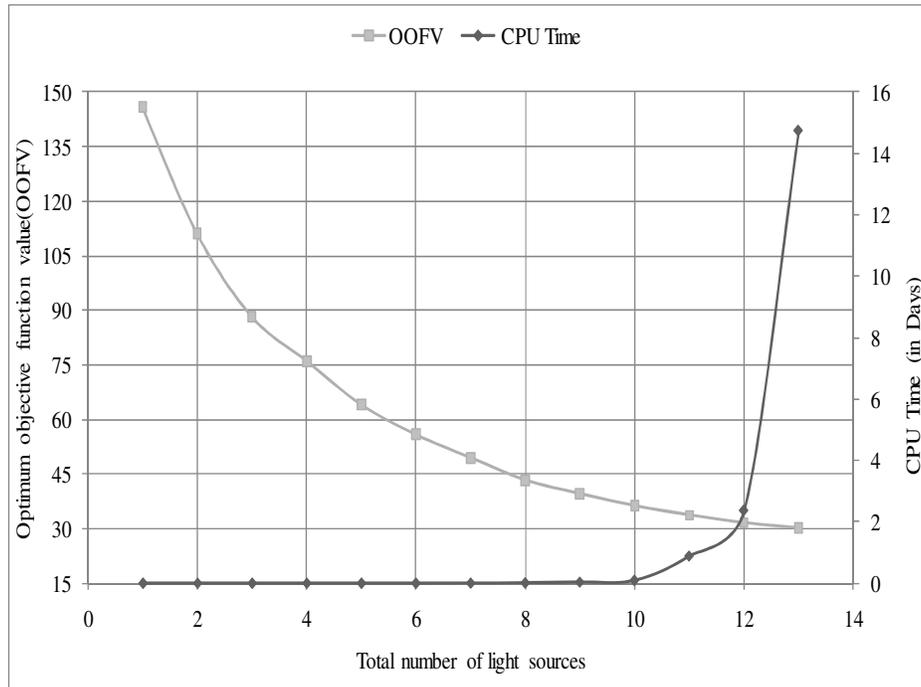


Figure 4.4: Objective function value and solution runtime for the GBLP on the 10x20 grid.

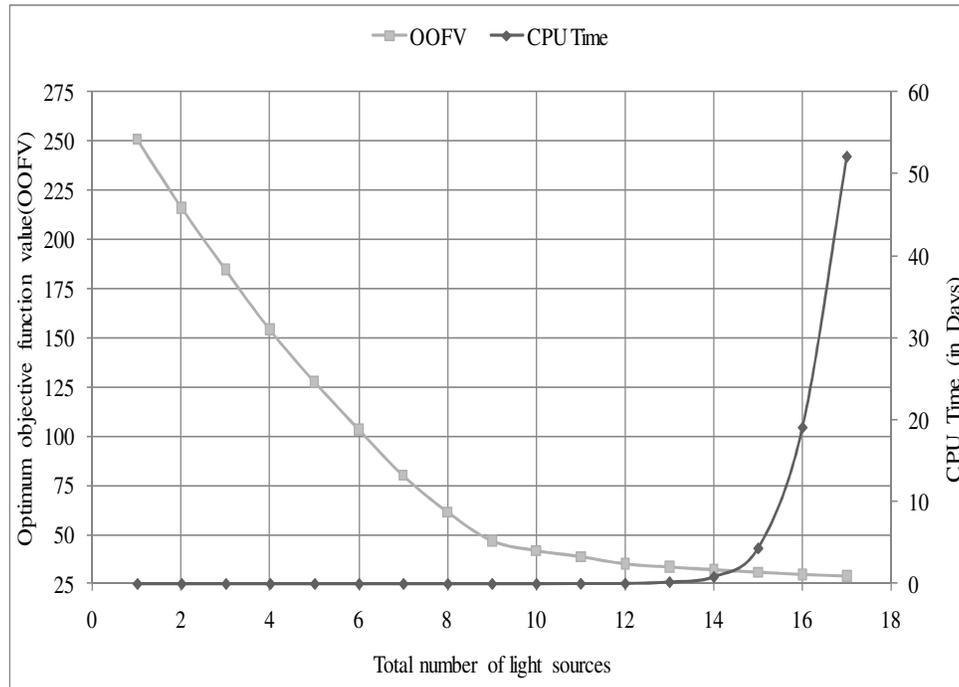


Figure 4.5: Objective function value and solution runtime for the GBLP on the 15x15 grid.

4.4. Relaxation-Based Decomposition Techniques

As discussed above, relaxation-based decomposition techniques have been widely used to solve large scale ILP instances. In many such computationally complex problems, that computational complexity arises from just a small subset of constraints or integrality requirements. In this case, relaxation-based decomposition techniques have proven to be very useful. By relaxing these constraints with Lagrangian relaxation, we can decompose the original problem into two easy sub-problems, which can be solved iteratively to obtain a near-optimal solution. However, this technique often requires an ILP model to have a

specific structure that can be taken advantage of. On the other hand, by relaxing the integrality requirement of selected decision variables, we can similarly decompose the original problem into two easy sub-problems, which can also be used to obtain a near-optimal solution (Wolsey, 1998). We first solve the semi-relaxed problem, where some integrality requirements are removed (i.e., those associated decision variables are permitted to take on real values). The solution to that problem will provide us with values for the remaining integer decision variables, which we can then fix in the second sub-problem. That subsequent problem is solved to provide a near-optimal solution to the original. The effectiveness of this decomposition technique depends on the careful selection of the set of integer variables to relax in the first sub-problem.

4.4.1. Relax-and-Fix-Based Decomposition (RFBD)

In our GBLP model, we can observe that our decision variables are in three classes, continuous decision variables (S_{ij} , UD_{ij} , and ES_{ij}), discrete integer decision variables (P_{xy}), and binary decisions variables (T_{xy}). By relaxing the integrality requirement of P_{xy} variables, we can obtain a sub-problem that will be easier to solve (there are fewer integer decision variables), but which will still permit precise identification of light source location(s). These precise locations (i.e., the binary T_{xy} decision variable values) from that sub-problem can then be fixed and the original problem solved with these fixed values. We call this

decomposition algorithm a *relax-and-fix-based decomposition* (RFBD) approach. This technique does not guarantee optimality; however, we will show that for our problems, it is able to provide high quality solutions within a reasonable amount of time.

Some might question whether the P_{xy} variables are strictly required to be integer in a real-world implementation of this problem; we acknowledge that some arguments can be made that the P_{xy} variables could be relaxed altogether and that there is no need for a decomposition approach such as the one described. However, we assert that since the P_{xy} variables represent sizes of light sources in our specific problem, these variables should not only remain integer, but would actually be an enumerated type integer since the permitted light source sizes would be limited.

4.4.2. Logical Restrictions

For some hard ILP problem, including ours, even the first semi-relaxed problem proves difficult to solve. Our second sub-problem, on the other hand is very easy to solve since all of the binary variables in the original problem have been fixed (i.e., they are no longer decision variables, rather they become parameters). As such, the proposed RFBD approach would solve much more efficiently if we could solve our first sub-problem more quickly. To do so, we add *logical restrictions* (LRs) to reduce the feasible region of the first sub-problem.

We can observe in our problem that, although placing excessive numbers of light sources indiscriminately throughout the grid may result in high ES_{ij} values, there is nothing explicitly in our ILP model that will prevent the solver from considering a high number of light sources, including solutions where light sources are placed in adjacent cells. While it is conceivable that in some cases it might be optimal to place light sources in adjacent cells, we feel that it is much more likely that such solutions would not arise. Even in cases where this might occur in the optimal solution, it is plausible to suggest that there would exist a sub-optimal feasible solution that would not include adjacent light sources but which would nonetheless have a very small optimality gap. We therefore introduce a set of logical restrictions that reduce the problem's feasible region by eliminating solutions with adjacent light sources, in hopes that it will speed up solution of the semi-relaxed sub-problem without unduly impacting optimality. More precisely, our restrictions take the form of the constraints in equation (4.12), where a light source in one cell, (x,y) , will exclude light sources in all cells immediately to the left $(x-1,y)$, right $(x+1,y)$, above $(x,y+1)$, or below $(x,y-1)$, as illustrated in Figure 4.6.

$$T_{x,y} + T_{x-1,y} + T_{x+1,y} + T_{x,y-1} + T_{x,y+1} \leq 1 \quad \forall x, y \quad (4.12)$$

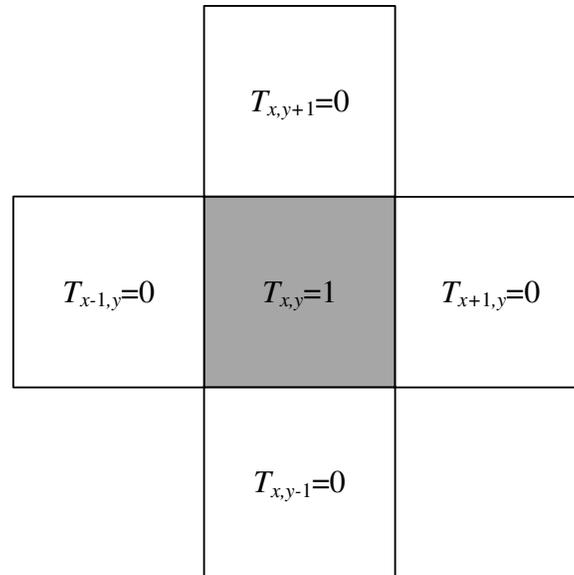


Figure 4.6: Illustration of the logical restrictions in equation (4.12).

As we mentioned earlier, for some hard ILP problem, the RFBD approach also takes a very long time to solve. From our preliminary experiments it has been observed that in such cases, the first semi-relaxed problem proved to be very difficult to solve, and the RFBD approach is unsuccessful in solving such instances within a reasonable amount of time. To improve the computational efficiency of the RFBD approach we add LR to the first sub-problem. The entire procedure is briefly illustrated in Figure 4.7.

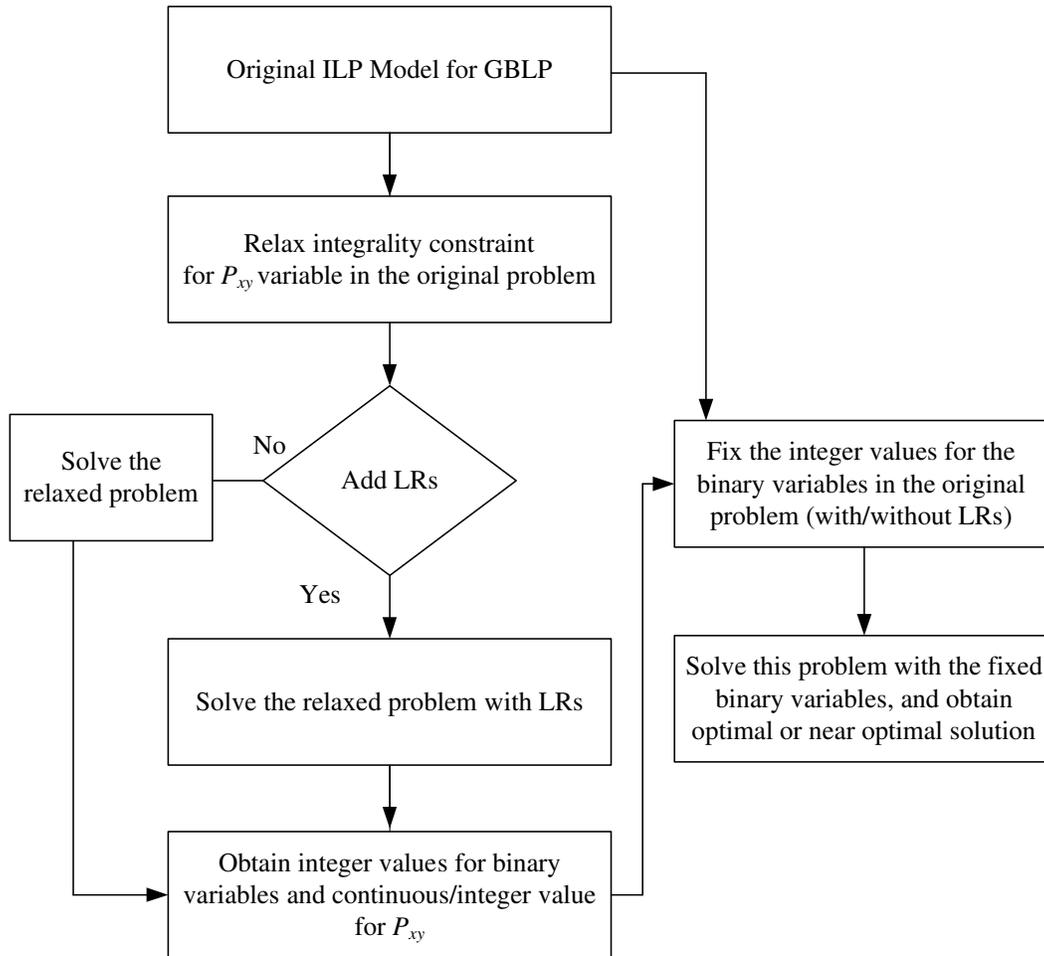


Figure 4.7: Illustration of the relax-and-fix-based decomposition approach.

4.5. Result Analysis

We solve our ILP problems with the same experimental setup described at the beginning of Section 4.3. Again, we note here that we use $UB = 10$ and $r = 2$, with a *mipgap* setting of 0.001. And in addition to the 10x20 and 15x15 grids used above, we also add the much larger 20x30 grid with the demand distribution

shown in Figure 4.8. In these experiments, the sum of the CPU times required to solve the two sub-problems will give us the CPU time for the RFBD approach as a whole.

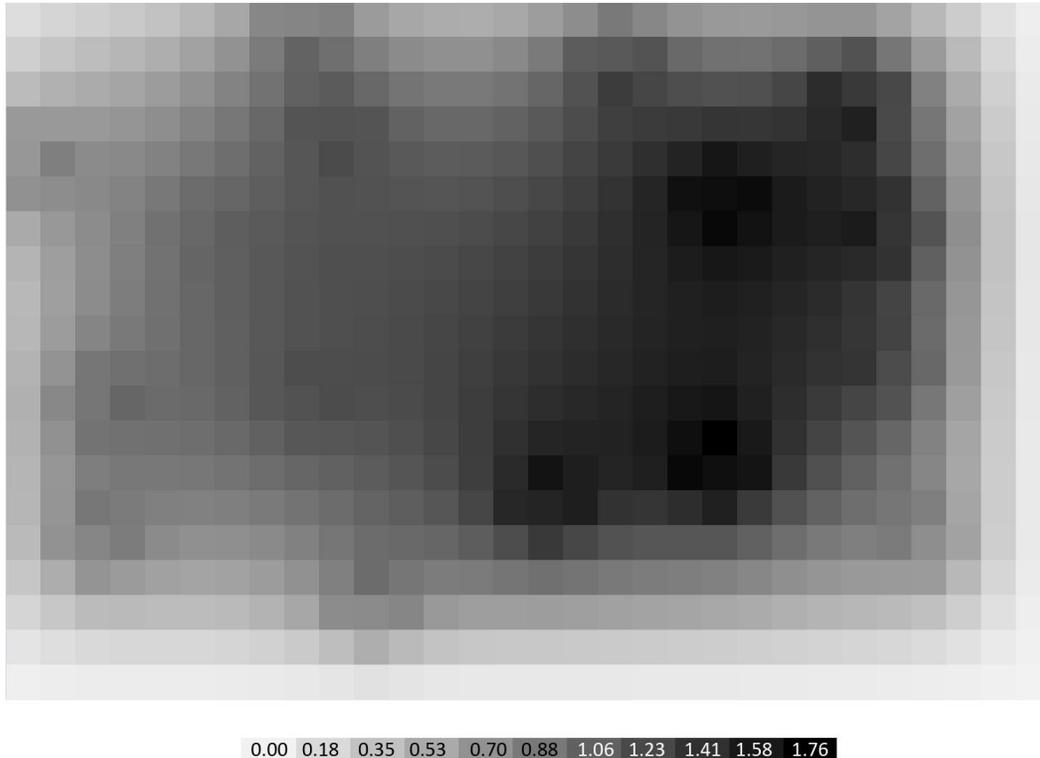


Figure 4.8: Demand distribution for 20x30 grid.

Table 4.1 through Table 4.3 show the comparative solution data for our ILP and algorithmic approaches on the three test-case grids. The “exact method” refers to the benchmark solution where the original ILP is solved to optimality (with $mipgap = 0.001$), while the “RFBD” and “RFBD with LR” columns refer to the solutions obtained using our relax-and-fix-based decomposition approach, without and with the logical restrictions of equation (4.12), respectively.

We see that for the hardest ILP instance of the 10x20 grid ($n = 13$), the RFBD approach could achieve a 95.1% runtime reduction with only a little over 2.38% increase in objective function values. On the other hand, the RFBD approach with logical restrictions reduces runtime by 99.6% with only a 2.64% increase in objective function value relative to the exact method. For the 15x15 test-case grid, we can observe in Table 4.2 that the hardest ILP instance ($n = 17$) is solved via the RFBD technique in only 50137.64 seconds and via RFBD with LR in only 1992.094 seconds, which represent 98.9% and 99.9% runtime reductions, again with only small impacts in optimality. And similarly in Table 4.3, we see that significant runtime reductions are made with the RFBD and RFBD with LR approaches on the 20x30 grid as well.

One might suggest that similar or perhaps even better runtime reductions might be obtained by simply increasing the *mipgap* setting on our solver. Doing so, with *mipgap* = 0.05, we find that in the 15x15 test grid with $n = 14$, $n = 15$, and $n = 16$, we obtain solution runtimes of over 78535 seconds, 282432 seconds, and 682080 seconds, respectively. These are improvements on the original runtimes in Table 4.2 with *mipgap* = 0.001, but they are still significantly longer than runtimes obtained from our RFBD approaches.

**Table 4.1: Comparison of Objective function value and CPU Time
(10x20 grid).**

Total number of light sources	Objective Function Value			CPU Time in Seconds			CPU Time reduction (%)	
	Exact Method	RFBD	RFBD with LR	Exact Method	RFBD	RFBD with LR	RFBD	RFBD with LR
1	145.962	145.962	145.962	0.828	0.219	0.266	73.6	67.9
2	111.174	111.174	111.174	1.094	1.406	1.328	-	-
3	88.306	88.306	88.306	4.5	5.469	6.297	-	-
4	76.094	76.094	76.094	29.297	32.547	28.61	-	2.3
5	64.106	64.218	64.218	81.625	41.984	49.047	48.6	39.9
6	55.836	55.836	55.836	148.359	121.703	100.454	18.0	32.3
7	49.494	49.662	49.662	382.938	250.969	228.141	34.5	40.4
8	43.354	43.792	43.654	1087.359	856.047	296.36	21.3	72.7
9	39.616	39.994	39.994	3280.094	1576.08	841.59	52.0	74.3
10	36.286	37.346	36.286	8224.734	4251.08	1973.95	48.3	76.0
11	33.734	33.794	34.374	76688.19	7930.5	2315.6	89.7	97.0
12	31.632	31.706	32.454	205382.16	26018.2	2128.3	87.3	99.0
13	30.13	30.846	30.926	1275670.6	62289.3	5351.6	95.1	99.6

**Table 4.2: Comparison of Objective function value and CPU Time
(15x15 grid).**

Total number of light sources	Objective Function Value			CPU Time in Seconds			CPU Time reduction (%)	
	Exact Method	RFBD	RFBD with LR	Exact Method	RFBD	RFBD with LR	RFBD	RFBD with LR
1	251.0	251.0	251.0	4.9	0.5	0.4	90.7	92.0
2	216.2	216.2	216.2	25.9	9.5	10.4	63.4	60.0
3	184.5	184.5	184.5	54.3	48.4	34.5	10.9	36.6
4	154.0	154.0	154.0	114.6	89.2	115.8	22.2	-
5	127.6	127.6	127.6	136.9	153.4	151.3	-	-
6	103.4	103.6	103.6	103.0	107.7	83.1	-	19.3
7	80.1	80.3	80.3	122.1	59.4	50.6	51.3	58.6
8	61.5	61.5	61.5	38.1	30.8	38.3	19.3	-
9	46.8	46.8	46.8	68.4	34.9	28.3	49.0	58.6
10	41.8	41.9	41.9	349.0	75.3	113.3	78.4	67.5
11	38.8	38.8	38.8	1313.3	136.7	201.4	89.6	84.7
12	35.1	36.0	36.0	3773.5	263.8	189.2	93.0	95.0
13	33.5	34.5	34.5	18024.9	1062.3	311.1	94.1	98.3
14	32.2	33.3	33.3	73152.0	5347.3	756.3	92.7	99.0
15	30.8	31.8	32.0	374235.7	11549	986.3	96.9	99.7
16	29.7	30.4	31.7	1643400.0	31501	2336.7	98.1	99.9
17	28.9	30.0	30.4	4491820.0	50138	1992.1	98.9	99.9

**Table 4.3: Comparison of Objective function value and CPU Time
(20x30 grid).**

Total number of light sources	Objective Function Value			CPU Time in Seconds			CPU Time reduction (%)	
	Exact Method	RFBD	RFBD with LR	Exact Method	RFBD	RFBD with LR	RFBD	RFBD with LR
1	495.3	495.3	495.3	95.0	2.6	2.5	97.3	97.3
2	460.5	460.5	460.5	176.3	39.0	35.5	77.9	79.9
3	428.8	428.8	428.8	1116.7	260.5	294.0	76.7	73.7
4	398.3	398.3	398.3	3390.6	1665.8	1229.0	50.9	63.8
5	371.1	371.1	371.1	20986.0	8924.4	8580.1	57.5	59.1
6	345.8	346.0	346.0	95002.9	58221.6	63990.0	38.7	32.6
7	319.5	319.5	319.5	180021.8	72138.7	93522.4	59.9	48.0
8	294.2	294.2	294.2	255978.8	100336	139757	60.8	45.4
9	269.6	269.6	269.6	368021.2	134579	153406	63.4	58.3
10	248.2	248.2	248.2	470075.2	264633	196605	43.7	58.2

It is clear from the runtime data above that solution runtimes appear to increase exponentially with increasing number of light sources. Initially, the RFBD approaches provide little or no improvements in runtime, but as the benchmark method's runtimes continue to increase, the RFBD runtimes increase much more slowly than runtime for the benchmark exact method. Looking closely at the runtimes of the two RFBD sub-problems, and continuing to solve for increasing numbers of light sources we gain additional insights. As we can observe in Table 4.4 (showing runtime breakdown of the RFBD with LR), it is the first sub-problem that is particularly difficult to solve, relative to the second sub-problem,

which appears to be exceedingly easy to solve. This suggests that if we can further reduce the runtime of this first sub-problem, we would likely further decrease runtime of the overall RFBD approach. We leave this for future work.

Furthermore, we can point out that the exact solution method was unable to obtain solutions for values of n greater than those shown in Table 4.1 through Table 4.3, we were able to obtain such solutions using the RFBD approaches, as shown in Table 4.4. And even more interestingly, runtimes eventually begin to decrease, quite substantially, as the number of light sources increases even further. Our interpretation of this is that when the number of light sources increases beyond a certain point, the solver begins seeking locations not to place a light source, which becomes smaller with increasing n .

The runtime data from Table 4.4 is also shown visually in Figure 4.9 through Figure 4.11. While solutions were generally easily obtainable for the 10x20 and 15x15 grids, even our decomposition technique was unable to obtain a complete solution for the more computationally complex instances of the 20x30 grid (those with $n \geq 14$) within a reasonable amount of time. We observe that the optimal objective function values decrease with increasing n until they reach an overall minimum, and then begin to increase (though we do not yet reach that overall minimum in the 20x30 test-case grid before the problem becomes intractable). We can also observe that the optimum number of light sources (n_{opt}) is 27 for the

10x20 grid and 24 for the 15x15 grid. To test the performance of our proposed ILP based heuristic approach, we compare the objective function value with a genetic algorithm (GA) approach. Our GBLP is implemented in MATLAB[®] (The MathWorks, 2010) and solved with GA for various grids. We used the following experimental setup to solve these problems:

Initial population function = Integer population (ROUND function was used to generate integer population)

Mutation function = Integer mutation (ROUND function was used to generate integer children)

Initial population range = Bounds for decision variables

Stall generations = 50

Generations = 50

Population size = 50

Figure 4.12 through Figure 4.14 are used to compare the RFBD approach with GA with respect to the objective function value. In each of those figures, the square data points represent the *objective function value* (OFV) obtained by RFBD with LRs and the diamond data points represent the OFV obtained by GA. These variations show that our RFBD approach outperforms GA.

Table 4.4: Details of CPU Time for RFBD with LR to solve large scale instances.

Total number of light source	CPU Time (in Seconds)					
	10x20 grid		15x15 grid		20x30 grid	
	First Sub-problem	Second sub-problem	First sub-problem	Second sub-problem	First sub-problem	Second sub-problem
1	0.219	0	0.391	0	2.547	0
2	1.312	0.016	10.344	0.016	35.453	0.016
3	6.281	0.016	34.453	0.016	293.969	0.016
4	28.125	0.016	115.734	0.016	1228.953	0.016
5	48.406	0.016	151.25	0.016	8580.078	0.047
6	98.969	0.031	83.094	0.016	63990.03	0
7	228.641	0.031	50.594	0.016	93522.39	0.031
8	292.953	0.031	38.266	0.031	139756.8	0.047
9	844.031	0.031	28.328	0.016	153406.3	0.031
10	1956.734	0.031	113.266	0.016	196604.7	0.031
11	2312.516	0.031	201.359	0.047	346319.2	0.031
12	2118.125	0.031	189.156	0.047	553120.5	0.031
13	5319.984	0.047	311.062	0.031	1040220	0.016
14	5896.828	0.047	756.266	0.062	-	-
15	6696.688	0.062	986.266	0.062	-	-
16	8003.547	0.094	2336.594	0.062	-	-
17	3285.234	0.062	1992.016	0.078	-	-
18	2434.109	0.109	406.906	0.062	-	-
19	1161	0.156	115.797	0.078	-	-
20	419.438	0.188	58.453	0.094	-	-
21	195.266	0.094	39.969	0.094	-	-
22	154.625	0.125	30.328	0.078	-	-
23	101.781	0.078	57.078	0.078	-	-
24	57.797	0.109	41.422	0.031	-	-
25	80.391	0.078	36.312	0.047	-	-
26	56.047	0.109	67.469	0.031	-	-
27	60.453	0.062	73.016	0.031	-	-
28	40.266	0.047	51.016	0.047	-	-
29	40.547	0.078	40.109	0.062	-	-
30	23.188	0.031	34.891	0.062	-	-
31	32.844	0.031	33.656	0.047	-	-
32	12.141	0.047	23.719	0.062	-	-
33	11.344	0.047	18.078	0.047	-	-
34	5.484	0.031	12.922	0.031	-	-
35	5.078	0.016	11.172	0.016	-	-
36	4.391	0.031	8.703	0.031	-	-
37	4.016	0.031	8.344	0.047	-	-
38	1.391	0.016	8.281	0.031	-	-
39	1.141	0.031	2.594	0.031	-	-
40	2.062	0.031	1.75	0.016	-	-

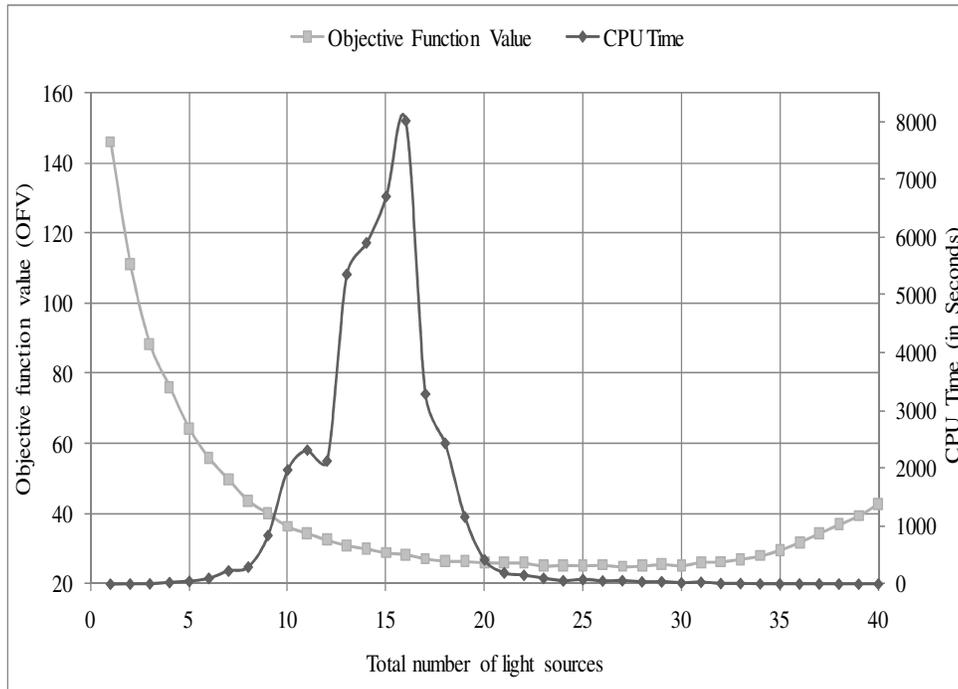


Figure 4.9: Objective function value and solution runtime of the GBLP using RFBD with LR on the 10x20 grid.

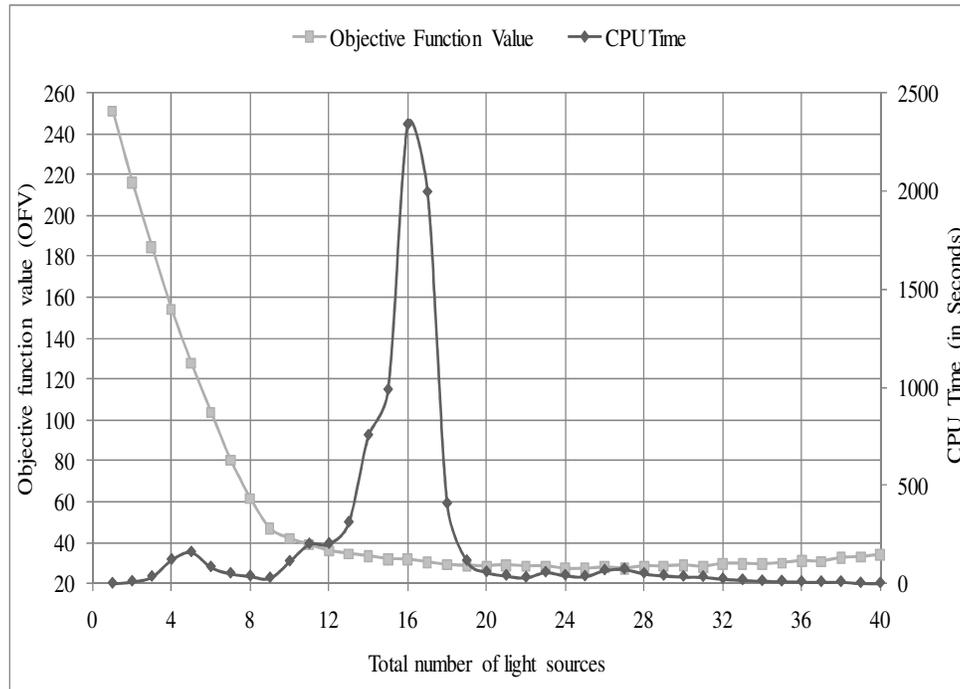


Figure 4.10: Objective function value and solution runtime of the GBLP using RFBD with LR on the 15x15 grid.

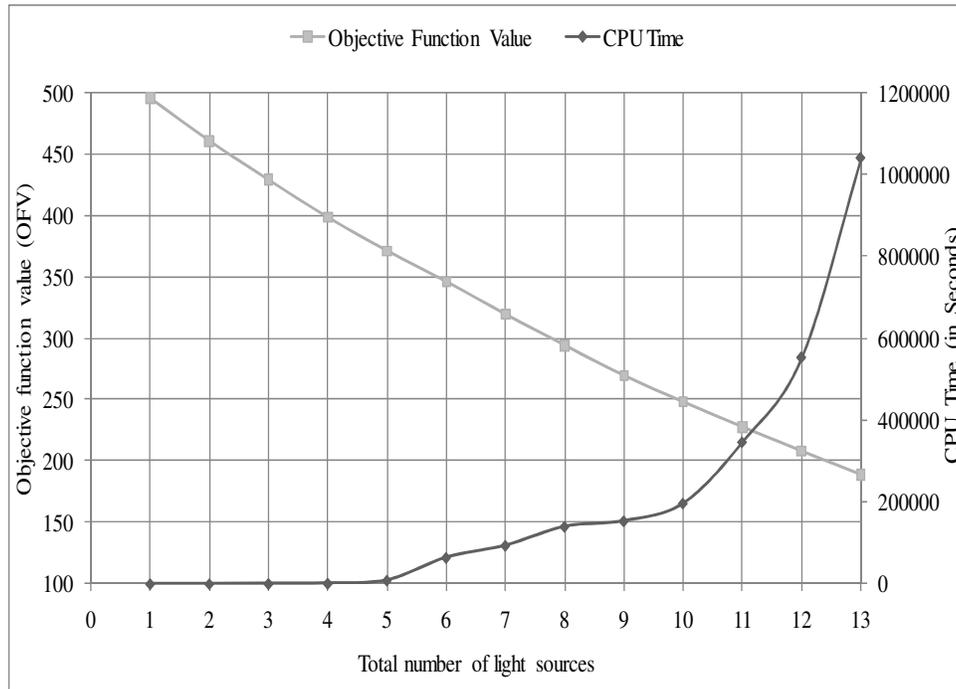


Figure 4.11: Objective function value and solution runtime of the GBLP using RFBD with LR on the 20x30 grid.

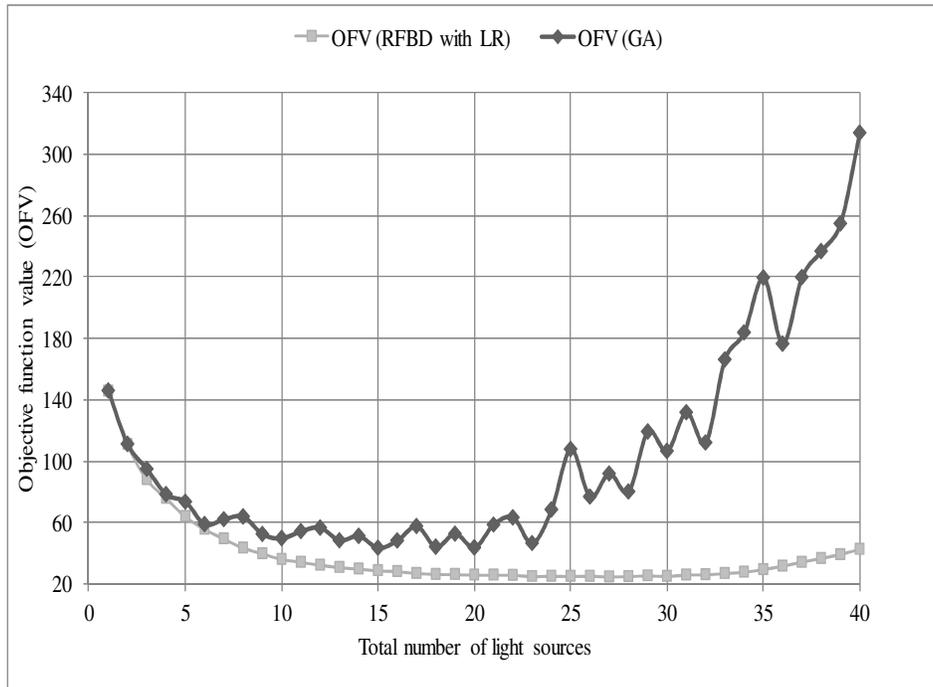


Figure 4.12: Objective function value of the GBLP using RFBD with LR and GA on the 10x20 grid.

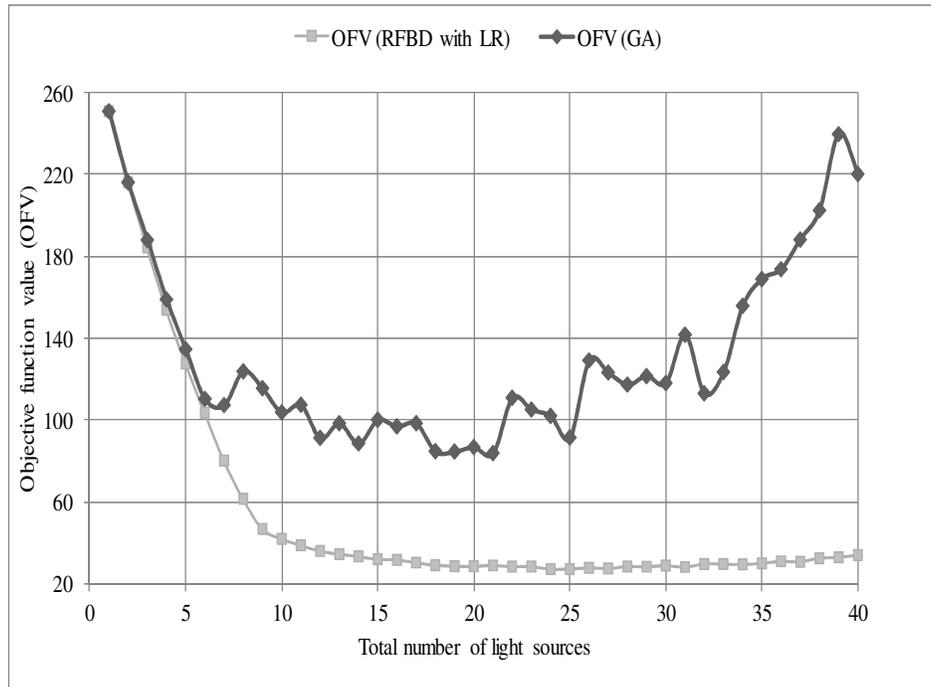


Figure 4.13: Objective function value of the GBLP using RFBD with LR and GA on the 15x15 grid.

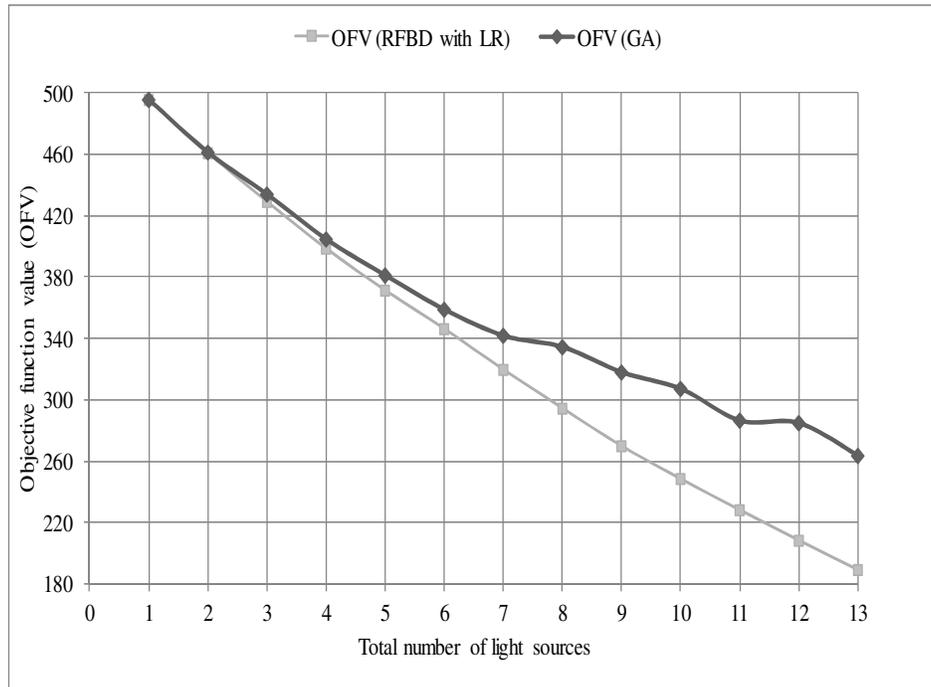


Figure 4.14: Objective function value of the GBLP using RFBD with LR and GA on the 20x30 grid.

We can also now return to our discussion above, regarding the progress of our problem through the branch-and-bound tree and the introduction of logical restrictions to reduce the feasible region and the number of branch-and-bound nodes in our problem solution. Table 4.5 shows the details on the number of Simplex iterations and branch-and-bound nodes for solutions using our RFBD and RFBD with LR approaches. We can observe, for instance, that in the 10x20 grid with 13 light sources ($n = 13$), solution of the first sub-problem in our RFBD approach requires nearly 577 million Simplex iterations and nearly 3 million branch-and-bound nodes without logical restrictions, while use of logical restrictions reduces those numbers to just over 43 million and just under 349 thousand, respectively. Clearly, the introduction of logical restrictions significantly reduces the complexity of the first sub-problem in our RFBD approach, which is consistent with runtime reductions observed in Table 4.1

through Table 4.3 (e.g., RFBD runtimes in the 10x20 grid with $n = 13$ dropped from 62,289 seconds without logical restriction to 5,351 seconds with logical restrictions).

Table 4.5: Branch-and-bound statistics for 10x20 grid solutions

n	RFBD				RFBD with LR			
	Number of MIP Simplex Iterations		Number of Branch-and-Bound Nodes		Number of MIP Simplex Iterations		Number of Branch-and-Bound Nodes	
	sub-prob. #1	sub-prob. #2						
1	1,494	25	102	0	1,494	25	102	0
2	10,403	50	513	0	10,494	50	522	0
3	27,755	92	560	0	27,042	92	549	0
4	179,932	119	1,468	0	131,797	119	1,036	0
5	203,907	140	1,457	0	263,165	140	1,935	0
6	787,562	156	4,757	0	620,937	156	4,090	0
7	1,635,738	205	8,989	0	1,395,123	205	7,039	0
8	4,877,580	206	23,521	0	1,874,634	216	10,179	0
9	9,914,572	299	54,396	10	5,476,282	299	26,736	10
10	27,503,543	398	145,484	15	13,656,892	239	70,549	0
11	55,826,720	318	298,043	6	17,854,132	424	113,308	18
12	179,826,722	338	1,053,518	0	16,614,079	350	120,894	3
13	576,961,471	508	2,988,967	22	43,274,511	387	348,969	0

4.6. Concluding Discussion

We have developed a relax-and-fix-based decomposition approach to solve large-scale grid-based location problems. We've tested our problem-specific decomposition technique with a light post placement problem using several test case grids and have shown that the RFBD technique is quite effective in reducing problem runtimes without significant loss of optimality. In our most

computationally complex test case on the 10x20 grid, we reduce runtime by 95.12% while increasing the obtained objective function value by only 2.38%, and in our most computationally complex test case on the 15x15 grid, we reduce runtime by 98.88% while increasing the obtained objective function value by only 3.58%. We then develop problem-specific logical restrictions to reduce the feasible region and the resulting branch-and-bound tree, thereby providing further reductions in runtime. When we add the logical restrictions to our RFBD, we are able to reduce runtimes as much as 99.58% in the most complex 10x20 test case with a 2.64% loss of optimality and as much as 99.96% in the most complex 15x15 test case with a 4.97% loss of optimality.

However, the numbers reported above are the best achieved in all of our test cases. Runtime reductions in test cases on our largest grid did not experience such drastically improved runtimes; reductions in the range of 50% are more typical, and the problem remains intractable for the most computationally complex of those test cases. We will extend our technique to include a fixed-charge component and additional logical restrictions with a modified decomposition to address this challenge in future work. And while we have tested our RFBD approach on a GBLP that seeks to optimally place light posts to minimize the sum of the total unmet demand and excess light supply, other GBLPs could be solved using this approach. Potential problems include optimal placement of retail outlets

and/or warehouses, sensor network configuration, resource exploitation, and even optimal delivery of radiation therapy.

References

- Aardal, K., Pochet, Y., Wolsey L. (1995), “Capacitated facility location: valid inequalities and facets”, *Mathematics of Operations Research*, 20, 552–582.
- Aardal, K. (1998), “Reformulation of capacitated facility location problems: How redundant information can help”, *Annals of Operations Research*, 82, 289-308.
- Anderson, E. J., Ferris, M.C. (1994), “Genetic algorithms for combinatorial optimization: The assembly line balancing problem”, *ORSA Journal on Computing*, 6 (2),161–173.
- Aytug, H., Saydam, C. (2002), “Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study”, *European Journal of Operational Research*, 141, 480–494.
- Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.F., Vance, P.H. (1998), “Branch-and-Price: Column Generation for Solving Huge Integer Programs”, *Operations Research*, 46 (3) , 316-329.
- Barnhart, C., Hane, C.A., Vance, P.H. (2000), “Using Branch-and-Price-and-Cut to Solve Origin-Destination Integer Multi commodity Flow Problems”, *Operations Research*, 48 (2), 318-326.
- Belov, G., Scheithauer, G. (2006), “A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting”, *European Journal of Operational Research*, 171, 85–106.
- Beraldi, P., Ghiani, G., Grieco, A. Guerriero, E. (2006), “Fix and Relax Heuristic for a Stochastic Lot-Sizing Problem”, *Computational Optimization and Applications*”, 33, 303–318.
- Chen, D.-S., Baton, R. G., Dang, Y. (2010), *Applied Integer Programming*, A John Willey & Sons, Inc., Publication, (Chapter 2).

Cheng, C.-P. Liu, C.-W., Liu, C.-C. (2000), “Unit Commitment by Lagrangian Relaxation and Genetic Algorithms”, *IEEE Transactions on Power Systems*, 15 (2), 707-714.

Ferreira, D., Morabito, R., Rangel, S. (2010), “Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants”, *Computers & Operations Research*, 37, 684 – 691.

Fourer, R., Gay, D. M. and Kernighan, B. W. (2002), *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press, Cole Publishing Co.

Ghiani, G., Grandinetti, L., Guerriero, F., Musmanno, R. (2002), “A Lagrangean Heuristic for the Plant Location Problem with Multiple Facilities in the Same Site”, *Optimization Methods and Software*, 17,1059–1076.

ILOG (2007), “ILOG CPLEX 11.0 User’s Manual”, ILOG, Inc.

Kelly, J. D., Mann, J. L. (2004), “Flow sheet decomposition heuristic for scheduling: a relax-and-fix method, *Computers and Chemical Engineering*”, 28, 2193–2200.

Klose, A., Görtz, S. (2007), “A branch-and-price algorithm for the capacitated facility location problem”, *European Journal of Operational Research*, 179, 1109–1125.

Lorena, L.A.N., Senne, E.L.F. (2004), “A column generation approach to capacitated p-median”, *Computers & Operations Research*, 31, 863–876.

Mauri, G. R., Ribeiro G. M., Lorena, L. A. N. (2010), “A new mathematical model and a Lagrangian decomposition for the point-feature cartographic label placement problem”, *Computers & Operations Research*, 37, 2164–2172.

Mohammadi, M., Ghomi, S. M. T. F., Karimi, B, Torabi, S. A. (2010), “Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lot sizing problem with sequence-dependent setups”, *Journal of Intelligent Manufacturing*, 21, 501–510.

- Noor-E-Alam, M., Ma, A., Doucette, J. (2012), “Integer Linear Programming Models for Grid-Based Light Post Location Problem”, *European Journal of Operational Research*, 222, 17-30.
- Rajagopalan S., Heragu S. S., Taylor, G. D. (2004), “A Lagrangian relaxation approach to solving the integrated pick-up/drop-off point and AGV flow path design problem”, *Applied Mathematical Modelling*, 28, 735–750.
- Roy, T .J. V., Wolsey, L.A. (1987), “Solving mixed integer programming using automatic reformulation”, *Operations Research*, 35, 45 – 57.
- Savelsbergh, M.W.P., Sigismondi, G.C., Nemhauser, G.L. (1995), “Functional description of MINTO, a Mixed INTEger Optimizer”, *Operations Research Letters*, 15, 47 – 58.
- Senne, E.L.F., Lorena, L.A.N., Pereira, M.A. (2005), “A branch-and-price approach to p-median location problems”, *Computers & Operations Research*, 32, 1655–1664.
- The MathWorks, Inc. (2010). MATLAB-The Language of Technical computing [online]. Available from: <http://www.mathworks.com/products/matlab/> [April 2010].
- Trick, M. A. (2005), “Formulations and Reformulations in Integer Programming”, Second International Conference on Integration of Artificial Intelligence and Operations Research, CPAIOR, Prague, Czech Republic, May 30 - June 1, 366-379.
- Williams, H. P. (1978), The Reformulation of Two Mixed Integer Programming Problems”, *Mathematical Programming*, 14, 325-331.
- Wolsey, L. A. (1998), *Integer Programming*, Wiley-Interscience, Series in Discrete Mathematics and Optimization, Toronto, (Chapter 1).
- Yamin, H.Y., Shahidehpour, S. M. (2004), Unit commitment using a hybrid model between Lagrangian relaxation and genetic algorithm in competitive electricity markets, *Electric Power Systems Research*, 68, 83-92.

Chapter 5³

Solving Large Scale Fixed Cost GBLPs

5.1. Introduction

Determining optimal locations for placement of warehouses, service centres, depots, etc. is a complex problem in business and engineering, and it plays a vital role in the success of an organization. Several methods have been developed in the area of location theory resulting in a number of notable mathematical models

³ A version of this chapter has been submitted for publication: Noor-E-Alam, M., Doucette, J. (2012), "Solving Large Scale Fixed Cost Integer Linear Programming Models for Grid-Based Location Problems with Heuristic Techniques", *Computers & Operations Research*, in review, submitted 30 August 2012.

and their solution methods. These mathematical models and the solution procedure are designed for a specific type of location problem. The complexity of the location problem depends on the nature of the problem and the considered criteria, which are selected by the decision maker from the problem description (Teixeira and Antunes, 2008). There are many types of location problems: Weber problem by Cooper (1963); Weber problem using rectilinear distances by Wesolowsky (1972); capacitated multi-facility Weber problem (CMFWP) by Sherali and Noradi (1988); probabilistic multi-facility Weber problem by Katz and Cooper (1974), which was later revisited by Altinel et al. (2009); coverage problem by Drezner et al. (1997) and Berman et al. (2010), and discrete location problem by Marín (2011).

Depending on the capacity restriction of a source, location problems can be classified as *uncapacitated facility location* (UFL) or *capacitated facility location* (CFL) problems, where the UFL becomes a CFL problem when there is an upper limit for the amount of supply (Ghiani et al., 2002; Chen, 2010). It is also important to recognize that mathematical models to solve the location problems can be classified into two distinct groups: *discrete* location analysis and *continuous* location analysis. In discrete location analysis, facilities and supply points are modeled as the vertices or nodes (Domschke and Krispin, 1996), whereas in continuous location analysis, customer demands are coordinate points

on a continuous plane and the feasible solution for the optimal placement of facilities can be any coordinate point in that plane.

In the present research we consider a special type of continuous location problem called *grid-based location problems* (GBLPs) that can be used to make optimal location decisions in business, engineering, resource exploitation, and even in the field of medical sciences (Noor-E-Alam et al., 2012). In these problems, locations can be approximated by a grid-based system of small-sized cells. To model a location problem as a GBLP, a heterogeneous demand distribution and a supply function can be established by using those cells. Based on these demand distributions, supply relationships and other constraints, we can then determine the optimum number, location(s), and size(s) of facilities simultaneously while fulfilling certain objectives. To do so, we need to develop mathematical models that consider a large number of discrete variables and complicated constraints.

The GBLP mathematical models proposed in (Noor-E-Alam et al., 2012) did not consider *fixed costs* (also known as *implementation cost*) incurred when installing a supply source. In practice, there is a significant amount of fixed cost involved in installing a supply source. As previous models did not consider fixed costs, the optimal decisions found from these models are not truly optimal. Combinations of these considerations contribute to producing large-scale and computationally difficult problems, which are generally not scalable and often become intractable

to solve with current methods. As such, this research plans to develop a model that includes fixed cost criteria for GBLPs. In addition to that, our goal in this research is to develop effective and efficient techniques for solving large scale fixed cost GBLP instances.

GBLP mathematical models are generally developed as *integer linear programming* (ILP) models. In practice, ILP problems are often easily solved using a variety of techniques as there is no known polynomial-time algorithm for solving them (Wolsey, 1998). One of the most popular techniques is *LP relaxation*, where the integer decision variables are permitted to take non-integer values, which results in a lower (upper) bound on the optimal solution for minimization (maximization) problems. An optimal solution to the LP relaxation version of an ILP is often quite a weak bound. The most widely used technique to solve ILPs is the *branch-and-bound* algorithm that makes use of LP relaxation. In the branch-and-bound algorithm, the lower bounds obtained from LP relaxation and upper bounds obtained from semi-relaxed problems are used to fathom the branch-and-bound search tree (Chen, 2010; Wolsey, 1998). Different versions of branch-and-bound techniques with cutting plane algorithms are widely used in ILP solvers such as CPLEX, Gurobi Optimization and MINTO. However, due to the combinatorial nature of some large scale ILPs, it is often hard to solve them with current branch-and-bound methods; many real-world ILP problems require

weeks or months of solution time to solve on the most powerful systems (Noor-E-Alam et al., 2012).

To improve the efficiency of the branch-and-bound algorithm, we need to reformulate our mathematical model in such a way that the LP relaxation produces very tight bound (i.e., small LP gap) and so that the solver requires less time to reach optimality. Such reformulation techniques for solving location problems are available in Aardal (1998), where an alternative way of modeling the CFL problem is discussed. In this model, a set of new decision variables and redundant constraints are added together such that the solver is able to generate stronger linear inequalities and takes less time to reach optimality. ILP models with extra logical restrictions are also discussed in Williams (1978) and Aardal et al. (1995), and reduced computational time significantly. Avella and Sassano (2001) investigated the polyhedral structure of *p*-median facility location problems to identify valid inequalities to get better formulation, whereas, related analysis is carried out for CFL in Leung and Magnanti (1989). Some branch-and-bound packages generate certain classes of inequalities for simple structures, such as the knapsack, single-node flow and path polytopes for efficient relaxation of many combinatorial optimization problems (Roy and Wolsey, 1987; Savelsbergh et al., 1995). In those cases, to help the solver's automatic reformulation, manual reformulation is often needed to redefine variables (Trick, 2005).

A number of decomposition techniques for solving large-scale ILP problems have been developed and evaluated in recent decades. The *Dantzig-Wolfe* decomposition technique is discussed in Barnhart et al. (1998) for solving large ILP problems, where *column generation* has been employed for implicit pricing of non-basic variables. A *branch-and-price-and-cut* algorithm is developed in Barnhart et al. (2000) that permits column generation and a cutting plane algorithm to be applied throughout the branch-and-bound search tree to reduce the computational time for ILPs. On the other hand, cutting plane and column generation algorithms are integrated with the branch-and-bound algorithm to improve relaxation of the problem and achieve price out efficiency in the *branch-cut-price* (BCP) algorithm (Belov and Scheithauer, 2006). However, column generation shows heavy tail convergence due to large fluctuations in the simplex multipliers (Valero, 2005). To improve this convergence rate of the column generation scheme, heuristic technique can be used to stabilize this procedure (Amor et al, 2006).

To solve CFL, a column generation procedure geared with a *branch-and-price* algorithm is used in Klose and Görtz (2007). In this research, a stabilized column generation method is used for solving the corresponding master problem to optimality. A similar branch-and-price algorithm is also used to solve the capacitated p-median problem in Lorena and Senne (2004) and Senne et al.

(2005). Sonmez and Lim (2012) developed an integer programming model for facility location problem, and a decomposition algorithm is proposed to produce near optimal solutions very quickly. García et al. (2012) proposed a branch-and-cut algorithm to solve large-scale ILP instances for uncapacitated multiple allocation p-hub median problems. Furthermore, Benders decomposition is used to solve CFL in (Magnanti and Wong, 1981) and (Wentges, 1996). This decomposition technique is also used to solve large scale ILP instance for uncapacitated hub location problems with multiple assignments (Contreras et al., 2011). While the above-mentioned methods often represent significant reductions in solution time, they are often suitable only for the specific types of problems having specific mathematical structure.

Another widely used decomposition technique is *relaxation-based decomposition*, where it is used to decompose the original problem into easier sub-problems by relaxing the complicating constraints or integrality restrictions. This relaxed problem is solved and its partial solutions are fixed in the original problem to generate another easier sub-problem, often called the *core problem*. This core problem is then solved to obtain a near-optimal solution to the original ILP (Wolsey, 1998). Such relax-and-fix strategies have been used in many works in the literature to solve large scale ILP problems (Beraldi et al., 2006; Ferreira and Morabito, 2010; Kelly and Mann, 2004; Mohammadi et al., 2010). *Lagrangian*

decomposition (LD) based decomposition creates easier sub-problems by relaxing the complicating constraints (Mauri et al., 2010). The solution of an easier sub-problem is then fixed into the original problem to create another sub-problem, which is then solved to obtain a near-optimal solution (Rajagopalan et al., 2004). Lee and Lee (2012) designed an LD-based heuristic technique to solve an ILP model to make facility location decisions with customer preferences. This heuristic was successful in yielding high quality solutions in a reasonable time. In other work, *dynamic programming* is used to calculate lower bounds in LD to solve CFL, and computational statistics show that this approach generates high quality solutions very quickly (Ghiani et al., 2002).

For solving large-scale location problems, *meta-heuristic* techniques are also used separately or combined with other decomposition approaches. A *genetic algorithm* (GA) is used to solve large-scale maximum expected covering location problems (Aytug and Saydam, 2002), and it is observed that a high quality solution is obtained very quickly. GA is shown to be very effective for some combinatorial optimization problems (Anderson and Ferris, 1994). For solving combinatorial optimization problems such as unit commitment problems, GA and LD have been combined together, where GA is used to update the Lagrangian multipliers and improve the performance of the LD method (Yamin and Shahidehpour, 2004; Cheng et al., 2000). You and Yamada (2011) proposed a

new branch-and-bound algorithm to solve *multiple knapsack problems* (MKP), where the LD approach is used to obtain an upper bound, and a greedy heuristic is used to obtain a lower bound. Sun (2012) implemented a *tabu search* heuristic method to solve a CFL problem. Computational tests show that this method was successful in solving intractable instances. Lee and Chang (2007) formulated an *unreliable discrete location problem* as an ILP to minimize the sum of fixed costs and expected operating costs. This research also proposed a dual-based heuristic technique to solve large-scale instances for such location problems. Noor-E-Alam and Doucette (2012) proposed a problem-specific *relax-and-fix-based decomposition* (RFBD) technique to solve large GBLP instances efficiently. To reduce solution time further, that work also proposed additional logical restrictions. It was observed from runtime statistics that the RFBD technique was successful in solving large-scale GBLPs.

5.1.1. Proposed Work

Much less effort has been made in studying the fixed cost ILP model for GBLPs and related solution procedures for large-scale instances. Mathematical models described in (Noor-E-Alam et al., 2012) and (Noor-E-Alam and Doucette, 2012) did not consider implementation costs or fixed costs; therefore the optimal decisions found from those models are not strictly optimal on the basis of total cost. As such, the present work is to develop an ILP model that considers the

fixed cost criteria for GBLPs. We find from our preliminary experiments that it becomes intractable for large-scale instances. The computational complexity of the fixed cost GBLP model is evaluated with several large-scale test-case grids and the structure of the mathematical model is studied to identify the cause of the exponential behavior of the CPU time. Based on findings from this investigation, we propose a problem-specific decomposition technique to solve large instances efficiently. In addition to that, we also propose an integrated RFBD technique with this decomposition approach to solve large-scale instances. Finally, the ILP model and proposed decomposition technique are implemented within a standard modeling language and tested on a number of large test-case grids to compare the performance of the proposed method.

The remainder of this chapter is organized as follows. Section 5.2 provides the description of the fixed cost GBLP model. In Section 5.3, we discuss preliminary results. In Section 5.4, we propose problem-specific decomposition techniques to solve large-scale instances. Section 5.5 describes the results and analysis, while conclusions and future research opportunities are described in Section 5.6.

5.2. Fixed Cost GBLP Model

In this work, to formulate a fixed cost model, we use the light post GBLP described in Noor-E-Alam et al. (2012). By using this problem, we'll compare the effectiveness of our proposed decomposition techniques for efficiently solving

large scale fixed cost GBLPs. To better understand the model, a brief discussion is provided herein, but detailed description and analysis of the problem is available in that prior work. This problem involves making optimum decisions on installing lights in a city park. Factors affecting the number of light sources, their sizes, and their locations are many and varied. In a city park, there are different areas used for a variety of purposes, requiring differing amounts of light. Trees and topography can also vary significantly throughout the park, further affecting the degree of brightness an area may require. It can also be assumed that the boundary region of this city park is not a feasible area where we can consider installing a light due to the various physical restrictions such as fences and underground utilities. This city park can be modeled as a fixed cost GBLP, where a heterogeneous demand distribution (i.e., the amount of light required at various locations) can be represented by cells. The goal is to find the optimal placement of light posts that would best satisfy the heterogeneous demand distribution while minimizing the total cost.

Noor-E-Alam et al. (2012) presented the light distribution model where the quantity of light delivered to neighbouring cells depends on the rectilinear distances from the point source to the respective cells, as shown in Figure 5.1. The set of equations shown in this figure is used to calculate the amount of light delivered in different cells, where S_a , S_b , S_c , S_d , S_e and S_f represent the total

amount of available supply at location a , b or b' , c , d or d' , e or e' and f , respectively. The distances in this model are calculated as the 3-dimensional Euclidean distances between the light source and the centers of the various cells. In the figure, the vertical distance between the point source, o , and centre point of the grid below it, a , is r . P is the luminosity of the point source. The angle between vertical line oa and line ob or ob' is α_b , the angle between vertical line oa and line oc is α_c , the angle between vertical line oa and line od or od' is α_d , the angle between vertical line oa and line oe or oe' is α_e , and the angle between vertical line oa and line of is α_f .

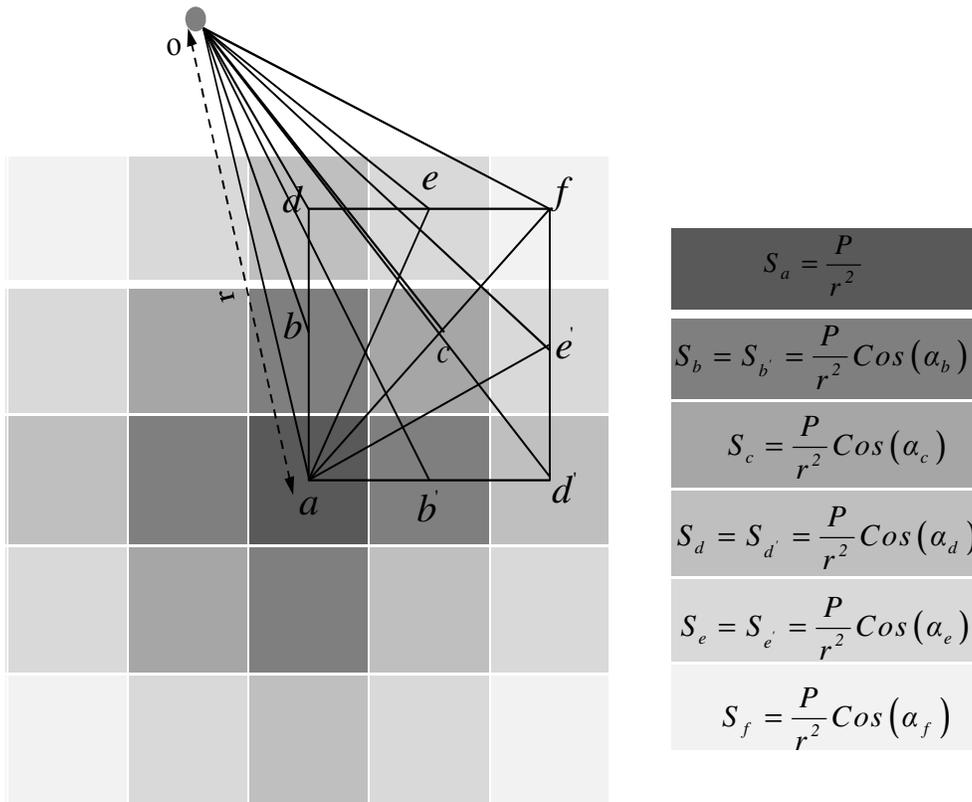


Figure 5.1: Light distribution model.

The GBLP models outlined in Noor-E-Alam et al. (2012) and Noor-E-Alam and Doucette (2012) did not consider the implementation costs or fixed costs required to install light sources. However, in practice, a significant amount of fixed cost is involved to install lights. Therefore, the optimal solutions from these models are not strictly optimal on the basis of total cost. In this present work, our goal is to develop a GBLP that includes fixed costs. To solve this new decision problem effectively by incorporating fixed cost criteria, we develop an ILP model. We first need to define the following notation:

Input Parameters:

D_{ij} is the demand at location (i,j)

Q is the set of all x-coordinates in the grid, indexed by i

R is the set of all y-coordinates in the grid, indexed by j

i_{max} is an upper limit or maximum value placed on i

i_{min} is a lower limit or minimum value placed on i

j_{max} is an upper limit or maximum value placed on j

j_{min} is a lower limit or minimum value placed on j

β is the system boundary constant

UB_{xy} is the upper bound on decision variable P_{xy}

X is the set of all x-coordinates of the light source, indexed by x

Y is the set of all y-coordinates of the light source, indexed by y

UB_{xy} is the upper bound on decision variable P_{xy}

C_f is the fixed cost to install a light source

C_v is the per unit variable cost

Decision Variables:

S_{ij} is the total supply at location (i,j)

P_{xy} is the integer size of the light source at location (x,y)

T_{xy} is a binary variable at location (x,y) : T_{xy} is 1 if $P_{xy}>0$; T_{xy} is 0 if $P_{xy}=0$

Finally, the fixed cost GBLP ILP model can be formulated as follows:

$$\text{Minimize } C_v \sum_{x \in X} \sum_{y \in Y} P_{xy} + C_f \sum_{x \in X} \sum_{y \in Y} T_{xy} \quad (5.1)$$

Subject to:

$$S_{ij} = \sum_{x \in X} \sum_{y \in Y} \frac{P_{xy}}{r^2} \cos \left(\tan^{-1} \left(\frac{\sqrt{(i-x)^2 + (j-y)^2}}{r} \right) \right), \quad \forall i \in Q, \forall j \in R \quad (5.2)$$

$$S_{ij} \geq D_{ij}, \quad \forall i \in Q, \forall j \in R \quad (5.3)$$

$$P_{xy} \leq MT_{xy}, \quad \forall x \in X, \forall y \in Y \quad (5.4)$$

$$P_{xy} = 0, \quad \forall x \in X, y \in \{1 \dots \beta\} \quad (5.5)$$

$$P_{xy} = 0, \quad \forall x \in X, y \in \{J_{max} - \beta \dots J_{max}\} \quad (5.6)$$

$$P_{xy} = 0, \quad x \in \{1 \dots \beta\}, \forall y \in Y \quad (5.7)$$

$$P_{xy} = 0, \quad x \in \{I_{max} - \beta \dots I_{max}\}, \forall y \in Y \quad (5.8)$$

$$0 \leq P_{xy} \leq UB_{xy}, \quad \forall x \in X, \forall y \in Y \quad (5.9)$$

$$T_{xy} \in \{0, 1\}, \quad \forall x \in X, \forall y \in Y \quad (5.10)$$

Our objective is to minimize the sum of the variable costs and fixed costs required to fulfill the demands throughout the grid. To fulfill this goal, equation (5.1) is used as an objective function. In this equation, the first and second terms express total variable cost (\$) and total fixed cost (\$) respectively. On the basis of our description of light distribution in Figure 5.1, equation (5.2) is used to calculate the total supply available in each cell (i,j) . For example, a light post is located at point a , and we then need to calculate the supply at point e . To calculate the supply at that point we need to find the value of angle α_e . This angle can be calculated by the ratio of ae and r . The value of ae is the distance between the point, a and the point, e . Moreover, the total supply in a particular cell (i,j) is the summation of all the individual supplies resulting from all light sources. In this model, we minimize the total cost while satisfying the demand constraint described in equation (5.3), which confirms that total supply in each cell should be greater or equal to the demand for each cell. Constraint (5.4) confirms that if $P_{xy} > 0$, then $T_{xy} = 1$ or 0 , otherwise. In this equation, M is a sufficiently large number to satisfy this restriction. Equations (5.5)-(5.8) are used to incorporate the boundary restrictions. Finally, the bounding constraint (5.9) is required to put bounds on decision variable P_{xy} and equation (5.10) defines T_{xy} as a binary variable.

5.3. Preliminary Results

To demonstrate the effectiveness of the proposed fixed cost GBLP model, we used 11 test-case grids: 10x10, 10x10a, 10x12, 12x12, 10x15, 10x17, 10x17a, 10x20, 15x15, 20x30 and 20x30a with demand distributions outlined in Figure 5.2 to Figure 5.12, respectively. Most of these test-case grids are used in Noor-E-Alam et al. (2012) and Noor-E-Alam and Doucette (2012). Note that, in this research, some of our test-case grids are identically sized, but have different demand distributions in order to observe the effects of those changes in demand distribution will have in runtime statistics. We solve our instances of the ILP problem on an 8 processor ACPI multiprocessor X64-based PC with Intel Xeon® CPU X5460 running at 3.16GHz with 32 GB memory. We have implemented our model in AMPL (Fourer et. al., 2002), and solved with the state-of-art solver CPLEX 11.2 (ILOG, 2007). In these experiments, other parameters are assumed as follows: $C_v = \$1/\text{Candela}$, $C_f = \$10$, $UB = 10$, $\beta = 2$ and $r = 2$. We used a CPLEX *mipgap* setting of 0.001 that means all test cases solved to full termination are provably within 0.1% of optimality.

0.36	0.55	0.73	0.84	0.98	1.21	1.45	0.96	0.46	0.12
0.53	0.84	1.17	1.23	1.38	1.82	1.66	1.49	0.66	0.16
0.66	1.11	1.31	1.52	1.51	1.68	1.90	1.35	0.68	0.17
0.68	1.06	1.42	1.43	1.45	1.51	1.55	1.35	0.70	0.17
0.68	1.02	1.30	1.35	1.34	1.36	1.45	1.14	0.76	0.19
0.67	1.05	1.42	1.32	1.20	1.13	1.09	1.01	0.53	0.13
0.63	1.10	1.21	1.30	1.02	0.88	0.77	0.62	0.34	0.08
0.45	0.73	1.01	0.86	0.71	0.60	0.50	0.38	0.20	0.05
0.23	0.35	0.44	0.41	0.35	0.30	0.24	0.18	0.10	0.02
0.06	0.09	0.11	0.10	0.09	0.07	0.06	0.04	0.02	0.00

Figure 5.2: Demand distribution for 10x10 grid.

0.63	0.63	0.63	0.62	0.62	0.62	0.63	0.62	0.58	0.54
1.01	0.97	0.94	0.92	0.92	0.94	0.96	0.96	0.89	0.81
1.22	1.31	1.23	1.21	1.22	1.24	1.30	1.37	1.22	1.08
1.41	1.43	1.45	1.48	1.49	1.52	1.64	1.57	1.53	1.31
1.42	1.55	1.67	1.77	1.74	1.71	1.72	1.76	1.59	1.45
1.48	1.67	1.92	2.17	2.00	1.86	1.78	1.73	1.63	1.53
1.51	1.76	2.15	2.19	2.24	1.94	1.80	1.73	1.68	1.56
1.49	1.69	1.93	2.19	2.01	1.85	1.76	1.73	1.64	1.56
1.44	1.56	1.70	1.80	1.77	1.70	1.65	1.62	1.59	1.46
1.37	1.42	1.50	1.56	1.56	1.54	1.52	1.50	1.47	1.36

Figure 5.3: Demand distribution for 10x10a grid.

0.76	0.81	0.93	0.93	0.95	0.88	0.90	0.97	1.02	0.89	0.73	0.58
1.03	1.12	1.22	1.30	1.23	1.22	1.30	1.49	1.72	1.39	1.09	0.86
1.27	1.42	1.55	1.54	1.47	1.46	1.58	1.96	1.89	1.86	1.37	1.10
1.48	1.74	2.03	1.83	1.65	1.56	1.60	1.78	2.01	1.69	1.42	1.26
1.62	2.03	2.05	2.10	1.73	1.55	1.50	1.53	1.57	1.47	1.37	1.37
1.53	1.75	2.05	1.86	1.60	1.42	1.32	1.28	1.26	1.24	1.24	1.37
1.25	1.40	1.60	1.67	1.41	1.20	1.08	1.01	0.98	0.97	1.00	1.09
0.87	1.02	1.26	1.22	1.18	0.91	0.77	0.70	0.67	0.67	0.68	0.73
0.45	0.53	0.64	0.76	0.61	0.48	0.40	0.36	0.34	0.34	0.35	0.36
0.11	0.13	0.16	0.19	0.15	0.12	0.10	0.09	0.09	0.08	0.09	0.09

Figure 5.4: Demand distribution for 10x12 grid.

1.05	1.12	1.21	1.31	1.33	1.35	1.38	1.37	1.40	1.46	1.53	1.23
1.13	1.19	1.27	1.35	1.42	1.51	1.61	1.54	1.49	1.48	1.44	1.26
1.20	1.25	1.31	1.39	1.49	1.65	1.66	1.68	1.57	1.51	1.48	1.40
1.24	1.29	1.35	1.42	1.50	1.61	1.71	1.63	1.57	1.54	1.57	1.38
1.28	1.33	1.38	1.44	1.50	1.56	1.60	1.57	1.53	1.50	1.46	1.40
1.31	1.35	1.40	1.45	1.49	1.53	1.55	1.53	1.50	1.45	1.39	1.27
1.33	1.38	1.42	1.46	1.50	1.53	1.54	1.51	1.47	1.42	1.37	1.28
1.36	1.40	1.45	1.48	1.52	1.55	1.55	1.51	1.46	1.40	1.38	1.21
1.38	1.44	1.47	1.50	1.54	1.59	1.61	1.53	1.43	1.34	1.26	1.17
1.41	1.50	1.51	1.52	1.56	1.65	1.76	1.58	1.41	1.27	1.15	1.02
1.46	1.62	1.54	1.50	1.54	1.71	1.65	1.67	1.35	1.18	1.06	0.94
1.48	1.50	1.55	1.40	1.38	1.44	1.53	1.35	1.18	1.05	0.97	0.91

Figure 5.5: Demand distribution for 12x12 grid.

0.76	0.81	0.93	0.93	0.95	0.88	0.90	0.97	1.02	0.89	0.73	0.58	0.45	0.31	0.16
1.03	1.12	1.22	1.30	1.23	1.22	1.30	1.49	1.72	1.39	1.09	0.86	0.67	0.48	0.25
1.27	1.42	1.55	1.54	1.47	1.46	1.58	1.96	1.89	1.86	1.37	1.10	0.90	0.68	0.37
1.48	1.74	2.03	1.83	1.65	1.56	1.60	1.78	2.01	1.69	1.42	1.26	1.15	0.99	0.54
1.62	2.03	2.05	2.10	1.73	1.55	1.50	1.53	1.57	1.47	1.37	1.37	1.46	1.57	0.81
1.53	1.75	2.05	1.86	1.60	1.42	1.32	1.28	1.26	1.24	1.24	1.37	1.78	1.48	1.14
1.25	1.40	1.60	1.67	1.41	1.20	1.08	1.01	0.98	0.97	1.00	1.09	1.27	1.44	0.75
0.87	1.02	1.26	1.22	1.18	0.91	0.77	0.70	0.67	0.67	0.68	0.73	0.77	0.74	0.42
0.45	0.53	0.64	0.76	0.61	0.48	0.40	0.36	0.34	0.34	0.35	0.36	0.36	0.32	0.19
0.11	0.13	0.16	0.19	0.15	0.12	0.10	0.09	0.09	0.08	0.09	0.09	0.09	0.08	0.05

Figure 5.6: Demand distribution for 10x15 grid.

0.96	0.92	0.88	0.84	0.80	0.78	0.75	0.72	0.68	0.67	0.64	0.61	0.55	0.49	0.42	0.39	0.37
0.95	0.89	0.83	0.79	0.77	0.76	0.76	0.74	0.73	0.72	0.71	0.68	0.63	0.56	0.50	0.45	0.41
0.93	0.85	0.77	0.71	0.72	0.75	0.78	0.79	0.79	0.80	0.80	0.78	0.71	0.63	0.56	0.50	0.45
0.92	0.82	0.70	0.57	0.66	0.75	0.81	0.84	0.87	0.89	0.91	0.93	0.81	0.70	0.60	0.54	0.49
0.93	0.82	0.62	0.61	0.60	0.77	0.86	0.91	0.95	0.98	1.04	0.98	0.91	0.74	0.62	0.56	0.51
0.98	0.89	0.78	0.66	0.77	0.87	0.95	1.00	1.03	1.05	1.05	1.04	0.89	0.73	0.60	0.55	0.52
1.04	0.99	0.93	0.90	0.94	1.00	1.06	1.11	1.13	1.12	1.08	1.01	0.87	0.69	0.50	0.51	0.53
1.12	1.09	1.07	1.06	1.09	1.13	1.18	1.23	1.27	1.22	1.15	1.05	0.89	0.65	0.57	0.48	0.55
1.20	1.19	1.19	1.20	1.22	1.26	1.31	1.37	1.37	1.35	1.25	1.14	1.00	0.83	0.65	0.65	0.64
1.28	1.29	1.30	1.32	1.34	1.38	1.41	1.45	1.48	1.42	1.35	1.26	1.15	1.03	0.91	0.83	0.76

Figure 5.7: Demand distribution for 10x17 grid.

0.19	0.23	0.27	0.33	0.39	0.47	0.55	0.63	0.63	0.63	0.62	0.62	0.62	0.63	0.62	0.58	0.54
0.27	0.33	0.40	0.48	0.57	0.70	0.85	1.01	0.97	0.94	0.92	0.92	0.94	0.96	0.96	0.89	0.81
0.36	0.43	0.51	0.61	0.73	0.89	1.13	1.22	1.31	1.23	1.21	1.22	1.24	1.30	1.37	1.22	1.08
0.44	0.52	0.61	0.71	0.84	1.00	1.20	1.41	1.43	1.45	1.48	1.49	1.52	1.64	1.57	1.53	1.31
0.52	0.60	0.69	0.80	0.93	1.08	1.25	1.42	1.55	1.67	1.77	1.74	1.71	1.72	1.76	1.59	1.45
0.60	0.67	0.76	0.86	0.99	1.13	1.30	1.48	1.67	1.92	2.17	2.00	1.86	1.78	1.73	1.63	1.53
0.67	0.73	0.81	0.90	1.02	1.17	1.33	1.51	1.76	2.15	2.19	2.24	1.94	1.80	1.73	1.68	1.56
0.74	0.77	0.84	0.93	1.04	1.18	1.33	1.49	1.69	1.93	2.19	2.01	1.85	1.76	1.73	1.64	1.56
0.78	0.79	0.84	0.92	1.03	1.18	1.34	1.44	1.56	1.70	1.80	1.77	1.70	1.65	1.62	1.59	1.46
0.73	0.75	0.81	0.89	1.00	1.15	1.25	1.37	1.42	1.50	1.56	1.56	1.54	1.52	1.50	1.47	1.36

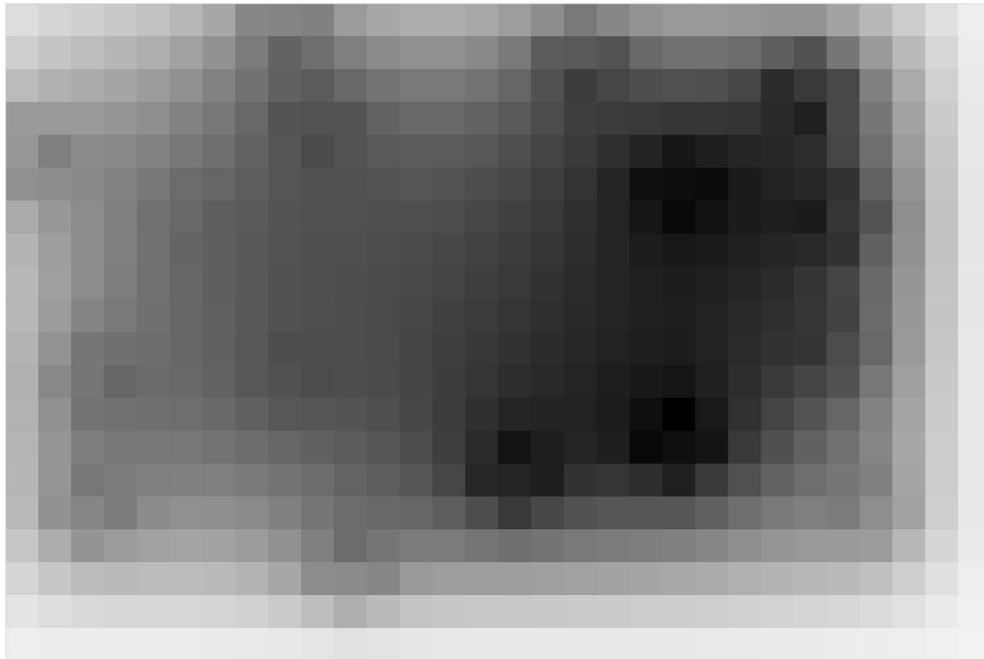
Figure 5.8: Demand distribution for 10x17a grid.

0.07	0.11	0.15	0.19	0.23	0.27	0.33	0.39	0.47	0.55	0.63	0.63	0.63	0.62	0.62	0.62	0.63	0.62	0.58	0.54
0.11	0.17	0.22	0.27	0.33	0.40	0.48	0.57	0.70	0.85	1.01	0.97	0.94	0.92	0.92	0.94	0.96	0.96	0.89	0.81
0.15	0.22	0.29	0.36	0.43	0.51	0.61	0.73	0.89	1.13	1.22	1.31	1.23	1.21	1.22	1.24	1.30	1.37	1.22	1.08
0.19	0.28	0.36	0.44	0.52	0.61	0.71	0.84	1.00	1.20	1.41	1.43	1.45	1.48	1.49	1.52	1.64	1.57	1.53	1.31
0.24	0.35	0.44	0.52	0.60	0.69	0.80	0.93	1.08	1.25	1.42	1.55	1.67	1.77	1.74	1.71	1.72	1.76	1.59	1.45
0.30	0.43	0.52	0.60	0.67	0.76	0.86	0.99	1.13	1.30	1.48	1.67	1.92	2.17	2.00	1.86	1.78	1.73	1.63	1.53
0.39	0.55	0.62	0.67	0.73	0.81	0.90	1.02	1.17	1.33	1.51	1.76	2.15	2.19	2.24	1.94	1.80	1.73	1.68	1.56
0.50	0.75	0.74	0.74	0.77	0.84	0.93	1.04	1.18	1.33	1.49	1.69	1.93	2.19	2.01	1.85	1.76	1.73	1.64	1.56
0.62	0.74	0.86	0.78	0.79	0.84	0.92	1.03	1.18	1.34	1.44	1.56	1.70	1.80	1.77	1.70	1.65	1.62	1.59	1.46
0.49	0.74	0.73	0.73	0.75	0.81	0.89	1.00	1.15	1.25	1.37	1.42	1.50	1.56	1.56	1.54	1.52	1.50	1.47	1.36

Figure 5.9: Demand distribution for 10x20 grid.

0.51	0.51	0.54	0.62	0.74	0.88	0.78	0.70	0.65	0.64	0.66	0.69	0.69	0.58	0.43
0.72	0.72	0.77	0.88	1.10	1.12	1.16	1.00	0.94	0.94	0.98	1.07	1.17	0.91	0.64
0.88	0.89	0.93	1.02	1.17	1.33	1.25	1.19	1.17	1.18	1.25	1.45	1.35	1.24	0.83
1.01	1.01	1.05	1.12	1.21	1.31	1.33	1.35	1.38	1.37	1.40	1.46	1.53	1.23	0.90
1.09	1.10	1.13	1.19	1.27	1.35	1.42	1.51	1.61	1.54	1.49	1.48	1.44	1.26	0.96
1.15	1.16	1.20	1.25	1.31	1.39	1.49	1.65	1.66	1.68	1.57	1.51	1.48	1.40	1.05
1.19	1.21	1.24	1.29	1.35	1.42	1.50	1.61	1.71	1.63	1.57	1.54	1.57	1.38	1.16
1.22	1.24	1.28	1.33	1.38	1.44	1.50	1.56	1.60	1.57	1.58	1.50	1.46	1.40	1.07
1.24	1.27	1.31	1.35	1.40	1.45	1.49	1.53	1.55	1.53	1.50	1.45	1.39	1.27	1.00
1.25	1.29	1.33	1.38	1.42	1.46	1.50	1.53	1.54	1.51	1.47	1.42	1.37	1.28	0.99
1.26	1.31	1.36	1.40	1.45	1.48	1.52	1.55	1.55	1.51	1.46	1.40	1.38	1.21	1.01
1.26	1.32	1.38	1.44	1.47	1.50	1.54	1.59	1.61	1.53	1.43	1.34	1.26	1.17	0.90
1.25	1.32	1.41	1.50	1.51	1.52	1.56	1.65	1.76	1.58	1.41	1.27	1.15	1.02	0.82
1.21	1.31	1.46	1.62	1.54	1.50	1.54	1.71	1.65	1.62	1.35	1.18	1.06	0.94	0.79
1.14	1.25	1.48	1.50	1.55	1.40	1.38	1.44	1.53	1.35	1.18	1.05	0.97	0.91	0.84

Figure 5.10: Demand distribution for 15x15 grid.



0.00 0.18 0.35 0.53 0.70 0.88 1.06 1.23 1.41 1.58 1.76

Figure 5.11: Demand distribution for 20x30 grid.

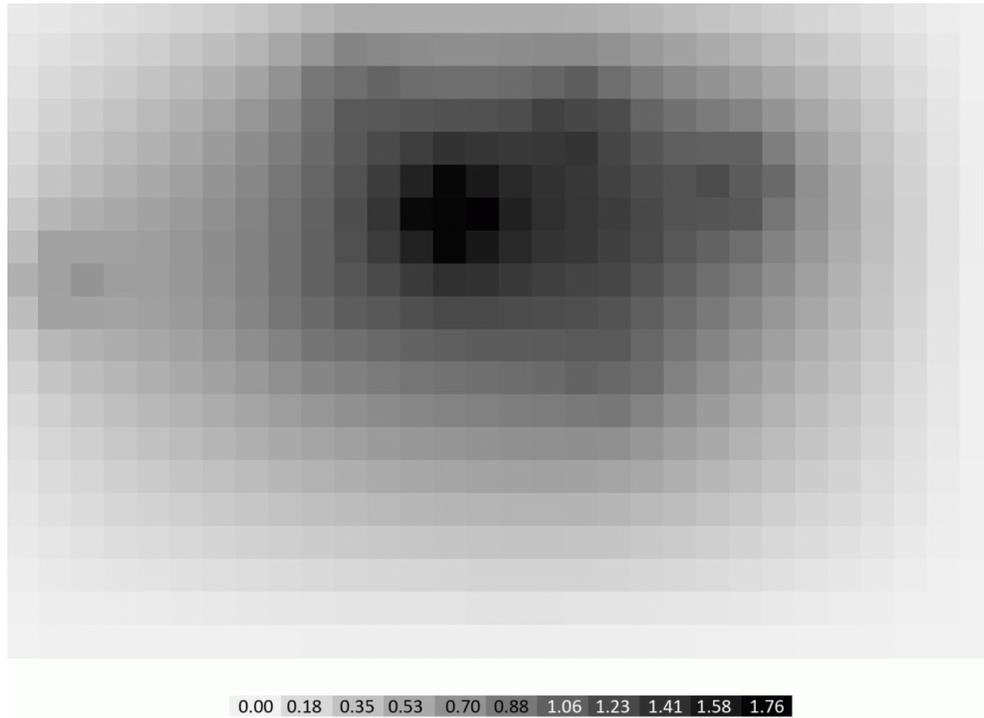


Figure 5.12: Demand distribution for 20x30a grid.

Table 5.1 shows the respective solution data for the above fixed cost ILP on these eleven test-case grids. In this table, various runtime statistics are indicated for the specified test-case grids (indicated in the first column) solved by the exact ILP method. The second and third columns show the *optimal objective function value* (OOFV) and associated optimum number of light sources, respectively. The other columns show the total number of simplex iterations, total number of branch-and-bound nodes, CPU time, and optimality gap. We see from this table that the developed fixed cost GBLP model can be solved for small and medium sized grids. For test-case grids outlined in this table, the exact method was able to reach the target optimality gap within a very short period of time. However, solution

time increases exponentially for the larger grids. The exact method was not able to reach the target optimality gap with more than a week of runtime for those larger grids; we terminated solution of the 20x30 and 20x30a grids after 865033 seconds (approximately 10 days) and 775886 seconds (approximately 9 days), respectively. For these two cases the CPLEX solver's branch-and-bound procedure made considerable progress early, with rapid improvements in the objective function values of the best-to-date branch-and-bound nodes. However, reductions in optimality gap slowed considerably, and even after many days of runtime with a very large number of simplex iterations (more than 1.5 billion) and the explosion of branch-and-bound nodes (more than 3 million), large optimality gaps remain (4.7% for 20x30 and 3.4% for 20x30a). While the smaller grids solved quickly, we feel that real-world GBLPs would be more representative of the larger grids. It is clear that these large grids with highly heterogeneous demand distributions will be intractable with the exact method. This drives our efforts to develop techniques to efficiently solve large-scale instances of this GBLP problem.

It is also worth mentioning here that the degree of the complexity of problem not only depends on the size of the grids, but also the degree of heterogeneity of the demand distributions. We see that for solving 10x17 grid, the solver takes 0.0625 second, whereas solving 10x17a takes 0.203125 second (three times greater than

the required CPU time of 10x17). This is because due to the heterogeneity of demand distribution, some instances of the problem might create much tighter LP relaxations than other instances, and/or algorithms used by CPLEX's internal branch-and-bound procedures might be better suited to some of those specific distributions. It is also noted that the OOFV and the total number of optimum lights are different for the same size grids. For example for 10x10, they are 81 and 5, where as for 10x10a, they are 113 and 7. This is because of the different demand distributions used in those grids (we know this because the demand distributions are the only differences between the two).

Table 5.1: Run time statistics for different test-case grids solved by the exact method.

Grids	OOFV	Optimum number of lights	# of MIP simplex iterations	# of branch and bound nodes	# of integer variables	# of constraints	CPU Time (Seconds)	MIP Gap
10x10	81	5	297	11	72	172	0.0469	0.001
10x10a	113	7	247	27	72	172	0.0469	0.001
10x12	126	7	364	66	96	216	0.0625	0.001
12x12	166	9	149	9	128	272	0.0625	0.001
10x15	138	8	1531	195	132	282	0.1719	0.001
10x17	137	8	234	21	156	326	0.0625	0.001
10x17a	166	10	2720	319	156	326	0.2031	0.001
10x20	177	11	324026	7405	192	392	14.578	0.001
10x20a	181	11	4310	602	192	392	0.3906	0.001
15x15	207	12	22607	2708	242	467	2.25	0.001
20x30	476	26	1629136831	3282912	832	1432	865033	0.047
20x30a	432	27	1548111145	3315281	832	1432	775886	0.034

5.4. Problem-Specific Decomposition Technique

In many hard ILP instances, computational complexity arises from a small subset of constraints or integrality requirements. In this case, relaxation-based decomposition techniques have been widely used. One of the popular techniques is Lagrangian relaxation that relaxes some “hard” constraints, and decomposes the original problem into two easy sub-problems. These sub-problems can be solved iteratively to obtain a near-optimal solution (Mauri et al., 2010). On the other hand, we can similarly decompose the original problem into two easy sub-problems by relaxing the integrality requirement of some decision variables (Wolsey, 1998). In this case, we first solve the partially relaxed problem using the exact method. The solution will provide us the values for the remaining integer decision variables, which we can fix in the original problem to produce a core problem. The core problem is solved to provide a near-optimal solution to the original. This technique is referred to as *relax-and-fix-based decomposition* (RFBD) technique in Noor-E-Alam and Doucette (2012). Another decomposition technique is proposed to solve large-scale ILP problems, in which the original problem is decomposed into a number of smaller sub-problems. Branch-and-bound is used to solve each of these sub-problems sequentially to obtain final solutions (Giortzis et al., 2000). The effectiveness of decomposition techniques are problem-specific, and depends on some specific structures that can be taken

advantage of. From our experiments described above and the investigation of the structure of our fixed cost GBLP model, we feel that the problem-specific decomposition techniques may be better used for solving large-scale instances. Therefore, we propose the following decomposition technique:

5.4.1. Partition-and-Fix-Based Decomposition (PFBD)

We have already seen above that the exact method takes a very long time to solve large-scale fixed cost GBLP instances. This drives our effort to develop a heuristic that could solve these instances quickly. From the above experiments, it is plausible to suggest that the optimal decision for a specific portion of the whole grid depends more on the demand distribution of that portion than the entire grid's demand distribution. This idea is also conceivable from the demand and supply constraints. We therefore propose to divide (by partitioning) the entire problem into a number of sub-problems, each representing a portion of the grid, so that the solver can consider each of these sub-problems independently. In this approach, first we solve all of these sub-problems by the exact method and obtain their solutions. The solution for a particular sub-problem is optimal only for that portion of the grid in isolation, but will still permit precise identification of light source location(s) and size(s) for that sub-problem. Furthermore, values of the decision variables within a partition may differ if they were obtained by solving sub-problems rather than the original (complete) problem. Therefore, the

optimum values of location decision variables, T_{xy} , throughout the partitions (α units) are not considered the optimum location of the GBLP as a whole. The remaining values of the location decision variables, T_{xy} , are fixed in the original problem to create a core problem, which is later solved to determine the complete solution for the entire grid. This method is referred to as *partition-and-fix-based decomposition* (PFBD). More precisely, the PFBD approach can be described as follows:

In order to decompose the entire grid, we need to determine the total number of sub-problems we may create. For instance, we are given two sets $V = \{1, 2, \dots, v_{max}\}$ and $W = \{1, 2, \dots, w_{max}\}$, where set V and W are the set of divisions of the grid in the horizontal and vertical directions, respectively. The maximum values that can be taken by V and W are v_{max} and w_{max} , respectively, which are the total number of divisions in the horizontal and vertical directions respectively. Therefore, the total number of sub-problems that can be created is $v_{max} \times w_{max}$. Any sub-problem created from these divisions is denoted as (v, w) , where $v \in V$ and $w \in W$. The coordinates of the cells at the four extreme corners are determined by the relationship outlined in equations (5.11)-(5.14), where L_x^v , L_y^w are the x and y coordinates of the cell located in the top left extreme corner, and U_x^v and U_y^w are the x and y coordinates of the cell located in the bottom right extreme corner. Figure 5.13 is used to describe this decomposition procedure briefly.

$$L_x^v = \left\lfloor (v-1) \frac{i_{max}}{v_{max}} + 1 \right\rfloor \quad (5.11)$$

$$L_y^w = \left\lfloor (w-1) \frac{j_{max}}{w_{max}} + 1 \right\rfloor \quad (5.12)$$

$$U_x^v = \left\lceil v \frac{i_{max}}{v_{max}} \right\rceil \quad (5.13)$$

$$U_y^w = \left\lceil w \frac{j_{max}}{w_{max}} \right\rceil \quad (5.14)$$

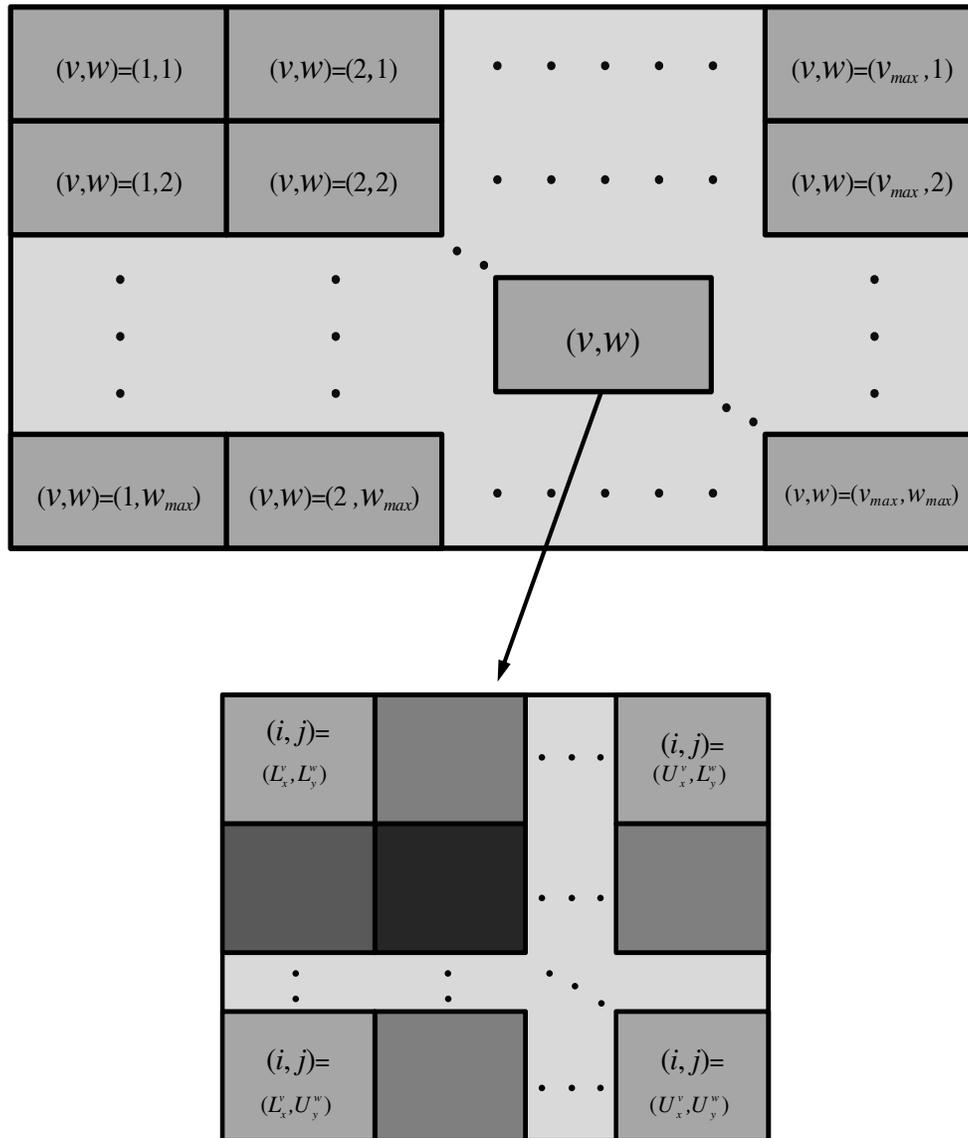


Figure 5.13: Illustration of the decomposition procedure.

Using this decomposition concept, we can create a sub-problem (v,w) with equations (5.15)-(5.23). The purpose of these equations is similar to that of equations (5.1)-(5.9) used in the original model, but they consider only a portion of the grid rather than the entire grid. All the sub-problems are then solved with

the exact method. We expect that the sub-problems will be comparatively easier to solve. By fixing the part of the solution of the location decision variable, T_{xy} , in the original problem, we create a core problem. This core problem is then solved with the exact method to find the entire solution. In the core problem, we fix most of the values of the T_{xy} variables. The resulting problem creates a very tight LP relaxation, and it becomes easier to solve. However, for some very large instances, even the core problem can possibly become intractable with the exact method. In such cases, we propose use of RFBD (we will explain this approach in a moment) to solve that core problem. Figure 5.14 illustrates this PFBD approach.

Sub-problem (v,w) :

$$\text{Minimize } \left\{ C_v \sum_{L_x^v}^{U_x^v} \sum_{L_y^w}^{U_y^w} P_{xy} + C_f \sum_{L_x^v}^{U_x^v} \sum_{L_y^w}^{U_y^w} T_{xy} \right\} \quad (5.15)$$

Subject to:

$$S_{ij} = \sum_{L_x^v}^{U_x^v} \sum_{L_y^w}^{U_y^w} \frac{P_{xy}}{r^2} \cos \left(\tan^{-1} \left(\frac{\sqrt{(i-x)^2 + (j-y)^2}}{r} \right) \right), \forall i \in Q \mid L_x^v \leq i \leq U_x^v, \forall j \in R \mid L_y^w \leq j \leq U_y^w \quad (5.16)$$

$$S_{ij} \geq D_{ij}, \quad \forall i \in Q \mid L_x^v \leq i \leq U_x^v, \forall j \in R \mid L_y^w \leq j \leq U_y^w \quad (5.17)$$

$$P_{xy} \leq MT_{xy}, \quad \forall x \in X \mid L_x^v \leq x \leq U_x^v, \forall y \in Y \mid L_y^w \leq y \leq U_y^w \quad (5.18)$$

$$0 \leq P_{xy} \leq UB_{xy}, \quad \forall x \in X \mid L_x^v \leq x \leq U_x^v, \forall y \in Y \mid L_y^w \leq y \leq U_y^w \quad (5.19)$$

$$P_{xy} = 0, \quad \forall x \in X \mid L_x^v \leq x \leq U_x^v, \forall y \in \{Y \mid L_y^w \leq y \leq U_y^w\} \cap \{Y \mid y \leq \beta\} \quad (5.20)$$

$$P_{xy} = 0, \quad \forall x \in X \mid L_x^v \leq x \leq U_x^v, \forall y \in \{Y \mid L_y^w \leq y \leq U_y^w\} \\ \cap \{Y \mid J_{max} - \beta \leq y \leq J_{max}\} \quad (5.21)$$

$$P_{xy} = 0, \quad \forall x \in \{X \mid L_x^v \leq x \leq U_x^v\} \cap \{X \mid x \leq \beta\}, \forall y \in Y \mid L_y^w \leq y \leq U_y^w \quad (5.22)$$

$$P_{xy} = 0, \quad \forall x \in \{X \mid L_x^v \leq x \leq U_x^v\} \cap \{X \mid I_{max} - \beta \leq x \leq I_{max}\}, \\ \forall y \in Y \mid L_y^w \leq y \leq U_y^w \quad (5.23)$$

We calculate the total time (CPU_t) required to solve a particular problem as the sum of the CPU runtimes of the individual sub-problems (CPU_{vw}) plus the CPU runtime of the core (or master) problem (CPU_m), as shown in Equation (5.24). These runtimes are the CPU runtimes reported by the CPLEX solver, not elapsed time (e.g., from a clock on the wall). We ignore the overhead to pre-process the sub-problems, to subsequently record their solutions, and then pass their solutions to the core problem, as this time (fractions of a second) is insignificant compared to the total CPU times reported.

$$CPU_t = \sum_{vw=1}^{V_m W_m} CPU_{vw} + CPU_m \quad (5.24)$$

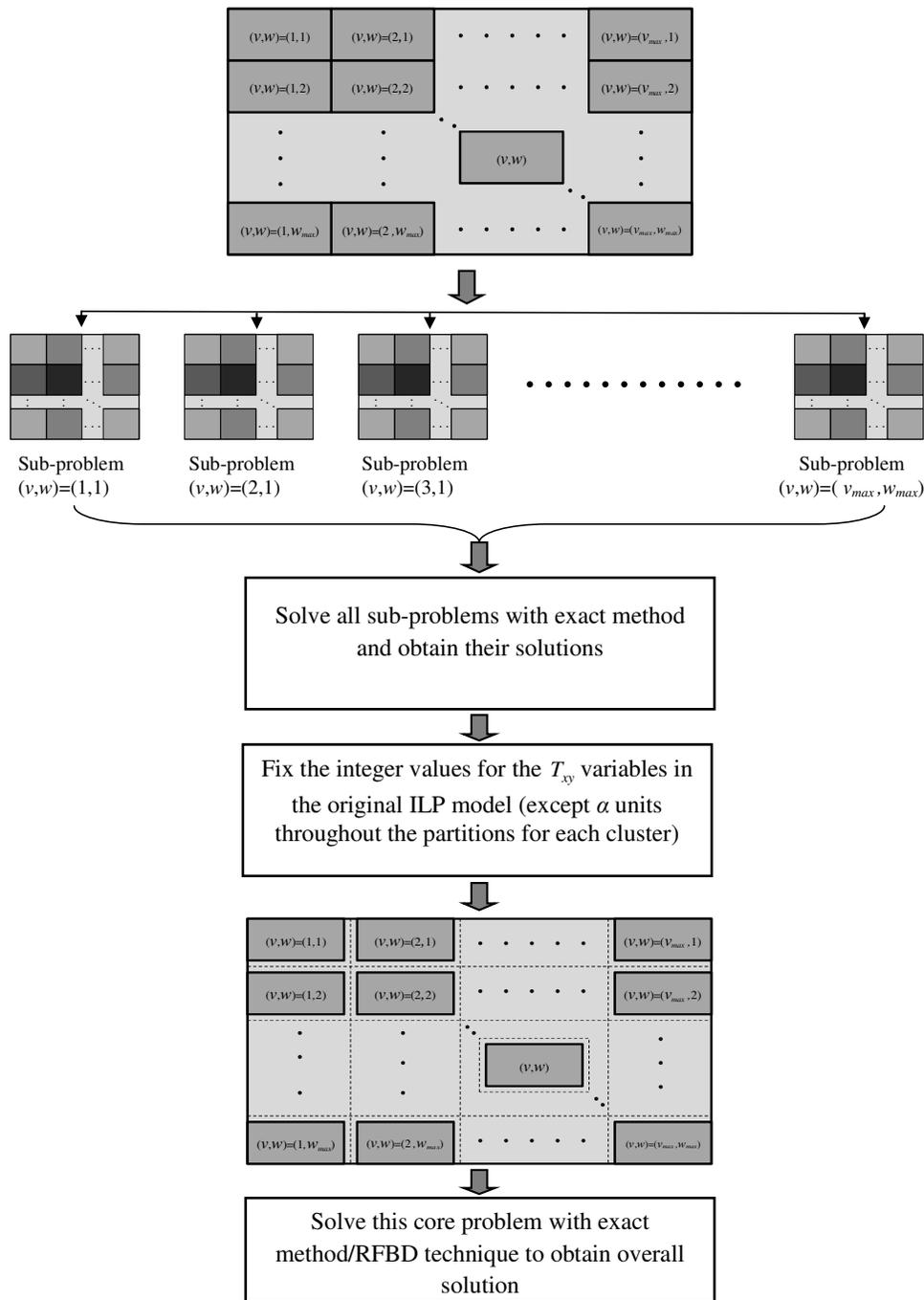


Figure 5.14: Illustration of the PFBD approach for fixed cost GBLP.

5.4.2. Relax-and-Fix-Based Decomposition (RFBD)

In the PFBD approach described above, we solve all sub-problems and the core problem by the exact method. However, in very large problems, even when the various sub-problems are easily solvable, the core problem can remain computationally difficult to solve. This is primarily due to the large number of unfixed T_{xy} variables that can remain throughout the partitions. Therefore, we further propose the use of the RFBD to solve these core problems. With the exact method, we solved our GBLP model to simultaneously determine the total number, location(s) and the size(s) of the supply sources. In RFBD, we are considering to make these decisions in two steps. In the first step, we determine the total number and location(s) of the sources. By relaxing the integrality requirement of the P_{xy} variables, we can obtain a sub-problem that will be easier to solve (there are fewer integer decision variables), but it will still permit precise identification of light source location(s). In the second step, the precise locations (i.e., T_{xy} decision variable values) obtained from the first sub-problem are fixed in the original problem, which is later solved to determine the size(s). This procedure is briefly illustrated in Figure 5.15. The RFBD approach does not guarantee optimality; however, it has been shown in the prior literature that it is able to provide high quality solutions in much less time compared to the exact method for large-scale instances (Noor-E-Alam and Doucette, 2012). We find that

the RFBD technique is not often effective in solving the original problem, as the first sub-problem takes a large amount of time to provide an optimal solution. On the other hand, the second sub-problem is very easy to solve since all of the T_{xy} variables in the original problem have been fixed (i.e., they are no longer decision variables, rather they have become parameters). As a result, the proposed RFBD approach would be solved much more efficiently if we could solve our first sub-problem more quickly by adding *logical restrictions* (LRs) to reduce the feasible region (Noor-E-Alam and Doucette, 2012). Unfortunately, it is very hard to devise such LRs for this problem due to the restrictions imposed by the existing constraints in the model. As such, we see from our preliminary experiments that the RFBD method takes a very long time to solve large-scale fixed cost GBLP instances. For instance, for a 20x30 test grid, it takes 2 days to reach optimality. However, we can integrate this approach with the PFBD to improve performance while solving very large problems. In that case, the core problem of the PFBD will be solved by the RFBD approach, instead of the exact method.

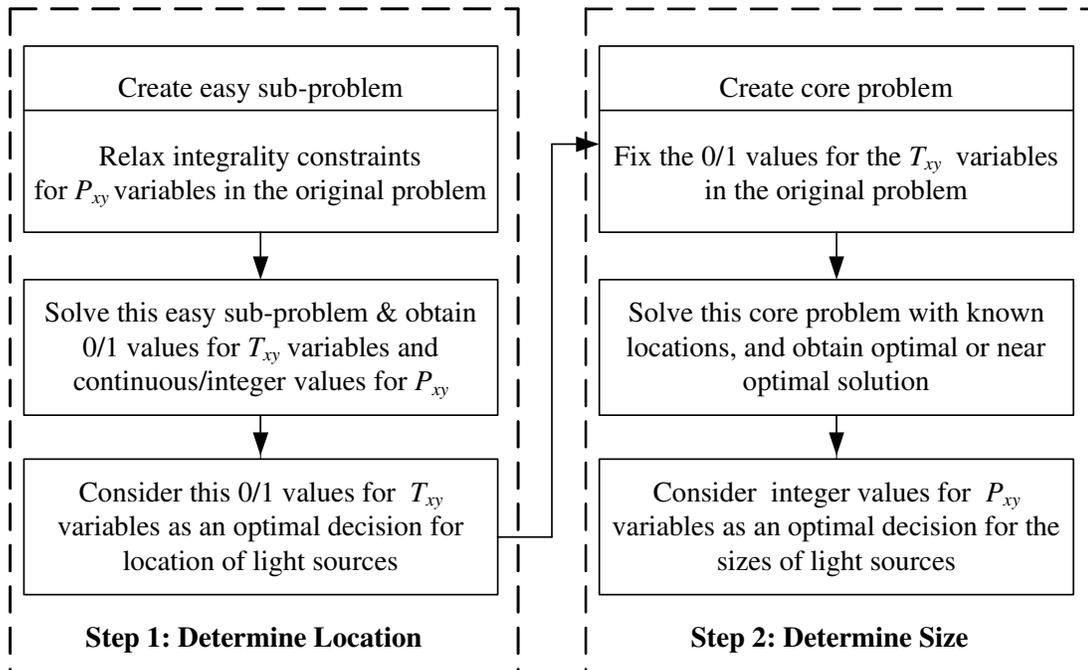


Figure 5.15: Illustration of the RFBD approach for fixed cost GBLP.

5.5. Result Analysis

We solve our ILP problems with the same experimental setup described at the beginning of Section 5.3. In addition to the grids used there, we also add much larger grids (30x40, 30x60, 40x60, 40x80, 40x100 and 50x100). Based on our light distribution model, in this experiment we have allowed α taking a value of 2 or 3. Our preliminary experiment suggests that if we choose α value less than 2, we may end up with a higher optimality gap. We have shown the demand distribution for only the first three grids in Figure 5.16 to Figure 5.18. We do not

show all the grids in this manuscript due to their large size, but will provide them as requested by any interested reader.

Table 5.2 shows the comparative solution data for the exact method and the PFBD approach on the seventeen test-case grids. The exact method refers to the benchmark solution where the original ILP is solved to optimality (with $mipgap = 0.001$). While we have already seen that the exact method takes a very long time (more than one week) to solve a 20x30 grid, it is conceivable that larger grids are almost nearly impossible to solve with the exact method. Therefore, we are not able to show the solution data for larger grids. We see that the solutions obtained from the PFBD approach are identical to the exact method for the most test-case grids, and nearly so for the larger cases where exact solutions are available. However, the PFBD approach takes fewer simplex iterations and branch-and-bound nodes in all cases. In the PFBD approach, the total number of simplex iterations and branch-and-bound nodes are the sum of the total number of simplex iterations and branch-and-bound nodes required to solve all sub-problems and the core problem. Each of the sub-problems becomes an easier ILP problem compared to the original problem. Therefore, they generally require a very small number of simplex iterations and branch-and-bound nodes to reach optimality. This illustrates the efficiency of our proposed PFBD approach with respect to the optimum solution. However, for some small grids, the CPU time is slightly higher

for the PFBD approach. This is because we have to solve several easy ILP problems in the PFBD approach, rather than a single already easy problem.

Not surprisingly, we see that solutions to large grids are obtainable with significant reduction in CPU time using the PFBD approach. PFBD achieved a 99.99% runtime reduction with only a 0.21% increase in objective function values for the 20x30 grid. For the 20x30a grid, PFBD reduced runtime by 99.98% and actually improved on the objective function value relative to the exact method (though we can note that those problems solution had been terminated prior to reaching optimality). Clearly, the PFBD approach significantly reduces the complexity of large-scale fixed cost GBLP, even with much larger test-case grids. The PFBD approach takes approximately 26 seconds to solve the 30x40 grid with only 160904 simplex iterations and 3662 branch-and-bound nodes. Similarly, the 40x80 grid was solved very quickly (only 11.421 seconds). Even the other very large grids are solved within several minutes. As we mentioned earlier, we generally solve all the PFBD sub-problems and core problem with the exact method. However, when solving some very large problems, the core problem may still become intractable. In those cases we take advantage of the RFBD approach to solve this core problem.

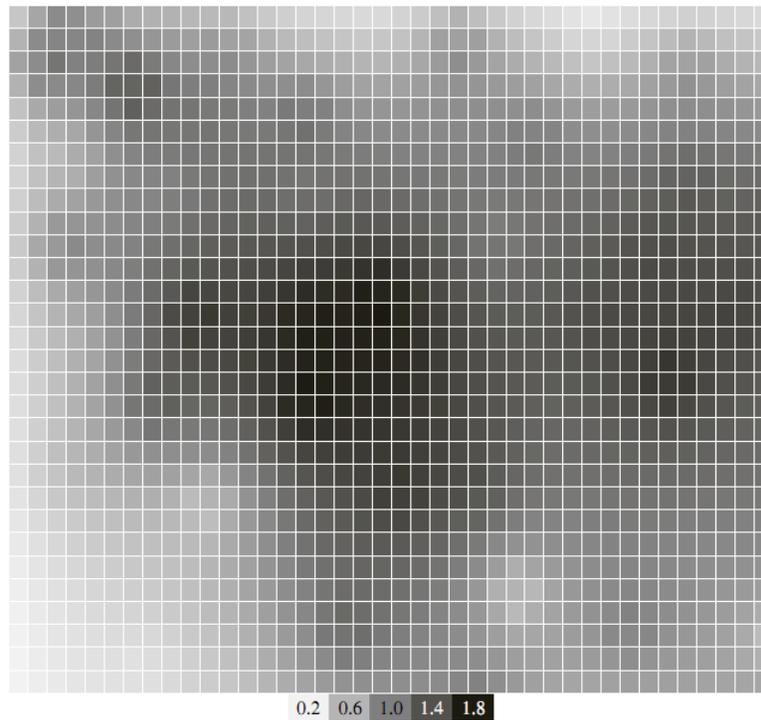


Figure 5.16: Demand distribution for 30x40 grid.

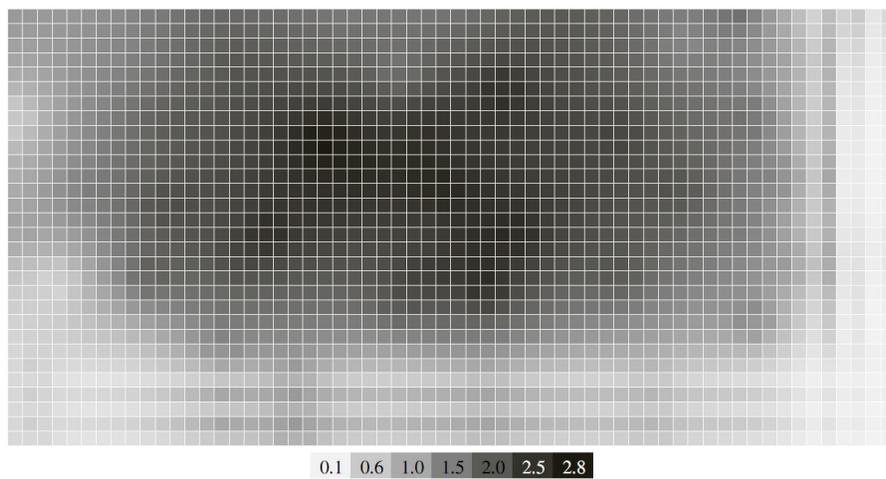


Figure 5.17: Demand distribution for 30x60 grid.

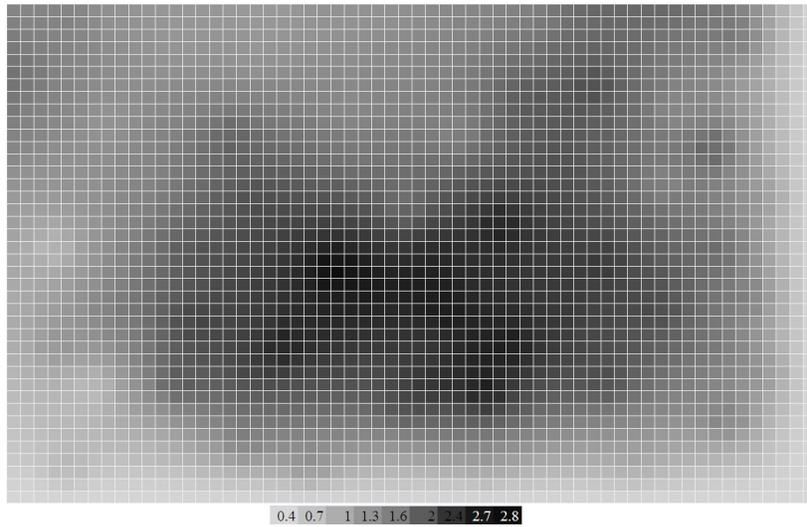


Figure 5.18: Demand distribution for 40x60 grid.

Table 5.2: Comparison of runtime statistics.*

Grids	OOFV		Optimum number of lights		# of MIP simplex iterations		# of branch and bound nodes		CPU Time (in Seconds)		
	Exact Method	PFBD	Exact Method	PFBD	Exact Method	PFBD	Exact Method	PFBD	Exact Method	PFBD	% Reduction
10x10	81	81	5	5	297	120	11	8	0.0469	0.062	-
10x10a	113	113	7	7	247	36	27	0	0.0469	0.048	-
10x12	126	126	7	7	364	201	66	1	0.0625	0.078	-
12x12	166	166	9	9	149	171	9	8	0.0625	0.109	-
10x15	138	138	8	8	1531	828	195	108	0.1719	0.156	9.25
10x17	137	137	8	8	234	34	21	0	0.0625	0.062	0.80
10x17a	166	166	10	10	2720	376	319	4	0.2031	0.156	23.19
10x20	181	181	11	11	4310	828	602	134	0.3906	0.156	60.06
15x15	207	207	12	12	22607	3590	2708	295	2.25	0.454	79.25
20x30	476	477	26	27	1629136831	918567	3282912	25934	865033	85.11	99.99
20x30a	432	430	27	26	1548111145	1141495	3315281	11765	775886	129.8	99.98
30x40	-	951	-	81	-	160904	-	3662	-	25.99	-
30x60	-	1934	-	115	-	728176	-	31483	-	114.7	-
40x60	-	2541	-	149	-	1459780	-	83665	-	221.7	-
40x80	-	1906	-	128	-	32440	-	307	-	11.42	-
40x100	-	3830	-	223	-	1147153	-	38701	-	166.1	-
50x100	-	4441	-	263	-	1567747	-	44765	-	289.4	-

*For the 40x60, 40x80, 40x100 and 50x100 test-case grids, we used the RFBD approach to solve the PFBD core problem.

We can also now return to our earlier discussion, regarding the efficiency of the PFBD approach. Table 5.3 shows the details on the number of simplex iterations and branch-and-bound nodes for solutions using PFBD approach on the 40x60

grid. It is clear from the runtime data that most of the sub-problems appear to be exceedingly easy to solve, and they do not require exploration of a large number of branch-and-bound nodes and simplex iterations to reach optimality. As a result, CPLEX takes less than a minute to solve them individually and just a few seconds for most. Moreover, in very large grids, where we solve core problem with the RFBD approach, the sub-problems are also trivial to solve, and CPLEX is able to solve each of them in less than a minute. A similar phenomenon is also illustrated in Table 5.4, where we demonstrate the number of simplex iterations and branch-and-bound nodes for solutions using our PFBD approach on the 40x100 grid. The solution is obtained on this grid in only 3 minutes of CPU time. The significant reduction in CPU time was possible due to the triviality of the sub-problems in the PFBD approach. Likewise, in the 40x60 grid, PFBD produces a set of trivial sub-problems that can be solved with very few branch-and-bound nodes. For example, sub-problems 1, 3, 16 and 17 actually require no branch-and-bound nodes at all (i.e., their LP relaxations produce integer solutions). It is also noticeable here that some sub-problems take much less time than others to reach optimality. This is because some sub-problems create much tighter LP relaxations than others, and/or algorithms used by CPLEX's internal branch-and-bound procedures might be better suited to some of the specific sub-problems.

Table 5.3: Details of PFBD runtime statistics for 40x60 grid.

Problems	OOFV	Optimum number of lights	# of MIP simplex iterations	# of branch and bound nodes	CPU Time (in Seconds)
Sub-problem 1 ($x < 11$ and $y < 21$)	176	10	447	21	0.156
Sub-problem 2 ($x < 11$ & $20 < y < 41$)	192	11	6053	250	1.391
Sub-problem 3 ($x < 11$ & $y > 40$)	210	12	9167	324	1.109
Sub-problem 4 ($10 < x < 21$ and $y < 21$)	218	13	76212	3426	10.219
Sub-problem 5 ($10 < x < 21$ & $20 < y < 41$)	294	17	765158	40439	113.391
Sub-problem 6 ($10 < x < 21$ & $y > 40$)	220	13	48633	1722	7.312
Sub-problem 7 ($20 < x < 31$ and $y < 21$)	232	14	256116	22752	35.484
Sub-problem 8 ($20 < x < 31$ & $20 < y < 41$)	319	18	35322	1615	7.984
Sub-problem 9 ($20 < x < 31$ & $y > 40$)	239	14	67775	3119	12.047
Sub-problem 10 ($x > 30$ and $y < 21$)	158	10	3567	113	0.562
Sub-problem 11 ($x > 30$ & $20 < y < 41$)	222	13	176850	9254	19.719
Sub-problem 12 ($x > 30$ & $y > 40$)	165	10	3095	72	0.688
Master- problem: Step 1 ($\forall x \in X$ & $\forall y \in X$)	2489.4	149	11277	558	11.609
Master- problem: Step 2 ($\forall x \in X$ & $\forall y \in X$)	2541	149	108	0	0.031

Table 5.4: PFBD runtime statistics for 40x100 grid.

Problems	OOFV	Optimum number of lights	# of MIP simplex iterations	# of branch and bound nodes	CPU Time (in Seconds)
Sub-problem 1 ($x < 11$ and $y < 21$)	145	8	458	0	0.062
Sub-problem 2 ($x < 11$ & $20 < y < 41$)	146	8	1087	1	0.312
Sub-problem 3 ($x < 11$ & $40 < y < 61$)	141	8	524	0	0.109
Sub-problem 4 ($x < 11$ & $60 < y < 81$)	140	8	1492	36	0.5
Sub-problem 5 ($x < 11$ and $y > 80$)	196	11	91518	1317	11.031
Sub-problem 6 ($10 < x < 21$ & $y < 21$)	192	11	149391	3189	19.531
Sub-problem 7 ($10 < x < 21$ & $20 < y < 41$)	191	10	5527	134	1.875
Sub-problem 8 ($10 < x < 21$ & $40 < y < 61$)	180	10	4355	284	0.922
Sub-problem 9 ($10 < x < 21$ & $60 < y < 81$)	225	13	132918	2535	17.312
Sub-problem 10 ($10 < x < 21$ & $y > 80$)	201	12	63951	2125	9.375
Sub-problem 11 ($20 < x < 31$ & $y < 21$)	144	9	1527	50	0.484
Sub-problem 12 ($20 < x < 31$ & $20 < y < 41$)	157	9	1217	17	0.688
Sub-problem 13 ($20 < x < 31$ & $40 < y < 61$)	240	14	79098	1586	12.953
Sub-problem 14 ($20 < x < 31$ & $60 < y < 81$)	308	18	369261	21964	55.875
Sub-problem 15 ($20 < x < 31$ & $y > 80$)	212	13	81350	1622	10.094
Sub-problem 16 ($x > 30$ & $y < 21$)	144	8	517	0	0.062
Sub-problem 17 ($x > 30$ & $20 < y < 41$)	125	8	726	0	0.141
Sub-problem 18 ($x > 30$ & $40 < y < 61$)	244	14	14673	595	3.891
Sub-problem 19 ($x > 30$ & $60 < y < 81$)	316	18	8302	371	2.609
Sub-problem 20 ($x > 30$ & $y > 80$)	211	12	133843	2411	15.297
Master- problem: Step 1 ($\forall x \in X$ & $\forall y \in X$)	3738.7	223	5286	464	2.938
Master- problem: Step 2 ($\forall x \in X$ & $\forall y \in X$)	3830	223	132	0	0.031

5.6. Concluding Discussion

We designed an ILP model to provide the optimal solution for GBLPs considering fixed cost criteria. Our preliminary results show that the ILP model is quite efficient at solving small to moderate sized problems. However, this ILP model becomes intractable for large-scale instances. As such, we then developed a partition-and-fix-based decomposition approach to solve large-scale instances. We carried out performance tests of our problem-specific decomposition technique with a light post placement problem using several test-case grids and have shown that the PFBD technique is effective in reducing problem runtimes with minimal loss of optimality. To benchmark our proposed heuristic, we compared our results with the exact method, and show that the proposed method significantly outperforms the exact method with respect to the CPU time. While it is nearly impossible to solve large grids by the exact method, the introduction of PFBD significantly reduces the complexity of the large test-case grids, and the ILP model becomes much easier to solve.

In the future, we will develop a nested PFBD approach to solve extremely large fixed cost GBLPs, where we will use the PFBD approach to solve the sub-problems. We could also solve these sub-problems simultaneously with the help of parallel computing environment to minimize the total solution time of a large-scale instance. In this research, we have tested our PFBD approach on a GBLP

that seeks to optimally place light posts to minimize total cost and to satisfy demand in each cell of a grid. Other GBLPs could also be solved using this approach. Potential applications of this proposed PFBD approach include optimal placement of sensors in a wireless sensor networks, outlets and/or warehouses, resource exploitation, and even optimal delivery of radiation therapy in the field of medical science.

References

- Aardal, K., Pochet, Y., Wolsey L. (1995), “Capacitated facility location: valid inequalities and facets”, *Mathematics of Operations Research*, 20, 552–582.
- Aardal, K. (1998), “Reformulation of capacitated facility location problems: How redundant information can help”, *Annals of Operations Research*, 82, 289-308.
- Aardal K. (1998), “Capacitated facility location: separation algorithm and computational experience”, *Mathematical Programming*, 81, 149–175.
- Amor, B., Desrosiers, Valero de Carvalho, J. M. (2006), “Dual-optimal inequalities for stabilized column generation”, *Operations Research*, 54 (3), 454–463.
- Anderson, E. J., Ferris, M.C. (1994), “Genetic algorithms for combinatorial optimization: The assembly line balancing problem”, *ORSA Journal on Computing*, 6 (2),161–173.
- Avella, P. and Sassano, A. (2001), “On the p-median polytope”, *Mathematical Programming*, 89, 395-411.
- Aytug, H., Saydam, C. (2002), “Solving large-scale maximum expected covering location problems by genetic algorithms: A comparative study”, *European Journal of Operational Research*, 141, 480–494.

- Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.F., Vance, P.H. (1998), “Branch-and-Price: Column Generation for Solving Huge Integer Programs”, *Operations Research*, 46 (3) , 316-329.
- Barnhart, C., Hane, C.A., Vance, P.H. (2000), “Using Branch-and-Price-and-Cut to Solve Origin-Destination Integer Multi commodity Flow Problems”, *Operations Research*, 48 (2), 318-326.
- Belov, G., Scheithauer, G. (2006), “A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting”, *European Journal of Operational Research*, 171, 85–106.
- Beraldi, P., Ghiani, G., Grieco, A. Guerriero, E. (2006), “Fix and Relax Heuristic for a Stochastic Lot-Sizing Problem”, *Computational Optimization and Applications*”, 33, 303–318.
- Chen, D.-S., Baton, R. G., Dang, Y. (2010), *Applied Integer Programming*, A John Willey & Sons, Inc., Publication, (Chapter 2).
- Cheng, C.-P. Liu, C.-W., Liu, C.-C. (2000), “Unit Commitment by Lagrangian Relaxation and Genetic Algorithms”, *IEEE Transactions on Power Systems*, 15 (2), 707-714.
- Contreras, I., Cordeau, J., Laporte, G. (2011), “Benders decomposition for large-scale uncapacitated hub location”, *Operations Research*, 59(6), 1477-1490.
- Ferreira, D., Morabito, R., Rangel, S. (2010), “Relax and fix heuristics to solve one-stage one-machine lot-scheduling models for small-scale soft drink plants”, *Computers & Operations Research*, 37, 684 – 691.
- Fourer, R., Gay, D. M. and Kernighan, B. W. (2002), *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press, Cole Publishing Co.
- García, S., Landete, M., Marín, A. (2012), “New formulation and a branch-and-cut algorithm for the multiple allocation p-hub median problem”, *European Journal of Operational Research*, 220, 48–57.

- Ghiani, G., Grandinetti, L., Guerriero, F., Musmanno, R. (2002), “A Lagrangean Heuristic for the Plant Location Problem with Multiple Facilities in the Same Site”, *Optimization Methods and Software*, 17,1059–1076.
- Giortzis, A. I., Turner, L.F. and Barria, J. A. (2000), “Decomposition technique for fixed channel assignment problems in mobile radio networks”, *IEE Proc., Commun*, 147(3), 187–194 .
- ILOG, CPLEX 11.0 User’s Manual, ILOG, Inc. 2007.
- Kelly, J. D., Mann, J. L. (2004), “Flow sheet decomposition heuristic for scheduling: a relax-and-fix method, *Computers and Chemical Engineering*”, 28, 2193–2200.
- Klose, A., Görtz, S. (2007), “A branch-and-price algorithm for the capacitated facility location problem”, *European Journal of Operational Research*, 179, 1109–1125.
- Lee, J. M., Lee, Y. H. (2012), “Facility location and scale decision problem with customer preference”, *Computers and Industrial Engineering*, 63, 184–191.
- Lee, S.-D., Chang, W.-T. (2007), “On solving the discrete location problems when the facilities are prone to failure”, *Applied Mathematical Modeling*, 31, 817–831.
- Leung, J. M. Y., Magnanti, T. L. (1989), “Valid inequalities and facets of the capacitated plant location problem. *Mathematical Programming*”, 44, 271–291.
- Lorena, L.A.N., Senne, E.L.F. (2004), “A column generation approach to capacitated p-median”, *Computers & Operations Research*, 31, 863–876.
- Mauri, G. R., Ribeiro G. M., Lorena, L. A. N. (2010), “A new mathematical model and a Lagrangian decomposition for the point-feature cartographic label placement problem”, *Computers & Operations Research*, 37, 2164–2172.
- Magnanti, T. L., Wong, R. T. (1981), “Accelerating Benders decomposition: algorithmic enhancement and model selection criteria”, *Operations Research*, 29, 464–484.

- Mohammadi, M., Ghomi, S. M. T. F., Karimi, B, Torabi, S. A. (2010), “Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lot sizing problem with sequence-dependent setups”, *Journal of Intelligent Manufacturing*, 21, 501–510.
- Noor-E-Alam, M., Ma, A., Doucette, J. (2012), “Integer Linear Programming Models for Grid-Based Light Post Location Problem”, *European Journal of Operational Research*, 222, 17-30.
- Noor-E-Alam, M., Doucette, J. (2012), “Relax-and-Fix-Based Decomposition Technique for Solving Large Scale GBLPs”, *Computers and Industrial Engineering*, 63, 1062-1073.
- Rajagopalan S., Heragu S. S., Taylor, G. D. (2004), “A Lagrangian relaxation approach to solving the integrated pick-up/drop-off point and AGV flow path design problem”, *Applied Mathematical Modelling*, 28, 735–750.
- Roy, T. J. V., Wolsey, L.A. (1987), “Solving mixed integer programming using automatic reformulation”, *Operations Research*, 35, 45 – 57.
- Savelsbergh, M.W.P., Sigismondi, G.C., Nemhauser, G.L. (1995), “Functional description of MINTO, a Mixed INTEger Optimizer”, *Operations Research Letters*, 15, 47 – 58.
- Senne, E.L.F., Lorena, L.A.N., Pereira, M.A. (2005), “A branch-and-price approach to p-median location problems”, *Computers & Operations Research*, 32, 1655–1664.
- Sonmez, A. D., Lim, G. J. (2012), “A decomposition approach for facility location and relocation problem with uncertain number of future facilities”, *European Journal of Operational Research*, 218, 327–338.
- Sun, M. (2012), “A tabu search heuristic procedure for the capacitated facility location problem”, *J. Heuristics*, 18, 91–118.
- Trick, M. A. (2005), “Formulations and Reformulations in Integer Programming”, Second International Conference on Integration of Artificial Intelligence and Operations Research, CPAIOR, Prague, Czech Republic, May 30 - June 1, 366-379.

- Valero de Carvalho, J. M. (2005), “Using extra dual cuts to accelerate convergence in column generation”, *INFORMS Journal of Computing*, 17 (2), 75–182.
- Wentges, P. (1996), “Accelerating Benders’ decomposition for the capacitated facility location problem”, *Mathematical Methods of Operations Research*, 44, 267–290.
- Williams, H. P. (1978), “The Reformulation of Two Mixed Integer Programming Problems”, *Mathematical Programming*, 14, 325-331.
- Wolsey, L. A. (1998), *Integer Programming*, Wiley-Interscience, Series in Discrete Mathematics and Optimization, Toronto, (Chapter 1).
- Yamin, H.Y., Shahidehpour, S. M. (2004), Unit commitment using a hybrid model between Lagrangian relaxation and genetic algorithm in competitive electricity markets, *Electric Power Systems Research*, 68, 83-92.
- You, B., Yamada, T. (2011), “An exact algorithm for the budget-constrained multiple knapsack problem”, *International Journal of Computer Mathematics*, 88(16), 3380-3393.

Chapter 6⁴

ILP Model for Wireless Transmitter Location Problem

6.1. Introduction

Communication industries have experienced a paradigm shift, where an exponential increasing trend in the use of wireless communication is observed in nearly every aspect of our lives, ranging from entertainment and personal communication to business transactions and on-line commerce. Wide area applications of sensor networks such as defense, air traffic control, industrial and

⁴ A version of this chapter has been submitted for publication: Noor-E-Alam, M., Todd, B. and Doucette, J. (2012) “Integer Linear Programming Model for Grid-Based Wireless Transmitter Location Problems”, *International Journal of Operational Research*, in review, submitted 10 October 2012.

manufacturing automation, distributed robotics, etc., are also on the rise (Chong and Kumar, 2003). All of these activities assume the underlying wireless network is secure and highly reliable. It is therefore imperative that the system should be specially engineered for extremely high performance service with minimum cost. Among all the components involved in wireless communication systems, transmitters are one of the key expensive components. Optimum location of these transmitters plays a vital role in successful operation of these systems. To make this communication effective and efficient, we have to place transmitters in a way that could provide reliable service with minimum cost. However, properly accomplishing this becomes a very difficult and computationally complex task when real-world considerations such as variation in signal strength due to distance and the propagation environment (different degrees of obstruction) are taken into account. Therefore, the objective of this research is to develop effective and efficient methodologies to design an optimum transmitter location strategy for a wireless network.

A wide variety of research interest has been attracted to the wireless sensor network localization problem due to the proliferation of wireless sensor network applications (Mao et al., 2007). Cao et al (2006) studied the localization problem with imprecise distance information in sensor networks and introduced a set of equality constraints from the geometric relations among distances between nodes.

Xiao et al. (2011) developed a *total least squares* (TLSs) algorithm for location estimation of a stationary source, whereas a distributed algorithm is proposed by Gentile (2007) to determine the locations of sensors in a network. Guney et al. (2010) described mixed-integer linear programming models to determine the optimal sink locations and information flow paths between sensors and sinks. In this research they assumed that the sensor locations are known. Jayabharathy et al. (2012) proposed a hybrid indoor wireless location method with unconstrained optimization technique.

Determining the optimum location of a transmitter is greatly affected by the path loss incurred due to the propagation environment and environmental fading. Sharma et al. (2010) has carried out a brief survey in different models to calculate path loss in various types of environments. The transmitted signal strength decays exponentially with distance and the degree of the obstacles that surround or interject between the transmitter and the receiver (Rodas and Escudero, 2010). The sensors may not be able to communicate through large distances because the transmission range of sensors is limited as a consequence of their energy and size limitations (Guney et al., 2012). For a summary of outdoor path loss models, readers are referred to Durgin et al. (1998).

The optimum location of a transmitter significantly influences smooth operation of a wireless system. A new adaptive clustering algorithm has been proposed for

energy efficiency wireless sensor network and it was found from the simulation results that the proposed adaptive clustering algorithm is efficient and effective for energy saving in wireless sensor networks (Ci et al., 2007). Ai et al. (2006) proposed an ILP model for the *maximum coverage with minimum sensors* (MCMS) problem that maximized the number of targets covered, while minimizing the number of sensors that were required. An approximate algorithm was proposed to solve the location-selection problem of wireless network (Lu and Zhang, 2011). Coluccia and Altman (2012) described a hierarchical decision making problem of base station (BS) placement in a wireless communication system. Marianov and Eiselt (2012) proposed a model for the optimum transmitter location of digital television that maximize coverage and minimize interfering signal reception.

From this above discussion, we find that that little effort has been made to determine the optimum location of transmitters, where the variation of signal strengths due to distance, propagation environment (different degrees of obstructions) and installation costs are taken into account. As such, this research aims to develop mathematical model to find optimum transmitters locations for a wireless communication system. It considers this decision problem as a *grid-based location problem* (GBLP), where the entire location area is approximated by a grid-based system of small-sized cells (Noor-E-Alam et al., 2012). These

cells can then be used to establish the signal strength's variation to model the decision problem. Based on this relationship, our goal was to develop an ILP model that was designed to provide the optimal solution for the transmitter location problem: the total number of transmitters, the location of each transmitter, and their capacities (i.e., signal strength).

From our experiments with GBLPs we see that the ILP instances become very hard to solve due to the combination of all the above considerations. In real-world GBLPs, computational complexity often arises from a small subset of constraints or integrality requirements. In such cases, relaxation-based decomposition techniques have been found to be very successful for solving hard instances. The *Lagrangian decomposition* (LD) technique is one of the most popular techniques, it relaxes some hard constraints, and decomposes the original problem into two easy sub-problems (Mauri et al., 2010). We can similarly decompose the original problem into two easy sub-problems by relaxing the integrality requirements of some decision variables (Wolsey, 1998). At first exact method solves the partially relaxed problem and provided us the values for the remaining integer decision variables, which we fixed in the original problem to produce a core problem. The core problem was solved to provide an optimal/near-optimal solution to the original. This technique is referred to as relax-and-fix-based decomposition (RFBD) technique in Noor-E-Alam and Doucette (2012).

On the other hand, the original ILP problem can be decomposed into a number of smaller sub-problems that are solved by the branch-and-bound technique sequentially to obtain a final solution (Giortzis et al., 2000). The effectiveness of these decomposition techniques are problem-specific and fully depended on some specific properties of the ILP models that can be taken advantage of. In prior literature it has been shown that the RFBD technique is quite successful in solving large-scale GBLPs. Therefore, in this research, we designed a problem-specific RFBD approach to solve such instances efficiently. To facilitate the solution process further, we integrated valid inequalities and logical restrictions in the first sub-problem. Finally, the ILP model and the RFBD approach were implemented within a standard modeling language and tested on a number of large test-case grids to compare the performance of the proposed technique with the exact method.

The remainder of this chapter is organized as follows. Section 6.2 provides the description of the problem and signal strength variation. In Section 6.3, we discuss ILP model. In Section 6.4, preliminary results are described. We propose a problem-specific RFBD technique in Section 6.5 to solve large-scale instances. Section 6.6 describes the results analysis. Finally, conclusions and future research opportunities are briefly described in Section 6.7.

6.2. Problem Description

The optimum location of transmitters plays a key role in successful operation of a wireless communication system. In this research, our objective was to develop methods to determine the optimum location of transmitters, considering cost, propagation loss, and other signal fading factors. Suppose we are considering a geographical location (that includes different degrees of obstructions, such as trees, building etc.), described as a 2-dimensional grid of known dimensions. The transmitters must be installed throughout this location to ensure reliable communication. The total number of transmitters must be determined, along with the location of each transmitter and their capacities to minimize cost. The greater the power of the transmitter, the more expensive it will be. There are many factors affecting this decision such as distance, the propagation environment, and different degrees of obstructions etc. The amount of path loss experienced by the signal strength depends on the degree of obstruction and the distance the radio signal travels to reach the receiver. As such, the objective is to locate the transmitter in such a way that reliable communication is ensured with minimum cost. This transmitter location problem can be represented by a grid-based area, where the variation of signal strength can be represented by each cell in the grid using the relationships described in the following sub-sections.

6.2.1. Signal Strength Variation

6.2.1.1. Log-distance Path Loss Model

The log-distance path loss model is extensively used in the literature, where this model indicates that average received signal power decreases logarithmically with distance (Rappaport, 1996). The average large-scale path loss $\overline{PL}(d)$ for an arbitrary T-R separation distance d can be expressed by following equation (6.1) or equation (6.2), where d_o is the close-in reference distance (close to the transmitter) and ξ is the attenuation factor that is the rate at which the path loss increases with distance.

$$\overline{PL}(d) \propto \left(\frac{d}{d_o} \right)^\xi \quad (6.1)$$

$$\overline{PL}(\text{dB}) = \overline{PL}(d_o) + 10\xi \log \left(\frac{d}{d_o} \right) \quad (6.2)$$

All the path losses in equation (6.2) are measured in decibels (dB). It is important to note that the value of ξ depends on the specific propagation environment and when there are obstructions, ξ will have a large value. The reference path loss $\overline{PL}(d_o)$ can be calculated through field measurements or by using equation (6.3), where C is a system loss constant. The reference distance is selected based on the

propagation environment. For example, for microcellular systems, 100 m or 1 m distances are used in the literatures.

$$\overline{PL}(d_o) = 10\xi \log(d_o) + C \quad (6.3)$$

6.2.1.2. Log-normal Shadowing Model

Equation (6.2) does not take into account fading due to the environment, which affects signal and leads to measured signal strengths very different from the calculated signal strengths obtained by equation (6.2). Researchers have shown that the measured path loss at any distance d is random and log-normally distributed about the mean distance-dependent value, which gives us equation (6.4):

$$PL(d)[\text{dB}] = \overline{PL}(d) + N_\sigma = \overline{PL}(d_o) + 10\xi \log\left(\frac{d}{d_o}\right) + N_\sigma \quad (6.4)$$

In this model, N_σ is a zero-mean Gaussian distributed random variable with standard deviation σ , where the random variable and the standard deviation are measured in dB.

Finally, the total received power at the distance d between two communication nodes can be expressed as in equation (6.5) (Zhang et al., 2011), where P_o is the received power at the reference distance d_o :

$$P_r(d)[\text{dBm}] = P_o - 10\xi \log\left(\frac{d}{d_o}\right) - N_\sigma, \quad d \geq d_o \quad (6.5)$$

The total received power at the reference distance d_o can be calculated by equation (6.6), where P_{xy} is the power of sensor located at (x,y) .

$$P_o = P_{xy} - \overline{PL}(d_o) \quad (6.6)$$

In this chapter, the system loss constant is assumed to be 0, and d_o is assumed to be 1 m. Using these assumptions in equation (6.3), we find that there is no path loss at the reference distance. Therefore equation (6.5) becomes equation (6.7).

$$P_r(d)[\text{dBm}] = P_{xy} - 10\xi \log(d) - N_\sigma, \quad d \geq d_o \quad (6.7)$$

6.2.2. Attenuation calculation process

In order to calculate attenuation between the transmitter and receiver with the presence of obstacles in the grid a free space attenuation factor, ξ , of 2 was used. Each cell has an interference factor that ranges between 0 and 10, representing a unit less degree with obstacles within that square affecting signal propagation. In order to reduce the computational complexity of the problem, interference between all cells in the grid were calculated assuming negligible multi-path effects. To illustrate how attenuations were calculated, a 4x4 grid is given in

Figure 6.1 with three served regions (S_1 - S_3), a transmitter (T), and an obstacle (O). The obstacle had an interference rating of 8.

To evaluate whether or not an obstacle affects a given path, the perpendicular distance between the center of the square which contains the obstacle and the straight line path between the centers of the squares serving as endpoints was evaluated. The grid was assumed to have a unit measurement of one (i.e., the distance from the squares indexed at 1,1 to 2,1 is one unit). If the perpendicular distance was found to be less than 0.5, then the obstacle was considered to be interfering. In the cases of T to S_1 and T to S_2 , the obstacle would factor into the interference calculations, but would not factor into the calculations for the path from T to S_3 .

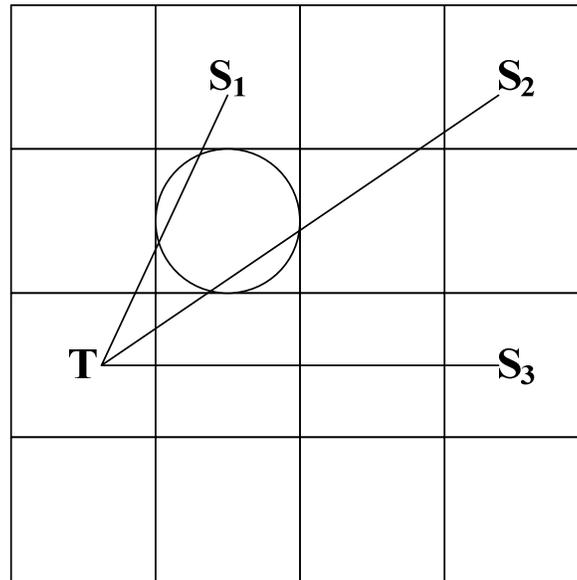


Figure 6.1: A sample 4x4 grid highlighting one transmitter (T), an obstacle (O), and three served regions (S₁-S₃).

If the source and destination were the same square, the interference was assumed to be zero, otherwise it was the greater of 2 and the largest obstacle that was interfering with the path between two points. In the example give, path (T,S₃) had an interference rating of 2, while (T,S₁) and (T,S₂) had an interference rating of 8. These interference ratings were considered as different levels of obstruction and translated into attenuation factors according to the schedule outlined in Table 6.1.

Table 6.1 : Translation of interference ratings to attenuation factors.

Interference Range	Level of Obstruction	Attenuation Factor
(0,2)	No obstruction	2
[2,4)	Low obstruction	4
[4,8)	Medium obstruction	6
[8,10)	High obstruction	8

6.3. ILP Model

We assume that each cell within the grid is uniform throughout the entire cell. In other words, the amount of signal strength at one point in the cell is the same as in all other points in the cell. In order to formulate our ILP model, we define the notation we will use, as follows:

Input Parameters:

D is the demand at each cell

Q is the set of all x-coordinates in the grid, indexed by i

R is the set of all y-coordinates in the grid, indexed by j

X is the set of all x-coordinates of the transmitter, indexed by x

Y is the set of all y-coordinates of the transmitter, indexed by y

UB_{xy} is the upper bound on decision variable P_{xy}

C_f is the fixed cost to install a transmitter

ξ_{xy}^{ij} is the attenuation factor between source (x, y) to destination (i, j)

d_c is the horizontal/vertical distance of the centres of adjacent cells

Decision Variables:

S_{ij} is the supply at location (i,j) , where $S_{ij} \geq 0$

P_{xy} is the integer size of the capacity of a sensor at location (x,y) , where

$P_{xy} \geq 0$

$T_{xy} \in \{0,1\}$: T_{xy} is 1 if $P_{xy} > 0$; T_{xy} is 0 if $P_{xy} = 0$

$\eta_{xy}^{ij} \in \{0,1\}$: η_{xy}^{ij} is 1 if transmitter at location (x,y) covers cell (i,j) , 0

otherwise

Finally, the proposed GBLP ILP model can be formulated as follows:

$$\text{Minimize } \left\{ \sum_{x \in X} \sum_{y \in Y} P_{xy} + C_f \left(\sum_{x \in X} \sum_{y \in Y} T_{xy} \right) \right\} \quad (6.8)$$

Subject to:

$$S_{ij} = \underset{\forall xy}{Max} \left\{ P_{xy} - 10 \xi_{xy}^{ij} \log \left(d_c \left(\sqrt{(i-x)^2 + (j-y)^2} \right) \right) - 3\sigma \right\} \quad (6.9)$$

$$\forall i \in Q, \forall j \in R$$

$$S_{ij} \geq D \quad \forall i \in Q, \forall j \in R \quad (6.10)$$

$$P_{xy} \leq M_1 \times T_{xy} \quad \forall x \in X, \forall y \in Y \quad (6.11)$$

$$0 \leq P_{xy} \leq UB_{xy} \quad \forall x \in X, \forall y \in Y \quad (6.12)$$

$$T_{xy} \in \{0,1\} \quad \forall x \in X, \forall y \in Y \quad (6.13)$$

As this model considers cost criteria, our objective is to minimize the sum of total variable cost and total fixed cost required to fulfill the demands throughout the grid. To fulfill this objective, equation (6.8) is used to minimize the total cost, where we are considering per unit variable cost is 1. On the basis of our description of signal strength variation in Section 6.2, equation (6.9) is used to calculate the total supply available in each cell (i,j) . In this model, we minimize the total cost while satisfying the demand constraint described in equation (6.10), which confirms that total supply in each cell should be greater or equal to the demand for each cell. Constraint (6.11) confirms that if $P_{xy} > 0$, then $T_{xy} = 1$ or 0, otherwise. In this equation M_1 is a sufficiently large number to satisfy this restriction. Equations (6.12) and (6.13) are required to put bounds on decision variables.

Our objective of this research is to develop ILP model to solve this decision problem optimally. However, we see that equation (6.9) is not a linear equation. Therefore we replaced equation (6.9) with the equivalent linear equations (6.14)-(6.17). In equation (6.15), M_2 is a large number, which is defined by equation (6.18).

$$S_{ij} \geq P_{xy} - 10\xi_{xy}^{ij} \log \left(d_c \left(\sqrt{(i-x)^2 + (j-y)^2} \right) \right) - 3\sigma \quad (6.14)$$

$$\forall i \in Q, \forall j \in R, \forall x \in X, \forall y \in Y$$

$$S_{ij} \leq P_{xy} - 10\xi_{xy}^{ij} \log \left(d_c \left(\sqrt{(i-x)^2 + (j-y)^2} \right) \right) - 3\sigma + M_2(1 - \eta_{xy}^{ij}) \quad (6.15)$$

$$\forall i \in Q, \forall j \in R, \forall x \in X, \forall y \in Y$$

$$\eta_{xy}^{ij} \in \{0,1\}, \quad \forall i \in Q, \forall j \in R, \forall x \in X, \forall y \in Y \quad (6.16)$$

$$\sum_{x \in X} \sum_{y \in Y} \eta_{xy}^{ij} = 1 \quad \forall i \in Q, \forall j \in R \quad (6.17)$$

$$M_2 \geq \max_{\forall ijxy} \left\{ P_{xy} - 10\xi_{xy}^{ij} \log \left(d_c \left(\sqrt{(i-x)^2 + (j-y)^2} \right) \right) - 3\sigma \right\} - \min_{\forall ijxy} \left\{ P_{xy} - 10\xi_{xy}^{ij} \log \left(d_c \left(\sqrt{(i-x)^2 + (j-y)^2} \right) \right) - 3\sigma \right\} \quad (6.18)$$

Finally, the objective function (6.8) and the constraint equations (6.10)-(6.17) constitute our ILP model.

6.4. Preliminary Results

We solve our instances of the above problem on an 8 processor ACPI multiprocessor X64-based PC with Intel Xeon® CPU X5460 running at 3.16GHz with 32 GB memory. We have implemented our model in AMPL (Fourer et. al., 2002), and used CPLEX 11.2 solver (ILOG, 2007) to solve them. To obtain preliminary results, we chose eight small to moderate size test-case grids: 5x5, 5x5a, 7x7, 7x7a, 8x8, 8x8a, 10x10 and 10x10a depicted in Figure 6.2 to Figure 6.6. In this solution, other parameters are assumed as follows: $d_c=10$, $UB_{xy}=200$, $C_f=10$ and $D=20$. Note that we have selected some grids with equal dimension, but with different distribution of obstructions to see the effect of this distribution in the decision outcome and solution statistics. We used a CPLEX *mipgap* setting of 0.001, which means all test cases solved to full termination are provably within 0.1% of optimality.

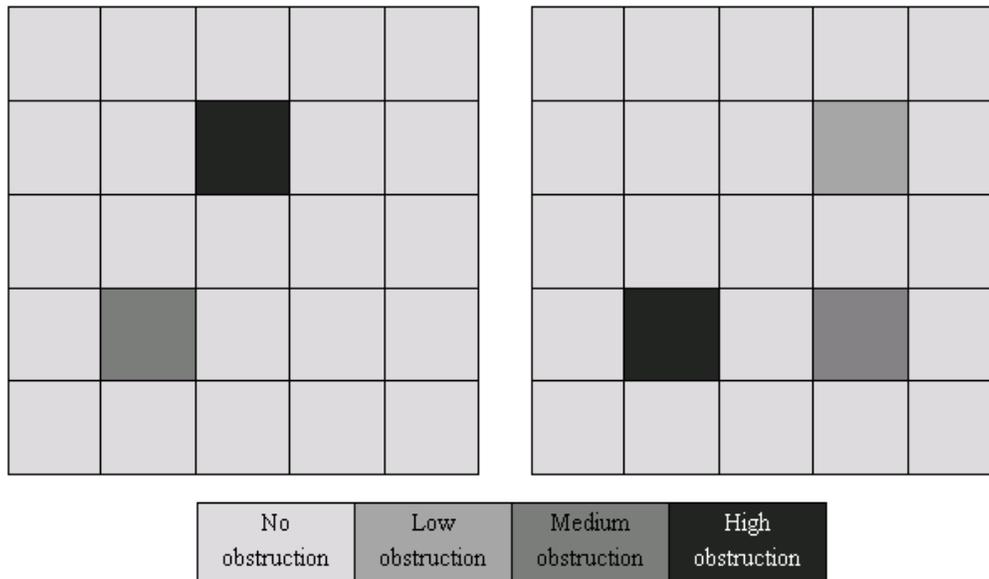


Figure 6.2: Obstructions' map for the 5x5 grid (left) and 5x5a grid (right).

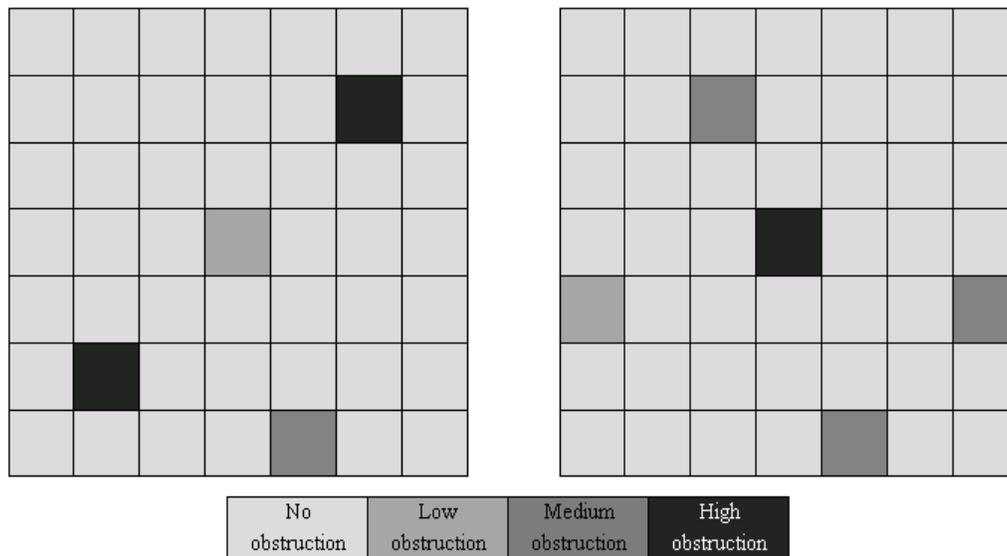


Figure 6.3: Obstructions' map for the 7x7 grid (left) and 7x7a grid (right).

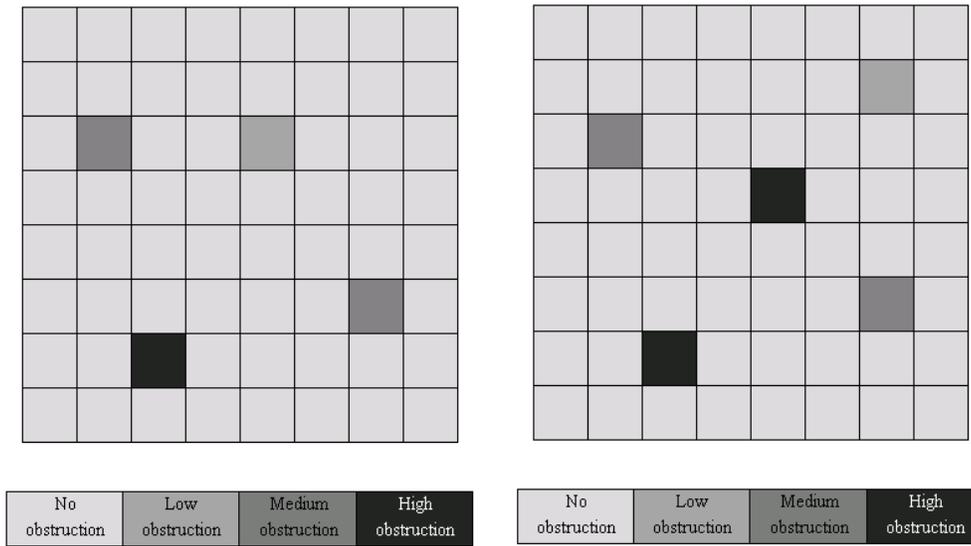
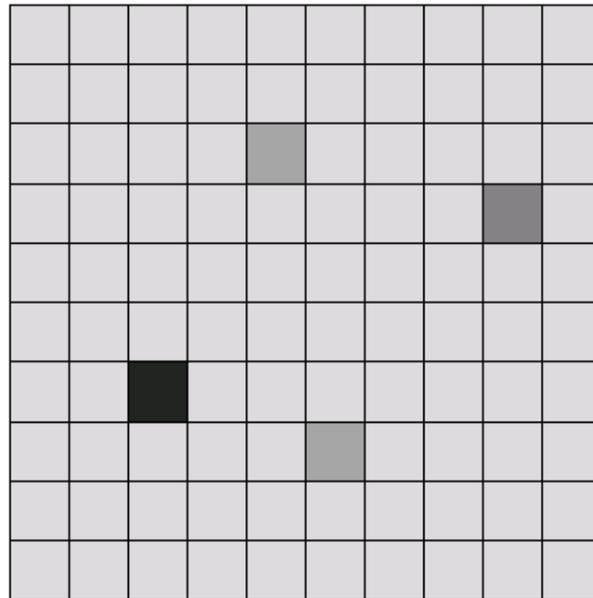


Figure 6.4: Obstructions' map for the 8x8 grid (left) and 8x8a grid (right).



No obstruction	Low obstruction	Medium obstruction	High obstruction
-------------------	--------------------	-----------------------	---------------------

Figure 6.5: Obstructions' map for 10x10 grid.

outlined in this table (except 5x5 and 5x5a), the exact method was not able to reach the target optimality gap within a reasonable amount of time. Overall, the solution time increases exponentially for the larger grids. The exact method takes a very long time to reach the target optimality gap for some grids; it took 247,952 seconds (2.87 days) and 394,598 seconds (4.57 days) to solve the 10x10 and 10x10a grids, respectively. For these test cases, the CPLEX solver's branch-and-bound procedure makes considerable progress at the early stages, with rapid improvements in the objective function values of the best-to-date branch-and-bound nodes. However, reductions in optimality gap slows down, and even after many days of runtime with a huge number of simplex iterations (more than 89 million for 10x10 grid and more than 116 million for 10x10a grid) and an explosion of branch-and-bound nodes (more than 104 thousand for the 10x10 grid and more than 88 thousand for the 10x10a grid). From our preliminary investigation, it is plausible that these larger grids with highly heterogeneous obstruction distributions will be very difficult to solve with the exact method. This was the driver behind our efforts to develop techniques to efficiently solve large instances of this ILP problem.

It is also worth mentioning here that the degree of the complexity of the problem not only depends on the size of the grids, but also the variations of obstructions. We see that for the 7x7 grid, the solver took 8077 seconds, whereas solving the

7x7a, took 19709 seconds (more than two times greater). This was due to the variation of obstructions in the map. Some instances of the problem might create much tighter LP relaxations than other instances, and/or algorithms used by CPLEX's internal branch-and-bound procedures, might be better suited to some of those specific instances. It is also noted that the OOFV and the total number of optimum transmitters are different for the same size grids. For example for 8x8, they were 153 and 1, whereas for 8x8a, they were 178 and 1. Similar results were found for the other pairs of grids.

Table 6.2: Run time statistics for different test-case grids solved by the exact method.

Grids	OOFV	Optimum # of Transmitters	# of MIP simplex iterations	# of branch and bound nodes	# of integer variables	# of constraints	CPU Time (Seconds)
5x5	147	2	181836	3002	675	1350	40.5312
5x5a	147	2	90847	1869	675	1350	20.6562
7x7	157	2	2086602	8077	2499	4998	1805.55
7x7a	154	2	4263524	19709	2499	4998	4063.12
8x8	153	1	6367467	14972	4224	8448	11124.9
8x8a	178	1	6311056	11394	4224	8448	11282.5
10x10	162	2	89431171	104495	10200	20400	247952
10x10a	181	1	116265208	88923	10200	20400	394598

6.5. Relax-and-Fix-Based Decomposition

In many hard ILP models, computational complexity arises from just a small subset of constraints or integrality requirements. Such ILP instances are often solved by the relaxation-based decomposition techniques. In that approach, we can decompose the original problem into two easy sub-problems by relaxing the complicating constraints via Lagrangian decomposition; those sub-problems can then be solved iteratively to obtain a near-optimal solution. Alternatively, we can relax the integrality requirement(s) of selected decision variables to decompose the original problem into sub-problems, which, again, will permit us to obtain a

near-optimal solution (Wolsey, 1998). Using this approach, we first solve the (partially) relaxed problem (i.e., some integrality requirements are ignored). The solution to that easy sub-problem will provide us with values for the remaining integer decision variables that can be then fixed in the original, thereby producing a second sub-problem. The second sub-problem is then solved to provide a near-optimal solution to the original. This technique is referred to as *relax-and-fix-based decomposition* (RFBD) (Noor-E-Alam and Doucette, 2012). RFBD is a problem-specific technique whose success depends on the careful selection of the set of integer variables to be relaxed in the first sub-problem.

We see from our preliminary experiments and investigations that the ILP model for this location problem becomes computationally intractable to solve by the exact method. Therefore we design the following problem-specific RFBD technique to solve large-scale instances, as RFBD has been found to be successful for solving GBLPs (Noor-E-Alam and Doucette, 2012). In the exact method, we solved our GBLP ILP models to determine the total number, location(s), and size(s) of the transmitter(s) simultaneously. In the RFBD approach, we make these decisions in the following two steps.

Step 1: At first, we determined the total number and location(s) of the transmitter(s). By relaxing the integrality requirement of the P_{xy} variables, we obtained a sub-problem that is easier to solve, as there are fewer

integer decision variables, but will still permit precise identification of the location of transmitters.

Step 2: In the second step, the precise locations of the transmitter(s) (i.e., T_{xy} decision variable values) found from the first sub-problem are fixed in the original problem, which is later solved to determine the size(s).

This approach does not guarantee optimality; however, in prior literature, it has been shown that it is quite successful in providing high quality solutions in less time (Noor-E-Alam and Doucette, 2012). Moreover, from our preliminary experiments with the ILP model, we see that even the first relaxed problem was very difficult to solve. Therefore, in the first step, we propose to integrate the valid inequalities and logical restrictions to solve the first sub-problem quickly and facilitate the entire solution time further.

Suppose for the following 4x4 grid problem shown in Figure 6.7, two transmitters ($T_{22}=1$ and $T_{43}=1$) are able to satisfy the demand for all cells. From this scenario, it is obvious that if there is a transmitter at any location (x,y) , then it will at least satisfy one demand cell. This phenomenon can be modelled with equation (6.19). It has been expected that these inequalities will create tight LP relaxations and the solver will take less time to solve the first sub-problem.

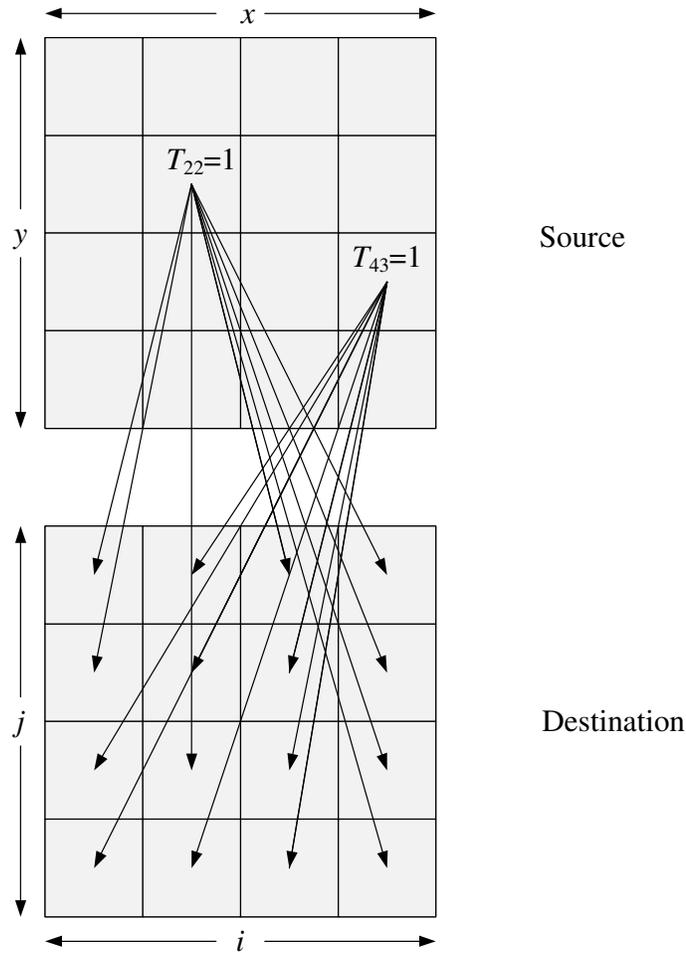


Figure 6.7: Illustration of valid inequalities.

$$\sum_{i \in Q} \sum_{j \in R} \eta_{x,y}^{i,j} \geq T_{xy}, \quad \forall x \in X, \forall y \in Y \quad (6.19)$$

Furthermore, to reduce the computational burden to solve the first sub-problem, we add the following *logical restriction* (LR) defined by equation (6.20), where T_{max} is the maximum number of allowable transmitter(s) considered in this model.

As we see in our ILP model, we minimize the total cost, and by choosing an appropriate value of this maximum threshold, we will help the solver to reduce the search region.

$$\sum_{x \in X} \sum_{y \in Y} T_{xy} \leq T_{max} \quad (6.20)$$

Finally, Figure 6.8 illustrates the details of this proposed problem-specific RFBD approach.

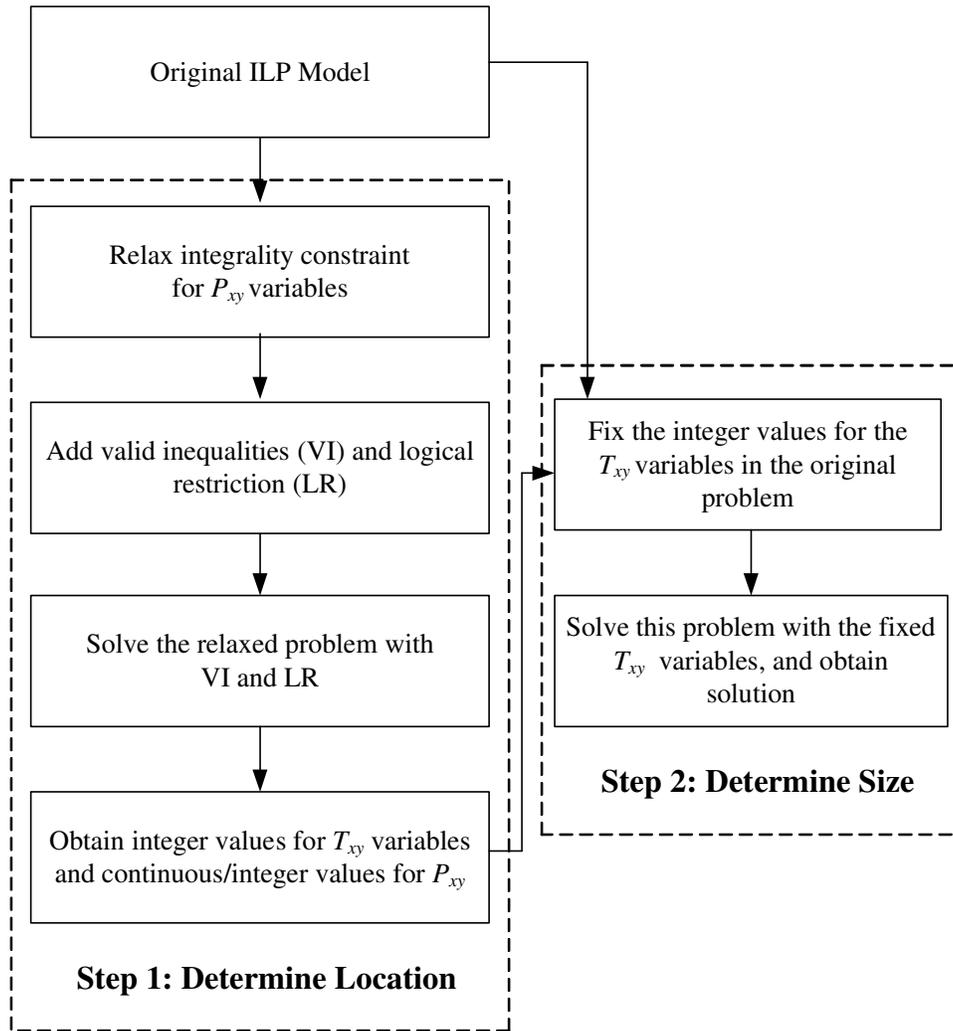


Figure 6.8: Illustration of the RFBD approach.

6.6. Results and Analysis

We solve our ILP model here with the same experimental setup outlined in Section 6.4. In addition to the grids used there, we also add much larger test grids: 10x15, 10x20, 15x15, 15x20 and 20x20, shown in Figure 6.9 to Figure 6.13. Table 6.3 shows the comparative solution data for the exact method and the

RFBD approach on the thirteen test-case grids. The exact method refers to the benchmark solution that was obtained via the ILP in Section 6.4 with $mipgap = 0.001$. In addition, the sum of the CPU times required to solve the two sub-problems gave us the CPU time for the RFBD approach as a whole.

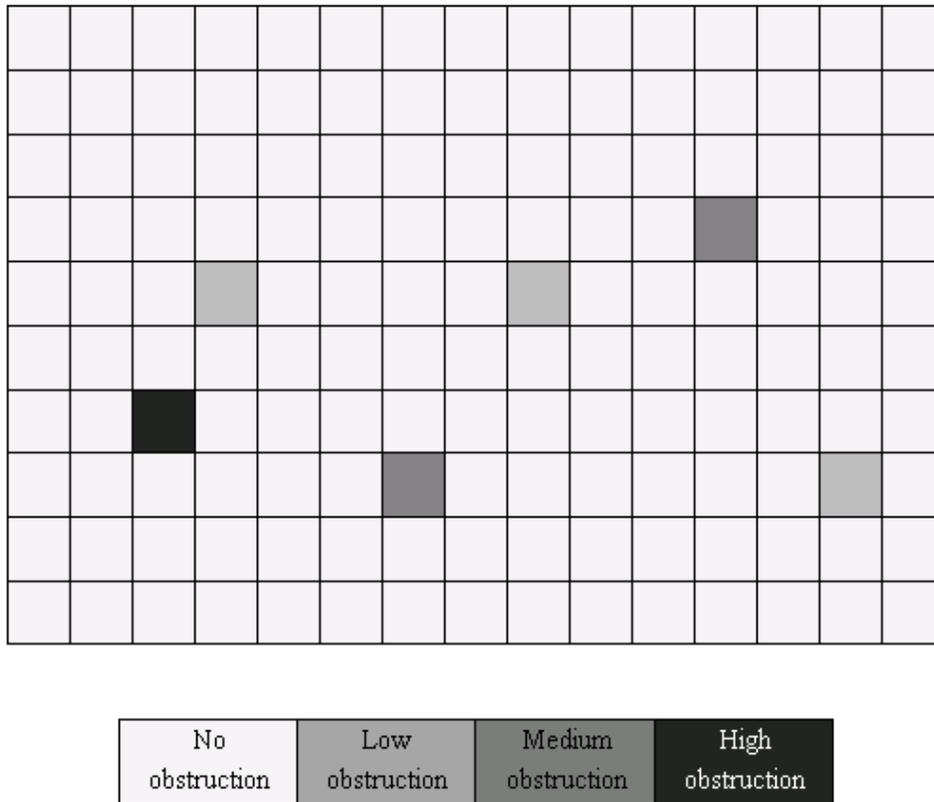


Figure 6.9: Obstructions' map for the 10x15 grid.

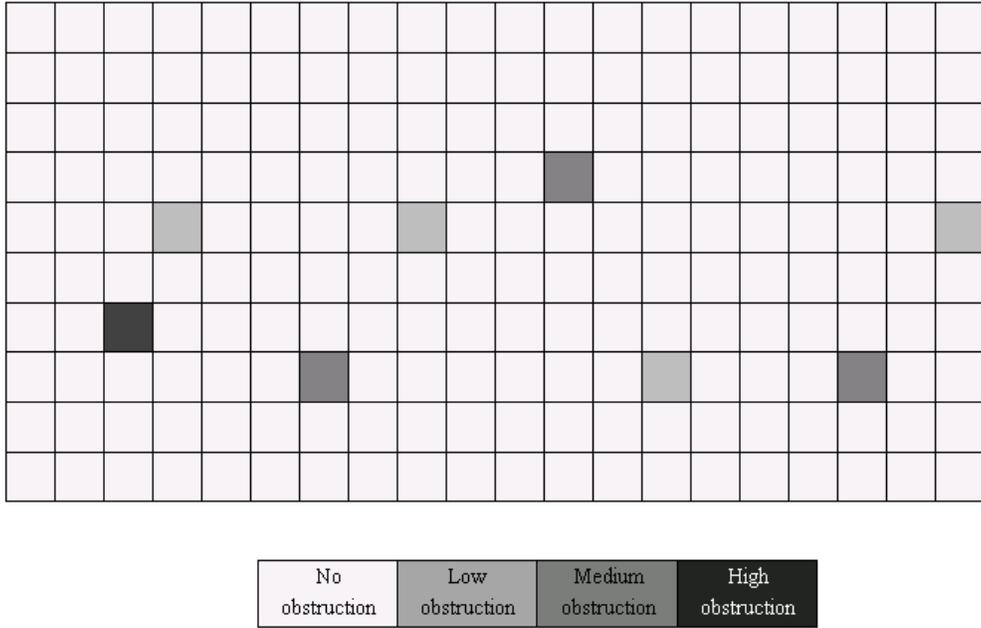


Figure 6.10: Obstructions' map for the 10x20 grid.

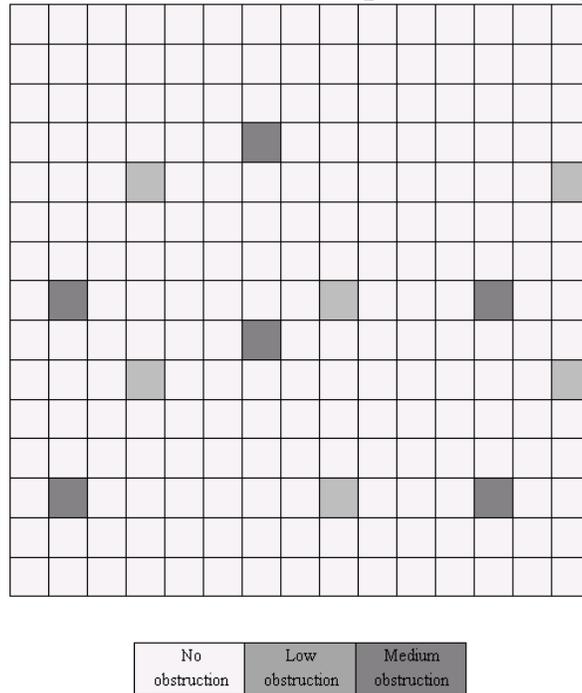


Figure 6.11: Obstructions' map for the 15x15 grid.

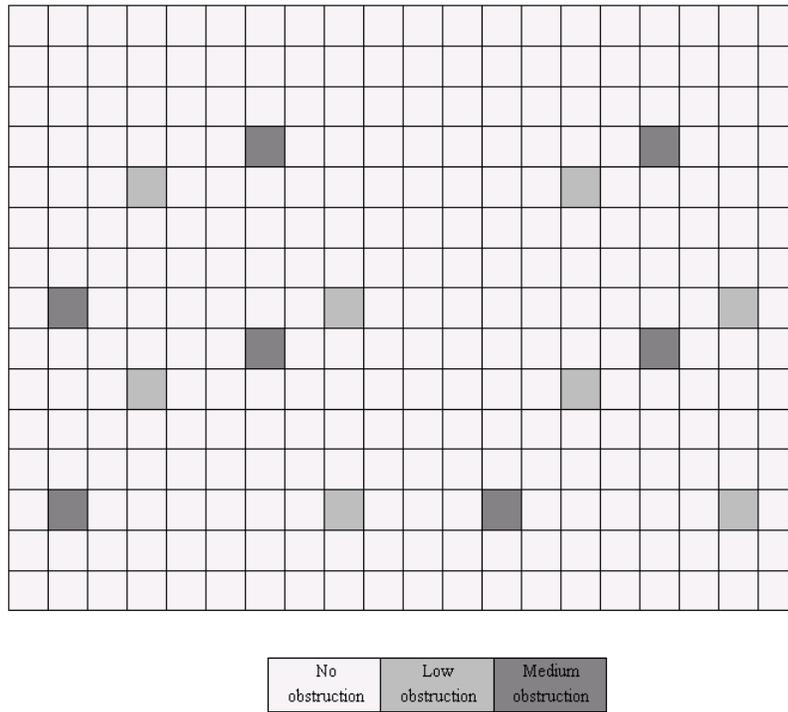


Figure 6.12: Obstructions' map for the 15x20 grid.

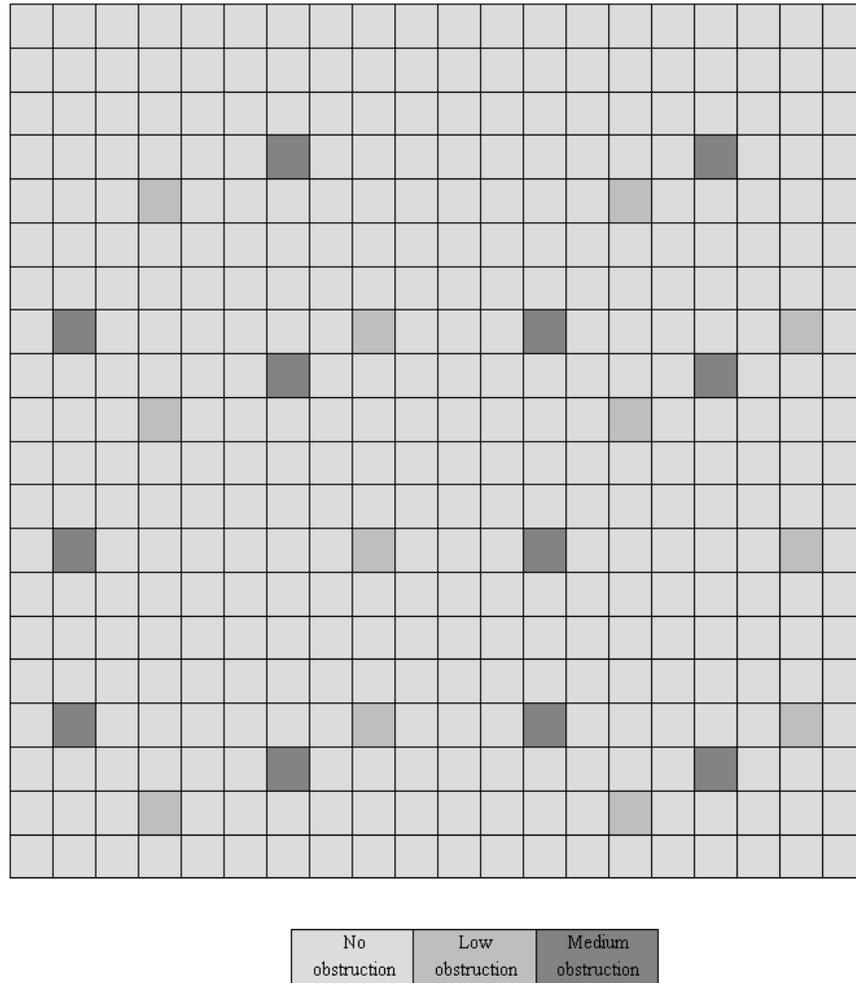


Figure 6.13: Obstructions' map for the 20x20 grid.

While we have already seen that the exact method takes a very long time (nearly 400 thousand CPU seconds) to solve a 10x10a grid, it is plausible that larger grids are nearly impossible to solve with the exact method within a reasonable amount of time. Therefore, we do not show the solution data for the five larger grids solved by the exact method. We see that the solutions (OOFV and optimum

number of transmitters) obtained from the RFBD approach were the same as the exact method for the test-case grids shown in this table. However, with the RFBD approach, significant run time reduction was observed compared to the exact method. We see that for the hard ILP instances, the RFBD approach achieved 85.7%, 94.4%, 96%, 94.7%, 91% and 98.5% runtime reductions for the 7x7, 7x7a, 8x8, 8x8a, 10x10 and 10x10a test-grids, respectively, without affecting optimality.

Furthermore, the RFBD approach took fewer simplex iterations and branch-and-bound nodes. In the RFBD approach, the total number of simplex iterations and branch-and-bound nodes were the sum of the total number of simplex iterations and branch-and-bound nodes required to solve the two sub-problems. Each of the sub-problems became an easier ILP problem compared to the original problem. Therefore, they required a very small number of simplex iterations and branch-and-bound nodes to reach optimality.

We can now return to our discussion above, regarding the introduction of valid inequalities and logical restriction to reduce the feasible region and the number of branch-and-bound nodes in our problem solution due to the tight LP relaxation. Table 6.3 shows the details on the number of simplex iterations and branch-and-bound nodes for solutions using our RFBD and the exact method. We observed, for instance, that in the 10x10a grid, solution of the exact method required nearly

116 million simplex iterations and nearly 89 thousand branch-and-bound nodes, while use of RFBD approach reduced those numbers to approximately 5 million simplex iterations and 6 thousand branch-and-bound nodes. Clearly, the introduction of logical restrictions and valid inequalities significantly reduced the complexity of the first sub-problem in our RFBD approach, which ultimately facilitated the reduction of CPU time to solve the entire problem.

While solutions were generally obtainable for all the test-case grids, even our proposed decomposition technique was unable to obtain a solution for the more computationally complex instances within a reasonable amount of time. The proposed RFBD technique took more than a million seconds (almost two weeks) of CPU time to solve the hardest 20x20 test-case grid (total number of integer variables involved in this grid is 160800, which make this problem difficult to solve with the proposed technique). It is also clear from the runtime data that CPU time appear to increase exponentially with the size and variation of obstructions in the test-case grid. In the future, our plan is to develop techniques to integrate with the first sub-problem so that we could achieve further reduction of CPU time and solve these large-scale instances quickly.

Table 6.3: Run time statistics for different test-case grids solved by the exact method.

Grids	OOFV		Optimum # of Transmitters		# of MIP simplex iterations		# of branch and bound nodes		CPU Time (in Seconds)		
	Exact Method	RFBD	Exact Method	RFBD	Exact Method	RFBD	Exact Method	RFBD	Exact Method	RFBD	% Reduction
5x5	147	147	2	2	181836	42615	3002	573	40.531	3.01	92.57
5x5a	150	150	2	2	89717	21048	1627	618	19.55	1.77	90.95
7x7	157	157	2	2	2086602	407825	8077	1349	1805.6	258.5	85.7
7x7a	154	154	2	2	4263524	348242	19709	1184	4063.1	229.5	94.4
8x8	153	153	1	1	6367467	805963	14972	1947	11125	444.9	96
8x8a	178	178	1	1	6311056	732548	11394	2022	11283	598.7	94.7
10x10	162	162	2	2	89431171	13858744	104495	13285	247952	22084	91
10x10a	181	181	1	1	116265208	5148209	88923	6096	394598	5899.3	98.5
10x15	-	170	-	2	-	35730280	-	22307	-	187318	-
10x20	-	178	-	1	-	36646305	-	26985	-	550757	-
15x15	-	163	-	1	-	16650200	-	24792	-	342549	-
15x20	-	170	-	1	-	17491921	-	14122	-	554048	-
20x20	-	172	-	1	-	33226950	-	23500	-	1165350	-

6.7. Conclusion

This research proposes an ILP model to optimally place wireless transmitter(s) to design cost-effective communication systems. Our ILP model was developed to optimize the number of transmitter(s), their location(s), and their size(s) by considering the variation of signal strengths due to distance, propagation environment, and establishment cost. To solve large-scale instances, we have developed a problem-specific RFBD approach that significantly reduces the

computational complexity of the problem (measured in CPU time). To further facilitate the reduction of CPU time, we have proposed valid inequalities and logical restrictions in the first sub-problem of this approach. We have carried out experiments to test RFBD with a number of grids and have shown that it is quite effective in reducing problem runtime without loss of optimality in test-case grids where exact solutions exist. In the most computationally complex test-case solved by the exact method, we reduce runtime by 98.5% without loss of any optimality. In the future, we plan to extend this work to develop more advanced optimization techniques (such as Lagrangian decomposition and/or integrating other strong cuts) to solve larger problems more efficiently.

While we have tested our ILP model and the RFBD approach on a GBLP that seeks to optimally place transmitter(s) to minimize the total cost, other GBLPs could be solved using this approach. Potential problems include optimal placement of retail outlets and/or warehouses, sensor network placement, resource exploitation, and even delivery of radiation therapy.

References

- Ai, J., Abouzeid, A. A. (2006), "Coverage by directional sensors in randomly deployed wireless sensor networks", *Journal of Combinatorial Optimization*, 11, 21-41.

- Cao, M., Anderson, B. D. O., Morse, A. S. (2006), "Sensor network localization with imprecise distances", *Systems & Control Letters*, 55, 887-893.
- Chong, C-Y., Kumar, S. P. (2003), "Sensor networks: Evolution, opportunities, and challenges," *Proceedings of The IEEE*, 91(8), 1247-1256.
- Ci, S., Guizani, M., Sharif, H. (2007), "Adaptive clustering in wireless sensor networks by mining sensor energy data", *Computer Communications*, 30, 2968-2975.
- Coluccia, A., Altman, E. (2012), "SINR Base Station Placement and Mobile Association Games under Cooperation", 9th Annual Conference on Wireless On-Demand Network Systems and Services (WONS), 9-11 January, 51-54.
- Durgin, G., Rappaport, T. S., Xu, H. (1998), "Measurements and Models for Radio Path Loss and Penetration Loss In and Around Homes and Trees at 5.85 GHz", *IEEE Transactions on Communications*, 46(11), 1484-1496.
- Fourer, R., Gay, D. M. and Kernighan, B. W. (2002), *AMPL: A Modeling Language for Mathematical Programming*, Duxbury Press, Cole Publishing Co.
- Gentile, C. (2007), "Distributed sensor location through linear programming with triangle inequality constraints", *IEEE Transactions on Wireless Communications*, 6(7), 2572-2581.
- Giortzis, A. I., Turner, L. F., Barria, J. A. (2000), "Decomposition technique for fixed channel assignment problems in mobile radio networks", *IEE Proceedings-Communication*, 147 (3), 187-194.

Guney, E., Aras, N., Altnel, I. K., Ersoy, C. (2012), “Efficient solution techniques for the integrated coverage, sink location and routing problem in wireless sensor networks”. *Computers and Operations Research*, 39(7), 1530-1539.

Guney, E., Aras, N., Altnel, I. K., Ersoy, C. (2010), “Efficient integer programming formulations for optimum sink location and routing in heterogeneous wireless sensor networks”, *Computer Networks*, 54, 1805-1822.

ILOG (2007), “ILOG CPLEX 11.0 User’s Manual”, ILOG, Inc.

Jayabharathy, R., Prithiviraj, V., Varadharajan, R. (2012), “Hybrid Range Based Indoor Wireless Localization using Unconstrained Nonlinear Optimization for UWB Applications” *Procedia Engineering*, 30, 562-568.

Lu, X.-X., Zhang, W. (2011), “An Approximate Algorithm to Solve the Location-Selection of Wireless Network Problem”, *Software Engineering and Knowledge Engineering*, 2, 827-832.

Mao, G., Fidan, B., Anderson, B. D. O. (2007), “Wireless sensor network localization techniques”, *Computer Networks*, 51, 2529-2553.

Marianov, V., Eiselt, H. A. (2012), “Transmitter location for maximum coverage and constructive–destructive interference management”, *Computers & Operations Research*, 39, 1441-1449.

Mauri, G. R., Ribeiro G. M., Lorena, L. A. N. (2010), “A new mathematical model and a Lagrangian relaxation for the point-feature cartographic label placement problem”, *Computers & Operations Research*, 37, 2164–2172.

Noor-E-Alam, M., Ma, A., Doucette, J. (2012), “Integer Linear Programming Models for Grid-Based Light Post Location Problem”, *European Journal of Operational Research*, 222, 17-30.

- Noor-E-Alam, M., Doucette, J. (2012), “Relax-and-Fix-Based Decomposition Technique for Solving Large Scale GBLPs”, *Computers and Industrial Engineering*, 63, 1062-1073.
- Rappaport, T. S. (1996), *Wireless Communications: Principles and Practice*, Prentice-Hall Inc., New Jersey.
- Rodas, J., Escudero, C. J. (2010), “Dynamic path-loss estimation using a particle filter”, *International Journal of Computer Science Issues*, 7(4), 1-5.
- Sharma, D., Sharma, P. K., Gupta, V., Singh, R. K. (2010), “A Survey on Path Loss Models Used In Wireless Communication System Design”, *International J. of Recent Trends in Engineering and Technology*, 3(2), 115-118.
- Weng, Y., Xiao, W., Xie, L. (2011), “Total least squares method for robust source localization in sensor networks using TDOA measurements”, *International Journal of Distributed Sensor Networks*, Article ID 172902, 8 pages, doi:10.1155/2011/172902
- Wolsey, L. A. (1998), *Integer Programming*, Wiley-Interscience, Series in Discrete Mathematics and Optimization, Toronto, (Chapter 1).
- Zhang, R.-B., Guo, J.-G., Chu, F.-H., Zhang, Y.-C. (2011), “Environmental-adaptive indoor radio path loss model for wireless sensor networks localization”, *Int. J. Electron. Commun. (AEÜ)* 65, 1023-1031.

Chapter 7

Concluding Discussion

7.1. Brief Summary of Thesis

The objective of this thesis was to develop methods for solving grid-based location problems (GBLPs) that could be derived from various business, engineering and medical science applications. To do so, we have developed a number of mathematical models with two real-world applications. We found that these mathematical models are computationally difficult and in many instances, completely intractable with conventional ILP solution methods. To overcome this computational difficulty, we have developed advanced ILP techniques and algorithmic approaches to solve-large scale instances in reasonable time.

We have started the thesis with an introductory discussion of different types of location problems and presented the concept of GBLPs in Chapter 1. We have described our motivation and objectives of this research.

In Chapter 2, we have discussed integer linear programming techniques with relevant literatures.

In Chapter 3, we have focused on developing integer linear programming models for solving grid-based location problems. We have used a real-world problem of placing lights in a city park to minimize the amount of darkness and excess supply. From the investigation of light physics and the location problem itself, we found that the supply function is non-linear and distribution of the demand is heterogeneous. Considering these demand and supply relationships, we have developed three GBLP ILP models that are designed to provide the optimal solution for the light post problem: the total number of light posts, the location of each light post, and their capacities, i.e., brightness. We implemented these ILP models within a standard modeling language and solved with the CPLEX solver. Our experiments showed that the ILP models are efficient in solving moderately-sized problems with a small optimality gap. Although this chapter represents just one specific GBLP, in reality, many other location problems can be modelled as GBLP, and can be solved using these methods.

In Chapter 4, we have developed a relax-and-fix-based decomposition (RFBD) approach to solve large-scale grid-based location problems. The RFBD approach is implemented within a standard modeling language and CPLEX solver, and is tested with a light post placement problem using several test-case grids. Our experiments showed a significant reduction in solution runtimes while not severely impacting optimality. To decrease the solution runtimes further, we have introduced problem-specific logical restrictions that reduce the feasible region, and the resulting branch-and-bound tree. In our most computationally complex test case on the 10x20 grid, we were able to reduce runtime by 95.1% while increasing the obtained objective function value by only 2.4%, and in our most computationally complex test case on the 15x15 grid, runtime was reduced by 98.9% while increasing the obtained objective function value by only 3.6%. Furthermore, when we add the logical restrictions to our RFBD, we were able to reduce runtimes as much as 99.6% in the most complex 10x20 test case with a 2.6% loss of optimality and as much as 99.9% in the most complex 15x15 test case with a 4.9% loss of optimality.

In Chapter 5, we have proposed a fixed cost ILP model to provide the optimal solution for GBLPs. Our preliminary results showed that the ILP model is successful in solving small to moderately-sized problems. However, this ILP model becomes intractable in solving large-scale instances. Therefore, we have

developed a partition-and-fix-based decomposition (PFBD) approach to solve large-scale instances. We have implemented our proposed approach within a standard modeling language and CPLEX solver, and tested with a light post placement problem using several test-case grids. We have compared our results with the exact method to benchmark our proposed PFBD heuristic. Our experiments showed that the PFBD technique is efficient in reducing problem runtimes without any significant loss of optimality. While it is very difficult to solve large grids by the exact method, the PFBD approach significantly reduces the complexity of the large test-case grids, and the ILP model becomes very scalable to solve. Furthermore, we have proposed to integrate the RFBD technique with this decomposition technique to solve very large-scale instances efficiently.

In Chapter 6, we have proposed another ILP model to optimally place wireless transmitter(s) to ensure a reliable and cost-effective communication system. We have considered variation of signal strength due to distance, propagation environment, and establishment cost to develop this ILP model. Our preliminary experiments showed that this ILP model becomes computationally intractable to solve large-scale instances. Therefore, we have developed a problem-specific RFBD approach that significantly reduced the computational complexity of the problem. To further assist in the reduction of CPU time, we have proposed to

integrate valid inequalities and logical restrictions in the first sub-problem of the RFBD approach. We have implemented our proposed approach within a standard modeling language and CPLEX solver, and tested with a number of test-case grids. Our experiments showed that the RFBD approach is efficient in reducing problem runtimes without loss of optimality. For example, in our most computationally complex test-case on the 10x10a grid solved by the exact method, we were able to reduce runtime by 98.5% without any loss of optimality.

7.1.1. Main Contributions

There are four main contributions of this thesis described in Chapter 3 through Chapter 6. These contributions are briefly listed as below:

1. Chapter 3: ILP Models for Light Post Location Problem
 - Developed a basic optimization model
 - Developed two enhanced ILP models
 - Implemented and tested these ILP models
2. Chapter 4: Solving Large Scale GBLPs
 - Proposed the RFBD approach to solve large-scale instances
 - Proposed problem-specific logical restrictions

- Implemented and tested RFBD approach

3. Chapter 5: Solving Large Scale Fixed Cost GBLPs

- Developed a fixed cost GBLP ILP model
- Proposed the PFBD approach to solve large-scale instances
- Integrated the RFBD approach with the PFBD approach
- Implemented and tested PFBD approach

4. Chapter 6: ILP Model for Wireless Transmitter Location Problem

- Developed an ILP model for wireless transmitter location problem
- Proposed the RFBD approach to solve large-scale instances
- Proposed problem-specific valid inequalities and logical restrictions
- Implemented and tested proposed approach

7.2. Other Contributions of Ph.D. Work

Besides the contributions discussed in herein, the overall Ph. D. work also made several other contributions. These contributions are listed as follows.

7.2.1. Journals Papers

1. Md. Noor-E-Alam, Andrew Ma, John Doucette, “Integer Linear Programming Models for Grid-Based Light Post Location Problem”, *European Journal of Operational Research*, vol. 222, pp. 17-30, October, 2012.
2. Md. Noor-E-Alam and John Doucette, “Relax-and-Fix-Based Decomposition Technique for Solving Large Scale GBLPs”, *Computers and Industrial Engineering*, vol. 63, pp. 1062-1073, December, 2012.
3. Md. Noor-E-Alam, John Doucette, “An Application of Infinite Horizon Stochastic Dynamic Programming in Multi Stage Project Investment Decision Making”, *International Journal of Operational Research*, vol. 13, No. 4, pp. 423-438, February, 2012.
4. Md. Noor-E-Alam, Ahmed Zaky Kasem and John Doucette, “ILP Model and Relaxation-Based Decomposition Approach for Incremental Topology Optimization in p -Cycle Networks”, *Journal of Computer Networks and Communications*, vol. 2012, pp. 1-10, November, 2012.
5. Md. Noor-E-Alam, John Doucette, “Solving Large Scale Fixed Cost Integer Linear Programming Models for Grid-Based Location Problems with Heuristic Techniques”, *Computers & Operations Research*, 2012. (in review)

6. Md. Noor-E-Alam, Brody Todd and John Doucette, “Integer Linear Programming Model for Grid-Based Wireless Transmitter Location Problems”, *International Journal of Operational Research*, 2012. (in review)

7.2.2. Refereed Conference Publication

1. Md. Noor-E-Alam and John Doucette, “Stochastic Investment Decision Making with Dynamic Programming”, *International Conference on Industrial Engineering and Operations Management (IEOM)*, Dhaka, Bangladesh, 9–10 January, 2010.

7.2.3. Other Peer Reviewed Publications and Presentations

1. Md. Noor-E-Alam, John Doucette, “Solving large scale ILP models for Grid-Based Location Problems with a Heuristic Decomposition Technique”, *International Conference of Manufacturing Engineering and Engineering Management (ICMEEM)*, 4-6 July, 2012, London, United Kingdom.

2. Md. Noor-E-Alam, John Doucette, “Mixed Integer Linear Programming Models for Grid-Based Location Problem”, *International Conference of Manufacturing Engineering and Engineering Management (ICMEEM)*, 6-8 July, 2011, London, United Kingdom.

3. Md. Noor-E-Alam, John Doucette “Stochastic Investment Decision Making with Dynamic Programming”, *Faculty of Engineering Graduate Research Symposium* (FEGRS), 17 June, 2010, Faculty of Engineering, University of Alberta, Edmonton, AB, Canada.

4. Md. Noor-E-Alam, John Doucette “Investment Decision Making under Uncertain Environment”, Poster, *Mechanical Engineering Graduate Symposium*, 19 March, 2009, Mechanical Engineering, University of Alberta, Edmonton, AB, Canada.

7.3. Future Research Avenues

Besides the current contributions discussed herein, some opportunities for future research on GBLPs are as follows. In the future, we plan to extend this work to apply our proposed ILP models to solve some other potential location problems such as optimal placement of retail outlets and/or warehouses. We could develop supply distributions of such facilities using the gravity model and demand distributions can be established through market research or surveys in the surrounding neighborhoods. We could also develop a multi-objective optimization model to solve such decision problems, where the *multi-criteria decision making* (MCDM) technique can be used to determine the weight of each objective.

Other topics where our models can potentially be applied include health/biological sciences (e.g., optimal delivery of drug/radiation therapy), communications (e.g., transmitter locations), real estate, and emergency service dispatching, physics, and resource exploration/exploitation. In case of optimal delivery of radiation therapy, it is important to determine appropriate location and dose of the treatment since the objective would be minimizing the dose to the healthy cell while applying a sufficient dose to the affected cell. This problem can be modeled as a GBLP. To develop GBLP ILP for this problem, we could develop a supply relationship from drug diffusion distribution.

To solve large-scale instances of the above problems, our proposed RFBD and PFBD approaches can be applied. To better facilitate the performance of the proposed RFBD and PFBD approaches, we will further investigate adding some strong cuts and logical restrictions to solve larger problems efficiently. We could develop a nested PFBD approach to solve extremely large fixed cost GBLPs, where we will use the PFBD approach to solve the sub-problems. We could also solve these sub-problems simultaneously with the advantage of parallel computing to minimize the total solution time.

Moreover, we will explore the opportunity of developing other advanced techniques such as Lagrangian decomposition and Dantzig-Wolfe decomposition to solve large-scale instances. Lagrangian decomposition techniques create an

easier sub-problem by relaxing complicating constraints from the original problem. This sub-problem will be solved by the exact method and the solution of this sub-problem will be fixed in the original problem to form the core problem. This core problem will be solved to get a near optimal solution. We could also reformulate our original problem using Dantzig-Wolfe decomposition technique so that LP relaxation will create tight bounds. Finally, column generation will be invoked for implicit pricing of non-basic variables.

Appendices

Appendix A: Following files are written and generated to carry out experiments with the ILP models discussed in Chapter 3.

AMPL Model File (for enhanced model without controlling number of sources):

```
set Q;
set R;
set X;
set Y;

var A {Q,R};
var B {Q,R};
var S{i in Q,j in R};
var p{w in X,z in Y} integer ;
# define decision variables in the model

minimize cost: sum {i in Q,j in R} (A[i,j]+B[i,j]);
# linear objective function

subject to upper_limit_ex_supply{i in Q, j in R}: A[i,j]>=S[i,j] - D[i,j];
subject to lower_limit_ex_supply {i in Q, j in R}: A[i,j]>=0;
```

```

subject to upper_limit_ex_demand {i in Q, j in R}: B[i,j]>=D[i,j]-S[i,j];
subject to lower_limit_ex_demand {i in Q, j in R}: B[i,j]>=0;
# above four constraints are used to linearize the objective function

```

```

subject to supply_relationship{i in Q, j in R}:S[i,j]=sum {w in X,z in
Y}((p[w,z]*C[i,j,w,z])*P[i,j,w,z]);
# calculates supply

```

```

subject to upper_limit_size{w in X, z in Y}: 0<=p[w,z]<=10;
#boundary condition of size variables

```

AMPL Model File (for enhanced model with controlling number of sources):

```

set Q;
set R;
set X;
set Y;

var A {Q,R};
var B {Q,R};
var S{i in Q,j in R};
var p{w in X,z in Y} integer ;
var T{w in X,z in Y} integer >=0,<=1 ;
var M{w in X,z in Y} integer >=0,<=1 ;
var N{w in X,z in Y} integer >=0,<=1 ;
# define decision variables in the model

var a1{X,Y} integer >=0,<=1 ;
var a2{X,Y} integer >=0,<=1 ;
var a3{X,Y} integer >=0,<=1 ;
var a4{X,Y} integer >=0,<=1 ;
var a5{X,Y} integer >=0,<=1 ;
var a6{X,Y} integer >=0,<=1 ;
var a7{X,Y} integer >=0,<=1 ;
var a8{X,Y} integer >=0,<=1 ;
# define decision variables used for linearization

minimize cost: sum {i in Q,j in R} (A[i,j]+B[i,j]);
# linear objective function

```

```

subject to upper_limit_ex_supply {i in Q, j in R}: A[i,j]>=S[i,j] - D[i,j];
subject to lower_limit_ex_supply {i in Q, j in R}: A[i,j]>=0;
subject to upper_limit_ex_demand {i in Q, j in R}: B[i,j]>=D[i,j]-S[i,j];
subject to lower_limit_ex_demand {i in Q, j in R}: B[i,j]>=0;
# above four constraints are used to linearize the objective function

subject to supply_relationship {i in Q, j in R}:S[i,j]=sum {w in X,z in
Y}((p[w,z]*C[i,j,w,z])*P[i,j,w,z]);
# calculates supply

subject to binary_cons1 {w in X, z in Y}:-T[w,z]+1<=10*(a1[w,z]);
subject to allowable_size1 {w in X, z in Y}: p[w,z]<=10*(1-a1[w,z]);
subject to binary_cons2 {w in X, z in Y}:T[w,z]-1<=10*(a2[w,z]);
subject to allowable_size2 {w in X, z in Y}: p[w,z]<=10*(1-a2[w,z]);

subject to binary_cons3 {w in X, z in Y}:-M[w,z]+1<=10*(a3[w,z]);
subject to allowable_size3 {w in X, z in Y}: p[w,z]+1<=11*(1-a3[w,z]);
subject to binary_cons4 {w in X, z in Y}:M[w,z]-1<=10*(a4[w,z]);
subject to allowable_size4 {w in X, z in Y}: p[w,z]+1<=11*(1-a4[w,z]);

subject to binary_cons5 {w in X, z in Y}:-N[w,z]+1<=10*(a5[w,z]);
subject to allowable_size5 {w in X, z in Y}: -p[w,z]+1<=11*(1-a5[w,z]);
subject to binary_cons6 {w in X, z in Y}:N[w,z]-1<=10*(a6[w,z]);
subject to allowable_size6 {w in X, z in Y}: -p[w,z]+1<=11*(1-a6[w,z]);

subject to binary_cons7 {w in X, z in Y}:-T[w,z]<=1*(a7[w,z]);
subject to all_binary1 {w in X, z in Y}: M[w,z]+N[w,z]-1<=1*(1-a7[w,z]);
subject to binary_cons8 {w in X, z in Y}:T[w,z]<=1*(a8[w,z]);
subject to all_binary2 {w in X, z in Y}: M[w,z]+N[w,z]-1<=1*(1-
a8[w,z]);
# above sixteen constraints are used to linearize the if then constraints

subject to maximum_allowable_sources : sum {w in X, z in Y}T[w,z]=ns;
subject to upper_limit_size {w in X, z in Y}: 0<=p[w,z]<=10;
#boundary condition of size variables

```

AMPL Data File (for 10x10 Grid):

```

set R := 1 2 3 4 5 6 7 8 9 10 ;

```

```
set Q := 1 2 3 4 5 6 7 8 9 10 ;
```

```
set Y := 3 4 5 6 7 8 ;
```

```
set X := 3 4 5 6 7 8 ;
```

```
param D:
      1      2      3      4      5      6      7      8      9      10 :=
1  0.35752  0.54987  0.73016  0.84070  0.98172  1.20709  1.44807  0.96481  0.4631  0.1157
2  0.52731  0.84154  1.17494  1.22878  1.38365  1.81815  1.66435  1.49454  0.6587  0.16467
3  0.65812  1.11477  1.3072  1.51691  1.50699  1.68254  1.89704  1.35475  0.6772  0.16931
4  0.68100  1.0601  1.4235  1.43293  1.44618  1.50892  1.55134  1.3504  0.6957  0.17393
5  0.67602  1.0224  1.30215  1.34661  1.3373  1.35669  1.44959  1.14130  0.7553  0.18883
6  0.66941  1.05239  1.41729  1.31546  1.2011  1.13197  1.09093  1.01008  0.5258  0.13144
7  0.62704  1.1011  1.20535  1.29786  1.02114  0.88003  0.7726  0.62386  0.3378  0.08446
8  0.44765  0.7256  1.005646  0.85553  0.70637  0.59511  0.49643  0.37510  0.2019  0.05049
9  0.22635  0.3483  0.44165  0.41280  0.35425  0.29808  0.24317  0.17831  0.0950  0.02376
10 0.0565  0.0870  0.11040  0.10318  0.08855  0.07451  0.06078  0.04457  0.0237  0.00000;
```

```
param P:=
```

```
[1,1,3,3] .15
[1,1,3,4] 1
[1,1,3,5] 1
[1,1,3,6] 1
[1,1,3,7] 1
.
.
.
[10,9,8,8] .17
[10,10,8,3] 1
[10,10,8,4] 1
[10,10,8,5] 1
[10,10,8,6] 1
[10,10,8,7] 1
[10,10,8,8] .15;
```

```
param C:=
```

```
[1,1,3,3] 1
[1,1,3,4] 0
[1,1,3,5] 0
[1,1,3,6] 0
```

```
[1,1,3,7] 0
[1,1,3,8] 0
[1,2,3,3] 1
[1,2,3,4] 1
[1,2,3,5] 0
.
.
.
[10,9,8,7] 1
[10,9,8,8] 1
[10,10,8,3] 0
[10,10,8,4] 0
[10,10,8,5] 0
[10,10,8,6] 0
[10,10,8,7] 0
[10,10,8,8] 1;
```

AMPL Run file (for 10x10 Grid):

```
option mipgap 0.001;
option omit_zero_rows 1;
option display_eps .001;
option solution_round 3;
option solver cplexamp;

model final.mod;
data light1010.dat;

for{1..20} {

    option cplex_options 'mipgap=0.001;

    solve;

    printf"\n %10.3f\n", _solve_time>>final1020.txt;
```

```
printf"\ %10.3f \", sum {i in Q,j in R} (A[i,j]+B[i,j])>>  
final1020.txt;  
  
let ns :=ns+1;  
  
}
```

Appendix B: Following files are written and generated to carry out experiments with the ILP model and RFBD approach discussed in Chapter 4.

RFBD approach only:

AMPL Model File: We used same model file that is described in Appendix A for Model 3.

AMPL Run File (for 10x20 grid):

```

option mipgap 0.001;
option omit_zero_rows 1;
option display_eps .001;
option solution_round 3;
option solver cplexamp;

model light1020.mod;
data light1020.dat;

for{1..13}{
option cplex_options 'mipgap=0.001';
let {w in X,z in Y}p[w,z].relax:=1;
solve;
printf"\n %10.3f \", _solve_time>>light1020_RFBDOnly.txt;

fix{w in X,z in Y}T[w,z];
fix{w in X,z in Y}M[w,z];
fix{w in X,z in Y}N[w,z];
fix{w in X,z in Y}a1[w,z];
fix{w in X,z in Y}a2[w,z];
fix{w in X,z in Y}a3[w,z];
fix{w in X,z in Y}a4[w,z];
fix{w in X,z in Y}a5[w,z];
fix{w in X,z in Y}a6[w,z];
fix{w in X,z in Y}a7[w,z];
fix{w in X,z in Y}a8[w,z];
let {w in X,z in Y}p[w,z].relax:=0;
solve;
printf"\ %10.3f \", _solve_time>>light1020_RFBDOnly.txt;

```

```

printf"\ %10.3f \", sum {i in Q,j in R} (A[i,j]+B[i,j])>>
light1020_RFBDonly.txt;
let ns :=ns+1;
unfix {w in X,z in Y}T[w,z];
unfix {w in X,z in Y}M[w,z];
unfix {w in X,z in Y}N[w,z];
unfix {w in X,z in Y}a1[w,z];
unfix {w in X,z in Y}a2[w,z];
unfix {w in X,z in Y}a3[w,z];
unfix {w in X,z in Y}a4[w,z];
unfix {w in X,z in Y}a5[w,z];
unfix {w in X,z in Y}a6[w,z];
unfix {w in X,z in Y}a7[w,z];
unfix {w in X,z in Y}a8[w,z];

}

```

RFBD with LRs:**AMPL Model File:**

We used same model file that is described in Appendix A for Model 3 with an addition of the following constraints equation (shown for 10x20 grid):

```

subject to LRs1020 {w in X, z in Y}: if w>3 and w< 7 and z>3 and z<17
then T[w-1,z]+T[w,z-1]+T[w,z]+T[w, z+1]+T[w+1,z]<=1;

```

AMPL Run File (for 10x20 grid):

```

option omit_zero_rows 1;
option display_eps .001;
option solution_round 3;
option solver cplexamp;

model lightlr1020.mod;
# this model file has Lrs with addition of all other constraints and
# objective function in the original model
data light1020.dat;

```

```

for{1..50}{
option cplex_options 'mipgap=0.001';
let {w in X,z in Y}p[w,z].relax:=1;
solve;
printf"\n %10.3f \", _solve_time>>light1020_RFBD.txt;

fix {w in X,z in Y}T[w,z];
fix {w in X,z in Y}M[w,z];
fix {w in X,z in Y}N[w,z];
fix {w in X,z in Y}a1[w,z];
fix {w in X,z in Y}a2[w,z];
fix {w in X,z in Y}a3[w,z];
fix {w in X,z in Y}a4[w,z];
fix {w in X,z in Y}a5[w,z];
fix {w in X,z in Y}a6[w,z];
fix {w in X,z in Y}a7[w,z];
fix {w in X,z in Y}a8[w,z];
let {w in X,z in Y}p[w,z].relax:=0;
solve;
printf"\ %10.3f \", _solve_time>>light1020_RFBD.txt;
printf"\ %10.3f \", sum {i in Q,j in R} (A[i,j]+B[i,j])>>
light1020_RFBD.txt;
let ns :=ns+1;
unfix {w in X,z in Y}T[w,z];
unfix {w in X,z in Y}M[w,z];
unfix {w in X,z in Y}N[w,z];
unfix {w in X,z in Y}a1[w,z];
unfix {w in X,z in Y}a2[w,z];
unfix {w in X,z in Y}a3[w,z];
unfix {w in X,z in Y}a4[w,z];
unfix {w in X,z in Y}a5[w,z];
unfix {w in X,z in Y}a6[w,z];
unfix {w in X,z in Y}a7[w,z];
unfix {w in X,z in Y}a8[w,z];

}

```

Appendix C: Following file is written and generated to carry out experiments with the ILP model and PFBD approach discussed in Chapter 5.

Fixed Cost GBLP ILP AMPL model:

```

set Q;
set R;
set X;
set Y;

var p{w in X,z in Y} integer ;
var S{i in Q,j in R};
var T{w in X,z in Y} integer >=0,<=1 ;

param D{Q,R};
param P{Q,R,X,Y};
param C{Q,R,X,Y};

minimize cost:

sum {w in X,z in Y} (10*T[w,z]+p[w,z]);
# objective function

subject to T12{i in Q, j in R}: sum {w in X,z in
Y}((p[w,z]*C[i,j,w,z])*P[i,j,w,z])>=D[i,j];

subject to S06 {w in X, z in Y}:p[w,z]<=10*T[w,z];

subject to pl{w in X, z in Y}: 0<=p[w,z]<=10;

```

Appendix D: Following files are written and generated to carry out experiments with the ILP model and RFBD approach discussed in Chapter 6.

AMPL Model (for 20x20 grid):

```

set Q;
set R;
set X;
set Y;

var p{w in X,z in Y} integer ;
var S{i in Q,j in R};
var T{w in X,z in Y} integer >=0,<=1 ;
var d{i in Q,j in R,w in X,z in Y} integer >=0,<=1 ;

param P{Q,R,X,Y};
param C{Q,R,X,Y};
param E{Q,R,X,Y};

minimize cost:

sum {w in X,z in Y} (10*T[w,z]+p[w,z]);

# objective function

subject to upper_supply {i in Q, j in R, w in X, z in Y}:S[i,j] >= p[w,z]-
E[i,j,w,z]*P[i,j,w,z]*C[i,j,w,z]-15;

subject to lower_supply{i in Q, j in R, w in X, z in Y}:S[i,j] <= p[w,z]-
E[i,j,w,z]*P[i,j,w,z]*C[i,j,w,z]-15+ 395*(1-d[i,j,w,z]);

subject to max_binary{i in Q, j in R}: sum {w in X, z in Y} d[i,j,w,z]=1;

subject to supply_demand{i in Q, j in R}: S[i,j]>=20;

subject to big_M {w in X, z in Y}:p[w,z]<=200*T[w,z];

subject to bounding_size{w in X, z in Y}: 0<=p[w,z]<=200;

```

```

subject to valid_inequalities{w in X, z in Y}: sum {i in Q, j in R}
d[i,j,w,z] >= T[w,z];

```

```

subject to LR: sum {w in X, z in Y} T[w,z]<=2;

```

AMPL Data File (for 20x20 grid):

```

set R := 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ;

```

```

set Q := 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ;

```

```

set Y := 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ;

```

```

set X := 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ;

```

```

param E:=

```

```

[1,1,1,1] 1

```

```

[1,1,1,2] 2

```

```

[1,1,1,3] 2

```

```

[1,1,1,4] 2

```

```

[1,1,1,5] 2

```

```

[1,1,1,6] 2

```

```

[1,1,1,7] 2

```

```

[1,1,1,8] 2

```

```

[1,1,1,9] 2

```

```

.

```

```

.

```

```

.

```

```

[20,20,20,12] 19.0309

```

```

[20,20,20,13] 18.451

```

```

[20,20,20,14] 17.7815

```

```

[20,20,20,15] 16.9897

```

```

[20,20,20,16] 16.0206

```

```

[20,20,20,17] 14.7712

```

```

[20,20,20,18] 13.0103

```

```

[20,20,20,19] 10

```

```

[20,20,20,20] 1

```

```

;

```

```

param C:=

```

```
[1,1,1,1] 0
[1,1,1,2] 1
[1,1,1,3] 1
[1,1,1,4] 1
[1,1,1,5] 1
[1,1,1,6] 1
[1,1,1,7] 1
[1,1,1,8] 1
[1,1,1,9] 1
[1,1,1,10] 1
[1,1,1,11] 1
[1,1,1,12] 1
[1,1,1,13] 1
[1,1,1,14] 1
.
.
.
[20,20,20,4] 1
[20,20,20,5] 1
[20,20,20,6] 1
[20,20,20,7] 1
[20,20,20,8] 1
[20,20,20,9] 1
[20,20,20,10] 1
[20,20,20,11] 1
[20,20,20,12] 1
[20,20,20,13] 1
[20,20,20,14] 1
[20,20,20,15] 1
[20,20,20,16] 1
[20,20,20,17] 1
[20,20,20,18] 1
[20,20,20,19] 1
[20,20,20,20] 0
;
```

AMPL Run File (for 20x20 grid):

```
option mipgap 0.001;
option omit_zero_rows 1;
```

```
option display_eps .001;
option solution_round 3;
option solver cplexamp;

model tlp2020.mod;
data tlp2020.dat;

option cplex_options 'mipgap=0.001;

let {w in X,z in Y}p[w,z].relax:=1;
solve;
printf"\n %10.3f \", _solve_time>>tlp2020.txt;

fix {w in X,z in Y}T[w,z];
let {w in X,z in Y}p[w,z].relax:=0;
solve;
printf"\ %10.3f \", _solve_time>> tlp2020.txt;
```