A Semidefinite Programming Model for the Facility Layout Problem

by

Elspeth Clair Adams

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2010

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

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

Abstract

The continuous facility layout problem consists of arranging a set of facilities so that no pair overlaps and the total sum of the pairwise connection costs (proportional to the center-to-center rectilinear distance) is minimized. This thesis presents a completely mixed integer semidefinite programming (MISDP) model for the continuous facility layout problem.

To begin we describe the problem in detail; discuss the conditions required for a feasible layout; and define quaternary variables. These variables are the basis of the MISDP model. We prove that the model is an exact formulation and a distinction is made between the constraints that semidefinite programming (SDP) optimization software can solve and those that must be relaxed. The latter are called exactness constraints and three possible exactness constraints are shown to be equivalent.

The main contribution of this thesis is the theoretical development of a MISDP model that is based on quaternary, as oppose to binary, variables; nevertheless preliminary computational results will be presented for problems with 5 to 20 facilities. The optimal solution is found for problems with 5 and 6 facilities, confirming the validity of the model; and the potential of the model is revealed as a new upper bound is found for an 11-facility problem.

Acknowledgements

First and foremost I would like to thank my supervisor Miguel Anjos. With his insight and support I found the direction for this thesis, with his patience and guidance it came to be and with his organization and recommendation to start typing early I am no longer still flipping through papers. But above all I appreciate the excitement he shared for math and matrices.

I would like to thank Professor Barlatt and Professor Mantin for their time and energy in reading my thesis and the helpful comments.

I am profoundly grateful to Jim Ostrowski and his help in everything from branching scheme improvements to LaTeXcommands. His door was always open whether I was confused or simply discouraged and without his motivation, support and his compassion this thesis would have been next to impossible.

Thank you Tiffany, Jen, Bissan, Christie, Joe, Juan and Alex for keeping me sane, it was no small task and I am infinitely grateful.

I would like to thank my mom, dad, Kelsey and Ryanne for their love and encouragement. Thank you for understanding when I was busy and for constantly trying to make things easier.

Last but not least Kyle, your contribution to this thesis is far greater than just 'port' and 'starboard'. Although you didn't always understand what I was saying, you were willing to listen. You were my sounding board, my first audience and as a result you were the voice in my head as I wrote this thesis.

You believed in me when I didn't, supported me when I couldn't and you were my voice of reason when I wasn't. You have been there every step of the way: through the ups and downs, the good, the bad and the downright ugly. You have been a constant and your unwavering friendship means the world to me.

Imagine you have a rectangle....

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Layout problems are concerned with the placement of facilities within a plant. A fundamental characteristic of layout problems is that the facilities are placed without overlap. Building from this basic definition gives rise to many variations and leads to numerous industrial applications; as a result this is a very active area of research [18],[9]. Layout models can be applied to the placement of machines within a manufacturing plant, very large scale integration (VLSI) design, the design of hospitals, schools and airports and backboard wiring, among many others [28] [11]. The effective placement of facilities impacts cost, productive and overall efficiency [30].

This thesis focuses on the Facility Layout Problem (FLP) which is concerned with arranging n indivisible facilities in n locations [18]. More specifically we focus on continuous facility layout problems (CFLP) with distance-based objectives as the layout problem addressed in this thesis falls into this category. However in order to distinguish this from discrete layout problems we begin by examining layout problems where the potential locations for facilities are given.

1.1 Facility Layout Problem

1.1.1 The Discrete vs Continuous Layout Problem

The quadratic assignment problem (QAP) was introduced by Koopmans and Beckmann [14] and is concerned with placing facilities at predefined locations to minimize the total flow between facilities. The condition that no facilities are overlapping is enforced by allowing only one facility to be placed at each location and restricting each facility to a single location. The QAP assumes that facilities have identical sizes. Relaxing this assumption gives the quadratic set covering problem (QSCP). In this problem the placement area is divided into blocks and the set of blocks that will be occupied for each assignment of a facility to a location is known. In addition to restricting each facility to one location and that each location only be assigned to one facility, the non-overlapping condition is enforced by ensuring that each block is only occupied by one facility [4].

The discrete QAP was extended for continuous layouts with distance-based objectives and was formulated as a MIP by Montreuil [20]. In continuous layout problems locations are not pre-defined and as a result the distances between each pair of facilities are continuous variables. This differs from both the QAP and QSCP in which the distance between any pair of facilities is based on the assignment of facilities to locations.

The assumption that facilities are rectangular is not required [13]. Under this assumption facilities can be defined as having fixed dimension or fixed area. In fixed/rigid block problems the width and height of each facility is given [7]. Fixed area formulations find the dimension of each facility in addition to finding its location. Meller, Narayanan and Vance [19], Sherali, Praticelli and Meller [27], Castillo and Westerlund [5] and others have proposed methods of linearizing the area constraint ($a_i = w_i h_i$); and non-linear methods have been developed by Anjos and Vannelli [3] and van Camp, Carter and Vannelli [31] and others. Aspect ratio constraints or upper and lower bounds on the area of each facility can be included in the model. These constraints prevent facilities from having unrealistic shapes, such as long and narrow.

In a recent survey Drira, Pierreval and Hajri-Gabouj [9] identified numerous characteristics that can be used to classify FLPs and proposed a tree representation to organize the different variations. Aside from the shape of the facilities, layout problems can be distinguished by the material handling path they describe. The material handling path is the path in which goods move and can restrict the placement of facilities. Drira et al. describes the following four potential paths:

1. Single-row problems require the placement of facilities to be in a row and only the length of each facility is defined. (Kim, Kim and Bobbie [12], Kumar, Hadjinicola and Lin [15], Anjos, Kennings and Vannelli [2], Amaral [1])

2. Multi-row problems place facilities in several rows with material able to transfer within and between rows. (Hassan [10])

3. In loop layouts facilities are placed in a closed ring and material is moved in only one direction. (Cheng, Gen and Tosawa [6], Potts and Whitehead [26])

4. Open field layouts allow facilities to be placed anywhere on a 2-dimensional plane. (Yang, Peters and Tu [34]).

With such a wide-array of facility layout problems, we will now refine out description even further to describe the variation that will be addressed in this thesis.

1.1.2 Problem Statement

In alignment with the tree representation of FLPs proposed by Drira et al. [9] we can describe the problem that this thesis will examine as the continuous facility layout problem for rectangular-rigid blocks on an open field. An exact solution approach will be taken via a mixed-integer semidefinite programming (MISDP) formulation and a branch-and-bound algorithm. We define this continuous facility layout problem and the required notation below

Problem Definition:

For n facilities, with given width w_i and height h_i, find the layout that will minimize the total pairwise connection cost so that no facilities are overlapping.

Parameters:

n = Number of facilities
w_i = Width of facility i
h_i = Height of facility i
c_{ij} = The per unit distance cost associated with the facilities i and j

Variables:

(x_i, y_i) = Coordinates of center of facility i
d^x_{ij} = Horizontal distance between the center of facilities i and j
d^y_{ij} = Vertical distance between the center of facilities i and j

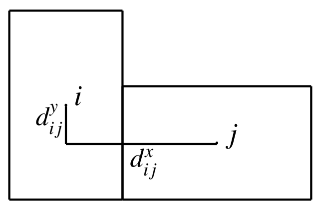


Figure 1.1: Horizontal and vertical distance between facilities i and j

There are two general types of constraints required for the facility layout problem. The first comes from the definition of variables and the second from the problem statement.

1. Facility coordinates and pairwise distance must be related, therefore using the definition of rectilinear distance Equations (1.1) must hold for a feasible layout.

$$d^x_{ij} = |x_j - x_i|, \qquad d^y_{ij} = |y_j - y_i| \tag{1.1}$$

2. According to the problem statement facilities cannot overlap. Intuitively this concept is extremely clear; however developing a mathematical representation describing what it means for two facilities to be non-overlapping is less intuitive. Section 2.1 will characterize the properties of non-overlapping facilities and Section 2.2 will translate these properties into the decision variables that will be used within the model.

The following details further specify the problem that will be studied.

1. Rectilinear distance will be used.

2. Facilities will be located by the coordinates of their center.

3. Distances will be measured from the center of each facility.

4. Pairwise connection costs are all non-negative.

In addition we make the following assumptions:

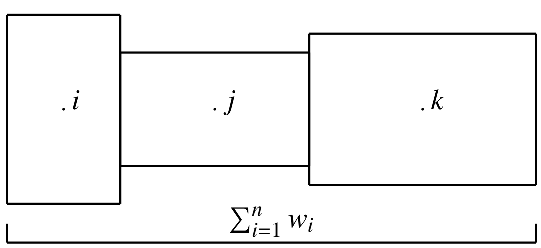
1. Facilities are rectangular with both strictly positive and finite dimensions.

2. The origin of the coordinate system is assumed to be in the lower left corner. This implies that for each facility i, $x_i, y_i \geq 0$. This assumption does not impact the layout as all facilities can be shifted on the plane and the relative arrangement (and total cost) is maintained. However, this assumption is important for computational purposes as this places all the continuous variables in the non-negative cone and they can easily be handled using semidefinite programming.

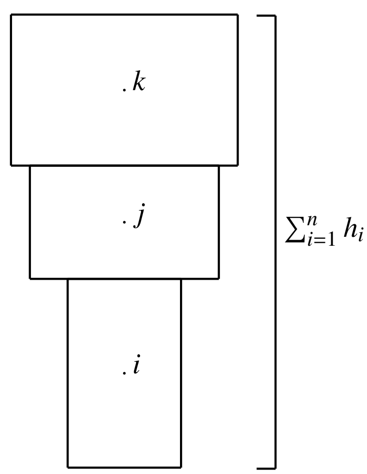
3. This formulation does not consider an outer perimeter in which all facilities are placed. We can, however, generate an outer perimeter that is finite and yet does not restrict the placement of the facilities. By considering the widest and highest potentially optimal layout we can create a perimeter without eliminating the optimal solution. An unrestictive outer perimeter is generated by placing facilites in a single row, either horizontally or vertically (Figure 1.2) and has dimensions:

$$\text{Maximum width required} = \sum_{i=1}^{n} w_i$$

$$\text{Maximum height required} = \sum_{i=1}^{n} h_i$$



(a) Widest Layout



(b) Highest Layout

Figure 1.2: Creating a finite and unrestrictive outer perimeter

1.2 Continuous Facility Layout Problem Formulations

A major difference between previous formulations and the one presented in this thesis is the method used to define the integer variables. As a result this section will examine how the definition of the decision variables has evolved.

1.2.1 Binary Variable Representations

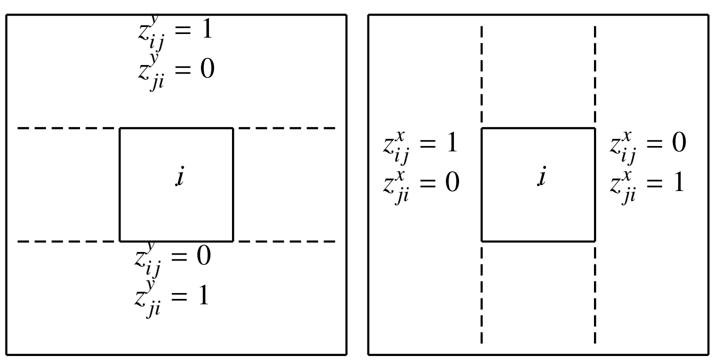
The facility layout problem requires a pair of facilities to be non-overlapping along the x or y axis; however it is also possible for facilities to be separated along both axes. In this section we will examine various binary variables used to formulate layout problems. For consistence we have used w_i and h_i for the width and height of facility i respectively since the notation for the dimensions of facilities varies between papers.

Montreuil [20] first formulated the CFLP as a mixed integer program (MIP) using two binary variables for every i and j where $i \neq j$. Based on the following definition a layout problem with n facilities requires $2n(n-1)$ binary variables:

$$z_{ij}^x = \begin{cases} 1 & \text{if } i \text{ precedes } j \text{ in } x \text{ direction} \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ij}^y = \begin{cases} 1 & \text{if } i \text{ precedes } j \text{ in } y \text{ direction} \\ 0 & \text{otherwise} \end{cases}$$

The non-overlapping conditions imply that at least one of $z_{ij}^x, z_{ji}^x, z_{ij}^y$ or z_{ji}^y equals 1. Figure 1.3 highlights this by indicating the regions in which facililty j would have to be completely within for the associated variable to equal 1. Note that it is possible for facility j to be completely within a region in both Figure 1.3(a) and 1.3(b). In this case facilities i and j are separated along both axes and the formulation will allow either (or both) of the variables to take the value 1.



(a) Non-overlapping along y-axis (b) Non-overlapping along x-axis

Figure 1.3: Variable definition for first MIP formulation of the CFLP

Defining variables in this manner implies each pair of facilities is represented by the indices ij and ji.

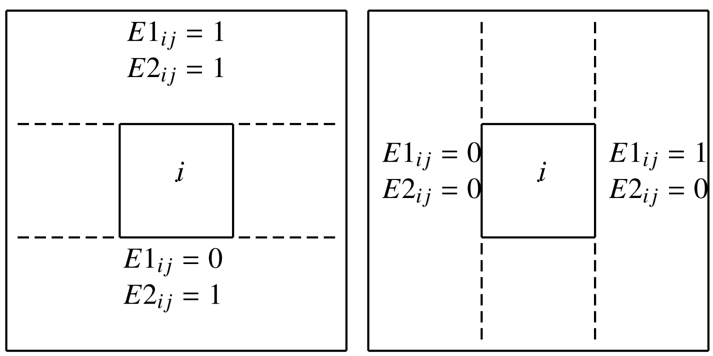
The duplication can be eliminated by only considering ij-pairs when $i < j$. Meller, Narayanan and Vance [19], reformulated Montreuil's model using this new definition of variables. In addition they tightened the area constraints and proposed valid inequalities.

Papageorgiou et al [25] proposed the following definition for the decision variables based on which non-overlapping condition is binding. In this model the area, as opposed to the width and height, of each facility was fixed.

$$\begin{array}{llll} If & x_i - x_j \geq \frac{1}{2}(w_i + w_j) & then & E1_{ij} = 0, E2_{ij} = 0 \\ If & x_j - x_i \geq \frac{1}{2}(w_i + w_j) & then & E1_{ij} = 1, E2_{ij} = 0 \\ If & y_i - y_j \geq \frac{1}{2}(h_i + h_j) & then & E1_{ij} = 0, E2_{ij} = 1 \\ If & y_j - y_i \geq \frac{1}{2}(h_i + h_j) & then & E1_{ij} = 1, E2_{ij} = 1 \end{array}$$

Castillo and Westerlund also created formulations using these variables and using cutting planes was able to solve problems to optimality with a guarantee that the area of each facility was within an ϵ-error of the required area [5].

Figure 1.4 show how the region around facility i is partitioned for the variables $E1_{ij}$ and $E2_{ij}$.



(a) Non-overlapping along y-axis (b) Non-overlapping along x-axis

Figure 1.4: Variable definition based on i, j pairs such that $i < j$

The formulation based on binary variables $E1_{ij}$ and $E2_{ij}$, (denoted LAYOUT2) includes the following non-overlapping constraints, where M is a large number.

$$\begin{array}{ll} x_i - x_j + M(E1_{ij} + E2_{ij}) \geq \frac{w_i + w_j}{2} & \forall i < j \\ x_j - x_i + M(1 - E1_{ij} + E2_{ij}) \geq \frac{w_i + w_j}{2} & \forall i < j \\ y_i - y_j + M(1 + E1_{ij} + E2_{ij}) \geq \frac{h_i + h_j}{2} & \forall i < j \\ y_j - y_i + M(2 - E1_{ij} - E2_{ij}) \geq \frac{h_i + h_j}{2} & \forall i < j \end{array}$$

LAYOUT2 included approximately 50% less binary variables and significantly less constraints than LAYOUT1, which was formulated using variables defined for all $i \neq j$. Computational results were run for both and LAYOUT2 was found to be computationally faster.

Özyurt and Realff [24] formulated the FLP without big M values by increasing the number of variables and constraints. The variables were:

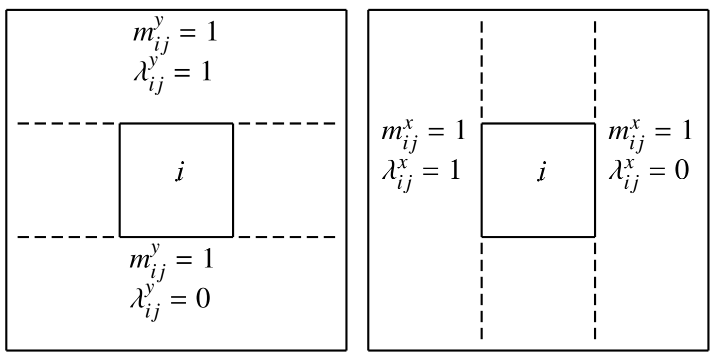
$$m_{ij}^x = \begin{cases} 1 & \text{if alignment disallowed in the } x\text{-direction} \\ 0 & \text{otherwise} \end{cases}$$

$$m_{ij}^y = \begin{cases} 1 & \text{if alignment disallowed in the } y\text{-direction} \\ 0 & \text{otherwise} \end{cases}$$

$$\lambda_{ij}^x = \begin{cases} 1 & \text{if unit } j \text{ to right of unit } i \\ 0 & \text{if unit } j \text{ to left of unit } i \end{cases}$$

$$\lambda_{ij}^y = \begin{cases} 1 & \text{if unit } j \text{ above unit } i \\ 0 & \text{if unit } j \text{ below unit } i \end{cases}$$

This formulation had $\frac{3}{2}n(n-1)$ binary variables and $5n^2 - n$ constraints and Figures 1.5(b) and 1.5(a) illustrates the partitioning of the region around facility i for the x and y variables respectively. Using both figures any undefined variables equal 0.



(a) Non-overlapping along y-axis (b) Non-overlapping along x-axis

Figure 1.5: Variable definition for MIP formulation without big-M values

Takouda, Anjos and Vannelli [29] formulated a semidefinite programming model in which the binary variables are described differently; however they partition the region around facility i in a manner similar to Figure 1.4. The first variable indicated the direction of separation and the second determined the relative arrangement of facilities i and j in the direction of separation:

$$\sigma_{ij} = \begin{cases} +1 & \text{if } i \text{ and } j \text{ are separated along } x \\ -1 & \text{if } i \text{ and } j \text{ are separated along } y \end{cases}$$

$$\alpha_{ij} = \begin{cases} +1 & \text{if } i \text{ precedes } j \text{ in the selected direction} \\ -1 & \text{if } j \text{ precedes } i \text{ in the selected direction} \end{cases}$$

This formulation also examined the CFLP with given areas and both the area and aspect ratio constraints were modelled using positive semidefinite matrices. Transitivity conditions were added and as a result the rank-one SDP relaxation matrix was formed from a vector of $1 + \binom{n}{2} + \binom{\binom{n}{2}}{2}$ binary variables.

1.2.2 Sequence-Pair Representations

Sequence-pair representation allows finitely many candidate solutions to be considered even though there exists infinately many feasible layouts. This method has been used by many authors to formulate CFLP with unequal sized facilites including Onodera, Taniguchi and Tamara [23]; Murata, Fujiyoshi, Nakatake and Kajitani [21]; and more recently by Meller, Chen and Sherali [17]; Liu and Meller [16]; and Xie and Sahinidis [32].

The two sequences in a sequence-pair representation were defined by Murata et al. [21] so that:
For any feasible layout Q, there always exists a pair of facility lists $(\alpha(Q), \beta(Q))$ that satisfies:

1) The lists $\alpha(Q)$ and $\beta(Q)$ are permutations of N

2) For any two facilities, their positions in the lists $\alpha(Q)$ and $\beta(Q)$ encode the relative location of these two facilities in the layout Q.

Xie and Sahinidis, using this definition, denoted the positions of facility i in $\alpha(Q)$ and $\beta(Q)$ as $\alpha^{-1}(Q,i)$ and $\beta^{-1}(Q,i)$, respectively. In addition, the following four sets were defined to refer to the four relative locations for any two facilities i and j:

$$\Gamma_{bb}(Q) := \{(i,j) : \alpha^{-1}(Q,i) < \alpha^{-1}(Q,j) \text{ and } \beta^{-1}(Q,i) < \beta^{-1}(Q,j)\} \text{where } (i,j) \in \Gamma_{bb} \text{ implies } x_i - x_j \le -\tfrac{1}{2}(w_i + w_j)$$
$$\Gamma_{ba}(Q) := \{(i,j) : \alpha^{-1}(Q,i) < \alpha^{-1}(Q,j) \text{ and } \beta^{-1}(Q,i) > \beta^{-1}(Q,j)\} \text{where } (i,j) \in \Gamma_{ba} \text{ implies } y_i - y_j \ge \tfrac{1}{2}(h_i + h_j)$$
$$\Gamma_{ab}(Q) := \{(i,j) : \alpha^{-1}(Q,i) > \alpha^{-1}(Q,j) \text{ and } \beta^{-1}(Q,i) < \beta^{-1}(Q,j)\} \text{where } (i,j) \in \Gamma_{ab} \text{ implies } y_i - y_j \le -\tfrac{1}{2}(h_i + h_j)$$
$$\Gamma_{aa}(Q) := \{(i,j) : \alpha^{-1}(Q,i) > \alpha^{-1}(Q,j) \text{ and } \beta^{-1}(Q,i) > \beta^{-1}(Q,j)\} \text{where } (i,j) \in \Gamma_{aa} \text{ implies } x_i - x_j \ge \tfrac{1}{2}(w_i + w_j)$$

This implies that for any two distinct facilities i and j: $(i,j) \in \Gamma_{aa}$ is equivalent to $(j,i) \in \Gamma_{bb}$ and $(i,j) \in \Gamma_{ba}$ is equivalent to $(j,i) \in \Gamma_{ab}$. The problem was formulated as a decoding problem with the non-overlapping condition described so that:

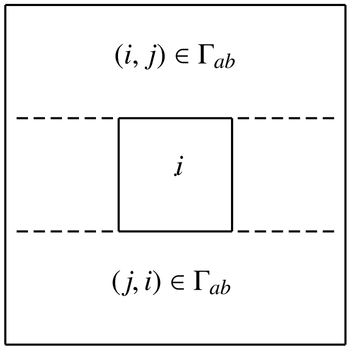
$$x_i - x_j \ge \tfrac{1}{2}(w_i + w_j) \;\; \text{if } (i,j) \in \Gamma_{bb}$$
$$y_i - y_j \ge -\tfrac{1}{2}(h_i + h_j) \;\; \text{if } (i,j) \in \Gamma_{ab}$$

Figure 1.6 partitions the region around facility i based on the sequence-pair representation if facility j was within that region. They developed a specialized branch-and-bound algorithm that was computationally faster then MIP formulations by Papageorgiou and Rotstein [25], Özyurt and Realff [24] and Yang and Peters [33]. In addition their algorithm was able to find the optimal solution for a previously unsolved problem with 9 facilities.

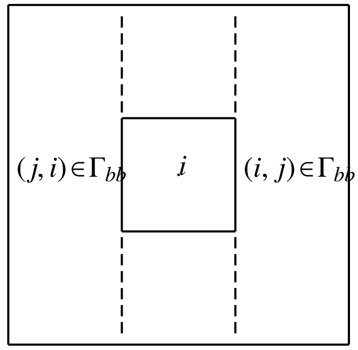
1.3 Semidefinite Programming

1.3.1 Facts of Positive Semidefinite Matrices

Semidefinite programming (SDP) is an optimization method that uses symmetric matrix variables. A linear objective is minimized subject to linear equality constraints on the matrix variables and the convex constraint



(a) Non-overlapping along y-axis



(b) Non-overlapping along x-axis

Figure 1.6: Variable definition for sequence-pair representation

that the matrices be positive semidefinite. The general form of an SDP problem is:

$$\begin{array}{ll} \min & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} x_{ij} \\ \text{s.t.} & \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{ij}^{k} x_{ij} = b^k \quad \text{for } k = 1 \dots m \\ & X \succeq 0 \end{array}$$

where x_{ij} refers to the $(i, j)^{th}$ element of matrix X. We will now examine positive semidefinite matrices. Let S^n denote the set of real symmetric $n \times n$ matrices.

Definition 1 (Positive Semidefiniteness)

For $X \in \mathbb{S}^n$, X is positive semidefinite if $x^T X x \geq 0$ for all $x \in \mathbb{R}^n$.

The following properties for positive semidefinite matrices hold:

Property 1 *If matrix $X \succeq 0$ then all principal minors are non-negative.*

Property 2 *If matrix $X \succeq 0$ then $Y^T X Y \succeq 0$ for any matrix $Y \in \mathbb{S}^n$.*

Property 3

$$\textit{Suppose } X = \begin{bmatrix} A & 0^{n_1 \times n_2} \\ 0^{n_2 \times n_1} & B \end{bmatrix}, \textit{ where } X \in \mathbb{S}^n, A \in \mathbb{S}^{n_1}, B \in \mathbb{S}^{n_2} \textit{ and } n_1 + n_2 = n, \textit{ then}$$
$$X \succeq 0 \Leftrightarrow A \succeq 0 \textit{ and } B \succeq 0$$

Property 4

$$\textit{Suppose } X = \begin{bmatrix} 1 & x_{12} & x_{13} \\ x_{12} & 1 & x_{23} \\ x_{13} & x_{23} & 1 \end{bmatrix} \succeq 0 \textit{ then } |x_{23}| = 1 \Rightarrow x_{12} x_{23} = x_{13}$$

Property 5 (Schur Complement)

$$\textit{Suppose } X = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}, \textit{ where } A \in \mathbb{S}^{n_1}, C \in \mathbb{S}^{n_2} \textit{ and } A \succ 0 \textit{ then}$$
$$X \succeq 0 \Leftrightarrow C - BA^{-1}B^T \succeq 0.$$

The final property we will discuss is an extension of the first property; however it is fundamental in understanding the construction of our constraints. As a result we have denoted it the ‘Zero Diagonal Property’.

Property 6 (Zero Diagonal)

$$\textit{Suppose } X \succeq 0 \textit{ and } x_{ii} = 0 \textit{ for some } i \in \{1 \ldots n\}, \textit{ then } x_{ij} = 0 \quad \forall j \in \{1 \ldots n\}$$

1.4 Concluding Remarks

Within this chapter we described the general class of facility layout problems and defined the specific variation that will be the focus of this thesis. The formulations and specifically the variable definitions of previous MIP, MISDP and sequence-pair models for continuous facility layout problems were examined. Finally semidefinite programming and properties of positive semidefinite matrices were introduced.

In Chapter 2 we will develop the conditions required for a layout to be feasible. Then, using characteristics of feasible layouts we will define a quaternary variable. Section 2.3 will present the complete model; with the latter part of this chapter focusing on examining the constraints that are relaxed to form the SDP relaxation (known as exactness constraints). Chapter 3 will examine the remaining constraints and show that the complete model is an exact formulation. Chapter 4 will focus on idenifying infeasible layouts by using transitivity conditions. These conditions will ultimately be defined in terms of the quaternary variable being used. A branch-and-bound algorithm was used to evaluate the SDP relaxation, with transitivity conditions being used to prune infeasible nodes. The specifics of the branch-and-bound algorithm and computational results will be presented in Chapter 5.

Chapter 2

The Model

2.1 Characterizing a Feasible Layout

Recall that the facility layout problem requires all pairs of facilities to be non-overlapping. In addition (1.1) must hold as it defines rectilinear distance. This leads to the following definition of a feasible layout.

Definition 2 (Feasible Layout)

The relative arrangement of facilities i and j is feasible if and only if all of the following conditions hold:

a) ***Non-overlapping Condition:*** *Facilities i and j are non-overlapping*

b) ***X-distance Condition:*** $d_{ij}^x = |x_j - x_i|$

c) ***Y-distance Condition:*** $d_{ij}^y = |y_j - y_i|$

Moreover, a layout of n facilities is feasible if the relative arrangement of every pair of facilities is feasible.

Within this section we will define the requirements for a pair of facilities to be non-overlapping and describe a method of classifying facilities that satisfy this definition. The following characteristics of a feasible layout will outline our discussions.

1. The axis along which facilities i and j do not overlap, which will be referred to as the **separation axis**.

2. The relative position of facility i with respect to facility j along the separation axis, which will be referred to as the **separation order**.

3. The relative position of facility i with respect to facility j along the axis that is not the separation axis, which will be referred to as the **non-separation order**.

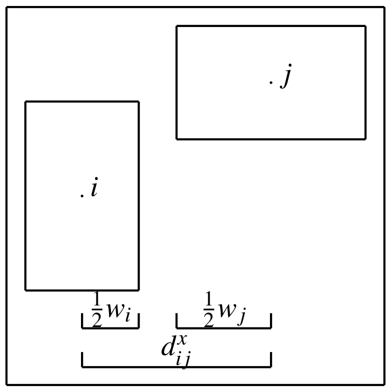
The separation axis relates to the non-overlapping requirement for a feasible layout. The separation order and non-separation order depend on the separation axis and enforces the relation between the distance and coordinate variables. These characteristics will be described and mathematically defined. We will then use this classification to introduce the quaternary variables β_{ij} that underpin the newly proposed model.

2.1.1 Separation Axis

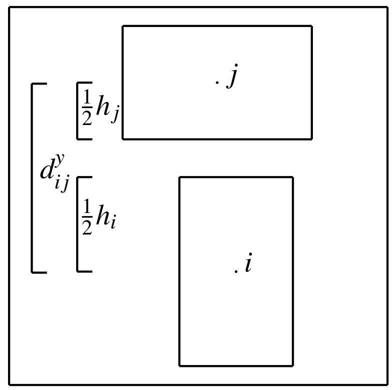
Within this section we will define the separation axis condition that will ensure that the relative arrangement of a pair of facilities is non-overlapping. To do this we will define what it means for facilities to be non-overlapping and then we will discuss how to break the symmetry that results from the non-overlapping definition. Finally the separation axis condition will be defined based on these two components.

Non-Overlapping

A pair of facilities is non-overlapping if they do not overlap along either the horizontal or vertical axis (Definition 3). This implies that either the horizontal distance is at least the sum of the half widths (Figure 2.1(a)) or the vertical distance is at least the sum of the half heights (Figure 2.1(b)).



(a) Horizontal



(b) Vertical

Figure 2.1: Separation Axis

Definition 3 (Non Overlapping Condition)
Facilities i and j are non-overlapping if and only if at least one of the following conditions holds:
a) $d^y_{ij} \geq \frac{1}{2}(h_i + h_j)$
b) $d^x_{ij} \geq \frac{1}{2}(w_i + w_j)$

Separation Axis Symmetry

Although only one condition is required for the facilities to be non-overlapping it is possible for both to be satisfied. The drawback of a pair of facilities satisfying both conditions is related to the number of feasible nodes within the branch-and-bound tree and will be discussed in depth in Chapter 4.
To examine when this happens consider Figure 2.2 where the region around a single facility i is divided into eight regions. The symbol $\updownarrow$ indicated that if facility j was placed completely within that region then facility i and j would satisfy the first non-overlapping condition. Likewise the symbol $\leftrightarrow$ indicates that if facility j was placed completely within that region then facility i and j would satisfy the second non-overlapping condition.

The figure shows that if facility j was placed in any of the four corner regions, facility i and j would satisfy both non-overlapping conditions. In order to make the non-overlapping conditions disjoint, we can define the conditions so that facilities are either a) separated vertically or b) not separated vertically but separated horizontally. In other words, facilities i and j are non-overlapping if and only if one of the following

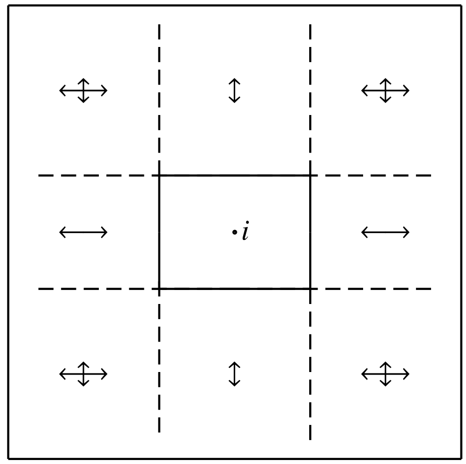


Figure 2.2: Symmetry of placing j around i

conditions hold:

a) $d^y_{ij} \geq \frac{1}{2}(h_i + h_j)$

b) $d^y_{ij} < \frac{1}{2}(h_i + h_j)$ and $d^x_{ij} \geq \frac{1}{2}(w_i + w_j)$

The problem with this classification is that condition b) requires defining an open set and introduces computational challenges. Therefore the separation axis is defined by taking the closure of condition b) and reduces but does not eliminate symmetry within the classification of the relative arrangement of a pair of facilities.

Definition 4 (Separation Axis Conditions)

Facilities i and j are non-overlapping if and only if at least one of the following separation axis conditions holds:

a) i and j have a ***Vertical Separation Axis*** $\Leftrightarrow \quad d^y_{ij} \geq \frac{1}{2}(h_i + h_j)$

b) i and j have a ***Horizontal Separation Axis*** $\Leftrightarrow \begin{cases} d^x_{ij} \geq \frac{1}{2}(w_i + w_j) \\ d^y_{ij} \leq \frac{1}{2}(h_i + h_j) \end{cases}$

If facilities i and j satisfy the separation axis conditions then they are non-overlapping; however in order for the layout to be feasible both distance conditions must also be satisfied. The X and Y distance conditions are nonlinear; however they can be linearized and subsequently enforced (if $c_{ij} \neq 0$) by minimizing the total sum of the separation distance.

$$\begin{array}{ll} \min & \sum_{\forall i,j} c_{ij} d^x_{ij} \\ d^x_{ij} & \geq x_j - x_i \\ d^x_{ij} & \geq -x_j + x_i \\ d^y_{ij} & \geq y_j - y_i \\ d^y_{ij} & \geq -y_j + y_i \end{array} \Leftrightarrow \begin{cases} d^x_{ij} = |x_j - x_i| \\ d^y_{ij} = |y_j - y_i| \end{cases} \forall i, j \text{ such that } c_{ij} \neq 0 \tag{2.1}$$

Since one of the separation axis conditions must hold for the layout to be feasible, additional constraints are added to the model, namely $d^x_{ij} \geq \frac{1}{2}(w_i + w_j)$ for a horizontal separation axis and $d^y_{ij} \geq \frac{1}{2}(h_i + h_j)$ for a vertical separation axis. In both of these cases the new constraint creates an additional lower bound on one of the distances and can prevent the linearization from enforcing the X or Y distance condition. The implications of adding the separation axis condition to the above model are as follows:

Case 1: Vertical Separation Axis

$$(2.1) \ \& \ d^y_{ij} \geq \tfrac{1}{2}(h_i + h_j) \Rightarrow \begin{cases} d^x_{ij} = \max[x_j - x_i, -x_j + x_i] \\ d^y_{ij} = \max[y_j - y_i, -y_j + y_i, \tfrac{1}{2}(h_i + h_j)] \end{cases}$$

Case 2: Horizontal Separation Axis

$$(2.1) \ \& \ d^x_{ij} \geq \tfrac{1}{2}(w_i + w_j) \Rightarrow \begin{cases} d^x_{ij} = \max[x_j - x_i, -x_j + x_i, \tfrac{1}{2}(w_i + w_j)] \\ d^y_{ij} = \max[y_j - y_i, -y_j + y_i] \end{cases}$$

For each case one of the distance conditions may not hold, namely the distance condition along the separation axis. Therefore the separation order condition explicitly states the relative order of facilities i and j so that the distance condition is satisfied. The separation order conditions are examined in Section 2.1.2. Although the traditional linearization may not work to enforce the distance-coordinate relationship along the separation axis, it will be used along the axis that is not the separation axis and defines the non-separation order conditions.

The horizontal and vertical separation order conditions each have two conditions (Section 2.1.2); however the horizontal and vertical non-separation order conditions are not of a disjunctive nature. This is because even once the separation axis has been determined the linearization discussed above can enforce the distance requirement along the axis that is not the separation axis.

2.1.2 Separation Order

The purpose of the separation order conditions are to relate distance variables (d^x_{ij}, d^y_{ij}) with coordinates (x_i, x_j or y_i, y_j) along the separation axis. Since the separation order is defined along the separation axis we will consider the explicit definition of separation order along each axis individually. We begin by considering the horizontal separation axis. Figure 2.3 illustrates both horizontal separation order conditions. In Figure 2.3(a) facility i is before j because preceeding along the x-axis one would reach i before j. Likewise facility i is after j in Figure 2.3(b) because preceeding along the x-axis one would reach i after j.

Since d^x_{ij} is required, by definition, to always be non-negative, knowing if x_i or x_j is larger allows us to know if $x_j - x_i \geq 0$ or if $-x_j + x_i \geq 0$, from this we define the horizontal separation order conditions. Moreover, the conditions of definition 5 are actually disjoint since if facilities i and j have a horizontal separation axis then d^x_{ij} is strictly positive (since the width of each facility is strictly positive).

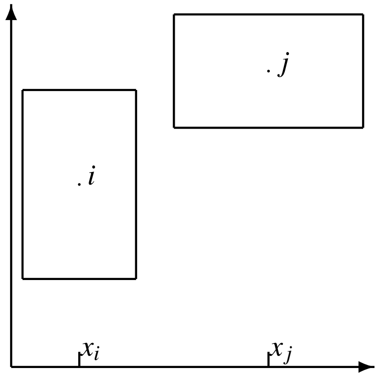
Definition 5 (Horizontal Separation Order Condition)

If facility i and j have a horizontal separation axis then the X-distance condition is satisfied if and only if one of the following horizontal separation order conditions holds:

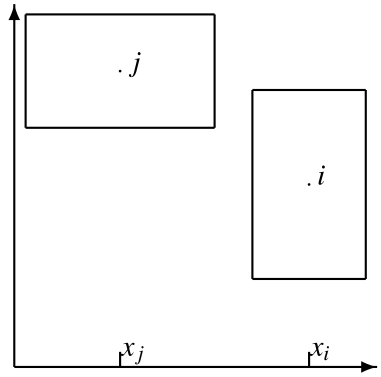
*a) facility i is **before** facility j* $\Leftrightarrow \ d^x_{ij} = x_j - x_i$

*b) facility i is **after** facility j* $\Leftrightarrow \ d^x_{ij} = -x_j + x_i$

Vertical separation order conditions are defined in the same way except that they relate vertical distance (d^y_{ij}) to vertical coordinates (y_i, y_j) and ensure the Y-distance feasibility condition.



(a) Facility i is before j



(b) Facility i is after j

Figure 2.3: Horizontal Separation Axis

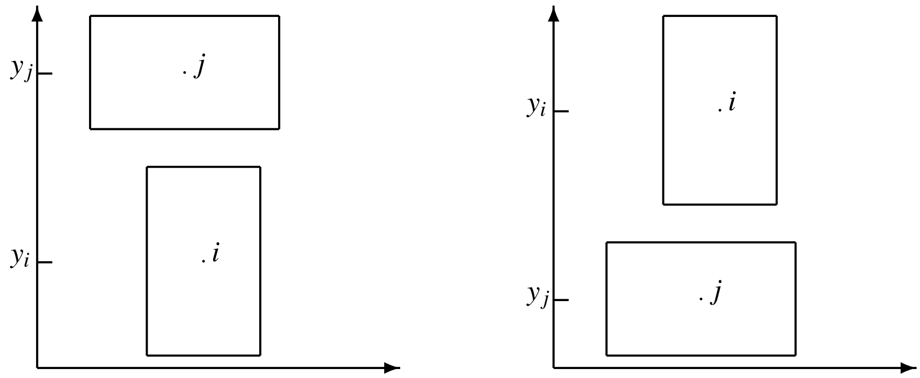
Definition 6 (Vertical Separation Order Condition)
If facility i and j have a vertical separation axis then the Y-distance condition is satisfied if and only if one of the following vertical separation order conditions holds:
*a) facility i is **before** facility j* $\Leftrightarrow$ $d^y_{ij} = y_j - y_i$
*b) facility i is **after** facility j* $\Leftrightarrow$ $d^y_{ij} = -y_j + y_i$

Figure 2.4 illustrate the vertical separation order conditions. When facility i and j have a vertical separation axis then i before j implies that proceeding along the y-axis facility i is before j indicating that $y_j - y_i \geq 0$. Likewise i after j implies that proceeding along the y-axis one would reach facility i after j and that $-y_j + y_i \geq 0$. These cases are disjoint because a vertical separation axis implies $d^y_{ij} > 0$. The reasoning is analogous to the discussion of $d^x_{ij} > 0$ except the assumption that the height of each facility is strictly positive is applied.

The motivation for using the terms "before" and "after" is to be able to refer to both horizontal and vertical separation simultanously. This will be useful in Chapter 3 when we examine the constraints that enforce these conditions. Generalizing our description of the terms "before" and "after" gives us the following: i before j implies proceeding along the separation axis one would reach i before j and i after j implies proceeding along the separation axis one would reach i after j.

2.1.3 Non-Separation Order

The purpose of separation axis and separation order conditions is to define the disjunctive nature of feasible layouts; however the purpose of the non-separation order conditions is to ensure feasibility for a range of distances. The horizontal non-separation order condition is defined when facilities i and j have a horizontal separation axis and relates vertical distance with vertical coordinates. Likewise the vertical non-separation order condition is defined when facilities i and j have a vertical separation axis and relates horizontal distance



(a) Facility i is before j (b) Facility i is after j

Figure 2.4: Vertical Separation Axis

with horizontal coordinates. The premise behind the horizontal non-separation order condition is to ensure d_{ij}^y is feasible for the entire range of 'acceptable' values and then the linearization will enforce $d_{ij}^y = |y_j - y_i|$. Similarly, the vertical non-separation order condition considers 'acceptable' values of d_{ij}^x.

For clarity we will defer defining the non-separation order conditions until Chapter 3 and the remainder of this section will be devoted to identifying the acceptable ranges for d_{ij}^y and d_{ij}^x.

- Distance is, by definition, positive; however these lower bounds can be strengthened by considering the separation axis condition.

$$\text{For horizontal separation axis:} \begin{cases} d_{ij}^x \geq \frac{1}{2}(w_i + w_j) \\ d_{ij}^y \geq 0 \end{cases}$$

$$\text{For vertical separation axis} \begin{cases} d_{ij}^x \geq 0 \\ d_{ij}^y \geq \frac{1}{2}(h_i + h_j) \end{cases}$$

- In Section 1.1.2 a finite yet unrestrictive outer perimeter was found. Regardless of the separation axis we can use this as an upper bound for the distances between facilities. However, in the case of horizontal separation a stronger upper bound exists for vertical distance. This is a result of the symmetry breaking decision that implies if facilities i and j have a horizontal separation axis then they are not separated vertically.

$$\text{For horizontal separation axis:} \begin{cases} d_{ij}^x \leq \sum_{k=1}^n w_k \\ d_{ij}^y \leq \frac{1}{2}(h_i + h_j) \end{cases}$$

$$\text{For vertical separation axis} \begin{cases} d_{ij}^x \leq \sum_{k=1}^n w_k \\ d_{ij}^y \leq \sum_{k=1}^n h_k \end{cases}$$

Combining these bounds we get the following ranges:

$$\text{For horizontal separation axis:}\begin{cases} d_{ij}^x \in [\frac{1}{2}(w_i + w_j), \sum_{k=1}^n w_k] \\ d_{ij}^y \in [0, \frac{1}{2}(h_i + h_j)] \end{cases}$$

$$\text{For vertical separation axis}\begin{cases} d_{ij}^x \in [0, \sum_{k=1}^n w_k] \\ d_{ij}^y \in [\frac{1}{2}(h_i + h_j), \sum_{k=1}^n h_k] \end{cases}$$

2.1.4 Summary of Non-Overlapping Conditions

In order to describe a feasible layout we examined 3 characteristics: separation axis, separation order and non-separation order. A feasible layout must satisfy one of the two separation axis conditions (horizontal or vertical) and one of the two separation order conditions (i before j or i after j). This produces four possible layout options, as shown in Figure 2.5. No distinction is needed with regards to the non-separation order since the linearization that enforces this condition is the same for all pairs of facilities.

		Separation Axis	
		Horizontal	Vertical
Separation Order	i before j	$d_{ij}^x \geq \frac{1}{2}(w_i + w_j)$ $d_{ij}^y \leq \frac{1}{2}(h_i + h_j)$ $d_{ij}^x = x_j - x_i$ $d_{ij}^y = \lvert y_j - y_i \rvert$	$d_{ij}^y \geq \frac{1}{2}(h_i + h_j)$ $d_{ij}^y = y_j - y_i$ $d_{ij}^x = \lvert x_j - x_i \rvert$
	i after j	$d_{ij}^x \geq \frac{1}{2}(w_i + w_j)$ $d_{ij}^y \leq \frac{1}{2}(h_i + h_j)$ $d_{ij}^x = -x_j + x_i$ $d_{ij}^y = \lvert y_j - y_i \rvert$	$d_{ij}^y \geq \frac{1}{2}(h_i + h_j)$ $d_{ij}^y = -y_j + y_i$ $d_{ij}^x = \lvert x_j - x_i \rvert$

Figure 2.5: The four feasible layouts produced from the separation axis and separation order conditions

Using separation axis and separation order conditions the region around facility i can be divided into four regions, corresponding to the four possible layout options. In Figure 2.6 each one of these layouts is represented by the relative arrangement between facility i and the facilities j, k, l or m.

		Separation Axis	
		Horizontal	Vertical
Separation Order	i before –	Pair i, k	Pair i, j
	i after –	Pair i, m	Pair i, l

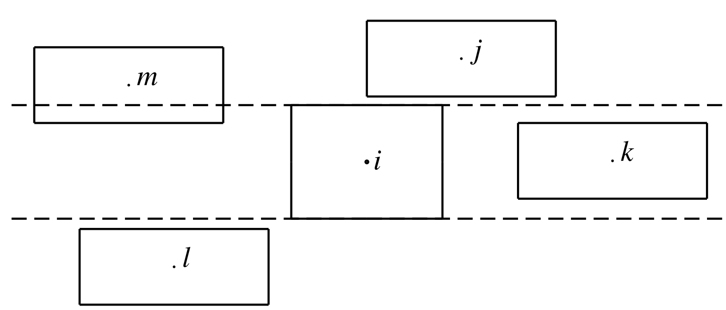


Figure 2.6: The Separation Axis and Separation Order as defined around facility i

Recall that in order to reduce symmetry, the vertical separation axis conditions is given preference over the horizontal separation axis. In Figure 2.6 pairs i, j and i, l have a vertical separation axis. This is regardless of the fact that pair i, l is also horizontally separated. Pairs i, k and i, m have a horizontal separation axis because both pairs are separated horizontally yet not separated vertically.

2.2 Defining a Decision Variable

The four possible arrangements are modelled by the quaternary variable β_{ij} for $i < j$ which will equal one of $\pm\sigma, \pm\mu$ where the parameters σ and μ satisfy $1 \geq \sigma > \mu > 0$. By relating the two layout characteristics previously discussed (separation axis and separation order) with two numerical characteristics (magnitude and sign) we obtain the following definitions for the discrete variable β_{ij}:

The separation axis for facilities i and j will be associated with the magnitude of β_{ij} :

$$\begin{array}{ll} |\beta_{ij}| = \sigma & \Rightarrow \text{horizontal separation axis} \\ |\beta_{ij}| = \mu & \Rightarrow \text{vertical separation axis} \end{array}$$

The separation order for facilities i and j will be associated to the sign of β_{ij} :

$$\begin{array}{ll} \beta_{ij} > 0 & \Rightarrow \text{facility } i \text{ is before facility } j \\ \beta_{ij} < 0 & \Rightarrow \text{facility } i \text{ is after facility } j \end{array}$$

Figure 2.7 combines the above definitions allowing each of the four possible arrangements to be associated with a value of β_{ij}.

		Separation Axis: Magnitude of Variable	
		Horizontal $|\beta_{ij}| = \sigma$	Vertical $|\beta_{ij}| = \mu$
Separation Order: Sign of Variable	i before j $\beta_{ij} > 0$	$\beta_{ij} = +\sigma$	$\beta_{ij} = +\mu$
	i after j $\beta_{ij} < 0$	$\beta_{ij} = -\sigma$	$\beta_{ij} = -\mu$

Figure 2.7: Chart for defining β_{ij}

Prior to examining the complete model, consider the following definition:

Definition 7 (Exactness Constraints)

Exactness constraints are the set of constraints that are:

1. *relaxed to form a SDP relaxation solvable by optimization software*

2. *but required for the model to be an exact formulation of the problem.*

Although the four possible arrangements of facility i and j are nicely represented by the four discrete values of β_{ij}, the assumption $\beta_{ij} \in \{\pm\sigma, \pm\mu\}$ is not an exactness constraint. In other words, although this "integer" requirement must be relaxed to form a SDP relaxation, this assumption alone is not sufficient for an exact formulation to hold. The formulation is driven by four forms of the discrete variables β_{ij}, namely $\beta_{ij}, \frac{1}{\beta_{ij}}, \beta_{ij}^2$ and $\frac{1}{\beta_{ij}^2}$ where the latter are linearized so that:

$$b_{ij} = \frac{1}{\beta_{ij}}, \alpha_{ij} = \beta_{ij}^2 \text{ and } a_{ij} = b_{ij}^2.$$

The purpose of requiring these specific variations is that together they form an exactness constraint for the model. Prior to examing the four dependent discrete variables and the relations required to form exactness constraints, we will present the model.

2.3 The Complete Model

Objective Function:

$$\min \quad \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij}(d_{ij}^x + d_{ij}^y) \tag{2.2}$$

Separation Axis Constraints, $\forall i < j$:

$$d_{ij}^x + s_{ij}^x - t_{ij}^x = \frac{1}{2}(w_i + w_j) \tag{2.3}$$

$$d_{ij}^y + s_{ij}^x - t_{ij}^x = \frac{1}{2}(h_i + h_j) \tag{2.4}$$

$$M_{sx} = \begin{bmatrix} \sigma^2 - \alpha_{ij} & s_{ij}^x \\ s_{ij}^x & \frac{[\frac{1}{2}(w_i+w_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \succeq 0 \tag{2.5}$$

$$M_{sy} = \begin{bmatrix} \alpha_{ij} - \mu^2 & s_{ij}^y \\ s_{ij}^y & \frac{[\frac{1}{2}(h_i+h_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \succeq 0 \tag{2.6}$$

$$M_{ty} = \begin{bmatrix} \sigma^2 - \alpha_{ij} & t_{ij}^y \\ t_{ij}^y & \frac{[(\sum_{i=1}^n h_i)-(h_i+h_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \succeq 0 \tag{2.7}$$

$$s_{ij}^x, s_{ij}^y, t_{ij}^x, t_{ij}^y \geq 0 \tag{2.8}$$

Separation Order Constraints, $\forall i < j$:

$$M_x = \begin{bmatrix} \sigma - \beta_{ij} & \alpha_{ij} - \sigma^2 & d_{ij}^x - x_j + x_i \\ \alpha_{ij} - \sigma^2 & \sigma + \beta_{ij} & d_{ij}^x + x_j - x_i \\ d_{ij}^x - x_j + x_i & d_{ij}^x + x_j - x_i & \max[\frac{2(\sum_{k=1}^n w_k)^2}{\sigma}, \frac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)}] \end{bmatrix} \succeq 0 \tag{2.9}$$

$$M_y = \begin{bmatrix} \frac{1}{\mu} - b_{ij} & \alpha_{ij} - \mu^2 & d^y_{ij} - y_j + y_i \\ \alpha_{ij} - \mu^2 & \frac{1}{\mu} + b_{ij} & d^y_{ij} + y_j - y_i \\ d^y_{ij} - y_j + y_i & d^y_{ij} + y_j - y_i & \max[\frac{(h_i+h_j)^2}{(\frac{1}{\mu}-\frac{1}{\sigma})(1-\sigma^4\mu^2+\sigma^2\mu^4)}, 2\mu(\sum_{k=1}^n h_k)^2] \end{bmatrix} \geq 0 \tag{2.10}$$

$$d^x_{ij} - x_j + x_i \geq 0,\ \ d^x_{ij} + x_j - x_i \geq 0,\ \ d^y_{ij} - y_j + y_i \geq 0,\ \ d^y_{ij} + y_j - y_i \geq 0 \tag{2.11}$$

$$d^x_{ij} \geq 0,\ \ d^y_{ij} \geq 0 \tag{2.12}$$

$$\sum_{k=1}^{n} w_k \geq d^x_{ij},\ \ \sum_{k=1}^{n} h_k \geq d^y_{ij} \tag{2.13}$$

Non-Negativity Constraints:

$$x_i \geq 0,\ \ y_i \geq 0 \qquad \forall i = \{1 \dots n\} \tag{2.14}$$

For Equivalence of Exactness Constraints, $\forall i < j$

$$M_\beta = \begin{bmatrix} 1 & \beta_{ij} & b_{ij} \\ \beta_{ij} & \alpha_{ij} & 1 \\ b_{ij} & 1 & a_{ij} \end{bmatrix} \geq 0 \tag{2.15}$$

$$\alpha_{ij} + \sigma^2\mu^2 a_{ij} = \sigma^2 + \mu^2 \tag{2.16}$$

$$\sigma^2 \geq \alpha_{ij} \geq \mu^2 \tag{2.17}$$

and any one of the following three exactness constraints, $\forall i < j$:

$$(\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) \in [(\sigma, \tfrac{1}{\sigma}, \sigma^2, \tfrac{1}{\sigma^2}), (-\sigma, -\tfrac{1}{\sigma}, \sigma^2, \tfrac{1}{\sigma^2}), (\mu, \tfrac{1}{\mu}, \mu^2, \tfrac{1}{\mu^2}), (-\mu, -\tfrac{1}{\mu}, \mu^2, \tfrac{1}{\mu^2})] \tag{ExC 1}$$

$$\text{rank}(M_\beta) = 1 \text{ and } \beta_{ij} \in [\pm\sigma, \pm\mu] \tag{ExC 2}$$

$$\beta_{ij} = \pm\sigma \text{ or } b_{ij} = \tfrac{1}{\pm\mu} \tag{ExC 3}$$

Since the three exactness assumptions (ExC 1), (ExC 2) and (ExC 3) are nonlinear, each of them would need to be relaxed if included in the model. The proof to show that they are all exactness constraints will proceed in two steps:

1. Given constraints (2.15), (2.16) and (2.17), we show that each of the exactness constraints are equivalent.

2. Given (ExC 1) and (2.2)-(2.14) we show that an exact formulation of the problem is achieved.

The first step will be shown in this chapter and the second will be the focus of the next chapter; however we begin by discussing the motivation for splitting the explanation of the model into these two steps. Figure 2.8 graphically represents the two steps we mentioned above and will be explained as we consider the following claims:

- Step 1, the equivalence of the three exactness constraints, is a result of constraints (2.15)-(2.17). The constraints are listed within the triangle as they are necessary for all three relationships.
- Given constraints (2.2)-(2.14), (ExC 1) is an exactness constraint. However, given only these constraints (ExC 2) and (ExC 3) are not. This feature is represented through the orientation of the triangle.
- In practice exactness constraints have to be relaxed and the SDP relaxation is computationally solved. Then using branch-and-bound, constraints are iteratively enforced until the global optimum is reached. Since these constraints are relaxed it is helpful to include constraints that will tighten the relaxation. Constraints (2.15)-(2.17) are included in the model to improve the relaxation.
- The purpose of separating the constraints in this manner and examining the model in two steps is solely to simplify the process of showing that an exact formulation is achieved. One could show that an exact formulation of the model holds starting from either (ExC 2) or (ExC 3) and using all constraints; however the process of relating the four discrete variables using constraints (2.15)-(2.17) would be repeated. To do this only once we will show that either exactness constraint is sufficient to find the explicit values of $\beta_{ij}, b_{ij}, \alpha_{ij}$, and a_{ij} and that the result is (ExC 1).

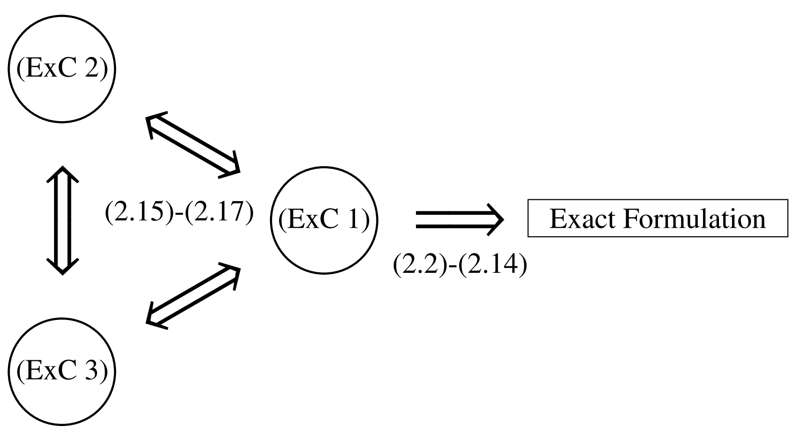


Figure 2.8: Representation of how the constraints and exactness constraints relate to achieve an exact formulation.

2.4 Equivalence of Exactness Constraints

We will show that both (ExC 2) and (ExC 3) are equivalent to (ExC 1). We will begin with the claim that (ExC 2) is sufficient for (ExC 1).

Lemma 1 (Exactness Constraints 1 & 2: Sufficiency)

Given (2.15), (2.16) *and* (2.17) *then:*
(ExC 2) ⇒ (ExC 1)

Proof

Let constraints (2.15), (2.16) and (2.17) hold and let $\text{rank}(M_\beta) = 1$ and $\beta_{ij} \in [\pm\sigma, \pm\mu]$.
By definition $\text{rank}(M_\beta) = 1$ implies all submatrices of dimension greater then 1 are singular. We begin by considering the determinant of the 2x2 submatrices.

$$\text{rank}(M_\beta) = 1 \Rightarrow \begin{cases} \alpha_{ij} - \beta_{ij}^2 & = 0 \quad \Rightarrow \alpha_{ij} = \beta_{ij}^2 \\ a_{ij} - b_{ij}^2 & = 0 \quad \Rightarrow a_{ij} = b_{ij}^2 \\ \alpha_{ij}a_{ij} - 1 & = 0 \quad \Rightarrow \alpha_{ij}a_{ij} = 1. \end{cases}$$

Now by taking the determinant of the full matrix and applying the substitutions above we get the following:

$$\begin{aligned} \det(M_\beta) = \alpha_{ij}a_{ij} + 2\beta_{ij}b_{ij} - 1 - \alpha_{ij}b_{ij}^2 - a_{ij}\beta_{ij}^2 &= 0 \\ \Rightarrow \quad \alpha_{ij}a_{ij} + 2\beta_{ij}b_{ij} - 1 - \alpha_{ij}a_{ij} - a_{ij}\alpha_{ij} &= 0 \quad (\because \alpha_{ij} = \beta_{ij}^2 \;\&\; a_{ij} = b_{ij}^2) \\ \Rightarrow \quad 2\beta_{ij}b_{ij} - 2 &= 0 \quad (\because \alpha_{ij}a_{ij} = 1) \\ \Rightarrow \quad b_{ij} &= \tfrac{1}{\beta_{ij}} \quad (\because \beta_{ij} \neq 0) \end{aligned}$$

We can now represent each variable in terms of β_{ij}:

$$b_{ij} = \frac{1}{\beta_{ij}}, \;\; \alpha_{ij} = \beta_{ij}^2, \;\; a_{ij} = b_{ij}^2 = \frac{1}{\beta_{ij}^2}.$$

Since $\beta_{ij} \in \{\pm\sigma, \pm\mu\}$ the four set of variables in (ExC 1) can be generated.

■

Now we will examine the necessary condition.

Lemma 2 (Exactness Constraints 1 & 2: Necessity)

(ExC 1) ⇒ (ExC 2)

Proof

Let (ExC 1) hold then $\beta_{ij} \in [\pm\sigma, \pm\mu]$. In addition for the four cases presented in (ExC 1) the following conditions hold:

$$b_{ij} = \frac{1}{\beta_{ij}}, \;\; \alpha_{ij} = \beta_{ij}^2, \;\; a_{ij} = \frac{1}{\beta_{ij}^2}.$$

To show that $\text{rank}(M_\beta) = 1$, we will first construct a rank-one matrix using just the variable β_{ij}. Since $\beta_{ij} \neq 0$ we can define the vector $v = \begin{bmatrix} 1 & \beta_{ij} & \frac{1}{\beta_{ij}} \end{bmatrix}$ and construct the following rank-one matrix:

$$vv^T = \begin{bmatrix} 1 & \beta_{ij} & \frac{1}{\beta_{ij}} \\ \beta_{ij} & \beta_{ij}^2 & 1 \\ \frac{1}{\beta_{ij}} & 1 & \frac{1}{\beta_{ij}^2} \end{bmatrix} = \begin{bmatrix} 1 & \beta_{ij} & b_{ij} \\ \beta_{ij} & \alpha_{ij} & 1 \\ b_{ij} & 1 & a_{ij} \end{bmatrix} = M_\beta.$$

Therefore $\beta_{ij} \in [\pm\sigma, \pm\mu]$ and $\text{rank}(M_\beta) = 1$ as required.

■

Now we will consider (ExC 1) and (ExC 3). We will not prove necessity since given (ExC 1) the condition that $\beta_{ij} = \pm\sigma$ or $b_{ij} = \pm\frac{1}{\mu}$ follows directly. It is not as straightforward to prove the sufficiency condition, therefore we will use the following steps as a guide to the proof. These steps act as motivation for the three lemmas that follow.

1. Bound a_{ij}
2. Define α_{ij} and a_{ij} under the exactness constraint
3. Relate β_{ij} and b_{ij}

Lemma 3 (Bound a_{ij})

$$(2.16) \textit{ and } (2.17) \Rightarrow \frac{1}{\mu^2} \geq a_{ij} \geq \frac{1}{\sigma^2}$$

Proof

Given constraint (2.16) we can solve for α_{ij}:

$$\alpha_{ij} + (\sigma^2\mu^2)a_{ij} = \sigma^2 + \mu^2 \Rightarrow \alpha_{ij} = \sigma^2 + \mu^2 - (\sigma^2\mu^2)a_{ij}.$$

Now using (2.17) we can find the bounds on a_{ij}:

$$\begin{array}{rrcl} & \sigma^2 \geq & \alpha_{ij} & \geq \mu^2 \\ \Rightarrow & \sigma^2 \geq & \sigma^2 + \mu^2 - (\sigma^2\mu^2)a_{ij} & \geq \mu^2 \\ \Rightarrow & -\mu^2 \geq & -(\sigma^2\mu^2)a_{ij} & \geq -\sigma^2 \\ \Rightarrow & \frac{1}{\sigma^2} \leq & a_{ij} & \leq \frac{1}{\mu^2}. \end{array}$$

■

Lemma 4 (Define α_{ij} and a_{ij})

$$\begin{array}{l} \textit{Given } (2.15), (2.16), (2.17) \textit{ then,} \\ 1) \quad \beta_{ij} = \pm\sigma \Rightarrow (\alpha_{ij}, a_{ij}) = (\sigma^2, \frac{1}{\sigma^2}) \\ 2) \quad b_{ij} = \frac{1}{\pm\mu} \Rightarrow (\alpha_{ij}, a_{ij}) = (\mu^2, \frac{1}{\mu^2}) \end{array}$$

Proof Let constraints (2.15), (2.16), (2.17) hold.

Since M_β is positive semidefinite, all the principal minors are non-negative (Property 1). Consider the following subset of prinicipal minors:

$$\alpha_{ij} - \beta_{ij}^2 \geq 0 \Rightarrow \alpha_{ij} \geq \beta_{ij}^2 \tag{2.18}$$

$$a_{ij}^2 - b_{ij}^2 \geq 0 \Rightarrow a_{ij} \geq b_{ij}^2. \tag{2.19}$$

Now consider the cases separately:

Case 1: Assume $\beta_{ij} = \pm\sigma$. First we will define α_{ij} by using the principal minor $\alpha_{ij} - \beta_{ij}^2 \geq 0$ and the upper bound on α_{ij}. Then with equation (2.16) we will define a_{ij}.

$$\left.\begin{array}{ll} (2.18) & \alpha_{ij} \geq \beta_{ij}^2 = (\pm\sigma)^2 \\ (2.17) & \sigma^2 \geq \alpha_{ij} \end{array}\right\} \Rightarrow \left.\begin{array}{l} \alpha_{ij} = \beta_{ij}^2 = \sigma^2 \\ \\ \alpha_{ij} + (\sigma^2\mu^2)a_{ij} = \sigma^2 + \mu^2 \end{array}\right\} \Rightarrow a_{ij} = \tfrac{1}{\sigma^2}.$$
$$(2.16)$$

Therefore $\beta_{ij} = \pm\sigma \Rightarrow (\alpha_{ij}, a_{ij}) = (\sigma^2, \frac{1}{\sigma^2})$.

Case 2: Assume $b_{ij} = \pm\frac{1}{\mu}$. By Lemma 3 constraints (2.16) and (2.17) imply $\frac{1}{\mu^2} \geq a_{ij}$. Now we will define a_{ij} by using the principal minor $a_{ij} - b_{ij}^2 \geq 0$ with the upper bound on a_{ij}. Then with equation (2.16) we will define α_{ij}.

$$\left.\begin{array}{ll} (2.19) & a_{ij} - b_{ij}^2 = a_{ij} - (\pm\frac{1}{\mu})^2 \geq 0 \\ (3) & \frac{1}{\mu^2} \geq a_{ij} \end{array}\right\} \Rightarrow \left.\begin{array}{l} a_{ij} = \frac{1}{\mu^2} \\ \\ \alpha_{ij} + (\sigma^2\mu^2)a_{ij} = \sigma^2 + \mu^2 \end{array}\right\} \Rightarrow \alpha_{ij} = \mu^2.$$
$$(2.16)$$

Therefore $b_{ij} = \pm\frac{1}{\mu} \Rightarrow (\alpha_{ij}, a_{ij}) = (\mu^2, \frac{1}{\mu^2})$.

∎

Lemma 5 (Relate β_{ij} and b_{ij})

$$\textit{Given } (2.15), (2.17) \textit{ and } \alpha_{ij}a_{ij} = 1 \textit{ then}$$
$$\alpha_{ij} = \beta_{ij}^2 \textit{ or } \alpha_{ij} = \frac{1}{b_{ij}^2} \Rightarrow \beta_{ij}b_{ij} = 1$$

Proof Using constraint (2.17) and Lemma 4 we can define the matrix X such that: $X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{a_{ij}} & 0 \\ 0 & 0 & \sqrt{\alpha_{ij}} \end{bmatrix}$.

Since $M_\beta \geq 0$ Property 2 implies $XM_\beta X^T$ is also positive semidefinite.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{a_{ij}} & 0 \\ 0 & 0 & \sqrt{\alpha_{ij}} \end{bmatrix} \begin{bmatrix} 1 & \beta_{ij} & b_{ij} \\ \beta_{ij} & \alpha_{ij} & 1 \\ b_{ij} & 1 & a_{ij} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{a_{ij}} & 0 \\ 0 & 0 & \sqrt{\alpha_{ij}} \end{bmatrix} \geq 0$$

$$\Rightarrow \begin{bmatrix} 1 & \beta_{ij}\sqrt{a_{ij}} & b_{ij}\sqrt{\alpha_{ij}} \\ \beta_{ij}\sqrt{a_{ij}} & \alpha_{ij}(\sqrt{a_{ij}})^2 & \sqrt{a_{ij}}\sqrt{\alpha_{ij}} \\ b_{ij}\sqrt{\alpha_{ij}} & \sqrt{a_{ij}}\sqrt{\alpha_{ij}} & a_{ij}(\sqrt{\alpha_{ij}})^2 \end{bmatrix} \geq 0$$

$$\Rightarrow \begin{bmatrix} 1 & \beta_{ij}\sqrt{a_{ij}} & b_{ij}\sqrt{\alpha_{ij}} \\ \beta_{ij}\sqrt{a_{ij}} & 1 & 1 \\ b_{ij}\sqrt{\alpha_{ij}} & 1 & 1 \end{bmatrix} \geq 0 \quad (\text{ since } \alpha_{ij} = \tfrac{1}{a_{ij}}).$$

The 2x2 submatrix of ones implies that $\beta_{ij}\sqrt{a_{ij}} = b_{ij}\sqrt{\alpha_{ij}}$ (Property 4).

Now using this equation consider the cases in turn.

Case 1: Let $\alpha_{ij} = \beta_{ij}^2$, then:

$$\begin{aligned} \beta_{ij}\sqrt{a_{ij}} &= b_{ij}\sqrt{\alpha_{ij}} & \\ \Rightarrow \quad \beta_{ij} &= b_{ij}\alpha_{ij} & (\because \sqrt{\tfrac{\alpha_{ij}}{a_{ij}}} = \alpha_{ij} > 0) \\ \Rightarrow \quad \beta_{ij} &= b_{ij}\beta_{ij}^2 & (\because \alpha_{ij} = \beta_{ij}^2 \neq 0) \\ \Rightarrow \quad 1 &= b_{ij}\beta_{ij}. & \end{aligned}$$

Case 2: Let $a = b_{ij}^2$. Then since $a_{ij} > 0$ we know that $b_{ij} \neq 0$, then:

$$\begin{aligned} \beta_{ij}\sqrt{a_{ij}} &= b_{ij}\sqrt{\alpha_{ij}} & \\ \Rightarrow \quad \beta_{ij}a_{ij} &= b_{ij} & (\because \sqrt{\tfrac{a_{ij}}{\alpha_{ij}}} = a_{ij} > 0) \\ \Rightarrow \quad \beta_{ij}b_{ij}^2 &= b_{ij} & (\because a_{ij} = b_{ij}^2 \neq 0) \\ \Rightarrow \quad \beta_{ij}b_{ij} &= 1. & \end{aligned}$$

■

We can now combine these lemmas to prove that (ExC 3) is sufficient for (ExC 1).

Lemma 6

Given (2.15), (2.16) *and* (2.17) *then*
(ExC 3) ⇒ (ExC 1)

Proof

Let constraints (2.15), (2.16) and (2.17) hold, then consider the cases in turn.

Case 1: Let $\beta_{ij} = \pm\sigma$.

Constraints (2.15), (2.16), (2.17) & $\beta_{ij} = \pm\sigma \Rightarrow (\alpha_{ij}, a_{ij}) = (\sigma^2, \frac{1}{\sigma^2})$ (by Lemma 4).
Now since $\alpha_{ij}a_{ij} = \sigma^2\frac{1}{\sigma^2} = 1$ & $\alpha_{ij} = \beta_{ij}^2$ Lemma 5 can be used to show that $\beta_{ij}b_{ij} = 1$.

$$\text{Therefore} \begin{cases} \beta_{ij} = \sigma & \Rightarrow (\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) = (\sigma, \frac{1}{\sigma}, \sigma^2, \frac{1}{\sigma^2}) \\ \beta_{ij} = -\sigma & \Rightarrow (\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) = (-\sigma, -\frac{1}{\sigma}, \sigma^2, \frac{1}{\sigma^2}). \end{cases}$$

Case 2: Let $b_{ij} = \pm\frac{1}{\mu}$.

Constraints (2.15), (2.16), (2.17) & $b_{ij} = \pm\frac{1}{\mu} \Rightarrow (\alpha_{ij}, a_{ij}) = (\mu^2, \frac{1}{\mu^2})$ (by Lemma 4).
Now since $\alpha_{ij}a_{ij} = \mu^2\frac{1}{\mu^2} = 1$ & $a_{ij} = b_{ij}^2$ Lemma 5 can be used to show that $\beta_{ij}b_{ij} = 1$.

$$\text{Therefore} \begin{cases} \beta_{ij} = \mu & \Rightarrow (\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) = (\mu, \frac{1}{\mu}, \mu^2, \frac{1}{\mu^2}) \\ \beta_{ij} = -\mu & \Rightarrow (\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) = (-\mu, -\frac{1}{\mu}, \mu^2, \frac{1}{\mu^2}). \end{cases}$$

■

Finally we can combine the above lemmas to prove that the exactness constraints are equivalent.

Theorem 1 *Given* (2.15),(2.16) *and* (2.17) *then* (ExC 1), (ExC 2) *and* (ExC 3) *are equivalent.*

Proof

By Lemma 1: (ExC 2) $\Rightarrow$ (ExC 1)
By Lemma 2: (ExC 1) $\Rightarrow$ (ExC 2)
therefore (ExC 1) $\Leftrightarrow$ (ExC 2).

By Lemma 6: (ExC 3) $\Rightarrow$ (ExC 1)
By (ExC 1) definition: (ExC 1) $\Rightarrow$ (ExC 3)
therefore (ExC 1) $\Leftrightarrow$ (ExC 3).

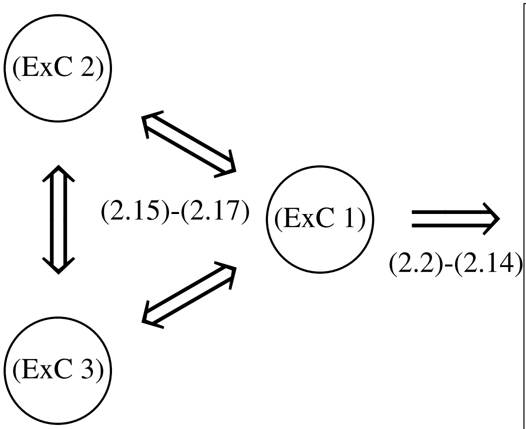
■

2.5 Concluding Remarks

Within this chapter we discussed the requirements for a layout to be feasible, stated our proposed semidefinite programming model and proved the equivalence of three exactness constraints. To show the connection between these topics consider the following:

- The conditions necessary for a feasible layout describe an exact formulation (Figure 2.5).
- An exactness constraint is used to enforce an exact formulation (Figure 2.8).

Both of these topics relate directly to exact formulations, therefore by combining the figures mentioned above we can highlight the connection between the topics examined in this chapter (Figure 2.9). In Section 2.2 the exactness constraints were proven to be equivalent given constraints (2.15)-(2.17) and we claim that one could be used to enforce an exact formulation. An exact formulation requires a feasible layout which was broken down into 3 conditions in Section 2.1.



		Separation Axis	
		Horizontal	Vertical
Separation Order	i before j	$d^x_{ij} \geq \frac{1}{2}(w_i + w_j)$ $d^y_{ij} \leq \frac{1}{2}(h_i + h_j)$ $d^x_{ij} = x_j - x_i$ $d^y_{ij} = \lvert y_j - y_i \rvert$	$d^y_{ij} \geq \frac{1}{2}(h_i + h_j)$ $d^y_{ij} = y_j - y_i$ $d^x_{ij} = \lvert x_j - x_i \rvert$
	i after j	$d^x_{ij} \geq \frac{1}{2}(w_i + w_j)$ $d^y_{ij} \leq \frac{1}{2}(h_i + h_j)$ $d^x_{ij} = -x_j + x_i$ $d^y_{ij} = \lvert y_j - y_i \rvert$	$d^y_{ij} \geq \frac{1}{2}(h_i + h_j)$ $d^y_{ij} = -y_j + y_i$ $d^x_{ij} = \lvert x_j - x_i \rvert$

Figure 2.9: The connection between exactness constraints and feasibility conditions.

(ExC 1) defines the set $(\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij})$ so that $b_{ij}, \alpha_{ij}, a_{ij}$ are dependent on the quaternary decision variable $\beta_{ij} \in \{\pm\sigma, \pm\mu\}$. These four sets correspond to the four layout options described by the separation axis and separation order of facilities i and j and, by definition, the separation axis and separation order require specific non-overlapping, X-distance and Y-distance condition to be satisfied. The focus of the following chapter will be to show that given (ExC 1), constraints (2.3)-(2.14) correctly define the feasible set for the CFLP of interest.

Chapter 3

Constructing the Constraints

The purpose of this chapter is to connect the exactness constraint (ExC 1) to a feasible solution using constraints (2.3)-(2.14) . Recall that although the model can be driven by different exactness constraints, (ExC 1) will be used for simplicity of notation.

The four elements of (ExC 1) correspond to the four possible layout options and each element consists of an assignment for the variables $(\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij})$. Figure 3.1 lists each element and associates the necessary conditions for a feasible layout. The constraints above each arrow indicate how each condition will be enforced. In addition separation axis, separation order and non-separation order conditions are aligned as this denotes the sequence within this chapter in which the conditions will be examined.

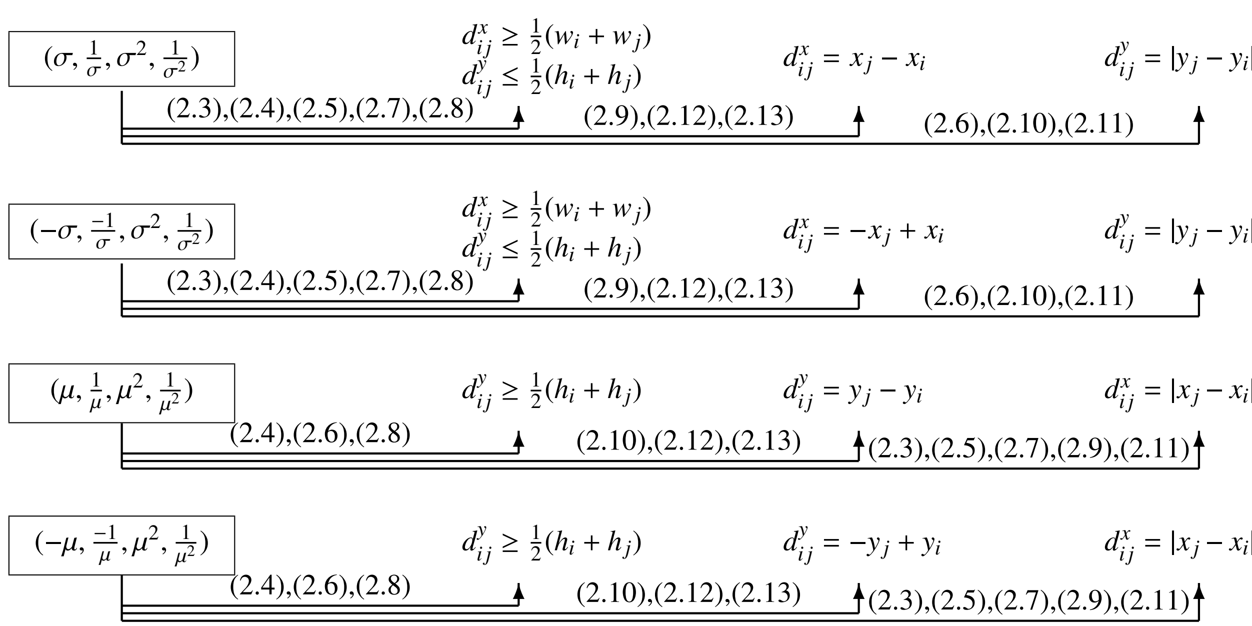


Figure 3.1: For each element of (ExC 1) the constraints required for separation axis, separation order and non-separation order conditions to hold

3.1 Separation Axis Constraints

Recall that the separation axis defines the axis (either horizontal or vertical) that facilities i and j are non-overlapping and that preference is given to the vertical axis if facilities are separated along both axes. In addition the separation axis is defined by the magnitude of the decision variable β_{ij}. Note that given (ExC 1) if $\beta_{ij} = \pm\sigma$ then $\alpha_{ij} = \sigma^2$ and if $\beta_{ij} = \pm\mu$ then $\alpha_{ij} = \mu^2$. Therefore for simplicity of notation we will use α_{ij} (as opposed to $|\beta_{ij}|$) to represent horizontal and vertical separation axes respectively. We are now able to define the purpose of the separation axis constraints as:

$$\begin{aligned} &\text{If } \alpha_{ij} = \sigma^2 \text{ then } d_{ij}^x \geq \tfrac{1}{2}(w_i + w_j) \ \ \& \ \ d_{ij}^y \leq \tfrac{1}{2}(h_i + h_j) \\ &\text{If } \alpha_{ij} = \mu^2 \text{ then } d_{ij}^y \geq \tfrac{1}{2}(h_i + h_j) \end{aligned}$$

To make constraints that hold for either $\alpha_{ij} = \sigma^2$ or $\alpha_{ij} = \mu^2$, slack and surplus variables are added to the inequalities. This gives us constraints (2.3) and (2.4), that are included in the model for each i, j pair. These constraints along with constraints (2.8) are shown below.

$$\begin{aligned} d_{ij}^x + s_{ij}^x - t_{ij}^x &= \tfrac{1}{2}(w_i + w_j) \\ d_{ij}^y + s_{ij}^y - t_{ij}^y &= \tfrac{1}{2}(h_i + h_j) \\ s_{ij}^x,\ t_{ij}^x \geq 0,\ s_{ij}^y,\ t_{ij}^y &\geq 0 \end{aligned}$$

The intuition behind the separation axis constraints lies within these constraints since given (2.3), (2.4) and (2.8), it is straightforward to observe that:

$$\begin{aligned} s_{ij}^x = 0 &\Rightarrow d_{ij}^x \geq \tfrac{1}{2}(w_i + w_j) \\ s_{ij}^y = 0 &\Rightarrow d_{ij}^y \geq \tfrac{1}{2}(h_i + h_j) \\ t_{ij}^y = 0 &\Rightarrow d_{ij}^y \leq \tfrac{1}{2}(h_i + h_j) \end{aligned}$$

This allows us to redefine the purpose of the separation axis constraints as:

$$\begin{aligned} &\text{If } \alpha_{ij} = \sigma^2 \text{ then } \begin{cases} s_{ij}^x = 0 \Rightarrow d_{ij}^x \geq \tfrac{1}{2}(w_i + w_j) \\ t_{ij}^y = 0 \Rightarrow d_{ij}^y \leq \tfrac{1}{2}(h_i + h_j) \end{cases} \\ &\text{If } \alpha_{ij} = \mu^2 \text{ then } s_{ij}^y = 0 \Rightarrow d_{ij}^y \geq \tfrac{1}{2}(h_i + h_j) \end{aligned}$$

Motivated by this new definition we can now turn our attention to relating α_{ij} to the slack and surplus variables. Each of the three implications are enforced through one of constraints (2.5)-(2.7). The 2×2 positive semidefinite matrices are shown below in the order we will examine them.

$$M_{sx} = \begin{bmatrix} \sigma^2 - \alpha_{ij} & s_{ij}^x \\ s_{ij}^x & \frac{[\frac{1}{2}(w_i + w_j)]^2}{\sigma^2 - \mu^2} \end{bmatrix} \succeq 0$$

$$M_{sy} = \begin{bmatrix} \alpha_{ij} - \mu^2 & s^y_{ij} \\ s^y_{ij} & \frac{[\frac{1}{2}(h_i+h_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \geq 0$$

$$M_{ty} = \begin{bmatrix} \sigma^2 - \alpha_{ij} & t^y_{ij} \\ t^y_{ij} & [(\sum_{i=1}^n h_i) - (h_i + h_j)]^2 \end{bmatrix} \geq 0$$

Each of these constraints works in a similar manner: when the (1,1)-element is set to zero then the slack/surplus variable on the off-diagonal is forced to zero. This effect is a direct consequence of the Zero Diagonal Property (Property 6). This defines the purpose of constraints (2.5),(2.6) and (2.7) for one of the two values of α_{ij}; however these constraints must still hold for the other value of α_{ij} (i.e. the value that does not fix the (1,1)-element to zero). The constraints, for this 2nd case, will generate upper bounds for the slack/surplus variable on the off-diagonal that will be used later in Chapter 3.

We will examine constraints (2.5)-(2.7) in turn each time considering α_{ij} equal to both σ^2 and μ^2 and showing the implication on the slack and surplus variables, namely that one will be fixed to zero while the other will be bounded above. The implication of the constraints are summarized in Table 3.1.

	If $\alpha_{ij} = \sigma^2$	If $\alpha_{ij} = \mu^2$
(2.5) ⇒	$s^x_{ij} = 0$	$s^x_{ij} \leq \frac{1}{2}(w_i + w_j)$
(2.6)⇒	$s^y_{ij} \leq \frac{1}{2}(h_i + h_j)$	$s^y_{ij} = 0$
(2.7)⇒	$t^y_{ij} = 0$	$t^y_{ij} \leq [(\sum_{i=1}^n h_i) - (h_i + h_j)]^2$

Table 3.1: Implication of separation axis constraints on the slack/surplus variables

The following three lemmas will prove these implications. We begin with constraint (2.5).

Lemma 7 (Bounding slack variable s^x_{ij})

Given (2.5) *and* $s^x_{ij} \geq 0$ *then*

$$\alpha_{ij} = \begin{cases} \sigma^2 & \Rightarrow s^x_{ij} = 0 \\ \mu^2 & \Rightarrow s^x_{ij} \leq \frac{1}{2}(w_i + w_j) \end{cases}$$

Proof

Let constraints (2.5) and $s^x_{ij} \geq 0$ hold. Now consider the cases in turn.

Case 1: Let $\alpha_{ij} = \sigma^2$.

Then constraint (2.5) becomes: $M_{sx} = \begin{bmatrix} 0 & s^x_{ij} \\ s^x_{ij} & \frac{[\frac{1}{2}(w_i+w_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \geq 0.$

The zero entry on the diagonal implies (by Property 6) that $s^x_{ij} = 0$.

Case 2: Let $\alpha_{ij} = \mu^2$.

Then constraint (2.5) becomes: $M_{sx} = \begin{bmatrix} \sigma^2 - \mu^2 & s^x_{ij} \\ s^x_{ij} & \frac{[\frac{1}{2}(w_i+w_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \geq 0.$

By examining the principal minors (Property 1) $M_{sy} \geq 0 \Leftrightarrow \begin{cases} \sigma^2 - \mu^2 \geq 0 \\ \frac{[\frac{1}{2}(w_i+w_j)]^2}{\sigma^2-\mu^2} \geq 0 \\ (\sigma^2-\mu^2)(\frac{[\frac{1}{2}(w_i+w_j)]^2}{\sigma^2-\mu^2}) - (s^x_{ij})^2 \geq 0. \end{cases}$

Both the first and second principal minors are constants that are, by definition, non-negative. Therefore only the third principal minor bounds s^x_{ij}. Simplifying this expression we get the following:

$$\begin{array}{rrll} & (\sigma^2-\mu^2)(\frac{[\frac{1}{2}(w_i+w_j)]^2}{\sigma^2-\mu^2}) - (s^x_{ij})^2 & \geq 0 & \\ \Rightarrow & (\frac{1}{2}(w_i+w_j))^2 & \geq (s^x_{ij})^2 & \\ \Rightarrow & \frac{1}{2}(w_i+w_j) & \geq s^x_{ij} & \text{since } s^x_{ij} \geq 0. \end{array}$$

■

A similar result can be shown for constraint (2.6) and slack variable s^y_{ij}.

Lemma 8 (Bounding slack variable s^y_{ij})

$$\begin{array}{rl} & \textit{Given } (2.6) \textit{ and } s^y_{ij} \geq 0. \\ \textit{Then} & \alpha_{ij} = \sigma^2 \Rightarrow s^y_{ij} \leq \frac{1}{2}(h_i+h_j) \\ \textit{and} & \alpha_{ij} = \mu^2 \Rightarrow s^y_{ij} = 0 \end{array}$$

Proof

Let constraints (2.6) and $s^y_{ij} \geq 0$ hold. Now consider the cases in turn.

Case 1: Let $\alpha_{ij} = \sigma^2$.

Then constraint (2.6) becomes: $M_{sy} = \begin{bmatrix} \sigma^2-\mu^2 & s^y_{ij} \\ s^y_{ij} & \frac{[\frac{1}{2}(h_i+h_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \geq 0.$

By examining the principal minors (Property 1) $M_{sy} \geq 0 \Leftrightarrow \begin{cases} \sigma^2 - \mu^2 \geq 0 \\ \frac{[\frac{1}{2}(h_i+h_j)]^2}{\sigma^2-\mu^2} \geq 0 \\ (\sigma^2-\mu^2)(\frac{[\frac{1}{2}(h_i+h_j)]^2}{\sigma^2-\mu^2}) - (s^y_{ij})^2 \geq 0. \end{cases}$

Both the first and second principal minors are constants that are, by definition, non-negative. Therefore only the third principal minor bounds s^y_{ij}. Simplifying this expression we get the following:

$$\begin{array}{rrll} & (\sigma^2-\mu^2)(\frac{[\frac{1}{2}(h_i+h_j)]^2}{\sigma^2-\mu^2}) - (s^y_{ij})^2 & \geq 0 & \\ \Rightarrow & (\frac{1}{2}(h_i+h_j))^2 & \geq (s^y_{ij})^2 & \\ \Rightarrow & \frac{1}{2}(h_i+h_j) & \geq s^y_{ij} & \text{since } s^y_{ij} \geq 0. \end{array}$$

Case 2: Let $\alpha_{ij} = \mu^2$.

Then constraint (2.6) becomes: $M_{sy} = \begin{bmatrix} 0 & s^y_{ij} \\ s^y_{ij} & \frac{[\frac{1}{2}(h_i+h_j)]^2}{\sigma^2-\mu^2} \end{bmatrix} \geq 0.$

The zero entry on the diagonal implies (by Property 6) that $s^y_{ij} = 0$.

■

Finally, a similar result can be shown for constraint (2.7) and surplus variable t^y_{ij}.

Lemma 9 (Bounding surplus variable t^y_{ij})

$$\begin{array}{rl} & \textit{Given } (2.7) \textit{ and } t^y_{ij} \geq 0. \\ \textit{Then} & \alpha_{ij} = \sigma^2 \Rightarrow t^y_{ij} \leq (\sum_{i=1}^n h_i) - (h_i + h_j) \\ \textit{and} & \alpha_{ij} = \mu^2 \Rightarrow t^y_{ij} = 0 \end{array}$$

Proof

Let constraints (2.7) and $t^y_{ij} \geq 0$ hold. Now consider the cases in turn.

Case 1: Let $\alpha_{ij} = \sigma^2$.

Then constraint (2.7) becomes: $M_{ty} = \begin{bmatrix} \sigma^2 - \mu^2 & t^y_{ij} \\ t^y_{ij} & \frac{[(\sum_{i=1}^n h_i) - (h_i + h_j)]^2}{\sigma^2 - \mu^2} \end{bmatrix} \geq 0.$

By examining the principal minors (Property 1) $M_{ty} \geq 0 \Leftrightarrow \begin{cases} \sigma^2 - \mu^2 \geq 0 \\ \frac{[(\sum_{i=1}^n h_i) - (h_i + h_j)]^2}{\sigma^2 - \mu^2} \geq 0 \\ (\sigma^2 - \mu^2)(\frac{[(\sum_{i=1}^n h_i) - (h_i + h_j)]^2}{\sigma^2 - \mu^2}) - (t^y_{ij})^2 \geq 0. \end{cases}$

Both the first and second principal minors are constants that are, by definition, non-negative. Therefore only the third principal minor bounds t^y_{ij}. Simplifying this expression we get the following:

$$\begin{array}{rrl} & (\sigma^2 - \mu^2)(\frac{[(\sum_{i=1}^n h_i) - (h_i + h_j)]^2}{\sigma^2 - \mu^2}) - (t^y_{ij})^2 & \geq 0 \\ \Rightarrow & [(\sum_{i=1}^n h_i) - (h_i + h_j)]^2 & \geq (t^y_{ij})^2 \\ \Rightarrow & (\sum_{i=1}^n h_i) - (h_i + h_j) & \geq t^y_{ij} \quad \text{since } t^y_{ij} \geq 0. \end{array}$$

Case 2: Let $\alpha_{ij} = \mu^2$.

Then constraint (2.7) becomes: $M_{ty} = \begin{bmatrix} 0 & t^y_{ij} \\ t^y_{ij} & \frac{[(\sum_{i=1}^n h_i) - (h_i + h_j)]^2}{\sigma^2 - \mu^2} \end{bmatrix} \geq 0.$

The zero entry on the diagonal implies (by Property 6) that $t^y_{ij} = 0$.

■

We are now able to show how the exactness constraint enforces the separation axis conditions.

Theorem 2 (Separation Axis Condition)

Given (2.3)-(2.7), (2.8) *and* (ExC 1) *then one of the following separation axis conditions must hold:*

$$\begin{array}{l} a)\ d^x_{ij} \geq \frac{1}{2}(w_i + w_j) \ \&\ d^y_{ij} \leq \frac{1}{2}(h_i + h_j) \\ b)\ d^y_{ij} \geq \frac{1}{2}(h_i + h_j). \end{array}$$

Proof

Let constraints (2.3)-(2.7), (2.8) and (ExC 1) hold.

(ExC 1) implies that $\alpha_{ij} = \sigma^2$ or $\alpha_{ij} = \mu^2$. Consider the cases in turn:

Case 1: Let $\alpha_{ij} = \sigma^2$.

By Lemma 7: (2.5) and (2.8) $\Rightarrow s^x_{ij} = 0$. Now with equation (2.3) we get that:

$$\begin{array}{rl} & d^x_{ij} + 0 - t^x_{ij} = \frac{1}{2}(w_i + w_j) \\ \Rightarrow & d^x_{ij} \geq \frac{1}{2}(w_i + w_j), \ \because (2.8) \Rightarrow t^x_{ij} \geq 0. \end{array}$$

By Lemma 9: (2.7) and (2.8) $\Rightarrow t^y_{ij} = 0$. Now with equation (2.4) we get that:

$$\begin{aligned} & d^y_{ij} + s^y_{ij} - 0 = \tfrac{1}{2}(h_i + h_j) \\ \Rightarrow \quad & d^y_{ij} \leq \tfrac{1}{2}(h_i + h_j), \ \because (2.8) \Rightarrow s^y_{ij} \geq 0. \end{aligned}$$

Since $d^x_{ij} \geq \frac{1}{2}(w_i + w_j)$ and $d^y_{ij} \leq \frac{1}{2}(h_i + h_j)$, separation axis condition a) holds.

Case 2: Let $\alpha_{ij} = \mu^2$.

By Lemma 7: (2.5) and (2.8) $\Rightarrow s^x_{ij} = 0$. Now with equation (2.3) we get that:

$$\begin{aligned} & d^x_{ij} + 0 - t^x_{ij} = \tfrac{1}{2}(w_i + w_j) \\ \Rightarrow \quad & d^x_{ij} \geq \tfrac{1}{2}(w_i + w_j), \ \because (2.8) \Rightarrow t^x_{ij} \geq 0. \end{aligned}$$

By Lemma 8: (2.6) and (2.8) $\Rightarrow s^y_{ij} = 0$. Now with equation (2.4) we get that:

$$\begin{aligned} & d^y_{ij} + 0 - t^y_{ij} = \tfrac{1}{2}(h_i + h_j) \\ \Rightarrow \quad & d^y_{ij} \geq \tfrac{1}{2}(h_i + h_j), \ \because (2.8) \Rightarrow t^y_{ij} \geq 0. \end{aligned}$$

Since $d^y_{ij} \geq \frac{1}{2}(h_i + h_j)$, separation axis condition b) holds.

■

3.2 Separation Order Constraints

Recall that the separation order defines the order of facilities i and j along the separation axis. This defines the purpose of the separation order constraints as:

$$(\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) = \begin{cases} (\sigma, \frac{1}{\sigma}, \sigma^2, \frac{1}{\sigma^2}) & \Rightarrow d^x_{ij} = x_j - x_i \\ (-\sigma, -\frac{1}{\sigma}, \sigma^2, \frac{1}{\sigma^2}) & \Rightarrow d^x_{ij} = -x_j + x_i \\ (\mu, \frac{1}{\mu}, \mu^2, \frac{1}{\mu^2}) & \Rightarrow d^y_{ij} = y_j - y_i \\ (-\mu, -\frac{1}{\mu}, \mu^2, \frac{1}{\mu^2}) & \Rightarrow d^y_{ij} = -y_j + y_i \end{cases}$$

Since the separation order is defined along the separation axis we will examine the conditions for the two separation axes individually.

3.2.1 Horizontal Separation Order Constraints

The purpose of constraint (2.9) is to force the appropriate separation order when there is a horizontal separation axis (i.e. $\beta_{ij} = \pm\sigma$). For the cases when there is a vertical separation axis ($\beta_{ij} = \pm\mu$) matrix M_x must be positive semidefinite for all acceptable values of d^x_{ij}, x_i and x_j. The constant C_x is selected so that matrix M_x will be positive for all values of $d^x_{ij} = |x_j - x_i| \in [0, \sum_{k=1}^n w_k]$. Recall constraint (2.9):

$$M_x = \begin{bmatrix} \sigma - \beta_{ij} & \alpha_{ij} - \sigma^2 & d^x_{ij} - x_j + x_i \\ \alpha_{ij} - \sigma^2 & \sigma + \beta_{ij} & d^x_{ij} + x_j - x_i \\ d^x_{ij} - x_j + x_i & d^x_{ij} + x_j - x_i & C_x \end{bmatrix} \geq 0$$

where $C_x = \max[\frac{2(\sum_{k=1}^n w_k)^2}{\sigma}, \frac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)}]$

We begin by showing the separation order conditions when there is a horizontal separation axis. First consider the following lemma.

Lemma 10

$$\textit{Given } (2.12) \textit{ and } (2.13)$$
$$C_x = \max[\tfrac{2(\sum_{k=1}^n w_k)^2}{\sigma}, \tfrac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)}] \Rightarrow \begin{bmatrix} 2\sigma & 2d_{ij}^x \\ 2d_{ij}^x & C_x \end{bmatrix} \geq 0$$

Proof

Let (2.12) and (2.13) hold, then $\sum_{k=1}^n w_k \geq d_{ij}^x \geq 0$. By Property 1: $\begin{bmatrix} 2\sigma & 2d_{ij}^x \\ 2d_{ij}^x & C_x \end{bmatrix} \geq 0 \Leftrightarrow \begin{cases} 2\sigma \geq 0 \\ C_x \geq 0 \\ 2\sigma C_x - (2d_{ij}^x)^2 \geq 0. \end{cases}$

Verify the principle minors in turn:

1. $2\sigma \geq 0$ by definition of σ .

2. $C_x = \max[\frac{2(\sum_{k=1}^n w_k)^2}{\sigma}, \frac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)}] \Rightarrow C_x \geq \frac{2(\sum_{k=1}^n w_k)^2}{\sigma} > 0.$

3. $C_x \geq \frac{2(\sum_{k=1}^n w_k)^2}{\sigma}$. Therefore $\begin{aligned} 2\sigma C_x - (2d_{ij}^x)^2 &\geq 2\sigma\Big(\frac{2(\sum_{k=1}^n w_k)^2}{\sigma}\Big) - (2d_{ij}^x)^2 \\ &= 4((\sum_{k=1}^n w_k)^2 - (d_{ij}^x)^2) \\ &\geq 0 \qquad \because \sum_{k=1}^n w_k \geq d_{ij}^x \geq 0. \end{aligned}$

■

Theorem 3 (Separation order for horizontal separation axis)

$$\textit{Given } (2.9), (2.12) \textit{ and } (2.13) \textit{ then, } \beta_{ij} = \begin{cases} \sigma \ \Rightarrow & \alpha_{ij} = \sigma^2 \ \ \& \ \ d_{ij}^x = x_j - x_i \\ -\sigma \Rightarrow & \alpha_{ij} = \sigma^2 \ \ \& \ \ d_{ij}^x = -x_j + x_i. \end{cases}$$

Proof

Let constraints (2.9), (2.12) and (2.13) hold. Now consider the cases in turn.

Case 1: Let $\beta_{ij} = \sigma$. Then constraint (2.9) becomes:

$$M_x = \begin{bmatrix} 0 & \alpha_{ij} - \sigma^2 & d_{ij}^x - x_j + x_i \\ \alpha_{ij} - \sigma^2 & 2\sigma & d_{ij}^x + x_j - x_i \\ d_{ij}^x - x_j + x_i & d_{ij}^x + x_j - x_i & C_x \end{bmatrix} \geq 0.$$

The (1,1)-element equals zero therefore, by Property 6, the following holds: $\begin{cases} \alpha_{ij} - \sigma^2 = 0 \\ d_{ij}^x - x_j + x_i = 0 \\ \begin{bmatrix} 2\sigma & d_{ij}^x + x_j - x_i \\ d_{ij}^x + x_j - x_i & C_x \end{bmatrix} \geq 0. \end{cases}$

Now examine the implications in turn:

1. $\alpha_{ij} - \sigma^2 = 0 \Rightarrow \alpha_{ij} = \sigma^2$.

2. $d^x_{ij} - x_j + x_i = 0 \Rightarrow d^x_{ij} = x_j - x_i$.

3. $d^x_{ij} + x_j - x_i = 2d^x_{ij}$ $(\because d^x_{ij} = x_j - x_i)$.
   Therefore, $\begin{bmatrix} 2\sigma & d^x_{ij} + x_j - x_i \\ d^x_{ij} + x_j - x_i & C_x \end{bmatrix} \geq 0$ and Lemma 10 implies $\begin{bmatrix} 2\sigma & 2d^x_{ij} \\ 2d^x_{ij} & C_x \end{bmatrix} \geq 0$.

Case 2: Let $\beta_{ij} = -\sigma$. Then constraint (2.9) becomes:

$$M_x = \begin{bmatrix} 2\sigma & \alpha_{ij} - \sigma^2 & d^x_{ij} - x_j + x_i \\ \alpha_{ij} - \sigma^2 & 0 & d^x_{ij} + x_j - x_i \\ d^x_{ij} - x_j + x_i & d^x_{ij} + x_j - x_i & C_x \end{bmatrix} \geq 0.$$

The (2,2)-element equals zero therefore, by Property 6, the following holds: $\begin{cases} \alpha_{ij} - \sigma^2 & = 0 \\ d^x_{ij} + x_j - x_i & = 0 \\ \begin{bmatrix} 2\sigma & d^x_{ij} - x_j + x_i \\ d^x_{ij} - x_j + x_i & C_x \end{bmatrix} & \geq 0. \end{cases}$

Now examine the implications in turn:

1. $\alpha_{ij} - \sigma^2 = 0 \Rightarrow \alpha_{ij} = \sigma^2$.

2. $d^x_{ij} + x_j - x_i = 0 \Rightarrow d^x_{ij} = -x_j + x_i$.

3. $d^x_{ij} - x_j + x_i = 2d^x_{ij}$ $(\because d^x_{ij} = -x_j + x_i)$.
   Therefore, $\begin{bmatrix} 2\sigma & d^x_{ij} - x_j + x_i \\ d^x_{ij} - x_j + x_i & C_x \end{bmatrix} \geq 0$ and Lemma 10 implies $\begin{bmatrix} 2\sigma & 2d^x_{ij} \\ 2d^x_{ij} & C_x \end{bmatrix} \geq 0$.

■

3.2.2 Vertical Separation Order Constraints

Next we will examine the separation order condition for facilities with a vertical separation axis. Constraint (2.10) is the vertical counterpart to constraint (2.9). Although they work in a similar manner there are two significant differences.

1. Matrix M_y is driven by the variable b_{ij} as opposed to β_{ij}.

2. The constant C_y is formed differently from C_x. This is a result of the symmetry-breaking condition that gives prefencial choice to the vertical separation axis.

$$M_y = \begin{bmatrix} \frac{1}{\mu} - b_{ij} & \alpha_{ij} - \mu^2 & d^y_{ij} - y_j + y_i \\ \alpha_{ij} - \mu^2 & \frac{1}{\mu} + b_{ij} & d^y_{ij} + y_j - y_i \\ d^y_{ij} - y_j + y_i & d^y_{ij} + y_j - y_i & C_y \end{bmatrix} \geq 0$$
$$\text{where } \max[\frac{(h_i+h_j)^2}{(\frac{1}{\mu}-\frac{1}{\sigma})(1-\sigma^4\mu^2+\sigma^2\mu^4)}, 2\mu(\textstyle\sum_{k=1}^n h_k)^2]$$

Now we show the vertical separation order conditions. First consider the following lemma.

Lemma 11

$$\textit{Given } (2.12) \textit{ and } (2.13), C_y = \max[\frac{(h_i+h_j)^2}{(\frac{1}{\mu}-\frac{1}{\sigma})(1-\sigma^4\mu^2+\sigma^2\mu^4)}, 2\mu(\textstyle\sum_{k=1}^n h_k)^2] \Rightarrow \begin{bmatrix} \frac{2}{\mu} & 2d_{ij}^y \\ 2d_{ij}^y & C_y \end{bmatrix} \geq 0.$$

Proof

Let (2.12) and (2.13) hold, then $\sum_{k=1}^n h_k \geq d_{ij}^y \geq 0$. By Property 1: $\begin{bmatrix} \frac{2}{\mu} & 2d_{ij}^y \\ 2d_{ij}^y & C_y \end{bmatrix} \geq 0 \Leftrightarrow \begin{cases} \frac{2}{\mu} \geq 0 \\ C_y \geq 0 \\ \frac{2}{\mu}C_y - (2d_{ij}^y)^2 \geq 0. \end{cases}$

Verify the principle minors in turn:

1. $\frac{2}{\mu} \geq 0$ by definition of μ.

2. $C_y = \max[\frac{(h_i+h_j)^2}{(\frac{1}{\mu}-\frac{1}{\sigma})(1-\sigma^4\mu^2+\sigma^2\mu^4)}, 2\mu(\sum_{k=1}^n h_k)^2] \Rightarrow C_x \geq 2\mu(\sum_{k=1}^n h_k)^2 > 0.$

3. $C_y \geq 2\mu(\sum_{k=1}^n h_k)^2$. Therefore $\begin{aligned} \frac{2}{\mu}C_y - (2d_{ij}^y)^2 &\geq \frac{2}{\mu}\left(2\mu(\textstyle\sum_{k=1}^n h_k)^2\right) - (2d_{ij}^y)^2 \\ &= 4((\textstyle\sum_{k=1}^n h_k)^2 - (d_{ij}^y)^2) \\ &\geq 0 \qquad \because \textstyle\sum_{k=1}^n h_k \geq d_{ij}^y \geq 0. \end{aligned}$

■

Theorem 4 (Separation Order for Vertical Separation Axis)

$$\textit{Given } (2.10),(2.12) \textit{ and } (2.13) \textit{ then, } b_{ij} = \begin{cases} \frac{1}{\mu} \Rightarrow & \alpha_{ij} = \mu^2 \;\; \& \;\; d_{ij}^y = y_j - y_i \\ \frac{1}{-\mu} \Rightarrow & \alpha_{ij} = \mu^2 \;\; \& \;\; d_{ij}^y = -y_j + y_i. \end{cases}$$

Proof

Let constraints (2.10),(2.12) and (2.13) hold. Now consider the cases in turn.

Case 1: Let $b_{ij} = \frac{1}{\mu}$. Then constraint (2.10) becomes:

$$M_y = \begin{bmatrix} 0 & \alpha_{ij} - \mu^2 & d_{ij}^y - y_j + y_i \\ \alpha_{ij} - \mu^2 & \frac{2}{\mu} & d_{ij}^y + y_j - y_i \\ d_{ij}^y - y_j + y_i & d_{ij}^y + y_j - y_i & C_y \end{bmatrix} \geq 0.$$

The (1,1)-element equals zero therefore, by Property 6, the following holds: $\begin{cases} \alpha_{ij} - \mu^2 = 0 \\ d_{ij}^y - y_j + y_i = 0 \\ \begin{bmatrix} \frac{2}{\mu} & d_{ij}^y + y_j - y_i \\ d_{ij}^y + y_j - y_i & C_y \end{bmatrix} \geq 0. \end{cases}$

Now examine the implications in turn:

1. $\alpha_{ij} - \mu^2 = 0 \Rightarrow \alpha_{ij} = \mu^2$.

2. $d_{ij}^y - y_j + y_i = 0 \Rightarrow d_{ij}^y = y_j - y_i$.

3. $d_{ij}^y + y_j - y_i = 2d_{ij}^y \;\; (\because d_{ij}^y = y_j - y_i)$.
   Therefore, $\begin{bmatrix} \frac{2}{\mu} & d_{ij}^x + x_j - x_i \\ d_{ij}^y + y_j - y_i & C_y \end{bmatrix} \geq 0 \quad \Rightarrow \begin{bmatrix} \frac{2}{\mu} & 2d_{ij}^y \\ 2d_{ij}^y & C_y \end{bmatrix} \geq 0$ (by Lemma 11 this holds).

Case 2: Let $b_{ij} = -\frac{1}{\mu}$. Then constraint (2.10) becomes:

$$M_x = \begin{bmatrix} \frac{2}{\mu} & \alpha_{ij} - \mu^2 & d^y_{ij} - y_j + y_i \\ \alpha_{ij} - \mu^2 & 0 & d^y_{ij} + y_j - y_i \\ d^y_{ij} - y_j + y_i & d^y_{ij} + y_j - y_i & C_y \end{bmatrix} \succeq 0.$$

The (2,2)-element equals zero therefore, by Property 6, the following holds: $\begin{cases} \alpha_{ij} - \mu^2 & = 0 \\ d^y_{ij} + y_j - y_i & = 0 \\ \begin{bmatrix} \frac{2}{\mu} & d^y_{ij} - y_j + y_i \\ d^y_{ij} - y_j + y_i & C_y \end{bmatrix} & \succeq 0. \end{cases}$

Now examine the implications in turn:

1. $\alpha_{ij} - \mu^2 = 0 \Rightarrow \alpha_{ij} = \mu^2$.

2. $d^y_{ij} + y_j - y_i = 0 \Rightarrow d^y_{ij} = -y_j + y_i$.

3. $d^y_{ij} - y_j + y_i = 2d^y_{ij}$ $(\because d^y_{ij} = -y_j + y_i)$.
   Therefore, $\begin{bmatrix} 2\sigma & d^y_{ij} - y_j + y_i \\ d^y_{ij} - y_j + y_i & C_y \end{bmatrix} \succeq 0 \quad \Rightarrow \begin{bmatrix} \frac{2}{\mu} & 2d^y_{ij} \\ 2d^y_{ij} & C_y \end{bmatrix} \succeq 0$ (by Lemma 11 this holds).

■

3.3 Non-Separation Order Constraints

The non-separation order conditions ensure the feasibility of the distance along the axis that is not the separation axis. Therefore we will examine the feasibility of $d^y_{ij} = |y_j + y_i| \in [0, \frac{1}{2}(h_i + h_j)]$ and $d^x_{ij} = |x_j + x_i| \in [0, \sum_{k=1}^n w_k]$ when there is a horizontal and vertical separation axis respectively. To do this we only need to examine the positive semidefinite constraints that relate to the variables in question. We are now able to define the non-separation order conditions.

Definition 8 (Horizontal non-separation order condition)
If facilities i and j have a horizontal separation axis then constraints (2.4)*,* (2.6),(2.7)*,* (2.8) *and* (2.10) *must be feasible for all* $d^y_{ij} = |y_j + y_i| \in [0, \frac{1}{2}(h_i + h_j)]$

Definition 9 (Vertical non-separation order condition)
If facilities i and j have a vertical separation axis then constraints (2.3)*,* (2.5)*,* (2.9)*,* (2.8) *must be feasible for all* $d^x_{ij} = |x_j + x_i| \in [0, \sum_{k=1}^n w_k]$

We will prove these conditions in turn.

3.3.1 Non-Separation Order Constraints for Horizontal Separation Axis

If there is a horizontal separation axis then the non-separation axis condition requires $d^y_{ij} = |y_j + y_i|$ to be feasible for all $d^y_{ij} \in [0, \frac{1}{2}(h_i + h_j)]$. To do so we will discuss the implications of constraints (2.4), (2.6),(2.7), (2.8) and (2.10) when $(\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) \in \{(\sigma, \frac{1}{\sigma}, \sigma^2, \frac{1}{\sigma^2}), (-\sigma, \frac{1}{-\sigma}, \sigma^2, \frac{1}{\sigma^2})\}$.

Recall the following bounds on the slack and surplus variables when $\alpha_{ij} = \sigma^2$:

1. By Lemma 8: (2.6) implies $s^y_{ij} \leq \frac{1}{2}(h_i + h_j)$
2. By Lemma 9: (2.7) implies $t^y_{ij} = 0$
3. (2.8) implies $s^y_{ij} \geq 0$

Therefore $s^y_{ij} \in [0, \frac{1}{2}(h_i + h_j)]$ and $t^y_{ij} = 0$. Using (2.4) we can translate this into bounds for d^y_{ij}.

$$\begin{aligned} (2.4) \Rightarrow & d^y_{ij} - \tfrac{1}{2}(h_i + h_j) = t^y_{ij} - s^y_{ij} \\ (2.6), (2.7), (2.8) \Rightarrow & -\tfrac{1}{2}(h_i + h_j) \leq d^y_{ij} - \tfrac{1}{2}(h_i + h_j) \leq 0 \\ \Rightarrow & 0 \leq d^y_{ij} \leq \tfrac{1}{2}(h_i + h_j) \end{aligned}$$

The bounds created are satisfied for all $d^y_{ij} \in [0, \frac{1}{2}(h_i + h_j)]$, therefore we can now turn our attention to the remaining implication, constraint (2.10). Instead of bounding d^y_{ij} using the constraint we will show that $d^y_{ij} = |y_j - y_i|$ is feasible for all $d^y_{ij} \in [0, \frac{1}{2}(h_i + h_j)]$.

Lemma 12

$$\alpha_{ij} = \frac{1}{b^2_{ij}} = \sigma^2 \Rightarrow \begin{bmatrix} \frac{1}{\mu} - b_{ij} & \alpha_{ij} - \mu^2 \\ \alpha_{ij} - \mu^2 & \frac{1}{\mu} + b_{ij} \end{bmatrix} \succ 0$$

Proof

Consider the principal minors: $\begin{bmatrix} \frac{1}{\mu} - b_{ij} & \alpha_{ij} - \mu^2 \\ \alpha_{ij} - \mu^2 & \frac{1}{\mu} + b_{ij} \end{bmatrix} \succ 0 \quad \Leftrightarrow \begin{cases} \frac{1}{\mu} - b_{ij} > 0 \\ \frac{1}{\mu} + b_{ij} > 0 \\ (\frac{1}{\mu} - b_{ij})(\frac{1}{\mu} + b_{ij}) - (\alpha_{ij} - \mu^2)^2 > 0 \end{cases}$

$$\Leftrightarrow \begin{cases} \frac{1}{\mu} \pm b_{ij} > 0 \\ \frac{1}{\mu^2} - b^2_{ij} - (\alpha_{ij} - \mu^2)^2 > 0 \end{cases} \quad \text{(obviously).}$$

Let $\alpha_{ij} = \frac{1}{b^2_{ij}} = \sigma^2$. Recall that $1 \geq \sigma > \mu > 0$. Now using this definition we will show that both of the principal minors are strictly positive.

1. $b^2_{ij} = \frac{1}{\sigma^2} \Rightarrow b_{ij} = \pm\frac{1}{\sigma}$. Therefore $\begin{aligned} \frac{1}{\mu} \pm b_{ij} &= \frac{1}{\mu} \pm \frac{1}{\sigma} \\ &= \frac{\sigma \pm \mu}{\mu\sigma} \\ &\geq \frac{\sigma - \mu}{\mu\sigma} \\ &> 0. \end{aligned}$

2. $\begin{aligned} \frac{1}{\mu^2} - b^2_{ij} - (\alpha_{ij} - \mu^2)^2 &= \frac{1}{\mu^2} - \frac{1}{\sigma^2} - (\sigma^2 - \mu^2)^2 \\ &= \frac{\sigma^2 - \mu^2}{\sigma^2\mu^2} - (\sigma^2 - \mu^2)^2 \\ &> \frac{\sigma^2 - \mu^2}{\sigma^2\mu^2} - (\sigma^2 - \mu^2), && \because 1 > (\sigma^2 - \mu^2) > (\sigma^2 - \mu^2)^2 > 0 \\ &= \frac{\sigma^2 - \mu^2}{\sigma^2\mu^2}(1 - \sigma^2\mu^2) \\ &> 0, && \because 1 \geq \sigma^2 > \sigma^2\mu^2 > 0 \text{ and} \sigma^2 - \mu^2 > 0. \end{aligned}$

∎

Theorem 5 (M_y for μ)

$$\textit{Suppose } b_{ij} = \pm\frac{1}{\sigma}, \alpha_{ij} = \sigma^2 \textit{ and } d^y_{ij} = |y_j - y_i| \textit{ then,}$$
$$\frac{1}{2}(h_i + h_j) \geq d^y_{ij} \Rightarrow M_y \geq 0.$$

Proof

Let $d^y_{ij} = |y_j - y_i| \Rightarrow d^y_{ij} = y_j - y_i$ or $d^y_{ij} = -y_j + y_i$.

Let $\alpha_{ij} = \sigma^2$, then Lemma 12 implies $\begin{bmatrix} \frac{1}{\mu} - b_{ij} & \alpha_{ij} - \mu^2 \\ \alpha_{ij} - \mu^2 & \frac{1}{\mu} + b_{ij} \end{bmatrix} \succ 0$.

By applying the Schur Complement (Property 5) we get that:

$M_y \geq 0 \Leftrightarrow C_y - \begin{bmatrix} d^y_{ij} - y_j + y_i & d^y_{ij} + y_j - y_i \end{bmatrix} \begin{bmatrix} \frac{1}{\mu} - b_{ij} & \sigma_{ij} - \mu^2 \\ \sigma^2 - \mu^2 & \frac{1}{\mu} + b_{ij} \end{bmatrix}^{-1} \begin{bmatrix} d^y_{ij} - y_j + y_i \\ d^y_{ij} + y_j - y_i \end{bmatrix} \geq 0.$

Let $b_{ij} = \pm\frac{1}{\sigma}$ then consider the two cases that result from $d^y_{ij} = |y_j - y_i|$ and simplify the right-hand-side expression.

Case 1: $d^y_{ij} = y_j - y_i \Rightarrow \begin{cases} d^y_{ij} - y_j + y_i = 0 \\ d^y_{ij} + y_j - y_i = 2d^y_{ij} \end{cases}$

The Schur Complement becomes:

$$\begin{aligned}
M_y \geq 0 &\Leftrightarrow C_y - \begin{bmatrix} 0 & 2d^y_{ij} \end{bmatrix} \left(\frac{1}{(\frac{1}{\mu^2} - b^2_{ij}) - (\sigma^2 - \mu^2)^2}\right) \begin{bmatrix} \frac{1}{\mu} + b_{ij} & \sigma^2 - \mu^2 \\ \sigma^2 - \mu^2 & \frac{1}{\mu} - b_{ij} \end{bmatrix} \begin{bmatrix} 0 \\ 2d^y_{ij} \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_y - \left(\frac{1}{(\frac{1}{\mu^2} - b^2_{ij}) - (\sigma^2 - \mu^2)^2}\right) \begin{bmatrix} 2d^y_{ij}(\sigma^2 - \mu^2) & 2d^y_{ij}(\frac{1}{\mu} - b_{ij}) \end{bmatrix} \begin{bmatrix} 0 \\ 2d^y_{ij} \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_y - \left(\frac{(2d^y_{ij})^2(\frac{1}{\mu} - b_{ij})}{(\frac{1}{\mu^2} - b^2_{ij}) - (\sigma^2 - \mu^2)^2}\right) \geq 0 \\
&\Leftrightarrow C_y - \left(\frac{(2d^y_{ij})^2(\frac{1}{\mu} \pm \frac{1}{\sigma})}{(\frac{1}{\mu^2} - \frac{1}{\sigma^2}) - (\sigma^2 - \mu^2)^2}\right) \geq 0 \quad (\because b_{ij} = \pm\tfrac{1}{\sigma}).
\end{aligned}$$

Case 2: $d^y_{ij} = -y_j + y_i \Rightarrow \begin{cases} d^y_{ij} - y_j + y_i = 2d^y_{ij} \\ d^y_{ij} + y_j - y_i = 0 \end{cases}$

The Schur Complement becomes:

$$\begin{aligned}
M_y \geq 0 &\Leftrightarrow C_y - \begin{bmatrix} 2d^y_{ij} & 0 \end{bmatrix} \left(\frac{1}{(\sigma^2 - \mu^2) - (\mu^2 - \sigma^2)^2}\right) \begin{bmatrix} \frac{1}{\mu} + b_{ij} & \sigma^2 - \mu^2 \\ \sigma^2 - \mu^2 & \frac{1}{\mu} - b_{ij} \end{bmatrix} \begin{bmatrix} 2d^y_{ij} \\ 0 \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_y - \left(\frac{1}{(\frac{1}{\mu^2} - b^2_{ij}) - (\sigma^2 - \mu^2)^2}\right) \begin{bmatrix} 2d^y_{ij}(\frac{1}{\mu} + b_{ij}) & 2d^y_{ij}(\sigma^2 - \mu^2) \end{bmatrix} \begin{bmatrix} 2d^y_{ij} \\ 0 \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_y - \left(\frac{(2d^y_{ij})^2(\frac{1}{\mu} + b_{ij})}{(\frac{1}{\mu^2} - b^2_{ij}) - (\sigma^2 - \mu^2)^2}\right) \geq 0 \\
&\Leftrightarrow C_y - \left(\frac{(2d^y_{ij})^2(\frac{1}{\mu} \pm \frac{1}{\sigma})}{(\frac{1}{\mu^2} - \frac{1}{\sigma^2}) - (\sigma^2 - \mu^2)^2}\right) \geq 0 \quad (\because b_{ij} = \pm\tfrac{1}{\sigma}).
\end{aligned}$$

We will consider both cases together as they both imply: $M_y \geq 0 \Leftrightarrow C_y - (\frac{(2d^y_{ij})^2(\frac{1}{\mu} \pm \frac{1}{\sigma})}{(\frac{1}{\mu^2} - \frac{1}{\sigma^2}) - (\sigma^2 - \mu^2)^2}) \geq 0$.

Now we will show that for any d^y_{ij} in the interval $[0, \frac{1}{2}(h_i + h_j)]$ $\quad C_y - (\frac{(2d^y_{ij})^2(\frac{1}{\mu} \pm \frac{1}{\sigma})}{(\frac{1}{\mu^2} - \frac{1}{\sigma^2}) - (\sigma^2 - \mu^2)^2}) \geq 0$.

$$
\begin{aligned}
C_y - (\frac{(2d^y_{ij})^2(\frac{1}{\mu} \pm \frac{1}{\sigma})}{(\frac{1}{\mu^2} - \frac{1}{\sigma^2}) - (\sigma^2 - \mu^2)^2}) &\geq C_y - (\frac{(2d^y_{ij})^2(\frac{1}{\mu} + \frac{1}{\sigma})}{(\frac{1}{\mu^2} - \frac{1}{\sigma^2}) - (\sigma^2 - \mu^2)^2}) \\
&\geq \frac{(h_i + h_j)^2}{(\frac{1}{\mu} - \frac{1}{\sigma})(1 - \sigma^4\mu^2 + \sigma^2\mu^4)} - (\frac{(2d^y_{ij})^2(\frac{1}{\mu} + \frac{1}{\sigma})}{(\frac{1}{\mu^2} - \frac{1}{\sigma^2}) - (\sigma^2 - \mu^2)^2}) \\
&\qquad \because \max[\frac{(h_i + h_j)^2}{(\frac{1}{\mu} - \frac{1}{\sigma})(1 - \sigma^4\mu^2 + \sigma^2\mu^4)}, 2\mu(\textstyle\sum_{k=1}^n h_k)^2] \geq \frac{(h_i + h_j)^2}{(\frac{1}{\mu} - \frac{1}{\sigma})(1 - \sigma^4\mu^2 + \sigma^2\mu^4)} \\
&= \frac{(h_i + h_j)^2 - (2d^y_{ij})^2}{(\frac{1}{\mu} - \frac{1}{\sigma})(1 - \sigma^4\mu^2 + \sigma^2\mu^4)} \\
&\geq 0 \qquad\qquad \forall d^y_{ij} \in [0, \tfrac{1}{2}(h_i + h_j)].
\end{aligned}
$$

■

3.3.2 Non-Separation Order Constraints for Vertical Separation Axis

If there is a vertical separation axis then the non-separation axis conditions requires $d^x_{ij} = |x_j - x_i|$ to be feasible for all $d^x_{ij} \in [0, \sum_{k=1}^n w_k]$. We will examine constraints (2.3), (2.5), (2.9), (2.8) when $(\beta_{ij}, b_{ij}, \alpha_{ij}, a_{ij}) \in \{(\mu, \frac{1}{\mu}, \mu^2, \frac{1}{\mu^2}), (-\mu, \frac{1}{-\mu}, \mu^2, \frac{1}{\mu^2})\}$.
Recall the following bounds on the slack and surplus variable when $\alpha_{ij} = \mu^2$:

1. By Lemma 7: (2.5) implies $s^y_{ij} \leq \frac{1}{2}(w_i + w_j)$
2. (2.8) implies $s^x_{ij} \geq 0$ and $t^x_{ij} \geq 0$.

Therefore $s^x_{ij} \in [0, \frac{1}{2}(w_i + w_j)]$ and $t^y_{ij} \geq 0$. Using (2.3) we can translate this into bounds on for d^x_{ij}.

$$
\begin{aligned}
(2.3) &\Rightarrow d^x_{ij} - \tfrac{1}{2}(w_i + w_j) = t^x_{ij} - s^x_{ij} \\
(2.5), (2.8) &\Rightarrow -\tfrac{1}{2}(w_i + w_j) \leq d^x_{ij} - \tfrac{1}{2}(w_i + w_j) \leq t^y_{ij} \\
&\Rightarrow 0 \leq d^x_{ij} \leq t^x_{ij} + \tfrac{1}{2}(h_i + h_j)
\end{aligned}
$$

These bounds are satisfied for all $d^x_{ij} \in [0, \sum_{k=1}^n w_k]$ since t^x_{ij} is an unbounded surplus variable. The interval for d^x_{ij} when there is a vertical separation axis is larger then the interval for d^y_{ij} when there is a horizontal separation axis. This is a result of our definition of a horizontal separation axis. Recall that if facilities i and j have a horizontal separation axis then they are not separated vertically. This limits the vertical distance between facilities. However facilities with a vertical separation distance can also be horizontally separated. As a result the interval for d^x_{ij} is larger.

Now we will show that $d^x_{ij} = |x_j - x_i|$ is feasible in (2.9) for all $d^x_{ij} \in [0, \sum_{k=1}^n w_k]$. We begin with the following lemma.

Lemma 13

$$\alpha_{ij} = \beta_{ij}^2 = \mu^2 \Rightarrow \begin{bmatrix} \sigma - \beta_{ij} & \alpha_{ij} - \sigma^2 \\ \alpha_{ij} - \sigma^2 & \sigma + \beta_{ij} \end{bmatrix} \succ 0$$

Proof

Consider the principal minors: $\begin{bmatrix} \sigma - \beta_{ij} & \alpha_{ij} - \sigma^2 \\ \alpha_{ij} - \sigma^2 & \sigma + \beta_{ij} \end{bmatrix} \succ 0 \quad \Leftrightarrow \begin{cases} \sigma - \beta_{ij} > 0 \\ \sigma + \beta_{ij} > 0 \\ (\sigma - \beta_{ij})(\sigma + \beta_{ij}) - (\alpha_{ij} - \sigma^2)^2 > 0 \end{cases}$

$$\Leftrightarrow \begin{cases} \sigma \pm \beta_{ij} > 0 \\ \sigma^2 - \beta_{ij}^2 - (\alpha_{ij} - \sigma^2)^2 > 0 \end{cases} \quad \text{(obviously).}$$

Let $\alpha_{ij} = \beta_{ij}^2 = \mu^2$. Recall that $1 \geq \sigma > \mu > 0$. Now using this definition we will show that both of the principal minors are strictly positive.

1. $\beta_{ij}^2 = \mu^2 \Rightarrow \beta_{ij} = \pm\mu$. Therefore $\begin{aligned} \sigma \pm \beta_{ij} &= \sigma \pm \mu \\ &\geq \sigma - \mu \\ &> 0. \end{aligned}$

2. $\begin{aligned} \sigma^2 - \beta_{ij}^2 - (\alpha_{ij} - \sigma^2)^2 &= \sigma^2 - \mu^2 - (\mu^2 - \sigma^2)^2 \\ &= (\sigma^2 - \mu^2)(1 - \sigma^2 + \mu^2) \quad \because 1 \geq \sigma \geq \sigma^2 \Rightarrow 1 - \sigma^2 + \mu^2 > 1 - \sigma^2 \geq 0 \\ &> 0. \end{aligned}$

■

Theorem 6 (M_x for μ)

$$\textit{Suppose } \beta_{ij} = \pm\mu, \alpha_{ij} = \mu^2 \textit{ and } d_{ij}^x = |x_j - x_i| \textit{ then,}$$
$$\textstyle\sum_{k=1}^n w_k \geq d_{ij}^x \Rightarrow M_x \succeq 0$$

Proof

Let $d_{ij}^x = |x_j - x_i| \Rightarrow d_{ij}^x = x_j - x_i$ or $d_{ij}^x = -x_j + x_i$.
Let $\alpha_{ij} = \mu^2$, then Lemma 13 implies $\begin{bmatrix} \sigma - \beta_{ij} & \alpha_{ij} - \sigma^2 \\ \alpha_{ij} - \sigma^2 & \sigma + \beta_{ij} \end{bmatrix} \succ 0$.
By applying the Schur Complement (Property 5) we get that:
$M_x \succeq 0 \Leftrightarrow C_x - \begin{bmatrix} d_{ij}^x - x_j + x_i & d_{ij}^x + x_j - x_i \end{bmatrix} \begin{bmatrix} \sigma - \beta_{ij} & \alpha_{ij} - \sigma^2 \\ \alpha_{ij} - \sigma^2 & \sigma + \beta_{ij} \end{bmatrix}^{-1} \begin{bmatrix} d_{ij}^x - x_j + x_i \\ d_{ij}^x + x_j - x_i \end{bmatrix} \geq 0.$
Let $\beta_{ij} = \pm\mu$ then consider the two cases that result from $d_{ij}^x = |x_j - x_i|$ and simplify the right-hand-side expression.

Case 1: $d_{ij}^x = x_j - x_i \Rightarrow \begin{cases} d_{ij}^x - x_j + x_i = 0 \\ d_{ij}^x + x_j - x_i = 2d_{ij}^x \end{cases}$

The Schur Complement becomes:

$$
\begin{aligned}
M_x \geq 0 &\Leftrightarrow C_x - \begin{bmatrix} 0 & 2d_{ij}^x \end{bmatrix} \left(\frac{1}{(\sigma^2 - \beta_{ij}^2) - (\mu^2 - \sigma^2)^2}\right) \begin{bmatrix} \sigma + \beta_{ij} & \sigma^2 - \mu^2 \\ \sigma^2 - \mu^2 & \sigma - \beta_{ij} \end{bmatrix} \begin{bmatrix} 0 \\ 2d_{ij}^x \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_x - \left(\frac{1}{(\sigma^2 - \beta_{ij}^2) - (\mu^2 - \sigma^2)^2}\right) \begin{bmatrix} 2d_{ij}^x(\sigma^2 - \mu^2) & 2d_{ij}^x(\sigma - \beta_{ij}) \end{bmatrix} \begin{bmatrix} 0 \\ 2d_{ij}^x \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_x - \left(\frac{(2d_{ij}^x)^2(\sigma - \beta_{ij})}{(\sigma^2 - \beta_{ij}^2) - (\mu^2 - \sigma^2)^2}\right) \geq 0 \\
&\Leftrightarrow C_x - \left(\frac{(2d_{ij}^x)^2(\sigma \pm \mu)}{(\sigma^2 - \mu^2) - (\mu^2 - \sigma^2)^2}\right) \geq 0 \quad (\because \beta_{ij} = \pm\mu).
\end{aligned}
$$

Case 2: $d_{ij}^x = -x_j + x_i \Rightarrow \begin{cases} d_{ij}^x - x_j + x_i = 2d_{ij}^x \\ d_{ij}^x + x_j - x_i = 0 \end{cases}$

The Schur Complement becomes:

$$
\begin{aligned}
M_x \geq 0 &\Leftrightarrow C_x - \begin{bmatrix} 2d_{ij}^x & 0 \end{bmatrix} \left(\frac{1}{(\sigma^2 - \beta_{ij}^2) - (\mu^2 - \sigma^2)^2}\right) \begin{bmatrix} \sigma + \beta_{ij} & \sigma^2 - \mu^2 \\ \sigma^2 - \mu^2 & \sigma - \beta_{ij} \end{bmatrix} \begin{bmatrix} 2d_{ij}^x \\ 0 \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_x - \left(\frac{1}{(\sigma^2 - \beta_{ij}^2) - (\mu^2 - \sigma^2)^2}\right) \begin{bmatrix} 2d_{ij}^x(\sigma + \beta_{ij}) & 2d_{ij}^x(\sigma^2 - \mu^2) \end{bmatrix} \begin{bmatrix} 2d_{ij}^x \\ 0 \end{bmatrix} \geq 0 \\
&\Leftrightarrow C_x - \left(\frac{(2d_{ij}^x)^2(\sigma + \beta_{ij})}{(\sigma^2 - \beta_{ij}^2) - (\mu^2 - \sigma^2)^2}\right) \geq 0 \\
&\Leftrightarrow C_x - \left(\frac{(2d_{ij}^x)^2(\sigma \pm \mu)}{(\sigma^2 - \mu^2) - (\mu^2 - \sigma^2)^2}\right) \geq 0 \quad (\because \beta_{ij} = \pm\mu).
\end{aligned}
$$

We will consider both cases together as they both imply: $M_x \geq 0 \Leftrightarrow C_x - (\frac{(2d_{ij}^x)^2(\sigma \pm \mu)}{(\sigma^2-\mu^2)-(\mu^2-\sigma^2)^2}) \geq 0$.

Now we will show that for any d_{ij}^x in the interval $[0, \sum_{k=1}^n w_k]$ $\quad C_x - (\frac{(2d_{ij}^x)^2(\sigma \pm \mu)}{(\sigma^2-\mu^2)-(\mu^2-\sigma^2)^2}) \geq 0$.

$$
\begin{aligned}
C_x - \tfrac{(2d_{ij}^x)^2(\sigma\pm\mu)}{(\sigma^2-\mu^2)-(\mu^2-\sigma^2)^2} &\geq C_x - \tfrac{(2d_{ij}^x)^2(\sigma+\mu)}{(\sigma^2-\mu^2)-(\mu^2-\sigma^2)^2} \\
&\geq \tfrac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)} - \tfrac{(2d_{ij}^x)^2(\sigma+\mu)}{(\sigma^2-\mu^2)(1-\sigma^2+\mu^2)} && \because C_x = \max\left[\tfrac{2(\sum_{k=1}^n w_k)^2}{\sigma}, \tfrac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)}\right] \geq \tfrac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)} \\
&= \tfrac{4(\sum_{k=1}^n w_k)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)} - \tfrac{4(d_{ij}^x)^2}{(\sigma-\mu)(1-\sigma^2+\mu^2)} \\
&= \tfrac{4}{(\sigma-\mu)(1-\sigma^2+\mu^2)}((\textstyle\sum_{k=1}^n w_k)^2 - (d_{ij}^x)^2) \\
&\geq 0 && \forall d_{ij}^x \in [0, \textstyle\sum_{k=1}^n w_k].
\end{aligned}
$$

■

3.4 Concluding Remarks

In Chapter 2 we defined separation axis, separation order and non-separation order to characterize the relative position of facitlities i and j. This chapter examined the theory required to prove each of these conditions and together the chapters showed that the complete model, as presented in Section 2.3, is an exact formulation.

The following chapter will consider how transitivity conditions can be derived for the complete model when defined for more than two facilities.

Chapter 4

Transitivity

Up to this point we have been dealing with the relative arrangement of two facilities and we have defined our decision variable β_{ij} based on the properties of this pairwise relationship. This chapter will examine the relative arrangement of 3 and 4 facilities. By examining these larger groups observations can be made that will hold regardless of the width, height and cost parameters for the problem. Specifically we will observe conditions, known as transitivity conditions, that will be used to describe infeasible layouts. Although these infeasible layouts could be found by running the model there is a tremendous computational advantage to being able to identify this beforehand. Ultimately the purpose of this chapter is to introduce a means to recognize infeasible combinations of fixed variables, in order to allow us to prune nodes within the branch and bound prior to solving the SDP relaxation for those nodes. For clarity we will examine groups of 3 and 4 facilities separately. The examination of transitivity conditions is limited to 3 and 4 facilities due to the increased complexity resulting from increasing the size. Within this chapter we will:

1. Discuss how and why it is helpful to identify infeasible layouts.
2. Introduce new definitions and notation to simplify the discussion of groups of facilities larger then two.
3. Identify the feasible and infeasible combinations for 3 facilities, referred to as **3-way Transitivity Conditions**.
4. Identify the feasible and infeasible combinations for 4 facilities, referred to as **4-way Transitivity Conditions**.

4.1 Purpose of Transitivity

Transitivity conditions are used to describe layouts that are infeasible regardless of the specific width and height of the facilities. This is done within the branch-and-bound algorithm by constructing the following potentially feasible set for all i, j pairs that are not yet fixed:

Definition 10 (Potentially Feasible Set)
Let $\hat{\Upsilon}_{ij}$ be the set of values of β_{ij} for which a transitivity condition implying the solution will be infeasible. Then the potentially feasible set $\Upsilon_{ij} = \{\pm\sigma, \pm\mu\} \setminus \hat{\Upsilon}_{ij}$.

Although each individual variable β_{ij} has four feasible values, namely $\beta_{ij} \in \{\pm\sigma, \pm\mu\}$, certain combinations of variables are not feasible. For example $(\beta_{ij}, \beta_{ik}, \beta_{jk}) = (\mu, -\mu, \mu)$ cannot produce a feasible layout. We examine why this combinations cannot be feasible in Section 4.3.1; however for now we ask the reader to accept that it is not possible so that we may focus on why identifying infeasible combinations is helpful.

At the beginning of the branch-and-bound algorithm $|\Upsilon_{ij}| = 4$ for all i, j pairs; however as variables are fixed transitivity conditions are used to identify infeasible layouts and as a result the size of potentially feasible sets are reduced. For example if $(\beta_{ij}, \beta_{ik}) = (\mu, -\mu)$ then using the transitivity condition stated above, $\mu \notin \Upsilon_{jk}$. Once the size of a potentially feasible set is reduced, constraints can be added to tighten the relaxation.

- If $|\Upsilon_{ij}| = 3$ then one of the following cuts (proven below in Theorem 7) is added to the relaxation:

$$\begin{array}{lll} \text{If } \sigma \notin \Upsilon_{ij} & \text{then} & 0 = \alpha_{ij} + \sigma\beta_{ij} - \mu^2 - \mu^2\sigma b_{ij} \\ \text{If } -\sigma \notin \Upsilon_{ij} & \text{then} & 0 = \alpha_{ij} - \sigma\beta_{ij} - \mu^2 + \mu^2\sigma b_{ij} \\ \text{If } \mu \notin \Upsilon_{ij} & \text{then} & 0 = \alpha_{ij} + \mu\beta_{ij} - \sigma^2 - \sigma^2\mu b_{ij} \\ \text{If } -\mu \notin \Upsilon_{ij} & \text{then} & 0 = \alpha_{ij} - \mu\beta_{ij} - \sigma^2 + \sigma^2\mu b_{ij} \end{array}$$

- If $|\Upsilon_{ij}| = 2$ then two of the preceeding cuts are added to the relaxation.
- If $|\Upsilon_{ij}| = 1$ then β_{ij} is fixed.
- If $|\Upsilon_{ij}| = 0$ then this node is infeasible and can be pruned without having to solving it.

Theorem 7 (Transitivity Cuts)

$$\begin{array}{c} \textit{Given } \text{(ExC 1)} \textit{then} \\ \sigma \notin \Upsilon_{ij} \Rightarrow 0 = \alpha_{ij} + \sigma\beta_{ij} - \mu^2 - \mu^2\sigma b_{ij} \\ -\sigma \notin \Upsilon_{ij} \Rightarrow 0 = \alpha_{ij} - \sigma\beta_{ij} - \mu^2 + \mu^2\sigma b_{ij} \\ \mu \notin \Upsilon_{ij} \Rightarrow 0 = \alpha_{ij} + \mu\beta_{ij} - \sigma^2 - \sigma^2\mu b_{ij} \\ -\mu \notin \Upsilon_{ij} \Rightarrow 0 = \alpha_{ij} - \mu\beta_{ij} - \sigma^2 + \sigma^2\mu b_{ij} \end{array}$$

Proof

Define ρ, τ so that $\beta_{ij} \in \{\pm\rho, \tau : |\rho| \neq |\tau|\} \subset \{\pm\sigma, \pm\mu\}$ if β_{ij} can only be 3 of the 4 discrete values.
Then for any $\beta_{ij} \in \{\pm\rho, \tau : |\rho| \neq |\tau|\} \subset \{\pm\sigma, \pm\mu\}$

$$\begin{aligned} & 0 = (\beta_{ij} - \rho)(\beta_{ij} + \rho)(\beta_{ij} - \tau) \\ \Rightarrow\ & 0 = (\beta_{ij}^2 - \rho^2)(\beta_{ij} - \tau) \\ \Rightarrow\ & 0 = \beta_{ij}^3 - \tau\beta_{ij}^2 - \rho^2\beta_{ij} + \rho^2\tau \\ \Rightarrow\ & 0 = \beta_{ij}^2 - \tau\beta_{ij} - \rho^2 + \rho^2\tau\frac{1}{\beta_{ij}}, \quad \because \beta_{ij} \neq 0 \\ \Rightarrow\ & 0 = \alpha_{ij} - \tau\beta_{ij} - \rho^2 + \rho^2\tau b_{ij}. \end{aligned}$$

By evaluating this equation for each case when β_{ij} can only be 3 of the 4 discrete values we get the following cuts:

$$\begin{aligned} 0 &= \alpha_{ij} - \mu\beta_{ij} - \sigma^2 + \sigma^2\mu b_{ij} && \text{if } \beta_{ij} \in \{\pm\sigma, \mu\} && (\rho = \sigma, \tau = \mu) \\ 0 &= \alpha_{ij} + \mu\beta_{ij} - \sigma^2 - \sigma^2\mu b_{ij} && \text{if } \beta_{ij} \in \{\pm\sigma, -\mu\} && (\rho = \sigma, \tau = -\mu) \\ 0 &= \alpha_{ij} - \sigma\beta_{ij} - \mu^2 + \mu^2\sigma b_{ij} && \text{if } \beta_{ij} \in \{\sigma, \pm\mu\} && (\rho = \mu, \tau = \sigma) \\ 0 &= \alpha_{ij} + \sigma\beta_{ij} - \mu^2 - \mu^2\sigma b_{ij} && \text{if } \beta_{ij} \in \{-\sigma, \pm\mu\} && (\rho = \mu, \tau = -\sigma). \end{aligned}$$

■

The goal of transitivity conditions is to identify nodes in which $|\Upsilon_{ij}| = 0$, allowing us to prune the node without spending the computational time required to solve it. If transitivity conditions cannot fathom the node they can still be of benefit by potentially providing constraints that can tighten the relaxation. We will now introduce definitions and notation that will be used within this chapter.

4.2 Definitions and Notation for Transitivity

The following definitions are required for this chapter. Each is further described using an example.

Definition 11 (Pairwise Variable Set)

For a set I of m facilities, the pairwise variable set, Π^m, is the set of discrete variables β for all pairs of facilities in I.

Example: For the set $I = \{i, j, k\}$ of size $m = 3$ the pairwise variable set is: $\Pi^3 = (\beta_{ij}, \beta_{ik}, \beta_{jk})$.

By assigning one of the four possible values $(\pm\sigma, \pm\mu)$ to each of the β variable in a pairwise variable we create a pairwise value combination.

Definition 12 (Pairwise Value Combination)

For a set I of m facilities, a pairwise value combination is a mapping from each β in the pairwise variable set Π^m to one of the discrete values $\pm\sigma$ or $\pm\mu$.

Example: For the set $I = \{i, j, k\}$ of size $m = 3$ one example of a pairwise value combination is $(\beta_{ij}, \beta_{ik}, \beta_{jk}) = (\sigma, \sigma, \mu)$.

Grouping all possible pairwise value combinations together results in a complete value combination set that is defined below.

Definition 13 (Complete Value Combination Set)
For a set I of m facilities, the complete value combination set, Φ^m, is the set of all pairwise value combinations.

Example: For the set $I = \{i, j, k\}$ of size $m = 3$ the complete value combination set can be written out as: $\Phi^3 = \{(\beta_{ij}, \beta_{ik}, \beta_{jk}) : \beta_{ij} \in \{\pm\sigma, \pm\mu\},\ \beta_{ik} \in \{\pm\sigma, \pm\mu\},\ \beta_{jk} \in \{\pm\sigma, \pm\mu\}\}$.

The complete value combination set for m facilities is composed of $\binom{m}{2}^4$ pairwise value combinations; however not all of these combinations make sense as potential layouts. Therefore the complete value combination set Φ can be partitioned into two disjoint sets: the pairwise value combinations that can represent feasible layouts and the combinations that cannot.

Definition 14 (Feasible and Infeasible Value Combination Set)
For the set $I = \{i, j, k\}$ of size $m = 3$:

Φ^F *= the set of pairwise value combinations that can represent feasible layouts*

Φ^I *= the set of pairwise value combinations that cannot represent feasible layouts*

where $\Phi^m = \Phi^F \cup \Phi^I$ *and* $\Phi^F \cap \Phi^I = \emptyset$.

The pairwise variable set and the size of the complete value combination set are included in Table 4.1.

Size m	Indices I	Pairwise Variable Set Π^m	Complete Value Combination Set Size $\lvert\Phi^m\rvert$
2	$I = \{i, j\}$	$\Pi^2 = (\beta_{ij})$	1
3	$I = \{i, j, k\}$	$\Pi^3 = (\beta_{ij}, \beta_{ik}, \beta_{jk})$	64
4	$I = \{i, j, k, l\}$	$\Pi^4 = (\beta_{ij}, \beta_{ik}, \beta_{il}, \beta_{jk}, \beta_{jl}, \beta_{kl})$	4096

Table 4.1: The Pairwise Variable Sets of sizes 2, 3 and 4

The purpose of the upcoming sections are to define the specific pairwise value combinations within the sets Φ^F and Φ^I for facilities of any width and height. This leaves the indices as the only distinguishing feature of these sets, in other words β_{ij} is defined only when $i < j$. (The sets Φ^F and Φ^I are made up of pairwise value combinations which are built from β_{ij}s.) Therefore we will describe transitivity conditions using the following steps:

1. Describe a property of layouts using unordered facilities. To distinguish between ordered and unordered facilities we will refer to unordered facilities a, b by using the terms 'above', 'below', 'left' and 'right'.

2. Consider every ordering of the unordered facilities. To do this we will map our unordered facilities to every possible sequence of ordered facilities.

3. Generate groups of pairwise value combinations. By combining a property of unordered facilities with all mappings we will produce a set of pairwise value combinations that satisfy the initial property.

By describing layouts using unordered facilities we group pairwise value combinations which prevents us from having to consider each one individually. In essence we will be describing all layouts that “look the same” and then considering all possible ways of labelling the layout using our ordered facility names i, j, k. We will portray transitivity conditions using separation illustrations. They are labelled using a, b and c to distinguish them and differ from previous figures since they represent the layout of a set of generic facilities in which the widths and heights are arbitrary. We will clarify the process to describe transitivity conditions, describe how separation illustrations represent layouts that “look the same” and explain how they generate multiple pairwise value combinations through an example. However we begin by introducing the following notation for unordered facilities that will simplfiy the discussions and proofs and can easily be related to the variable β_{ij} by ordering the unordered facilities a, b.

All forms of transitivity that we will discuss are based on the relationship between the edges of the facilities. Although these edges can be defined using the coordinate of the centers and the half-heights or half-widths, the notation can be cumbersome. For this reason we will introduce the following notation:

$$\begin{aligned}
B_i &= \text{Bottom edge of facility } i \Rightarrow B_i = y_i - \tfrac{1}{2}h_i \\
T_i &= \text{Top edge of facility } i \Rightarrow T_i = y_i + \tfrac{1}{2}h_i \\
L_i &= \text{Left edge of facility } i \Rightarrow L_i = x_i - \tfrac{1}{2}w_i \\
R_i &= \text{Right edge of facility } i \Rightarrow R_i = x_i + \tfrac{1}{2}w_i
\end{aligned}$$

Using this notation we can describe layouts using both sets of notation:

- Definition 15 uses unordered facilities a, b to define the terms ‘above’, ‘below’, ‘left of ’ and ‘right of’.
- Definition 16 uses ordered facilities i and j and the decision variable β_{ij}.

Definition 15 (a above b)

$$\begin{aligned}
&a \textit{ above } b && \Leftrightarrow B_a \geq T_b \\
&a \textit{ below } b && \Leftrightarrow B_b \geq T_a \\
&a \textit{ left of } b && \Leftrightarrow \begin{cases} L_b \geq R_a \\ T_a > B_b \textit{ and } T_b > B_a \end{cases} \\
&a \textit{ right of } b && \Leftrightarrow \begin{cases} L_a \geq R_b \\ T_a > B_b \textit{ and } T_b > B_a \end{cases}
\end{aligned}$$

Note that describing a layout using the terms ‘above’ and ‘below’ take precedence over the terms ‘left’ and ‘right’. This is in alignment with the vertical separation axis taking precedent over horizontal separation. In addition these definitions clearly imply that a above b is equivalent to b below a.

Definition 16 (Top and Bottom Edges related to β)

$$\begin{aligned}
B_j \geq T_i &\quad \Rightarrow \beta_{ij} = \mu \\
B_i \geq T_j &\quad \Rightarrow \beta_{ij} = -\mu \\
\left.\begin{array}{c} L_j \geq R_i \\ T_i > B_j \;\&\; T_j > B_i \end{array}\right\} &\Rightarrow \beta_{ij} = \sigma \\
\left.\begin{array}{c} L_i \geq R_j \\ T_i > B_j \;\&\; T_j > B_i \end{array}\right\} &\Rightarrow \beta_{ij} = -\sigma
\end{aligned}$$

Note that strict inequalities have been used for the relation of the top and bottom edges of facilities with a horizontal separation axis. This is because any layout only needs to be described by one pairwise value combination. This was done when horizontal and vertical separation axes were defined; however due to computational challenges of open sets the closure was used. In this instance open sets will not cause difficultity as transitivity conditions are theoretically proven using Definition 15 and then pairwise value combinations are explicitly produced using Definition 16.

The relation between these definitions depends on the mapping of unordered to ordered facilities. We relate the terms 'above', 'below', 'right of' and 'left of' to the value of β_{ij} by considering the two ordering of a, b in turn.

Lemma 14

$$\textit{For the mapping } (a,b) \rightarrow (i,j): \begin{cases} a \textit{ is below } b & \Rightarrow \beta_{ij} = \mu \\ a \textit{ is above } b & \Rightarrow \beta_{ij} = -\mu \\ a \textit{ is left of } b & \Rightarrow \beta_{ij} = \sigma \\ a \textit{ is right of } b & \Rightarrow \beta_{ij} = -\sigma \end{cases}$$

$$\textit{For the mapping } (a,b) \rightarrow (j,i): \begin{cases} a \textit{ is above } b & \Rightarrow \beta_{ij} = \mu \\ a \textit{ is below } b & \Rightarrow \beta_{ij} = -\mu \\ a \textit{ is right of } b & \Rightarrow \beta_{ij} = \sigma \\ a \textit{ is left of } b & \Rightarrow \beta_{ij} = -\sigma \end{cases}$$

We will now present an example to describe separation illustrations and outline the process we will use to generate pairwise value combinations that satisfy transitivity conditions.

An Example Using Separation Illustrations

Recall the steps we outlined previously to describe transitivity conditions. In this example we will use a separation illustration to show six feasible pairwise value combinations.

1. Describe using unordered facilities

Consider the layout when a is right of c, a is left of b and b is above c (shown in Figure 4.1). Note that the focus of Figure 4.1 is not the coordinates of the facilities but the relative placement of them. In this case both pairs a, b and a, c are horizontally separated (and therefore vertically overlapping) while the pair b, c is vertically separated. The dotted line highlights this vertical separation.

2. Map to ordered facilities

For three facilities there are six mappings of a, b, c to i, j, k. The six mappings are:

$$(a,b,c) \rightarrow \{(i,j,k),(i,k,j),(j,i,k),(j,k,i),(k,i,j),(k,j,i)\}$$

3. Generate the pairwise value combinations

Figure 4.2 shows each of the six possible mappings and highlights how the layouts generated will "look the same". Despite their similar appearance each layout denotes a different pairwise value combination. This is

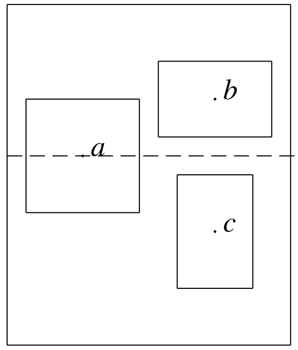
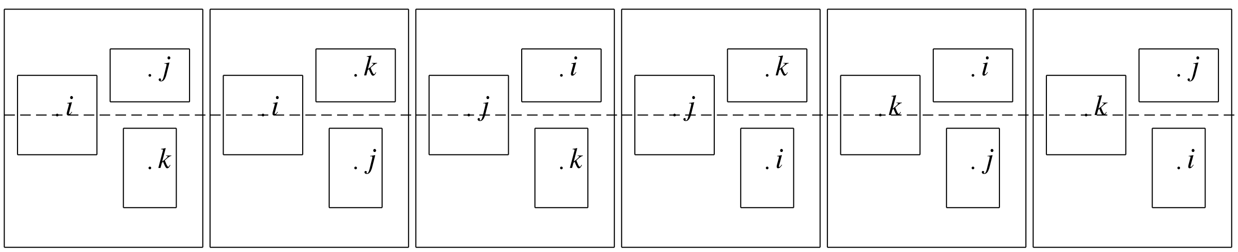


Figure 4.1: Separation Illustration Example - a is right of c and a is left of b and b is above c

because β_{ij} is always defined for $i < j$. Therefore by considering each mapping from an unordered separation illustration to ordered facilities different values of $(\beta_{ij}, \beta_{ik}, \beta_{jk})$ are produced. In addition since a, b and c are generic facilities, regardless of how they are mapped to facilities i, j and k this layout is feasible.



(a) $(\sigma, \sigma, -\mu)$ (b) (σ, σ, μ) (c) $(-\sigma, -\mu, \sigma)$ (d) $(-\sigma, \mu, \sigma)$ (e) $(-\mu, -\sigma, -\sigma)$ (f) $(\mu, -\sigma, -\sigma)$

Figure 4.2: The six pairwise value combinations generated from the separation illustration

As oppose to showing steps 2 and 3 in full, we will combine the steps and summarize the results in Table 4.2.

Feasible			
(a, b, c)	β_{ij}	β_{ik}	β_{jk}
(i, j, k)	σ	σ	$-\mu$
(i, k, j)	σ	σ	μ
(j, i, k)	$-\sigma$	$-\mu$	σ
(j, k, i)	$-\sigma$	μ	σ
(k, i, j)	$-\mu$	$-\sigma$	$-\sigma$
(k, j, i)	μ	$-\sigma$	$-\sigma$

Table 4.2: The six mappings and the pairwise value combinations generated from the separation illustration

Although this example was for 3 facilities the process can be extended for 4 facilities. The difference is simply that there are 24 mappings of unordered facilities a, b, c and d to i, j, k and l, as oppose to six. We can now describe 3-way transitivity conditions.

4.3 3-Way Transitivity Conditions

Given 3 facilities the complete value combination set has 64 pairwise value combinations, where 48 of these combinations are feasible layouts. The remaining 16 are infeasible for facilities of any width and height. The focus of this section will be to identify the pairwise value combinations in each set, beginning with the set of infeasible pairwise value combinations.

4.3.1 Infeasible Pairwise Value Combinations

To show that 16 of the 64 pairwise value combinations are infeasible we will need to examine two infeasibility conditions. For each of these conditions we will describe the condition using unordered facilities and map these to ordered facilities to generate pairwise value combinations that are infeasible.

Lemma 15 (3-Way Infeasibility Condition 1) *If a is above b and b is above c then* $\left\{\begin{array}{c} a \text{ is below } c \\ a \text{ is left of } c \\ a \text{ is right of } c \end{array}\right\}$ *all describe infeasible layouts.*

Proof

Let a be above b and b be above c then by Lemma 15 the following inequalities hold:

$$\begin{array}{rll} a \text{ above } b & \Rightarrow B_a \geq T_b & (1) \\ b \text{ above } c & \Rightarrow B_b \geq T_c & (2) \\ \text{and by definition} & T_b > B_b & (3) \end{array}$$

Combining these inequalities we get: $B_a \underbrace{\geq}_{(1)} T_b \underbrace{>}_{(3)} B_b \underbrace{\geq}_{(2)} T_c \Rightarrow B_a > T_c$.

We will now show that $B_a > T_c$ contradicts with each of the descriptions of a and c.

Consider the descriptions and their implications according to definition 15 in turn:

1. Assume a is below c. $\Leftrightarrow B_c \geq T_a$. Then since $T_c > B_c$ and $T_a > B_a$ by definition then we get: $T_c > B_c \geq T_a \geq B_a \Rightarrow T_c > B_a$, a contradiction since $B_a > T_c$.

2. Assume a is left of c. $\Leftrightarrow T_a > B_c$ and $T_c > B_a$. Then the second inequality implies a contradiction since $B_a > T_c$.

3. Assume a is right of c. $\Leftrightarrow T_a > B_c$ and $T_c > B_a$. Then the second inequality implies a contradiction since $B_a > T_c$.

■

For each orderings of facilities a, b and c we are able to rewrite Lemma 15 in terms of β, (using Lemma 14) and identify three pairwise value combinations that are infeasible. For the mapping $(a, b, c) \rightarrow (i, j, k)$ translating Lemma 15 generates the following infeasible combinations.

For $(a, b, c) \rightarrow (i, j, k)$:

$$\text{If } \underbrace{\beta_{ij} = -\mu}_{(a \text{ is above } b)} \text{ and } \underbrace{\beta_{jk} = -\mu}_{(b \text{ is above } c)} \text{ then } \left\{\begin{array}{ll} \beta_{ik} = \sigma & (a \text{ is left of } c) \\ \beta_{ik} = -\sigma & (a \text{ is right of } c) \\ \beta_{ik} = \mu & (a \text{ is below } c) \end{array}\right\} \text{ describe infeasible layouts.}$$

Therefore $(\beta_{ij}, \beta_{ik}, \beta_{jk}) \in \{(-\mu, \sigma, -\mu), (-\mu, -\sigma, -\mu), (-\mu, \mu, -\mu)\}$ is infeasible.

The other mappings follow similarily, therefore the pairwise value combinations for a is left of, right of and below c are summarized in Tables 4.3(a)-4.3(c), respectively. After eliminating the repetition from Table 4.3(c) we have a total of 14 infeasible pairwise value combinations.

(a) Infeasible for a left of c

Infeasible			
(a,b,c)	β_{ij}	β_{ik}	β_{jk}
(i,j,k)	$-\mu$	σ	$-\mu$
(i,k,j)	σ	$-\mu$	μ
(j,i,k)	μ	$-\mu$	σ
(j,k,i)	$-\sigma$	μ	$-\mu$
(k,i,j)	$-\mu$	μ	$-\sigma$
(k,j,i)	μ	$-\sigma$	μ

(b) Infeasible for a right of c

Infeasible			
(a,b,c)	β_{ij}	β_{ik}	β_{jk}
(i,j,k)	$-\mu$	$-\sigma$	$-\mu$
(i,k,j)	$-\sigma$	$-\mu$	μ
(j,i,k)	μ	$-\mu$	$-\sigma$
(j,k,i)	σ	μ	$-\mu$
(k,i,j)	$-\mu$	μ	σ
(k,j,i)	μ	σ	μ

(c) Infeasible for a below c

Infeasible			
(a,b,c)	β_{ij}	β_{ik}	β_{jk}
(i,j,k)	$-\mu$	μ	$-\mu$
(i,k,j)	μ	$-\mu$	μ
(j,i,k)	μ	$-\mu$	μ
(j,k,i)	$-\mu$	μ	$-\mu$
(k,i,j)	$-\mu$	μ	$-\mu$
(k,j,i)	μ	$-\mu$	μ

$$(\beta_{ij},\ \beta_{ik},\ \beta_{jk}) \in \left\{ \overbrace{\begin{array}{lll} (\pm\sigma,\ \mu\ ,-\mu), & (\ \mu\ ,\pm\sigma,\ \mu\ ), & (\ \mu\ ,-\mu,\pm\sigma), \\ (\pm\sigma,-\mu,\ \mu\ ), & (-\mu,\pm\sigma,-\mu), & (-\mu,\ \mu\ ,\pm\sigma), \end{array}}^{a \text{ left of or right of } c} \quad \overbrace{\begin{array}{l} (\ \mu\ ,-\mu,\ \mu\ ) \\ (-\mu,\ \mu\ ,-\mu) \end{array}}^{a \text{ below } c} \right\} \text{ is infeasible}$$

Table 4.3: Infeasible Pairwise Value Combinations

We will now consider the second condition that will be used to find infeasible pairwise value combinations. This condition is similar to the third layout described in Lemma 15. Analogous conditions for the first two layouts do not exist because the vertical separation axis (i.e. ‘above’ and ‘below’) is preferred.

Lemma 16 (3-Way Infeasibility Condition 2) *If a is left of b and b is left of c then a is right of c describes an infeasible layout.*

Proof

Let a be left of b and b be left of c then by Lemma (15) the following inequalities hold:

a left b $\Rightarrow L_b \geq R_a$ (1)

b left c $\Rightarrow L_c \geq R_b$ (2)

and by definition $R_b > L_b$ (3)

Combining these inequalities we get: $L_c \underbrace{\geq}_{(2)} R_b \underbrace{>}_{(3)} L_b \underbrace{\geq}_{(1)} R_a \Rightarrow L_c > R_a$.

We will now show that $L_c > R_a$ contradicts with the descriptions of a and c.

Assume a is right of c. $\Rightarrow L_a \geq R_c$. Then since $R_a > L_a$ and $R_c > L_c$ (by definition) then we get: $R_a > L_a \geq R_c > L_c \Rightarrow R_a > L_c$, a contradiction since $L_c > R_a$.

■

By considering all orderings and eliminating the repetition we get that the following two pairwise value combinations are infeasible:

To confirm that there are only 16 infeasible pairwise value combinations we will show that remain 48 combinations are feasible.

Infeasible			
(a,b,c)	β_{ij}	β_{ik}	β_{jk}
(i,j,k)	σ	$-\sigma$	σ
(i,k,j)	$-\sigma$	σ	$-\sigma$
(j,i,k)	$-\sigma$	σ	$-\sigma$
(j,k,i)	σ	$-\sigma$	σ
(k,i,j)	σ	$-\sigma$	σ
(k,j,i)	$-\sigma$	σ	$-\sigma$

$$(\beta_{ij},\ \beta_{ik},\ \beta_{jk}) \in \begin{Bmatrix} (\ \sigma\ , -\sigma,\ \ \sigma\ ) \\ (-\sigma,\ \ \sigma\ , -\sigma) \end{Bmatrix} \text{is infeasible}$$

Table 4.4: Infeasible Pairwise Value Combinations

4.3.2 Feasible Pairwise Value Combinations

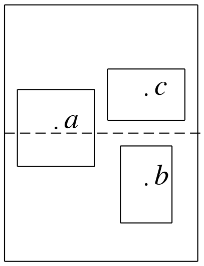
A pairwise value combinations belongs to the feasible value combination set if a layout exists for facilities of any widths and heights. To generate feasible combinations we will once again begin with unordered facilities and then consider all mappings to generate the combinations. We will use separation illustrations to portray the layout of generic facilities, as oppose to proving feasibility using Lemma 14 (as was done for infeasible layouts) since the proofs follow directly.

For 3 unordered facilities there are 8 distinct separation illustrations. Figure 4.3 shows each of the separation illustrations and the six pairwise value combinations that each produce. The mappings of (a,b,c) have been removed from the summary tables due to space, however they are identical to the previous tables shown.

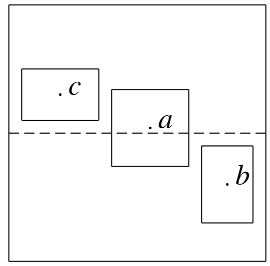
- Port: a is left of b, a is right of c, b is below c
- Slope: a is left of b, a is right of c, b is below c
- Valley: a is below of b, a is below of c, b is left of c
- Starboard: a is right of b, a is right of c, b is below c
- Hill: a is right of b, a is left of c, b is below c
- Mountain: a is above b, a is above c, b is left of c
- Horizontal Line: a is left of b, a is left of c, b is left of c
- Vertical Line: a is below b, a is below c, b is below c

The 48 pairwise value combinations are unique and with the 16 infeasible pairwise value combinations identified in section 4.3.1 yield the complete value combination set.

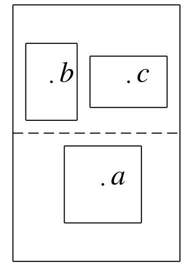
Port		
β_{ij}	β_{ik}	β_{jk}
σ	σ	μ
σ	σ	$-\mu$
$-\sigma$	μ	σ
$-\sigma$	$-\mu$	σ
μ	$-\sigma$	$-\sigma$
$-\mu$	$-\sigma$	$-\sigma$



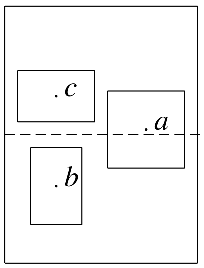
Slope		
β_{ij}	β_{ik}	β_{jk}
σ	$-\sigma$	μ
$-\sigma$	σ	$-\mu$
$-\sigma$	μ	$-\sigma$
σ	$-\mu$	σ
μ	$-\sigma$	σ
$-\mu$	σ	$-\sigma$



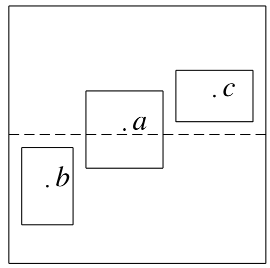
Valley		
β_{ij}	β_{ik}	β_{jk}
μ	μ	σ
μ	μ	$-\sigma$
$-\mu$	σ	μ
$-\mu$	$-\sigma$	μ
σ	$-\mu$	$-\mu$
$-\sigma$	$-\mu$	$-\mu$



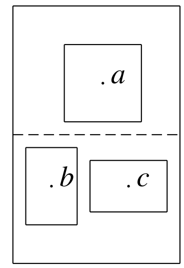
Starboard		
β_{ij}	β_{ik}	β_{jk}
$-\sigma$	$-\sigma$	μ
$-\sigma$	$-\sigma$	$-\mu$
σ	μ	$-\sigma$
σ	$-\mu$	$-\sigma$
μ	σ	σ
$-\mu$	σ	σ



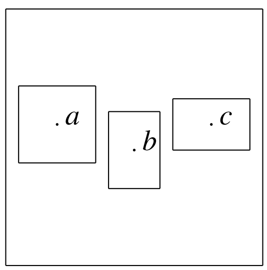
Hill		
β_{ij}	β_{ik}	β_{jk}
$-\sigma$	σ	μ
σ	$-\sigma$	$-\mu$
σ	μ	σ
$-\sigma$	$-\mu$	$-\sigma$
μ	σ	$-\sigma$
$-\mu$	$-\sigma$	σ



Mountain		
β_{ij}	β_{ik}	β_{jk}
$-\mu$	$-\mu$	σ
$-\mu$	$-\mu$	$-\sigma$
μ	σ	$-\mu$
μ	$-\sigma$	$-\mu$
σ	μ	μ
$-\sigma$	μ	μ



Horizontal Line		
β_{ij}	β_{ik}	β_{jk}
σ	σ	σ
σ	σ	$-\sigma$
$-\sigma$	σ	σ
$-\sigma$	$-\sigma$	σ
σ	$-\sigma$	$-\sigma$
$-\sigma$	$-\sigma$	$-\sigma$



Vertical Line		
β_{ij}	β_{ik}	β_{jk}
μ	μ	μ
μ	μ	$-\mu$
$-\mu$	μ	μ
$-\mu$	$-\mu$	μ
μ	$-\mu$	$-\mu$
$-\mu$	$-\mu$	$-\mu$

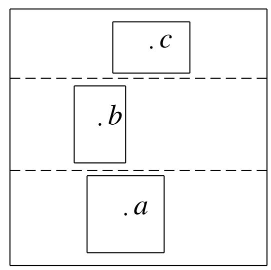


Figure 4.3: All separation illustrations for 3 facilities with the pairwise value combinations generated

4.3.3 Summary of 3-way Transitivity

For 3 facilities there are 64 possible pairwise value combinations; however each combination does not represent a feasible layout. Using unordered facilities we described two 3-way infeasibility conditions and then generated 16 infeasible pairwise value combinations by mapping the unordered facilities to ordered facilities. The mapping of unordered facilities to ordered facilities was used in conjunction with separation illustrations to identify the 48 feasible pairwise value combinations. The feasible and infeasible pairwise value combinations are listed in Figure 4.3 and Tables 4.3 and 4.4 respectively.

For facility layout problems of more then three facilities, 3-way infeasibility can be applied to every group of three facilities. For example, for 4 facilities the complete value combination set has 4096 pairwise value combinations, of which 2632 are infeasible by 3-way transitivity. The next section will examine additional transitivity conditions related to 4 facilities.

4.4 4-Way Transitivity Conditions

Given 4 facilities the complete value combination set has 4096 pairwise value combinations. The purpose of this section is to identify combinations that are infeasible. Unlike 3-way transitivity conditions studied in the previous section, 4-way transitivity conditions can be caused by different factors. Within this section we will consider the following 3 causes of infeasibility.

1. **Already 3-way transitivity infeasible:** These are pairwise value combinations where 3 of the 4 facilities are infeasible by 3-way transitivity and clearly, adding a 4^{th} facility cannot make it feasible.

2. **4-Way Transitivity Infeasible:** These pairwise value combinations cannot generate a feasible layout for facilities of any width and height; however taking any 3 of the 4 facilities the combination can produce a feasible layout. In other words these pairwise value combinations are strictly 4-way transitivity infeasible and therefore this set is disjoint from cause 1.

3. **Infeasibility with respect to height:** The previous two conditions held for facilities of any height; however the pairwise value combinations in thie subset are only infeasible when a general height relationship (such as $h_i \geq h_j$) holds.

4.4.1 Infeasible Pairwise Value Combinations Independent of Facility Heights

The following lemma will identify 96 infeasible layouts; however some of these layouts are infeasible by 3-way transitivity. The goal of this section is to identify the layouts that are infeasible by 4-way transitivity, therefore after we examine the lemma we will consider which layouts are already 3-way infeasible.

Lemma 17 (4-way Transitivity Condition)

$$\textit{If } a \textit{ is horizontally separated with } b, a \textit{ is below } d \textit{ and } b \textit{ is above } c \textit{ then} \left\{ \begin{array}{c} c \textit{ is above } d \\ c \textit{ is left of } d \\ c \textit{ is right of } d \end{array} \right\} \textit{described infeaible layouts}.$$

Proof

Let a is horizontally separated with b, a is below d and b is above c. Then by Lemma (15) the following inequalities hold:

$$\begin{array}{ll} b \text{ above } c \Rightarrow B_b \geq T_c & (1) \\ a \text{ horizontally separated from } b \Rightarrow T_a > B_b \text{ and } T_b > B_a & (2) \\ a \text{ below } d \Rightarrow B_d \geq T_a & (3) \end{array}$$

Combining these inequalities we get:

$$B_d \underbrace{\geq}_{(3)} T_a \underbrace{>}_{(2)} B_b \underbrace{\geq}_{(1)} T_c \Rightarrow B_d > T_c.$$

We will now show that $B_d > T_c$ contradicts with each of the descriptions of c and d. Consider the following two cases:

1. Assume c is above d then

$$\left.\begin{array}{ll} c \text{ is above } d & \Rightarrow B_c \geq T_d \\ \text{by definition} & \Rightarrow T_d > B_d \;\&\; T_c > B_c \end{array}\right\} \Rightarrow T_c > B_c \geq T_d > B_d \Rightarrow T_c > B_d, \text{ a contradiction since } B_d > T_c.$$

2. Assume c is either left of or right of d then by Lemma (15) $T_c > B_d$ and $T_d > B_c$. Then the first inequality implies a contradiction since $B_d > T_c$.

■

We will partition the pairwise value combinations from Lemma 17 into 2 sets by:

1. identifying the combinations that satisfy 3-way infeasibility conditions
2. showing the remaining combinations are not 3-way infeasible.

In order to identify which pairwise value combinations are 3-way infeasible we will use Lemma 15 and various groups of three facilities to show the combinations which are infeasible by only considering 3 facilities. We will split our discussion into the same two cases examined in the previous proof, beginning with when c is above d.

Case 1: When c is above d, applying Lemma 15 to:

- facilities *acd* implies all layouts described are infeasible by 3-way transitivity.

Case 2: When c is left of or right of d, applying Lemma 15 to:

- facilities *abc* implies layouts that also have a below c are infeasible by 3-way transitivity.
- facilities *acd* implies layouts that also have a above c are infeasible by 3-way transitivity.
- facilities *abd* implies layouts that also have b below d are infeasible by 3-way transitivity.
- facilities *abd* implies layouts that also have b above d are infeasible by 3-way transitivity.

For the mapping $(a, b, c, d) \rightarrow (i, j, k, l)$ Lemma 17 can be rewritten in terms of β. The pairwise value combinations for this mapping are listed in Figure 4.4(a). No relation between a and c is given in the lemma because the layouts are infeasible for all 4 arrangements. For this mapping this implies $\beta_{ik} \in \{\pm\sigma, \pm\mu\}$. The same is true for the relation between b and d.

The mapping can also be applied to the aforementioned facts. This generates the 80 pairwise value combinations that are infeasible by 3-way transitivity listed in Figure 4.4(b). Figure 4.4(c) lists the remaining 16 pairwise value combinations identified in Lemma 17.

Before considering the 24 mappings that exist for 4 facilities we will show that the pairwise value combinations identified in Figure 4.4(c) are 4-way infeasible. Recall that this means they are only infeasible when all 4 facilities are considered. Therefore a conclusion of 4-way infeasible can be drawn if all of the

β_{ij}	β_{ik}	β_{il}	β_{jk}	β_{jl}	β_{kl}
$\pm\sigma$	$\{\pm\sigma, \pm\mu\}$	μ	$-\mu$	$\{\pm\sigma, \pm\mu\}$	$\{\pm\sigma, -\mu\}$

(a) Infeasible layouts from Lemma 17

β_{ij}	β_{ik}	β_{il}	β_{jk}	β_{jl}	β_{kl}
$\pm\sigma$	$\{\pm\sigma, \pm\mu\}$	μ	$-\mu$	$\{\pm\sigma, \pm\mu\}$	$-\mu$
$\pm\sigma$	$\pm\sigma$	μ	$-\mu$	$\pm\sigma$	$\pm\sigma$
$\pm\sigma$	$\pm\sigma$	μ	$-\mu$	$\pm\mu$	$\pm\sigma$
$\pm\sigma$	$\pm\mu$	μ	$-\mu$	$\pm\sigma$	$\pm\sigma$

(b) Infeasible by 3-way Transitivity

β_{ij}	β_{ik}	β_{il}	β_{jk}	β_{jl}	β_{kl}
$\pm\sigma$	$\pm\mu$	μ	$-\mu$	$\pm\mu$	$\pm\sigma$

(c) Infeasible by 4-way Transitivity

Figure 4.4: For the mapping $(a, b, c, d) \rightarrow (i, j, k, l)$ the infeasible pairwise value combinations identified by Lemma 17 categorized based on why they are infeasible.

pairwise value combinations generated by considering every group of three facilities are feasible. Table 4.5 identifies the pairwise value combination generated by each group of three facilities for the mapping $(a, b, c, d) \rightarrow (i, j, k, l)$, as the other mappings would follow similarily, and the 3-way feasibility tables in Figure 4.3 are used to show that each pairwise value combination is feasible.

	β_{ij}	β_{ik}	β_{il}	β_{jk}	β_{jl}	β_{kl}
Group of 3 Facilities	$\pm\sigma$	$\pm\mu$	μ	$-\mu$	$\pm\mu$	$\pm\sigma$
i, j, k	$\pm\sigma$	$\pm\mu$		$-\mu$		
i, j, l	$\pm\sigma$		μ		$\pm\mu$	
i, k, l		$\pm\mu$	μ			$\pm\sigma$
j, k, l				$-\mu$	$\pm\mu$	$\pm\sigma$

Table 4.5: Considering Groups of 3 facilities

We can now consider the 24 mappings of (a, b, c, d). Table 4.6 lists only the pairwise value combinations from Lemma 17 that are infeasible by 4-way transitivity.

4.4.2 Infeasibility with respect to Height

In the previous section we constructed the feasible and infeasible value combination set for 3 facilities and showed that, regardless of the height of the facilities, the pairwise value combinations within each set are the same. The feasible and infeasible value combination sets for 4 facilities are not independent of the height of the facilities. Feasibility is not affected by widths because the disjunction of the separation axis condition is along the vertical axis. This section will identify pairwise value combinations that are infeasible only under certain height conditions by first proving infeasibility for unordered facilities and then applying this to ordered facilities; however we begin with a motivating example.

Suppose that 4 facilities are arranged so that b is above d and d is above c. Now we will examine is if all facilities can be to the right of facility a. Figure 4.5 illustrates the facilities as the height of facility d

(a,b,c,d)	β_{ij}	β_{ik}	β_{il}	β_{jk}	β_{jl}	β_{kl}
(i,j,k,l)	$\pm\sigma$	$\pm\mu$	μ	μ	$\pm\mu$	$\pm\sigma$
(i,j,l,k)	$\pm\sigma$	μ	$\pm\mu$	$\pm\mu$	μ	$\pm\sigma$
(i,k,j,l)	$\pm\mu$	$\pm\sigma$	μ	$-\mu$	$\pm\sigma$	$\pm\mu$
(i,k,l,j)	$\pm\mu$	μ	$\pm\sigma$	$\pm\sigma$	$-\mu$	$\pm\mu$
(i,k,j,l)	μ	$\pm\sigma$	$\pm\mu$	$\pm\mu$	$\pm\sigma$	μ
(i,k,l,j)	μ	$\pm\mu$	$\pm\sigma$	$\pm\sigma$	$\pm\mu$	$-\mu$
(j,i,k,l)	$\pm\sigma$	μ	$\pm\mu$	$\pm\mu$	μ	$\pm\sigma$
(j,i,l,k)	$\pm\sigma$	$\pm\mu$	μ	μ	$\pm\mu$	$\pm\sigma$
(j,k,i,l)	μ	$\pm\sigma$	$\pm\mu$	$\pm\mu$	$\pm\sigma$	μ
(j,k,l,i)	μ	$\pm\mu$	$\pm\sigma$	$\pm\sigma$	$\pm\mu$	$-\mu$
(j,l,i,k)	$\pm\mu$	$\pm\sigma$	μ	$-\mu$	$\pm\sigma$	$\pm\mu$
(j,l,k,i)	$\pm\mu$	μ	$\pm\sigma$	$\pm\sigma$	$-\mu$	$\pm\mu$
(k,i,j,l)	$\pm\mu$	$-\mu$	$\pm\sigma$	$\pm\sigma$	μ	$\pm\mu$
(k,i,l,j)	$\pm\mu$	$\pm\sigma$	$-\mu$	μ	$\pm\sigma$	$\pm\mu$
(k,j,i,l)	$-\mu$	$\pm\mu$	$\pm\sigma$	$\pm\sigma$	$\pm\mu$	μ
(k,j,l,i)	$-\mu$	$\pm\sigma$	$\pm\mu$	$\pm\mu$	$\pm\sigma$	$-\mu$
(k,l,i,j)	$\pm\sigma$	$\pm\mu$	$-\mu$	$-\mu$	$\pm\mu$	$\pm\sigma$
(k,l,j,i)	$\pm\sigma$	$-\mu$	$\pm\mu$	$\pm\mu$	$-\mu$	$\pm\sigma$
(l,i,j,k)	$-\mu$	$\pm\mu$	$\pm\sigma$	$\pm\sigma$	$\pm\mu$	μ
(l,i,k,j)	$-\mu$	$\pm\sigma$	$\pm\mu$	$\pm\mu$	$\pm\sigma$	$-\mu$
(l,j,i,k)	$\pm\mu$	$-\mu$	$\pm\sigma$	$\pm\sigma$	μ	$\pm\mu$
(l,j,k,i)	$\pm\mu$	$\pm\sigma$	$-\mu$	μ	$\pm\sigma$	$\pm\mu$
(l,k,i,j)	$\pm\sigma$	$-\mu$	$\pm\mu$	$\pm\mu$	$-\mu$	$\pm\sigma$
(l,k,j,i)	$\pm\sigma$	$\pm\mu$	$-\mu$	$-\mu$	$\pm\mu$	$\pm\sigma$

Table 4.6: Pairwise value combinations that are infeasible by 4-way transitivity

increases. Recall that that for facility b to be right of a then $T_a > B_b$ and $T_b > B_a$, similarily for c and d right of a. Therefore Figure 4.5(a) is feasible as $T_a > B_b$ and $T_c > B_a$; however as the height of d increases it no longer becomes possible for both b and c to be right of facility a (Figures 4.5(b) and 4.5(c)).

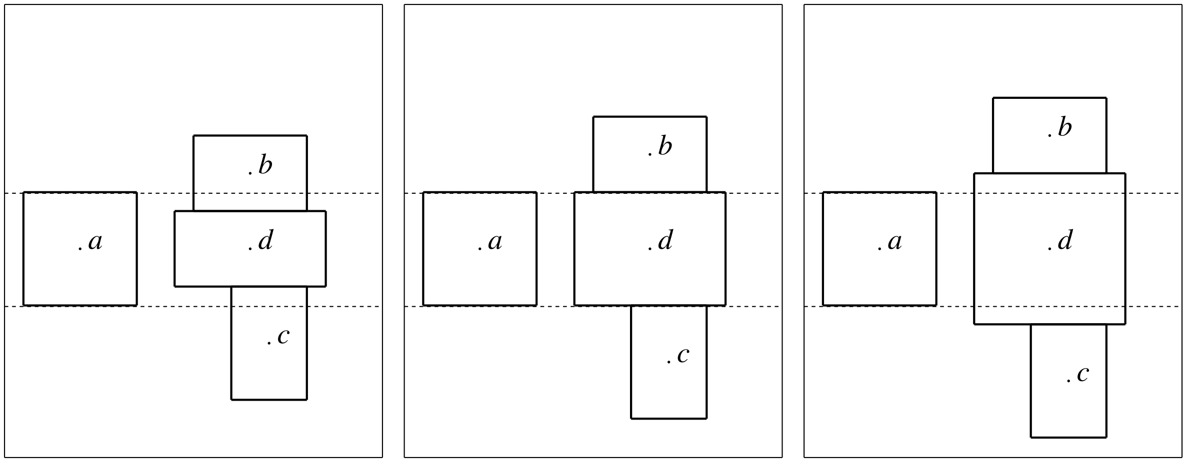
This is not the only example of the effect of the height of facility d. Similar examples can be seen by allowing each of facilities b, c and d to be left of or right of facility a while facility d is below b and above c. We will now prove this result formally.

Lemma 18 (Transitivity with respect to height)

$$\text{If } h_d \geq h_a \text{ then } \left\{ \begin{array}{l} a \text{ is left of or right of } b \\ a \text{ is left of or right of } c \\ a \text{ is left of or right of } d \\ b \text{ is above } d \\ d \text{ is above } c \end{array} \right\} \text{ describes an infeasible layout.}$$

Proof

Let a is left of or right of b , a is left of or right of c, a is left of or right of d, b is above d, d is above c and



(a) Feasible Layout (b) Infeasible Layout (c) Infeasible Layout

Figure 4.5: The effect the height of facilities has on feasibility

let $h_d \geq h_a$. Then by Lemma (15) the following inequalities hold:

$$
\begin{array}{llr}
a \text{ is left of or right of } b & \Rightarrow T_a > B_b \;\;\&\;\; T_b > B_a & (1) \\
a \text{ is left of or right of } c & \Rightarrow T_a > B_c \;\;\&\;\; T_c > B_a & (2) \\
d \text{ is above } c & \Rightarrow B_d \geq T_c & (3) \\
b \text{ is above } d & \Rightarrow B_b \geq T_d & (4)
\end{array}
$$

Combining these inequalities we get:

$$
\left.
\begin{array}{lcl}
T_a \underbrace{>}_{(1)} B_b \underbrace{\geq}_{(4)} T_d & \Rightarrow & T_a > T_d \\
-B_a \underbrace{>}_{(2)} -T_c \underbrace{\geq}_{(3)} -B_d & \Rightarrow & -B_a > -B_d
\end{array}
\right\} \Rightarrow h_a = T_a - B_a > T_d - B_d = h_d
$$

$$\Rightarrow h_a > h_d, \text{ which is a contradiction since } h_d \geq h_a.$$

Therefore the given description with the height condition describes an infeasible layout.

■

As for our previous infeasibility conditions, we can generate infeasible pairwise value combinations by considering Lemma 18 under every order of facilities a, b, c and d. It is important to note that the height condition is dependent on the assignment of facilities a and d. Table 4.7 summarizes the infeasible pairwise value combinations with the required height condition.

4.5 Concluding Remarks

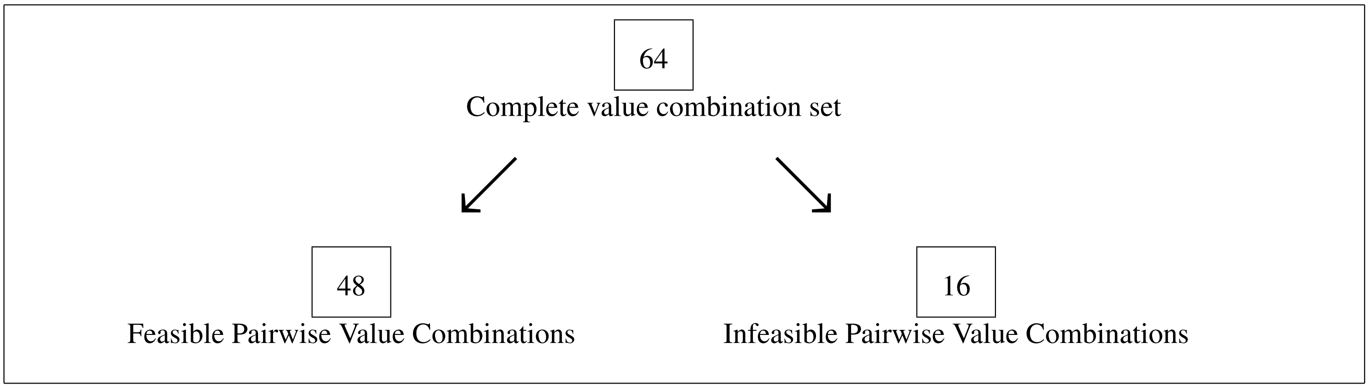
Within this chapter we examined transitivity conditions that are used to identify infeasible layouts. For 3 facilities the complete value combination set can be partitioned into feasible and infeasible combinations and this partition holds for facilities of any width and height. Similarily we constructed the set of pairwise

(a,b,c,d)	β_{ij}	β_{ik}	β_{il}	β_{jk}	β_{jl}	β_{kl}	Height Condition
(i,j,k,l)	$\pm\sigma$	$\pm\sigma$	$\pm\sigma$	$-\mu$	$-\mu$	μ	$h_i \geq h_j$
(i,j,l,k)	$\pm\sigma$	$\pm\sigma$	$\pm\sigma$	$-\mu$	$-\mu$	$-\mu$	$h_i \geq h_j$
(i,k,j,l)	$\pm\sigma$	$\pm\sigma$	$\pm\sigma$	μ	μ	$-\mu$	$h_i \geq h_k$
(i,k,l,j)	$\pm\sigma$	$\pm\sigma$	$\pm\sigma$	μ	μ	μ	$h_i \geq h_k$
(i,k,j,l)	$\pm\sigma$	$\pm\sigma$	$\pm\sigma$	μ	$-\mu$	$-\mu$	$h_i \geq h_l$
(i,k,l,j)	$\pm\sigma$	$\pm\sigma$	$\pm\sigma$	$-\mu$	μ	μ	$h_i \geq h_l$
(j,i,k,l)	$\pm\sigma$	$-\mu$	$-\mu$	$\pm\sigma$	$\pm\sigma$	μ	$h_j \geq h_i$
(j,i,l,k)	$\pm\sigma$	$-\mu$	$-\mu$	$\pm\sigma$	$\pm\sigma$	$-\mu$	$h_j \geq h_i$
(j,k,i,l)	$-\mu$	$\pm\sigma$	$-\mu$	$\pm\sigma$	μ	$\pm\sigma$	$h_j \geq h_k$
(j,k,l,i)	$-\mu$	$-\mu$	$\pm\sigma$	μ	$\pm\sigma$	$\pm\sigma$	$h_j \geq h_k$
(j,l,i,k)	$-\mu$	$\pm\sigma$	$-\mu$	$\pm\sigma$	$-\mu$	$\pm\sigma$	$h_j \geq h_l$
(j,l,k,i)	$-\mu$	$-\mu$	$\pm\sigma$	$-\mu$	$\pm\sigma$	$\pm\sigma$	$h_j \geq h_l$
(k,i,j,l)	$\pm\sigma$	μ	μ	$\pm\sigma$	$\pm\sigma$	$-\mu$	$h_k \geq h_i$
(k,i,l,j)	$\pm\sigma$	μ	μ	$\pm\sigma$	$\pm\sigma$	μ	$h_k \geq h_i$
(k,j,i,l)	μ	$\pm\sigma$	μ	$\pm\sigma$	$-\mu$	$\pm\sigma$	$h_k \geq h_j$
(k,j,l,i)	μ	μ	$\pm\sigma$	$-\mu$	$\pm\sigma$	$\pm\sigma$	$h_k \geq h_j$
(k,l,i,j)	μ	$\pm\sigma$	μ	$\pm\sigma$	μ	$\pm\sigma$	$h_k \geq h_l$
(k,l,j,i)	μ	μ	$\pm\sigma$	μ	$\pm\sigma$	$\pm\sigma$	$h_k \geq h_l$
(l,i,j,k)	$\pm\sigma$	μ	$-\mu$	$\pm\sigma$	$\pm\sigma$	$-\mu$	$h_l \geq h_i$
(l,i,k,j)	$\pm\sigma$	$-\mu$	μ	$\pm\sigma$	$\pm\sigma$	μ	$h_l \geq h_i$
(l,j,i,k)	μ	$\pm\sigma$	$-\mu$	$\pm\sigma$	$-\mu$	$\pm\sigma$	$h_l \geq h_j$
(l,j,k,i)	μ	$-\mu$	$\pm\sigma$	$-\mu$	$\pm\sigma$	$\pm\sigma$	$h_l \geq h_j$
(l,k,i,j)	$-\mu$	$\pm\sigma$	μ	$\pm\sigma$	μ	$\pm\sigma$	$h_l \geq h_k$
(l,k,j,i)	$-\mu$	μ	$\pm\sigma$	μ	$\pm\sigma$	$\pm\sigma$	$h_l \geq h_k$

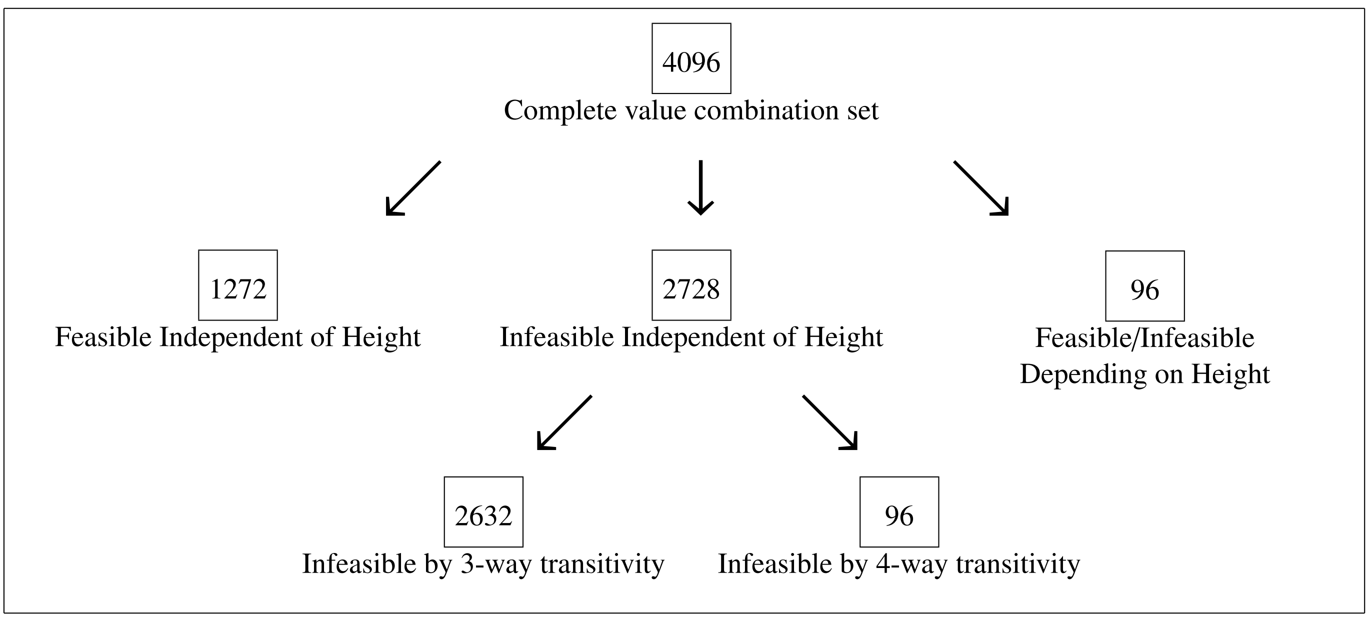
Table 4.7: Infeasible Pairwise Value Combinations under height conditions

value combinations that are infeasible for 4 facilities of any width and height; however this set could be further partitioned into combinations that are 3-way infeasible and those that are 4-way infeasible. Finally we identified combinations whose feasibility depends on the height of the facilities. The breakdown of the number of pairwise value combinations in each category is shown in Figure 4.6.

The purpose of identifying infeasible combinations is that knowing which values of β_{ij} are infeasible allows cuts to be added to the relaxation or entire nodes pruned.



(a) Three Facilities



(b) Four Facilities

Figure 4.6: Number of pairwise value combinations within each category

Chapter 5

Computational Results

A depth-first search branch-and-bound algorithm was used to combine solving the SDP relaxation and applying transitivity conditions. The relaxations were solved using SDP T3 and the branch-and-bound algorithm was implemented in Matlab and the discrete parameters σ and μ were set to 1 and $\frac{1}{2}$ respectively.

This chapter will present preliminary computational results regarding upper and lower bounds, percentage gaps and classification of nodes visited. The focus of computational experimentation, for the purpose of this thesis, was twofold: first the successful implementation shows proof of concept and second, it allows the dynamic nature of this unique SDP relaxation to be examined. The goal at this stage, was not to produce competitive results but to verify the applicability of themodel and present preliminary results. We begin by discussing the details of the branch-and-bound algorithm used.

5.1 Branch-and-Bound

The branching decision is based on exactness constraint (ExC 1) implying each branching for an i, j pair and each node can have four children.The general branching decisions vary slightly for the first level therefore we begin by examining how the branching variables and directions are selected for the first level of the tree.

Initial Branching Decision

Reflection along the x or y axis produces layouts with identical objective values. Therefore to reduce this symmetry the first pair of i, j facilities is only branched on $\beta_{ij} = \sigma$ and $\beta_{ij} = \mu$. Both of these branches are required because each facility has a fixed orientation and fixed dimensions, meaning rotational symmetry does not exist. Selecting only $\beta_{ij} = \sigma$ or $\beta_{ij} = \mu$ would determine the axis along which facilities i and j would be separated and since the height and width of each facility is fixed the separation axes are not interchangeable.

The first level of the branch-and-bound tree was produced by selecting two pairs that related only 3 facilities. This was done so that transitivity conditions would be more likely to hold. The two pairs were selected so that the total weighted cost for separation along the cheapest axis is maximized:

$$\max_{ij} c_{ij} \min[\tfrac{1}{2}(w_i + w_j), \tfrac{1}{2}(h_i + h_j)]$$

One variable is branched so that $\beta_{ij} = \{\sigma, \mu\}$, while the other is branched for $\beta_{jk} = \{\pm\sigma, \pm\mu\}$. Figure 5.1 illustrates the nodes solved in the inital level of the branch-and-bound tree. Of the 20 nodes shown, half are eliminated by reflectional symmetry and an additional 2 are eliminated because the first branching decision is based on 2 pairs. This leaves only 8 nodes that are solved which are marked with ✓. Reflectional symmetry eliminates the nodes where $\beta_{ij} = -\sigma$ or $-\mu$ and their children. The first branching decision is based on two pairs so that a wider array of nodes could be used to begin the dives. This implies that the nodes where $\beta_{ij} = \sigma$ or μ do not need to be solved.

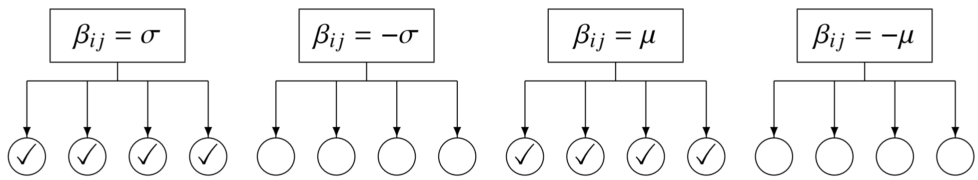


Figure 5.1: The nodes in the first two levels of the branch-and-bound tree that are solved

General Branching Decisions

The branching decisions for nodes that are not in the initial stage of the branch-and-bound algorithm are related to transitivity, therefore we begin by recalling the following two purposes of transitivity:

1. Identify infeasible layouts prior to solving the relaxation

2. Include additional constraints to tighten the relaxation

Then the branching decision is made so that:

- The i, j pair to branch on is selected by first finding the smallest increase in distance for each overlapping pair. Then selecting the i, j pair that maximizes the minimum weigthed increase in distance.

$$\max_{\forall i,j:|\Upsilon_{ij}|\neq 1} c_{ij} \min[\tfrac{1}{2}(w_i + w_j) - d^x_{ij}, \tfrac{1}{2}(h_i + h_j) - d^y_{ij}]$$

- The branching direction is randomly selected from the potentially feasible set (Definition 10). Recall that this set initially contains $\pm\sigma$ and $\pm\mu$ and then is reduced as transitivity conditions identify branching options as being infeasible.

- The potentially feasible set for each i, j pair that is not fixed is created. This is done by starting with the potentially feasible set from the parent node and eliminating infeasible options by using the branching rule for the current node and the transitivity conditions presented in Chapter 4. Then, for any i, j pair that is not fixed if $|\Upsilon_{ij}| < 4$ then cuts are added; moreover if $|\Upsilon_{ij}| = 0$ the node is pruned because it is infeasible.

5.2 Depth First Search

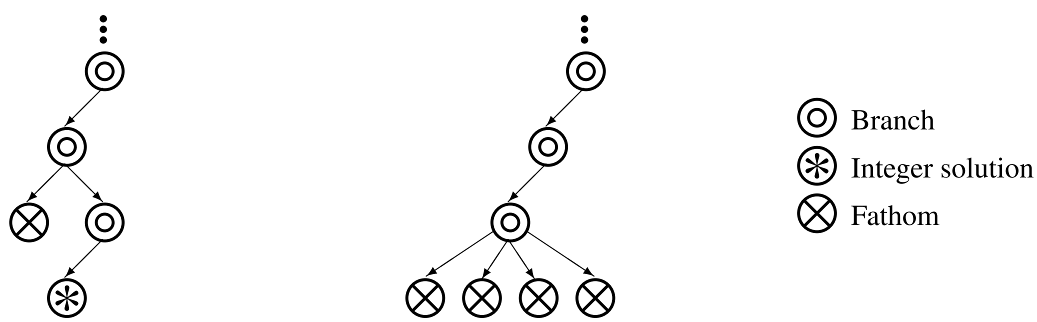
A branch-and-bound algorithm was implemented with the intention of finding feasible layouts. As a result depth-first search (DFS) was used. A dive refers to the sequence of nodes solved at successively lower levels. The specifics of the DFS method used are detailed below:

- The starting node for each dive was randomly selected from the 10% of nodes with the lowest objective value.

- For nodes that were found to be feasible for the SDP relaxation but did not produce a non-overlapping layout the dive continued by branching on the node that was most recently solved. The branching decisions presented earlier were used.

- If a node was fathomed for any reason other then that it was an integer solution the dive continued by selecting one of the fathomed nodes siblings.

- A dive ended when either an integer solution was found or the last considered node had no more unexamined siblings. Figure 5.2 illustrates the two cases that end a dive.



(a) A dive ending with Integer Solution (b) A dive ending with all children fathomed

Figure 5.2: Examples of how dives end

5.3 Test Cases

Xie and Sahinidis [32] created 6 test problems from a data set originally presented by Yang and Peters [33]. The first k facilities and their respective costs and dimensions for $k = 5, \ldots, 11$ were selected, and were denoted _T5,... _T11. [32] We have used these six problems. The cost matrix and the width and height of each facility is given in Appendix B.1.

Xie and Sahinidis placed a time limit of 6000 seconds on these six problems and the optimal solution was not reached for YA2_T10 and YA2_T11 within this time. In order to gain a better understanding of the performance of our model we used a time limit of 2 hours (7200 seconds).

In order to further examine the performance of our model problems ranging in size from 12 to 20 facilities were tested. The cost matrix for problems of size 12,14,15,16,17, 18 and 20 was constructed by Clausen and Perregaard [8] by deleting facilities from larger instances presented by Nugent et al [22]. The literature includes two problems with 16 facilities, we have used Nug16a. The cost matrix for Nug12 are listed in Table B.2; Nug14 and Nug15 are listed in Table B.3; and the remaining five instances are listed in Table B.4. For each table the smaller data sets consider the first k facilities. Facility dimensions do not exist since this data was initially developed for the QAP, therefore the dimensions of facilities from test instance YA2_T10 were applied as indicated in Table 5.1.

Instance	Nug12	Nug14	Nug15	Nug16	Nug17	Nug18	Nug20
Facility dimensions from YA2_T10 used	1-6,1-6	1-7,1-7	1-7,1-8	1-8,1-8	1-8-1,9	1-9,1-9	1-10,1-10

Table 5.1:

The selection of the starting dive node and the branching direction introduced randomness into the branch-and-bound algorithm. Due to this randomness we have completed three runs for tests limited at 2 hours. We have run longer tests (3 to 12 hours) in order to get a sense of the model's performance; however only a single run was completed for each of these test problems.

5.4 Results

5.4.1 Upper and Lower Bounds

This section will examine the upper and lower bound for the YA2_T test cases in 10 minute intervals. Each test case was run three times with a time limit of 2 hours. Test cases YA2_T6, _T7, _T8 and _T9 will be compared to the known optimal solution (plotted with a dotted line) as presented by Xie et al [32]. For test cases YA2_T10 and _T11 the optimal solution is not known, therefore the best known upper and lower bound are plotted. We will now examine three trends for either the upper and lower bound (Figure 5.3).

1. The bound remains constant throughout the 2 hours. In Figure 5.3(a) both the upper and lower bound are constant throughout the 2 hours.

2. The bound converges towards but does not achieve the known optimal solution nor improve on the best known bound. In Figure 5.3(b) both the upper and lower bounds converge; however neither bound achieves the known optimal solution.

3. The bound either achieves the known optimal solution or improves on the best known bound. In Figure 5.3(c) the upper and lower bounds converge with the upper bound achieving the known optimal solution prior to the 2 hour time limit. In Figure 5.3(d) the upper bounds improves on the best known upper bound.

Combinations of the aforementioned results are possible, consider Figure 5.3(e) in which the lower bound remains constant however the upper bound converges. Table 5.2 lists the trend for each of the 18 test instances; the terms 'constant', 'converges' and 'achieves/improves' are used to refer to trends 1,2 and 3 respectively. All 18 plots all shown in Appendix B.2.

	_R1		_R2		_R3	
	Upper	Lower	Upper	Lower	Upper	Lower
YA2_T6	**Achieves**	Converges	Converges	Converges	**Achieves**	Converges
YA2_T7	Converges	Constant	Converges	Constant	Converges	Constant
YA2_T8	Converges	Converges	Converges	Constant	Converges	Constant
YA2_T9	Converges	Constant	Constant	Constant	Converges	Constant
YA2_T10	Converges	Constant	Converges	Constant	Converges	Converges
YA2_T11	**Improves**	Constant	Converges	Constant	Converges	Constant

Table 5.2: Upper and lower bound trends

Observations

- In the 18 test instances the lower bound never reached the known optimal solution or best known lower bound. In addition, in 13 of the instances the lower bound remained constant. By using depth first search and since each node has four potential children it is difficult to advance the lower level of the tree.

- In one instance, YA2_T11_R1, the upper bound improves on the current best known upper bound. Three layouts with improved objective values were found. The most improved layout is presented in Figure 5.4 with the other two layouts in Appendix B.8. It is important to note that the weak lower bound results in a larger percentage gap then the gap found by the branch-and-bound method proposed by Xie et al [32].

5.4.2 Node Type

Table 5.3 lists the number of nodes processed in 2 hours for test cases YA2_T6_R1 to YA2_T11_R3. The number of nodes are classified under the five following categories:

1. Fathomed after solving - Integer Solution: Nodes that produce a feasible, non-overlapping layout.

2. Fathomed after solving - Other: Nodes that are either infeasible or the objective value is greater then the current upper bound.

3. Fathomed before solving - Infeasible by Transitivity: Nodes that are identified as being infeasible by the transitivity conditions presented in Chapter 4.

4. Fathomed before solving - Other: Nodes that are prunded because either the parent node or the lower guarentee is greater then the current upper bound.

5. Branched on: Nodes in which the relaxation is feasible with an objective value less then the current upper bound, however the layout has facilities that overlap.

Pruning nodes prior to solving them is favourable as not only is the branch-and-bound tree reduced but the computational time required to solve the node is saved.

	Fathomed After Solving		Fathomed Before Solving		
Test Run	Integer Solution	Other	Infeasible by transitivity	Other	Branched on
YA2_T6_R1	395	217	41	86	412
YA2_T6_R2	368	132	56	119	472
YA2_T6_R3	377	147	66	126	446
YA2_T7_R1	147	72	55	48	486
YA2_T7_R2	120	76	55	43	479
YA2_T7_R3	125	68	72	72	512
YA2_T8_R1	21	61	21	19	312
YA2_T8_R2	22	59	20	16	313
YA2_T8_R3	55	39	34	42	279
YA2_T9_R1	23	14	24	24	206
YA2_T9_R2	19	21	18	8	234
YA2_T9_R3	12	14	21	18	220
YA2_T10_R1	12	7	15	8	153
YA2_T10_R2	19	9	12	8	158
YA2_T10_R3	10	11	23	19	180
YA2_T11_R1	4	6	9	12	130
YA2_T11_R2	15	3	16	14	133
YA2_T11_R3	7	5	14	22	131

Table 5.3: Classification of nodes processed in 2 hours for test cases YA2_T6_R1 to YA2_T11_R3

Observations

- Less nodes are processed as the size of the problem increases. This results from the increased time required to solve a single node.
- In every instance nodes are fathomed prior to solving them as a result of transitivity conditions.

- As the number of unfixed i, j pairs decreases the time to solve the node also decreases. The number of unfixed i, j pairs is plotted against the time to solve a node in Figure 5.5.

5.4.3 Results: Percentage Gap

Figure 5.6 shows the average percentage gap for test instances YA2_T6 to YA2_T11. The percentage gap was calculated at 10 minute intervals over 2 hours. The upper and lower bounds and the following formula was used:

$$\text{Percentage Gap} = \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}} \times 100$$

Since both an upper and lower bound are required to satisfy this definition the gap could not be calculated until an upper bound was found. The first percentage gap for test instances of size 9 was calculated at the 30 minute interval and for instances of size 10 and 11 it took 40 minutes until an upper bound was found.

The following observations can be noted:

- For instances of size 6, 9, 10 and 11, the percentage gap increases as the size of the problem increases. This is the trend that one would expect.

- For instances 7 and 8 the average percentage is relatively similar. To further examine this similarity the percentage gap for test instance YA2_T7 and YA2_T8 are shown in Figure 5.7(a) and 5.7(b) respectively. In each graph the three test runs are shown separately. Although the reason for the similarity is not known, the variability in the percentage gap over instances YA2_T8_R1,_R2 and _R3 suggest that additional runs are required to further gage the trend in the percentage gap.

5.4.4 Optimality

Within 2 hours the test instances YA2_T5 could be solved to optimality. Figure 5.3(a) shows the trend in the upper and lower bound in 2 minute intervals for run 2. Appendix B.2 shows all 3 runs. Table 5.5 lists the number of nodes processed during each run in terms of the classification introduced in Section 5.4.2.

Within 4 hours the test instance YA2_T6 could be solved to optimality; however optimality could not be reached within 12 hours for test instance YA2_T7. Figure 5.5(a) and 5.5(b), respectively, shows the trend in the upper and lower bound in 10 minute intervals.

5.4.5 Larger Instances

The model was also tested on larger problems, specifically the Nug test instances ranging from 12 to 20 facilities. Table 5.7 lists the percentage gap for instances of 12 to 15 facilities. Percentage gap was calculated as in Section 5.4.3. The gap could not be found for larger instances since no upper bound was found. The categorization of nodes is listed in Table 5.8.

(a) Test instance YA2_T5, run until Optimality

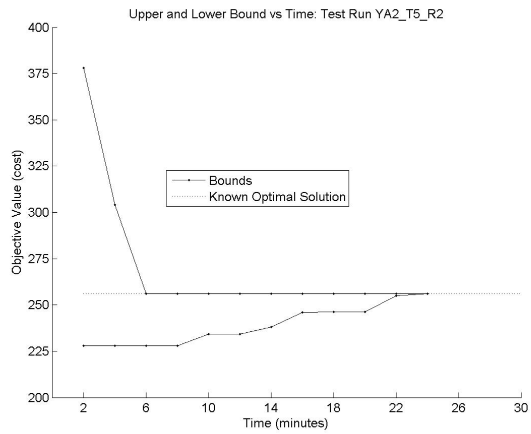
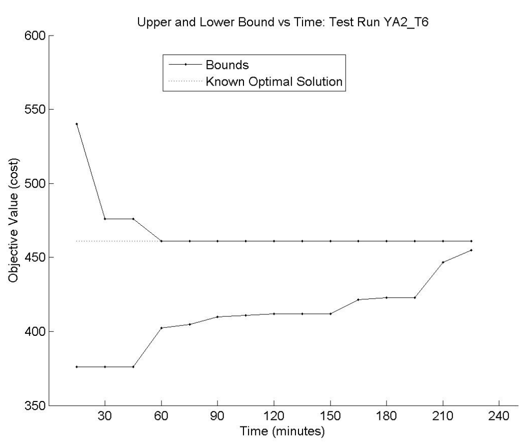


Table 5.4: Upper and lower bounds for test instance YA2_T5

	Fathomed After Solving		Fathomed Before Solving		
Test Run	Integer Solution	Other	Infeasible by transitivity	Other	Branched on
YA2_T5_R1	134	95	16	84	105
YA2_T5_R2	125	93	14	67	97
YA2_T5_R3	135	116	29	116	128

Table 5.5: Classification of nodes processed for test case YA2_T5

(a) Test instance YA2_T6, run until Optimality



(b) Test instance YA2_T7, run for 12 hours

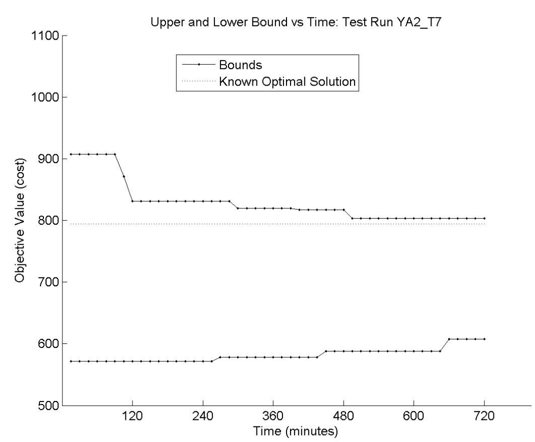


Table 5.6: Upper and lower bounds for test instances YA2_T6 and YA2_T7

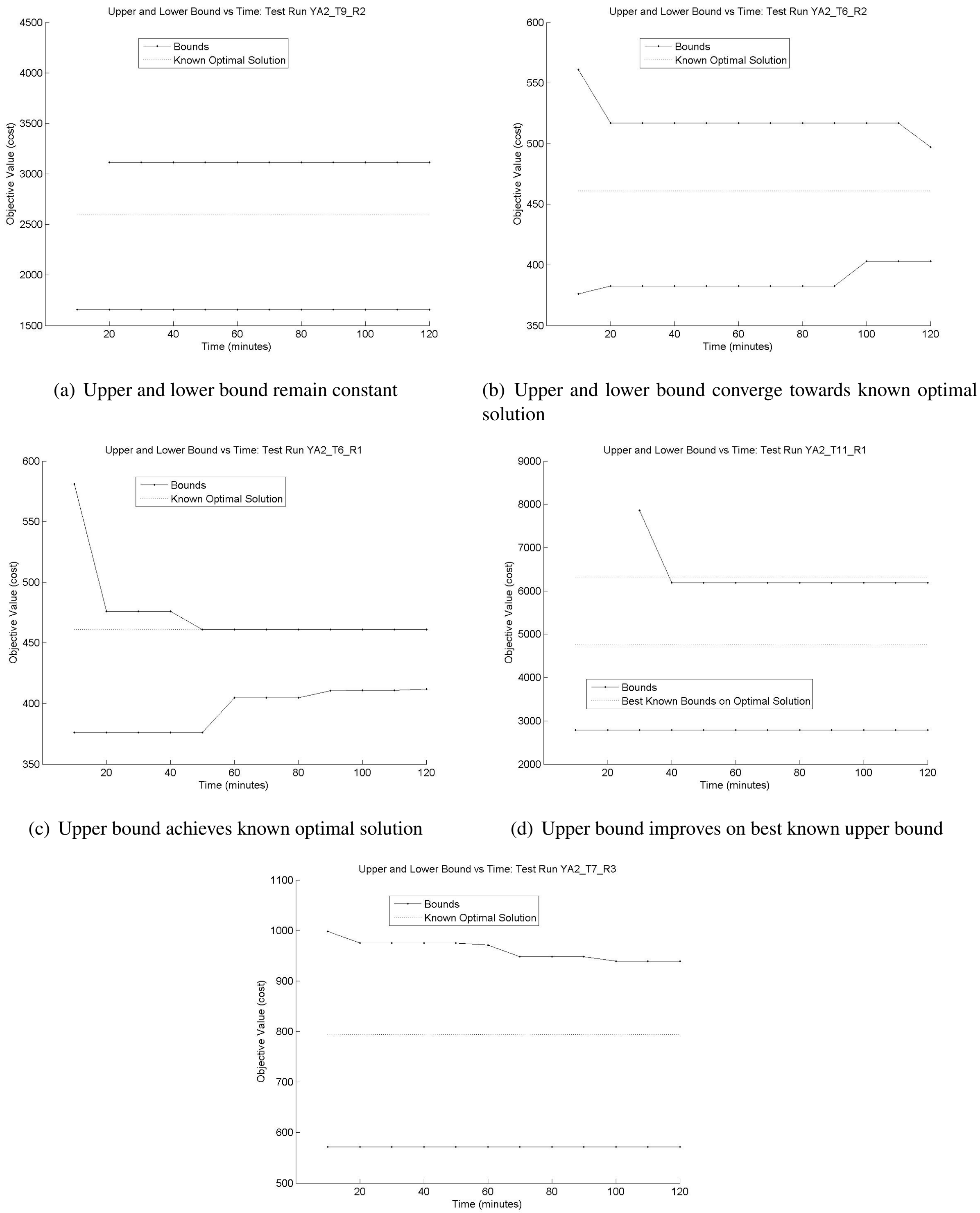
A gap was not able to be calculated for larger instances since an integer solution was not found within the 3 hour time limit and as a result no upper bound could be calculated. The full classification of the type of node processed is shown in Table 5.8. This table uses the same five node categories as in Section 5.4.2

Test Run	Nug12	Nug14	Nug15
Gap after 3 hours	58.04	72.48	73.58

Table 5.7: Percentage Gap after 3 hours for test instances Nug12 to Nug16

	Fathomed After Solving		Fathomed Before Solving		
Test Run	Integer Solution	Other	Infeasible by transitivity	Other	Branched on
Nug12	11	4	12	13	143
Nug14	7	1	7	2	71
Nug15	6	0	2	4	49
Nug16	0	0	7	0	44
Nug17	0	0	0	0	28
Nug18	0	0	1	0	24
Nug20	0	0	0	0	16

Table 5.8: Classification of nodes processed in 3 hours for test instances Nug12 to Nug20



(a) Upper and lower bound remain constant

(b) Upper and lower bound converge towards known optimal solution

(c) Upper bound achieves known optimal solution

(d) Upper bound improves on best known upper bound

(e) Lower bound converge towards known optimal solution

Figure 5.3: Upper and lower bound in comparison to known optimal solutions for a representive sample of test cases, run for 2 hours

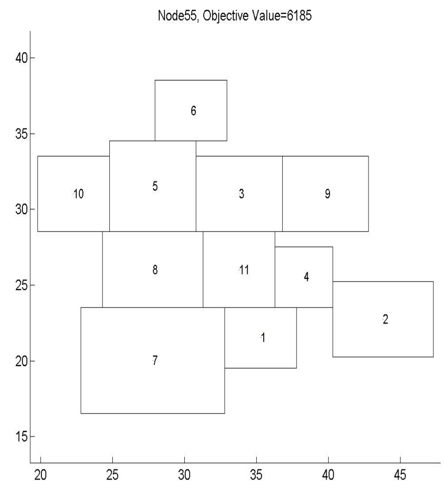


Figure 5.4: Layout for test case YA2_T11 with improved upper bound

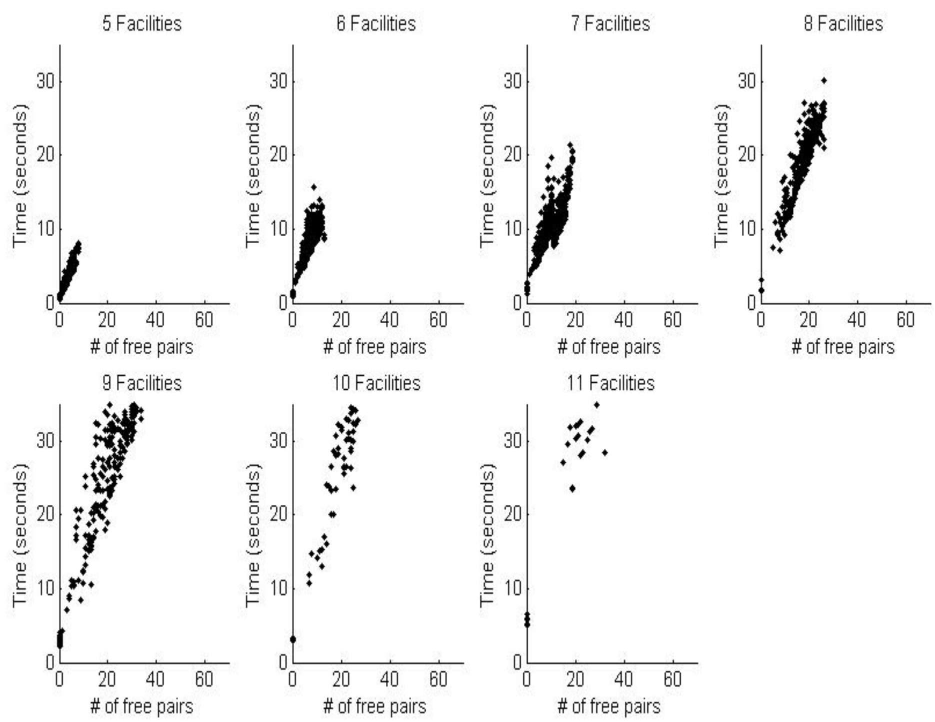


Figure 5.5: Number of free i, j pairs vs time to solve one node, for test cases YA2_T5 to YA2_T11

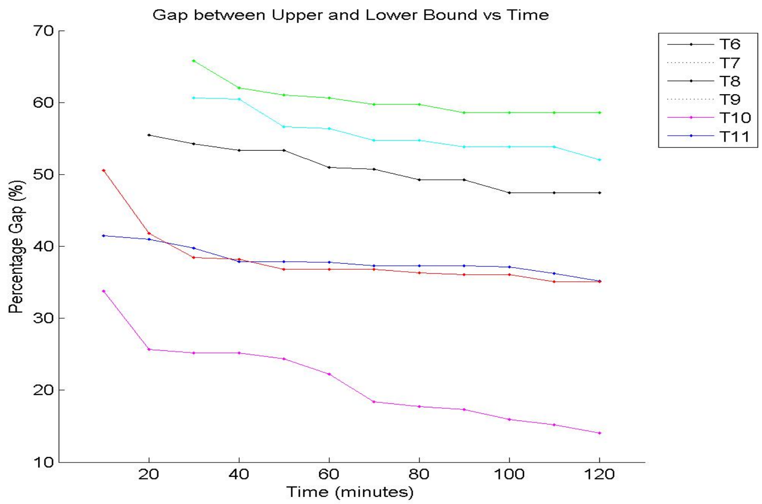


Figure 5.6: Average Percentage Gap over 3 runs for data sets YA2T_6 to YA2T_11

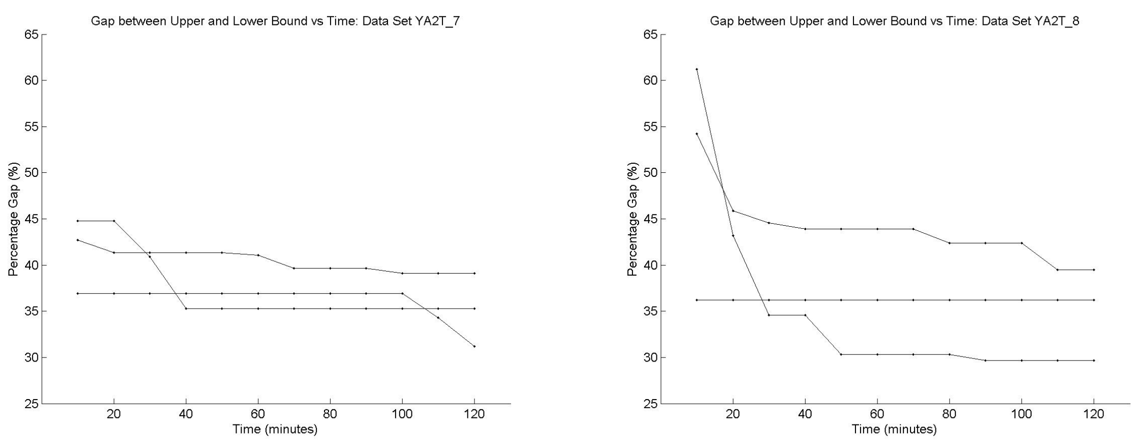


Figure 5.7: Percentage gap for test instance YA2_T7 and YA2_T8

Chapter 6

Concluding Remarks

The main contribution of this thesis is the theoretical development of a MISDP model. Three layout characteristics (separation axis, separation order and non-separation order) guided the discussion. They were used to defined conditions that ensured a feasible layout; then were used to define quaternary variables. These variables became the foundation of the MISDP model introduced.

A distinction was made between the constraints that SDP optimization software can solve and those that must be relaxed. The latter were called exactness constraints because they are required for an exact formulation; however they are not included in the SDP relaxation. Three possible exactness constraints were shown to be equivalent. Then given an exactness constraint the remaining constraints, that form the SDP relaxation, were shown to imply that the three layout characteristics are satisfied.

Transitivity conditions were examined; these are used to identify infeasible layouts during a branch-and-bound algorithm. By identifying these layouts additional cuts can be added or nodes can be pruned prior to solving them.

Finally, preliminary computational results were presented. Results regarding upper and lower bounds, gaps and a classification of the nodes during branch-and-bound were reported.

6.1 Future Work

This thesis described a MISDP formulation that uses quaternary variables. Now with a theoretical framework in place future work can focus on simplifing and/or tightening the SDP relaxation to attempt to make the theory competitive with other approaches in terms of either computational time or optimality gap. Branching in such a way that only 2 children are produced may help improve the lower bound. Two potential branching decisions are:

$$\begin{bmatrix} \beta_{ij} + \sigma\mu b_{ij} & = \sigma + \mu \\ -\beta_{ij} - \sigma\mu b_{ij} & = \sigma + \mu \end{bmatrix} \text{ or } \begin{bmatrix} -\beta_{ij} + \sigma\mu b_{ij} & = \sigma - \mu \\ \beta_{ij} - \sigma\mu b_{ij} & = \sigma - \mu \end{bmatrix}$$

These branching decisions are motivated by the graph produced by plotting b_{ij} vs β_{ij} for the four discrete values (Figure 6.1), where the aforementioned equations define the lines. The first set of equations is branching on the two solid lines while the second is branching on the dashed lines.

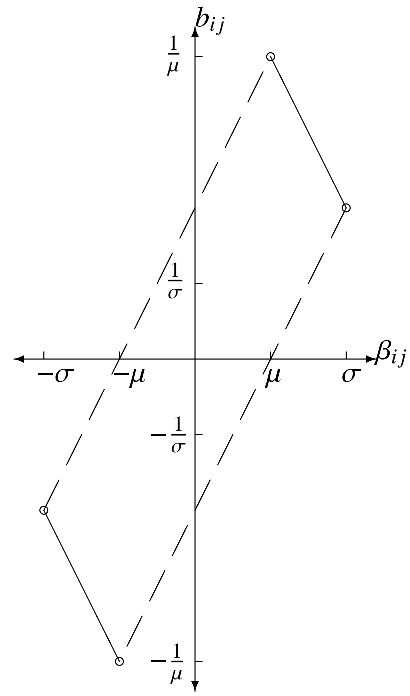


Figure 6.1: Convex hull of feasible values of β_{ij} and b_{ij}

In addition the convex hull of these points can be defined by the intersection of the following 4 halfspaces and may be able to tighten the relaxation:

$$\beta_{ij} + \sigma\mu b_{ij} \leq \sigma + \mu$$
$$-\beta_{ij} - \sigma\mu b_{ij} \leq \sigma + \mu$$
$$-\beta_{ij} + \sigma\mu b_{ij} \leq \sigma - \mu$$
$$\beta_{ij} - \sigma\mu b_{ij} \leq \sigma - \mu$$

Less computational time is required to solve each node if the SDP relaxation can be simplified. This may be able to be done by reformulating the 2×2 positive semidefinite constraints (2.5)-(2.7) as linear inequalities using big M values.

Finally the effect of varying the parameters σ and μ should be examined.

Appendices

Appendix A

Transitivity

A.1 Transitivity Charts

	A	B	R	L
A	✓	⋫	✓	✓
B	⋫	⋫	⋫	⋫
R	⋫	⋫	∅	∅
L	⋫	⋫	∅	∅

	A	B	R	L
A	⋫	⋫	⋫	⋫
B	⋫	✓	⋫	⋫
R	⋫	✓	∅	∅
L	⋫	✓	∅	∅

	A	B	R	L
A	b≥a	d≥a	✓	⋫
B	⋫	c≥a	⋫	⋫
R	⋫	✓	✓	⋫
L	⋫	✓	✓	⋫

	A	B	R	L
A	b≥a	d≥a	✓	✓
B	⋫	c≥a	⋫	⋫
R	⋫	⋫	⋫	⋫
L	⋫	✓	✓	✓

	A	B	R	L
A	✓	⋫	⋫	⋫
B	⋫	⋫	⋫	⋫
R	✓	⋫	∅	∅
L	✓	⋫	∅	∅

	A	B	R	L
A	⋫	⋫	⋫	⋫
B	⋫	✓	✓	✓
R	⋫	⋫	∅	∅
L	⋫	⋫	∅	∅

	A	B	R	L
A	c≥a	⋫	⋫	⋫
B	d≥a	b≥a	✓	⋫
R	✓	⋫	✓	⋫
L	✓	⋫	✓	⋫

	A	B	R	L
A	c≥a	⋫	⋫	⋫
B	d≥a	b≥a	✓	✓
R	⋫	⋫	⋫	⋫
L	✓	⋫	✓	✓

	A	B	R	L
A	✓	⋫	✓	✓
B	⋫	⋫	⋫	⋫
R	⋫	⋫	∅	∅
L	⋫	⋫	∅	∅

	A	B	R	L
A	⋫	⋫	⋫	⋫
B	⋫	✓	⋫	⋫
R	⋫	✓	∅	∅
L	⋫	✓	∅	∅

	A	B	R	L
A	b≥a	d≥a	✓	⋫
B	⋫	c≥a	⋫	⋫
R	⋫	✓	✓	⋫
L	⋫	⋫	⋫	⋫

	A	B	R	L
A	b≥a	d≥a	✓	✓
B	⋫	c≥a	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	✓	✓	✓

	A	B	R	L
A	✓	⋫	✓	✓
B	⋫	⋫	⋫	⋫
R	⋫	⋫	∅	∅
L	⋫	⋫	∅	∅

	A	B	R	L
A	⋫	⋫	⋫	⋫
B	⋫	✓	⋫	⋫
R	⋫	✓	∅	∅
L	⋫	✓	∅	∅

	A	B	R	L
A	b≥a	d≥a	✓	✓
B	⋫	c≥a	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	✓	✓	✓

	A	B	R	L
A	b≥a	d≥a	⋫	✓
B	⋫	c≥a	⋫	⋫
R	⋫	⋫	⋫	⋫
L	⋫	✓	⋫	✓

	A	B	R	L
A	✓	✓	✓	✓
B	⋫	⋫	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	✓	✓	✓

	A	B	R	L
A	⋫	⋫	⋫	⋫
B	⋫	✓	⋫	⋫
R	⋫	✓	⋫	⋫
L	⋫	✓	⋫	⋫

	A	B	R	L
A	⋫	✓	✓	✓
B	⋫	✓	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	⋫	⋫	⋫

	A	B	R	L
A	⋫	✓	✓	✓
B	⋫	✓	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	✓	✓	✓

	A	B	R	L
A	✓	✓	✓	✓
B	⋫	⋫	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	✓	✓	✓

	A	B	R	L
A	⋫	⋫	⋫	⋫
B	⋫	✓	⋫	⋫
R	⋫	✓	⋫	⋫
L	⋫	✓	⋫	⋫

	A	B	R	L
A	⋫	✓	✓	✓
B	⋫	✓	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	⋫	⋫	⋫

	A	B	R	L
A	⋫	✓	✓	✓
B	⋫	✓	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	✓	✓	✓

	A	B	R	L
A	✓	⋫	✓	✓
B	⋫	⋫	⋫	⋫
R	✓	⋫	✓	✓
L	✓	⋫	⋫	✓

	A	B	R	L
A	⋫	⋫	⋫	⋫
B	⋫	✓	✓	✓
R	⋫	✓	✓	✓
L	⋫	✓	⋫	✓

	A	B	R	L
A	✓	⋫	✓	⋫
B	⋫	✓	✓	⋫
R	✓	✓	✓	⋫
L	⋫	⋫	⋫	⋫

	A	B	R	L
A	✓	⋫	✓	✓
B	⋫	✓	✓	✓
R	✓	✓	✓	✓
L	✓	✓	⋫	✓

	A	B	R	L
A	✓	⋫	⋫	⋫
B	⋫	⋫	⋫	⋫
R	⋫	⋫	⋫	⋫
L	⋫	⋫	⋫	⋫

	A	B	R	L
A	✓	✓	✓	✓
B	⋫	✓	⋫	⋫
R	⋫	✓	✓	✓
L	⋫	✓	✓	✓

	A	B	R	L
A	✓	⋫	✓	✓
B	⋫	⋫	⋫	⋫
R	⋫	⋫	$d \geq b$	$d \geq b$
L	⋫	⋫	$d \geq b$	$d \geq b$

	A	B	R	L
A	✓	⋫	✓	✓
B	⋫	⋫	⋫	⋫
R	⋫	⋫	$d \geq b$	$d \geq b$
L	⋫	⋫	$d \geq b$	$d \geq b$

⋫=3-way infeasible
∅= 4-way infeasible
✓ = Feasible

Appendix B

Computational Results

B.1 Test Cases

(a) Cost Matrix

c_{ij}	1	2	3	4	5	6	7	8	9	10	11	12
1	-	18	6	12	2	20	18	10	38	20	26	26
2	-	-	0	0	0	0	0	0	0	0	18	0
3	-	-	-	0	4	4	14	30	16	36	32	38
4	-	-	-	-	8	0	0	0	0	0	0	0
5	-	-	-	-	-	10	2	34	30	6	14	24
6	-	-	-	-	-	-	0	0	0	14	0	0
7	-	-	-	-	-	-	-	36	12	20	4	28
8	-	-	-	-	-	-	-	-	0	0	6	0
9	-	-	-	-	-	-	-	-	-	8	22	12
10	-	-	-	-	-	-	-	-	-	-	0	0
11	-	-	-	-	-	-	-	-	-	-	-	6
12	-	-	-	-	-	-	-	-	-	-	-	-

(b) Facility Dimensions

Facility	Width	Height
1	5	4
2	7	5
3	6	5
4	4	4
5	6	6
6	5	4
7	10	7
8	7	5
9	6	5
10	5	5
11	5	5
12	6	4

Table B.1: The YA2_T data: from Yang and Peters (1998) [33]

c_{ij}	1	2	3	4	5	6	7	8	9	10	11	12
1	-	5	2	4	1	0	0	6	2	1	1	1
2	-	-	3	0	2	2	2	0	4	5	0	0
3	-	-	-	0	0	0	0	5	5	2	2	2
4	-	-	-	-	5	2	2	10	0	0	5	5
5	-	-	-	-	-	10	0	0	0	5	1	1
6	-	-	-	-	-	-	5	1	1	5	4	0
7	-	-	-	-	-	-	-	10	5	2	3	3
8	-	-	-	-	-	-	-	-	0	0	5	0
9	-	-	-	-	-	-	-	-	-	0	10	10
10	-	-	-	-	-	-	-	-	-	-	5	0
11	-	-	-	-	-	-	-	-	-	-	-	2
12	-	-	-	-	-	-	-	-	-	-	-	-

Table B.2: Cost Matrix for Nug12

c_{ij}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-	10	0	5	1	0	1	2	2	2	2	0	4	0	0
2	-	-	1	3	2	2	2	3	2	0	2	0	10	5	0
3	-	-	-	10	2	0	2	5	4	5	2	2	5	5	5
4	-	-	-	-	1	1	5	0	0	2	1	0	2	5	0
5	-	-	-	-	-	3	5	5	5	1	0	3	0	5	5
6	-	-	-	-	-	-	2	2	1	5	0	0	2	5	10
7	-	-	-	-	-	-	-	6	0	1	5	5	5	1	0
8	-	-	-	-	-	-	-	-	5	2	10	0	5	0	0
9	-	-	-	-	-	-	-	-	-	0	10	5	10	0	2
10	-	-	-	-	-	-	-	-	-	-	0	4	0	0	5
11	-	-	-	-	-	-	-	-	-	-	-	5	0	5	0
12	-	-	-	-	-	-	-	-	-	-	-	-	3	3	0
13	-	-	-	-	-	-	-	-	-	-	-	-	-	10	2
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.3: Cost Matrix for Nug14 and Nug15

c_{ij}	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	-	0	5	0	5	2	10	3	1	5	5	5	0	0	5	4	4	0	0	1
2	-	-	3	10	5	1	5	1	2	4	2	5	0	10	10	3	0	5	10	5
3	-	-	-	2	0	5	2	4	4	5	0	0	0	5	1	0	0	5	0	0
4	-	-	-	-	1	0	5	2	1	0	10	2	2	0	2	1	5	2	5	5
5	-	-	-	-	-	5	6	5	2	5	2	0	5	1	1	1	5	2	5	1
6	-	-	-	-	-	-	5	2	1	6	0	0	10	0	2	0	1	0	1	5
7	-	-	-	-	-	-	-	0	0	0	5	10	2	2	5	1	2	1	0	10
8	-	-	-	-	-	-	-	-	1	1	10	10	2	0	10	2	5	2	2	10
9	-	-	-	-	-	-	-	-	-	2	0	3	5	5	0	5	0	0	0	2
10	-	-	-	-	-	-	-	-	-	-	5	5	0	5	1	0	0	5	5	2
11	-	-	-	-	-	-	-	-	-	-	-	5	2	5	1	10	0	2	2	5
12	-	-	-	-	-	-	-	-	-	-	-	-	2	10	5	0	1	1	2	5
13	-	-	-	-	-	-	-	-	-	-	-	-	-	2	2	1	0	0	0	5
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	5	1	5	5	0
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	0	5	10	10
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	2	0
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	2	0
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	6
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table B.4: Cost Matrix for Nug16,Nug17,Nug18 and Nug20

B.2 Results

B.2.1 Test Instance YA2_T5

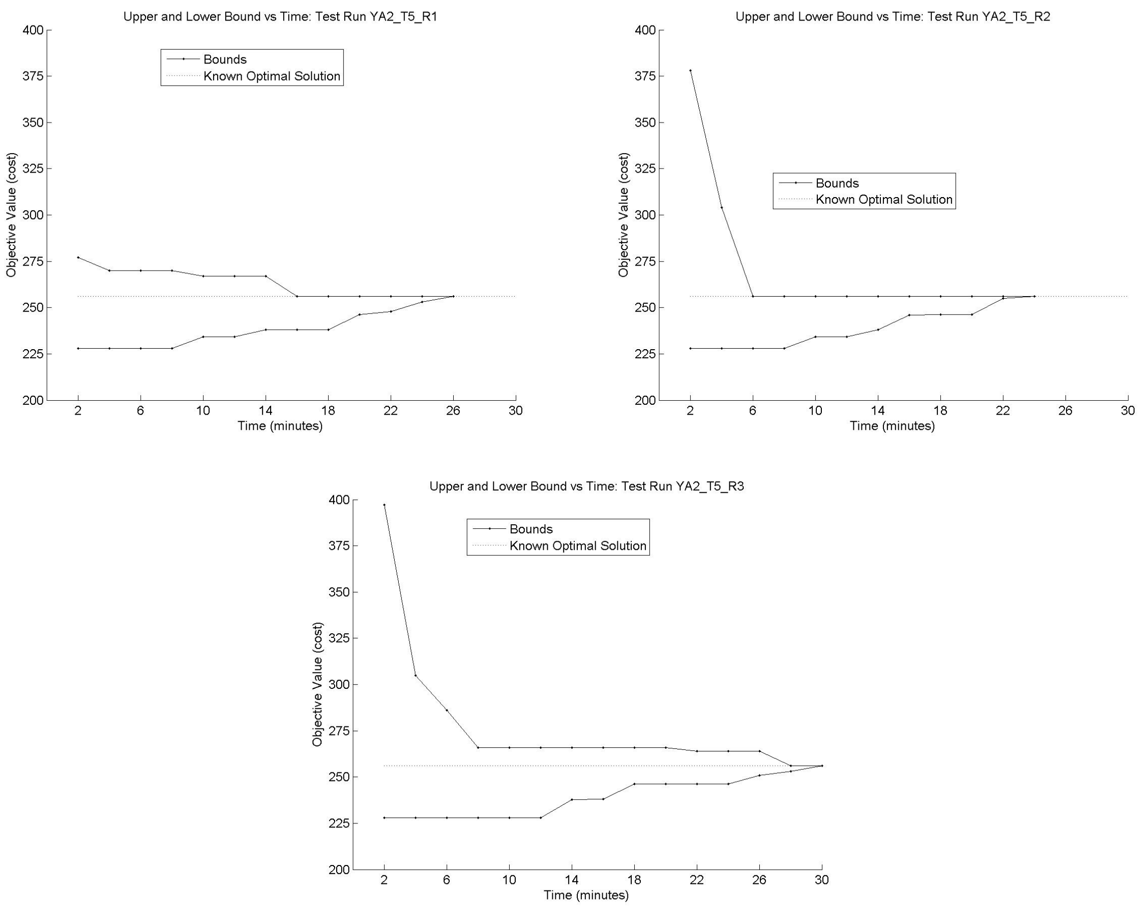


Figure B.1: Upper and lower bounds for test case YA2_T5, run for 2 hours

B.2.2 Test Instance YA2_T6

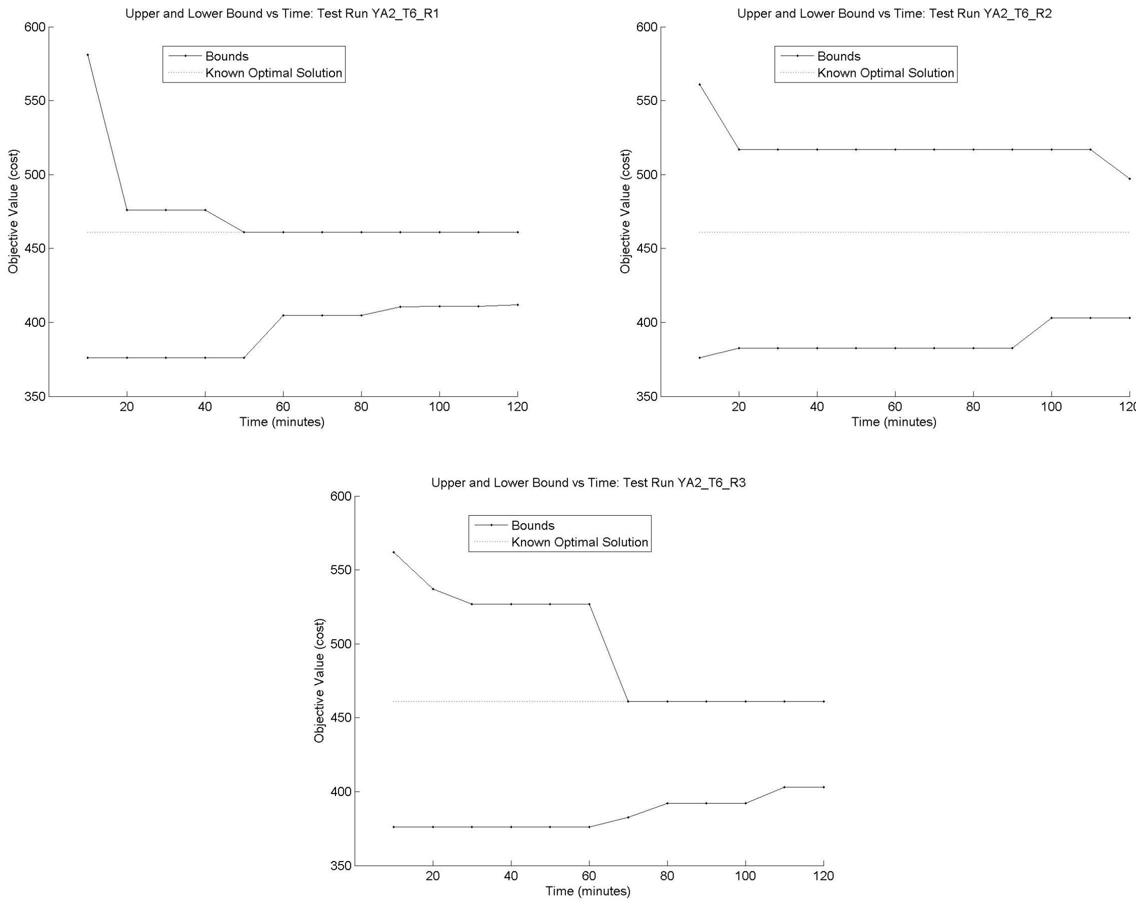


Figure B.2: Upper and lower bounds for test case YA2_T6, run for 2 hours

B.2.3 Test Instance YA2_T7

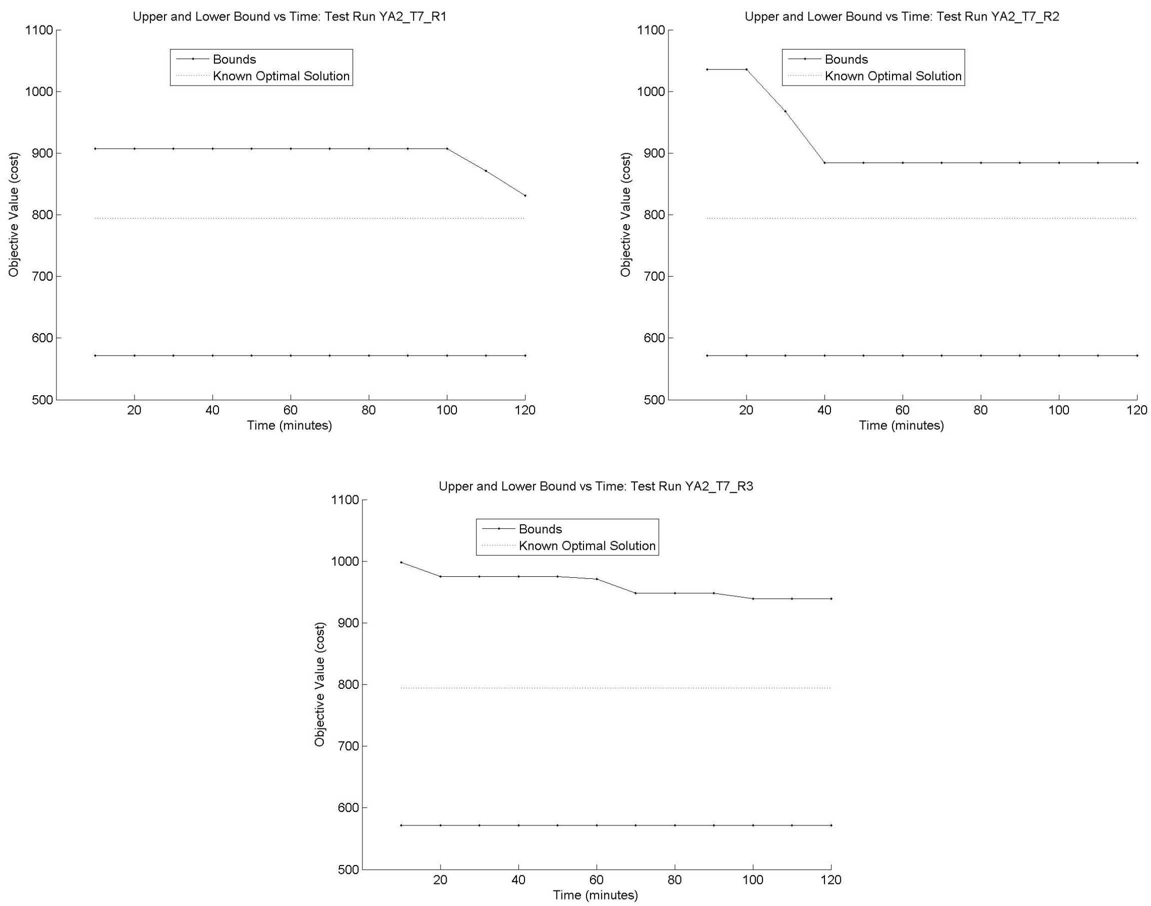


Figure B.3: Upper and lower bounds for test case YA2_T7, run for 2 hours

B.2.4 Test Instance YA2_T8

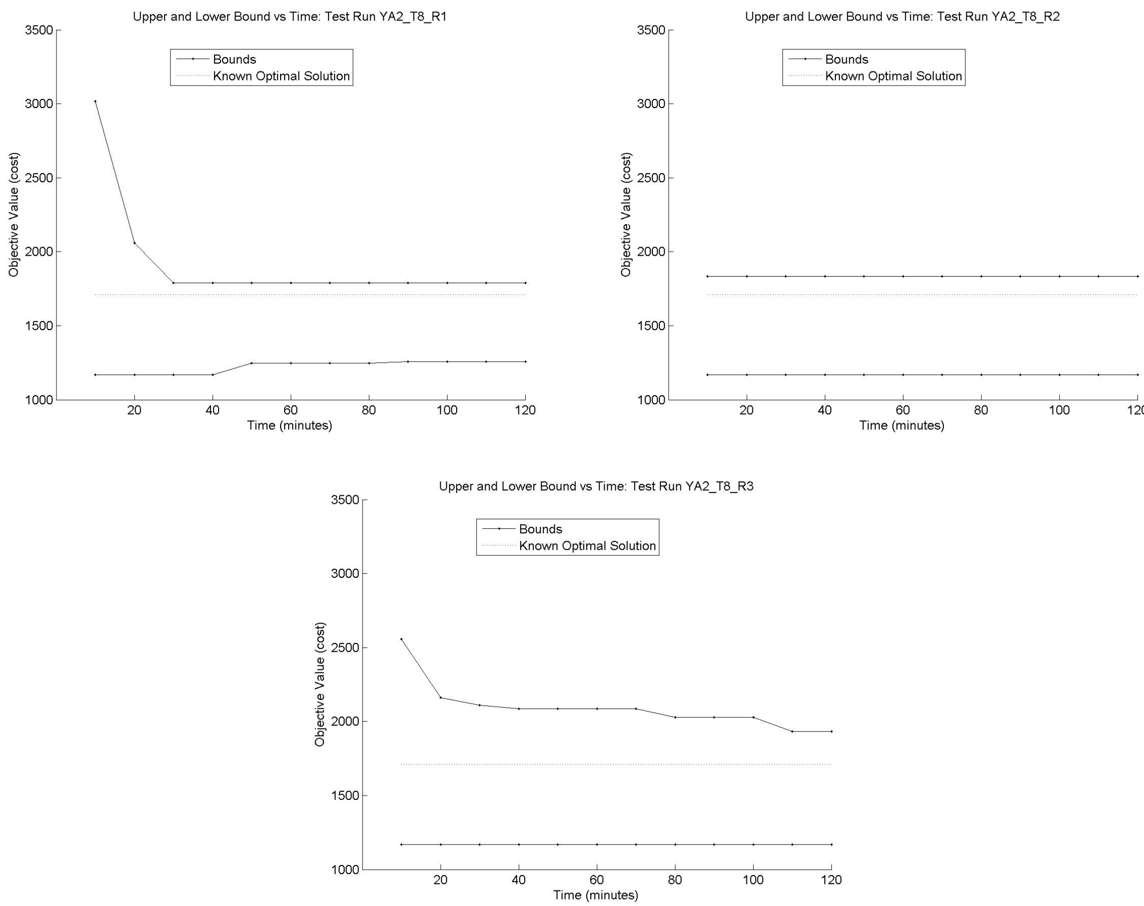


Figure B.4: Upper and lower bounds for test case YA2_T8, run for 2 hours

B.2.5 Test Instance YA2_T9

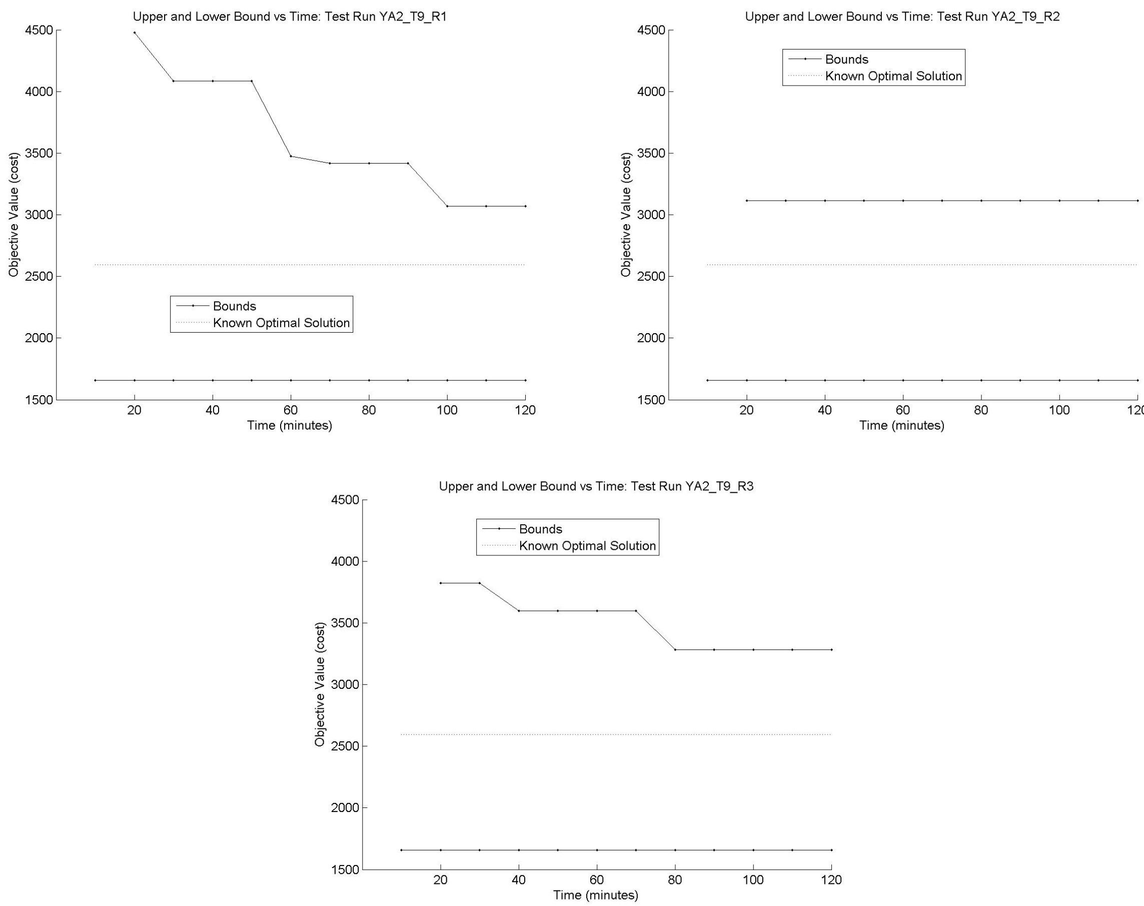


Figure B.5: Upper and lower bounds for test case YA2_T9, run for 2 hours

B.2.6 Test Instance YA2_T10

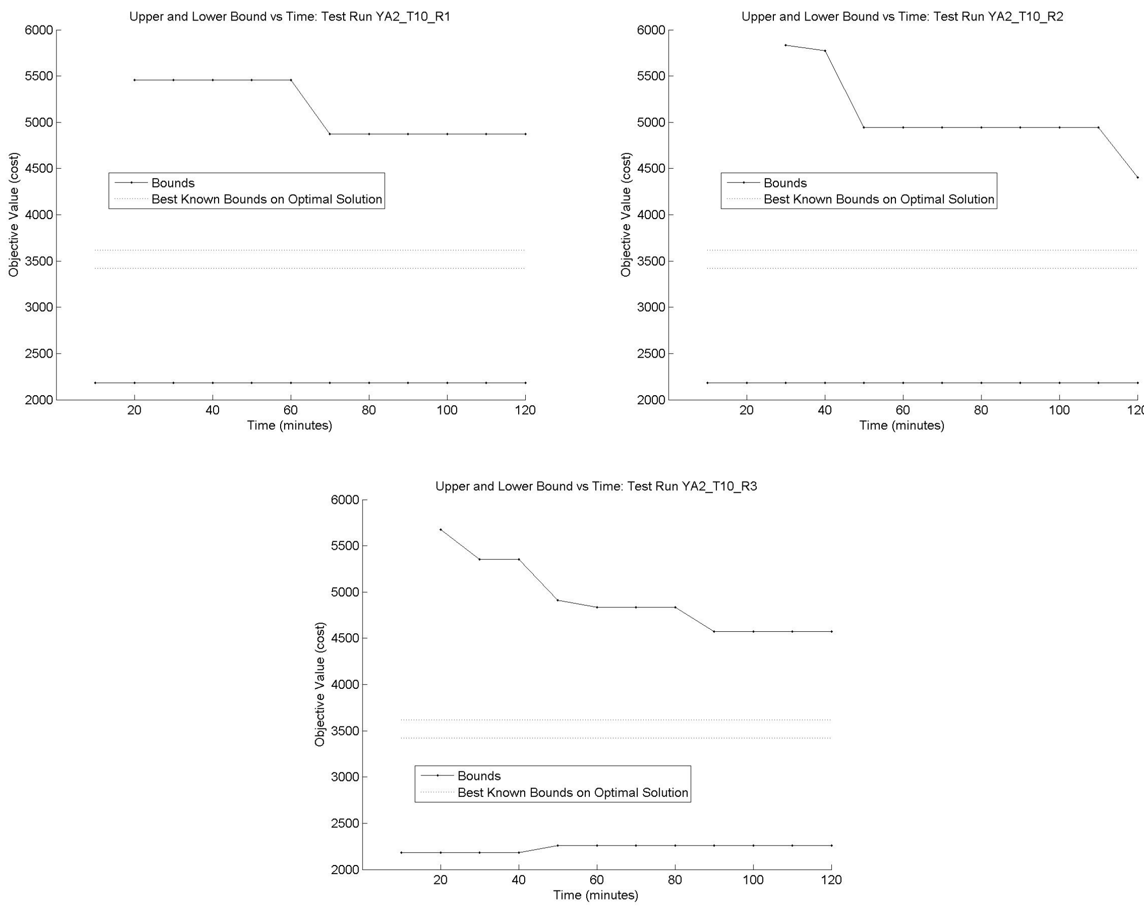
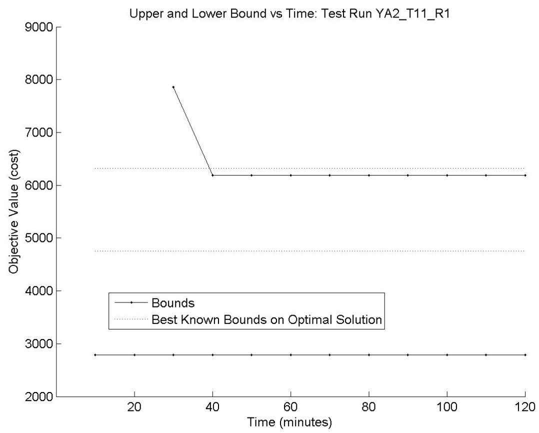
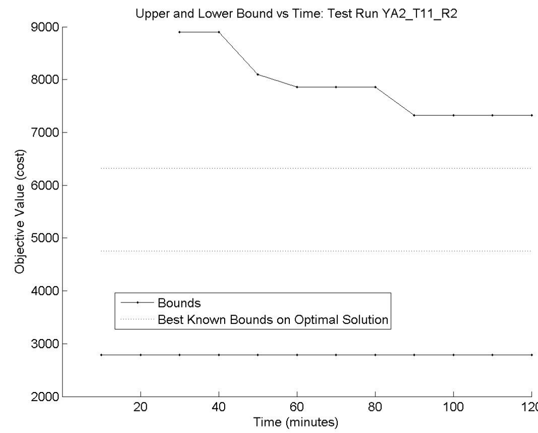


Figure B.6: Upper and lower bounds for test case YA2_T10, run for 2 hours

B.2.7 Test Instance YA2_T11





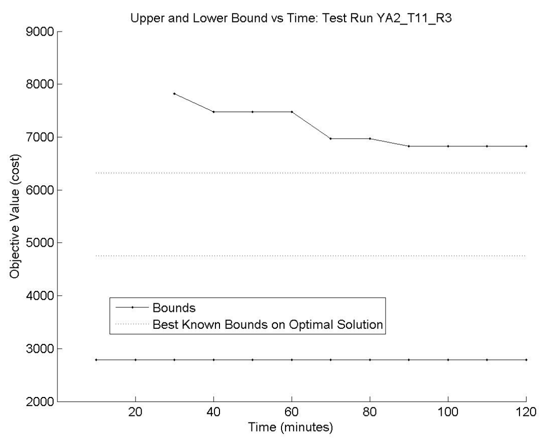


Figure B.7: Upper and lower bounds for test case YA2_T11, run for 2 hours

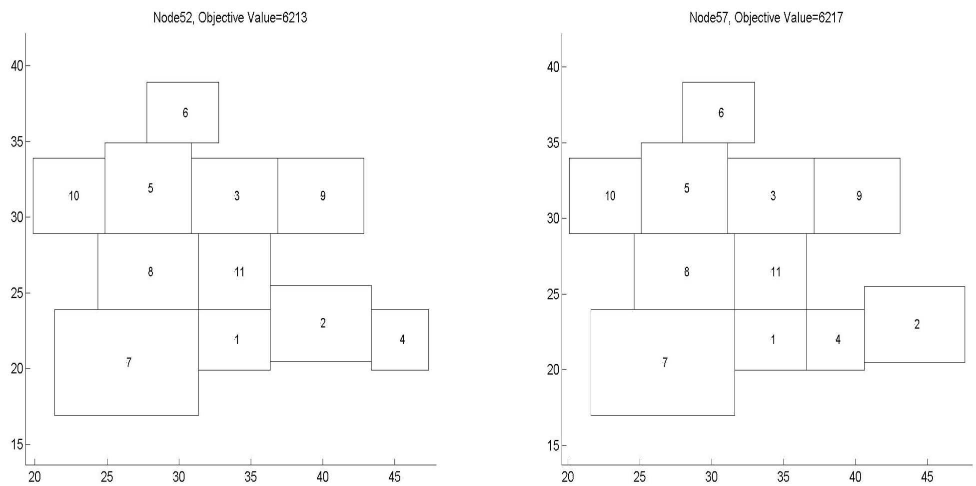


Figure B.8: Layouts for test case YA2_T11 with improved upper bound

Bibliography

[1] A. Amaral. A new lower bound for the single row facility layout problem. *Discrete Applied Mathematics*, 157:183–190, 2009.

[2] M.F. Anjos, A. Kennings, and A. Vannelli. A semidefinite optimization approach for the single-row layout problem with unequal dimensions. *Discrete Optim.*, 2:113–122, 2005.

[3] M.F. Anjos and A. Vannelli. A new mathematical programming framework for facility layout design. *INFORMS Journal on Computing*, 18(1):111–118, 2006.

[4] M.S. Bazaraa. Computerized layout design: A branch and bound approach. *AIIE Transactions*, 7(4):432–437, 1975.

[5] I. Castillo and T. Westerlund. An ϵ-accurate model for optimal unequal-area block layout design. *Computers & Operations Research*, 32(3):429–447, 2005.

[6] R. Cheng, M. Gen, and T. Tosawa. Genetic algorithms for designing loop layout manufacturing systems. *Computers & Industrial Engineering*, 31(3-4):587–591, 1996.

[7] L. Chwif, M.R. Pereira Barretto, and L.A. Moscato. A solution to the facility layout problem using simulated annealing. *Computers in Industry*, 36(1-2):125–132, 1998.

[8] J. Clausen and M. Perregaard. Solving large quadratic assignment problems in parallel. *Computational Optimization and Applications*, 8(2):111–127, 1997.

[9] A. Drira, H. Pierreval, and S. Hajri-Gabouj. Facility layout problems: A survey. *Annual Reviews in Control*, 31:255–267, 2007.

[10] M.M.D. Hassan. Machine layout problem in modern manufacturing facilities. *International Journal of Production Research*, 32(11):2559–2584, 1994.

[11] S.S. Heragu and A. Kusiak. Efficient models for the facility layout problem. *European J. Oper. Res.*, 53:1–13, 1991.

[12] C.B. Kim, S.S. Kim, and L.F. Bobbie. Assignment problems in singlerow and double-row machine layouts during slow and peak periods. *Computers & Industrial Engineering*, 30(3):411–422, 1996.

[13] J.G. Kim and Y.D. Kim. Layout planning for facilities with fixed shapes and input and output points. *International Journal of Production Research*, 38(18):4635–4653, 2000.

[14] T.C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25(1):53–76, 1957.

[15] K.R. Kumar, G.C. Hadjinicola, and T.L. Lin. A heuristic procedure for the single-row facility layout problem. *Eurpoean Journal of Operational Research*, 87(1):65–73, 1995.

[16] Q. Liu and R.D. Meller. A sequence-pair representation and MIP model based heuristic for the facility layout problem with rectangular departments. *IIE Transactions*, 39:377–394, 2007.

[17] R.D. Meller, W. Chen, and H.D. Sherali. Applying the sequence-pair representation to optimal facility layout designs. *Operations Research Letters*, 35(5):651–659, 2007.

[18] R.D. Meller and K. Y. Gau. The facility layout problem: Recent and emerging trends and perspectives. *J. Manufacturing Systems*, 15(5):351–366, 1996.

[19] R.D. Meller, V. Narayanan, and P.H. Vance. Optimal facility layout design. *Operations Research Letters*, 23(3-5):117–127, 1999.

[20] B. Montreuil. A modelling framework for integrating layout design and flow network design. *Proceedings from the Material Handling Research Colloquium*, pages 43–58, 1990.

[21] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitannil. VLSI module placement based on rectangle-packing by sequence-pair. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(12):1518–1524, 1996.

[22] C.E. Nugent, T.E. Vollmann, and J. Ruml. An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research*, 16:150–173, 1968.

[23] H. Onodera, Y. Taniguchi, and K. Tamaru. Branch and bound placement for building block layout. *Proceedings of the 28th Conference on ACM/IEEE Design Automation*, pages 433–439, 1991.

[24] D.B. Özyurt and M.J. Realff. Geographic and process information for chemical plant layout problems. *AIChE Journal*, 45(10):2161–2174, 1999.

[25] L.G. Papageorgiou and G.E. Rotstein. Continuous-domain mathematical models for optimal process plant layout. *Industrial & Engineering Chemistry Research*, 37:3631–3639, 1998.

[26] C.N. Potts and J.D. Whitehead. Workload balancing and loop layout in the design of a flexible manufacturing system. *European Journal of Operational Research*, 129(2):326–336, 2001.

[27] H.D. Sherali, B.M.P. Fraticelli, and R.D. Meller. Enhanced model formulations for optimal facility layout. *Operations Research*, 51 (4):629–644, 2003.

[28] S.P. Singh and R.R.K.Sharma. A review of different approaches to the facility layout problem. *International Journal of Advanced Manufacturing Technology*, 30:425–433, 2006.

[29] P. Takouda, M.F. Anjos, and A. Vannelli. Global lower bounds for the VLSI macrocell floorplanning problem using semidefinite optimization. *Proceedings Fifth International Workshop on System-on-Chip for Real-Time Applications*, pages 275–280, 2005.

[30] J.A. Tompkins, J.A White, Y.A Bozer, E.H. Frazelle, J.M.A. Tanchoco, and J. Trevino. *Facilities Planning*. Wiley, New York, second edition, 1996.

[31] D.J. van Camp, M.W. Carter, and A. Vannelli. A nonlinear optimization approach for solving facility layout problems. *European Journal of Operational Research*, 57:174–189, 1991.

[32] W. Xie and N.V. Sahinidis. A branch and bound algorithm for the continuous facility layout problem. *Comput. Chem. Eng.*, 32:1016–1028, 2008.

[33] T. Yang and B. Peters. Flexible machine layout design for dynamic and uncertain production environments. *European Journal of Operational Research*, 108:49–64, 1998.

[34] T. Yang, B.A. Peters, and M. Tu. Layout design for flexible manufacturing systems considering single-loop directional flow patterns. *European Journal of Operational Research*, 164(2):440–455, 2005.