PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Metaheuristic Approach for the Facility Layout Problem in a Hospital
Auteur: Author:	Mehran Mehri
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Mehri, M. (2021). Metaheuristic Approach for the Facility Layout Problem in a Hospital [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10014/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/10014/
Directeurs de recherche: Advisors:	Jean-Marc Frayret, & Nadia Lahrichi
Programme: Program:	Maîtrise recherche en génie industriel

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Metaheuristic Approach for the Facility Layout Problem in a Hospital

MEHRAN MEHRI

Département de mathématique et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie industriel

Décembre 2021

© Mehran Mehri, 2021.

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Metaheuristic Approach for the Facility Layout Problem in a Hospital

présenté par **Mehran MEHRI**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Samira KEIVANPOUR, présidente
Jean-Marc, FRAYRET membre et directeur de recherche
Nadia, LAHRICHI membre et codirectrice de recherche
Louis-Martin ROUSSEAU, membre

DEDICATION

To My first teacher in life, my devoted mother, who taught me to fight for my ambitions and never give up when faced with adversities.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere gratefulness to my research director, Dr. Jean-Marc Frayret, for trusting me and allowing me to work on this project under his close supervision on a topic that I am passionate about. His knowledge, insightful advice, patience, and morals have not only aided in the successful completion of my Master's Degree but have also provided me with a great deal of professional and personal enrichment.

I am grateful to my research co-director, Dr. Nadia Lahrichi, for the thoroughness she has shown toward me. Thanks to the extent of her knowledge and references, the exchange of ideas we had during the meeting and discussions was highly informative. Her encouragement was decisive in the accomplishment of this project.

My great appreciation also goes to le Centre Intégré de Santé et de Service Sociaux de la Montérégie-Ouest et Projet Hôpital Vaudreuil-Soulanges for sponsoring this project.

I am further indebted to my friend Benoît Forest, IT responsible at the Department of Mathematics and Industrial Engineering, for his patience, accountability, and generosity. His professional advice and technical assistance in keeping the servers up and running, as well as the remote desktop connection during the pandemic, have been critical to the computation part of my project. I could never imagine the project's progress to this point without him.

RÉSUMÉ

Les impacts d'un aménagement sur la performance d'une organisation dans les différentes industries et secteurs de services sont indéniables. Par conséquent, le besoin d'un système d'aide à la décision parmi les décideurs dans la planification de l'aménagement des installations a amené les problèmes d'aménagement des installations (FLP) à l'épicentre des recherches dans le domaine de l'optimisation. En tant que première partie d'une approche en deux étapes de la planification de l'aménagement des installations, cette recherche vise à concevoir et à mettre en œuvre un cadre d'optimisation pour proposer la meilleure allocation des services aux bâtiments et aux étages d'un futur hôpital près de Montréal, au Québec.

Pour cela, une approche de programmation mathématique et plus précisément des problèmes de semi-affectation quadratique (QSAP) est utilisée pour modéliser ce problème d'aménagement d'installations. Le modèle QSAP adopte les quatre types de contraintes : affectation unique des points de service aux étages, disponibilité de l'espace d'un étage, affectation restreinte d'un point de service aux étages, et la contiguïté et la proximité au sein des points de service. L'objectif de ce problème d'optimisation est de minimiser le flux entre les points de service, ce qui peut inclure les déplacements des patients et du personnel et la manutention du matériel. Le modèle QSAP a été reformulé comme un problème linéaire à nombres entiers mixtes pour s'appliquer à l'un des solveurs MIP existants. La complexité du problème, qui est attribuée à la nature des problèmes combinatoires, empêche l'optimisation du problème de fournir une solution optimale en un temps polynomial. Ce problème réaffirme le fait que, malgré les formidables avancées observées dans le domaine du calcul numérique, qu'il soit logiciel ou matériel, l'utilisation des méthodes exactes dans les problèmes d'optimisation en taille réelle est encore inefficace.

La recherche est passée de l'utilisation d'une méthode exacte au développement d'un optimiseur méta heuristique basé sur l'algorithme génétique, qui peut proposer une solution presque optimale avec une qualité acceptable dans un délai raisonnable. L'optimiseur a été mis en œuvre à l'aide d'une architecture multicouche et dans un environnement de développement de avant-garde. Une analyse statistique descriptive approfondie des résultats des expériences démontre :

• Avec quelle efficacité l'optimiseur converge vers une solution quasi optimale.

• Comment le processus de recherche optimal et les solutions sont sensibles aux modifications des paramètres de l'GA tels que la taille de la population, le taux de croisement, le taux de mutation, la taille du pool d'élite et la taille du tournoi.

• Et avec quelle efficacité l'optimiseur applique les contraintes aux solutions.

ABSTRACT

The impacts of a layout on the performance of an organization in the different industries and service sectors are undeniable. Therefore, the need for a decision support system amongst decision-makers in the facility layout planning has brought the facility layout problems (FLPs) to the epicenter of the researches in the optimization arena. As the first part of a two-stage approach to facility layout planning, this research aims to design and implement an optimization framework to propose the best departments allocation to buildings and floors in a future hospital near Montreal, Quebec.

For this purpose, an approach of mathematical programming and specifically quadratic semi-assignment problems (QSAP) is used to model this facility layout problem. The QSAP model adopts the four types of constraints: unique assignment of service points to the floors, floor space availability, and restricted assignment of a service point to floors, and the adjacency and proximity within the service points. The objective of this optimization problem is to minimize the flow between the service points, which may include traveling of patients and personnel and material handling. The QSAP model was reformulated as a mixed-integer linear problem to apply to one of the existing MIP solvers. The complexity of the problem, which attributes to the nature of combinatorial problems, prevents optimizing the problem from providing an optimum solution in a polynomial-time. This problem reasserts the fact that, despite the tremendous breakthroughs observed in the field of digital computation, whether software or hardware, using the exact methods in real-size optimization problems is still inefficient.

The research has shifted its focus from using an exact method to developing a metaheuristic optimizer based on the genetic algorithm, which can propose a near-optimal solution with acceptable quality in a reasonable time. The optimizer has been implemented using multi-layer architecture and in a state-of-art development environment. A thorough descriptive statistical analysis on the results of the experiments demonstrates:

- How effectively the optimizer converges to a near-optimum solution.
- How the optimal search process and the solutions are sensitive to changes in GA parameters such as population size, crossover rate, mutation rate, the size of the elite pool, and the tournament size.
- And how effectively the optimizer applies the constraints to the solutions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

FLP Facility Layout Problem

SLP Systematic Layout Planning

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 General Context and Background

Facility layout problems have long been an epicenter of researches in optimization. According to research on the production systems, as reported by (Heragu, 2016), material handling costs account for 30 percent to 75 percent of a product's cost (Sule, 1991). The estimated contribution of material handling-related workloads to a company's overall operating budget is 20%–50% (Tompkins & White, 1994). A proper facility arrangement may enhance the overall operational efficiency and cut the total functional expenses to 50% (Tompkins et al., 1996). The abovementioned facts may explain why organizations in many industries and service sectors are interested in finding the best facility planning in their new or existing businesses.

Furthermore, effective facility planning may assist businesses to make the best use of limited resources, such as land and capital, when building new facilities or expanding an existing one. Most researchers attempted to propose optimum solutions for the FLPs through the approach of mathematical programming, such as Quadratic Assignment Problem (QAP) and Mixed Integer Linear Problems (MIPS), but the complicated nature of this approaches left them with disappointment for not being feasible for real-size projects. As a result of this problem, a new age of heuristics and metaheuristics for addressing complicated optimization problems has emerged. Despite not guaranteeing the optimum solution, these new approaches have successfully proven their capability to propose near-optimal solutions.

1.2 Problem

The need of decision-makers for having an effective tool to assist them in discovering the best layout that helps minimize the travels between departments, including patients, personnel, and material at a future hospital near Montreal, QC, is the cornerstone of the definition of this project. Therefore, the definition of the problem entails the development of a decision-making support framework that is able to provide the best possible layout for a hospital. The hospital includes some multi-floor buildings, which each of the floors may provide different areas given the fact that the smaller floor places over the top of a larger one. An architectural plan that can provide the number of buildings, floors, and elevators that existed. Some of the floors are interconnected through a

bridge. Vertical movement, i.e., travel between floors in elevators, is avoided as much as possible. This framework must take into account considerations such as requiring or restricting various departments from being allocated to specific floors, as well as enforcing desired adjacency and proximity between departments.

1.3 Purpose

As part of a research project for obtaining a master's degree, this thesis aims to develop an optimization framework that uses a metaheuristic approach to propose the best solution for the problem described in the section of the problem (1.2) in a reasonable time. The best solution represents a distribution of departments across the buildings and floors with the minimum overall interactions between the departments. This framework must effectively apply the considerations mentioned in the previous section (1.2) to the solution as the problem's constraints. The framework also provides a convenient way to receive and modify the client-side data.

1.4 Thesis Structure

The second chapter covers the definitions, main concepts, and distinguishing aspects of facility layout planning (FLP), as well as several mathematical approaches for formulating an FLP. Following that, it addresses several heuristic and metaheuristic approaches and the reason for choosing the genetic algorithm (GA) in this optimization framework. Finally, it explains the principle concepts of the GA in more detail.

The third chapter addresses the requirements and expectations raised by the project partner, as well as the expected data elements to construct the FLP model. Then it goes into detail on the multi-layer methodology that is used in the development of the framework, which includes the front-end layer, the model constructor layer, and the optimizer layer.

Chapter four provides some notions and definitions of hypotheses, decision variables, objective function, and constraints, as well as a mathematical representation of the problem using the approach of Quadratic semi-assignment problems (QSAP). Following that, it presents a linearized reformulation of the problem along with several techniques that may help lower the computation

efforts. Afterward, this chapter delves into details of how the model and the navigation network are implemented in the Julia programming environment.

Chapter five describes the metaheuristic GA-based optimizer in great depth. It begins with definitions and then explains the GA representation of the project's FLP problem. Then, it explains the implemented GA operators, including population, selection, elitism, coupling, crossover, mutation, as well as the process termination criteria. It also goes into detail on how the constraints are implemented and evaluated in this optimization framework.

Chapter six provides the experiment scenarios and analyses of the sensitivity of the optimization process and the solution to the changes in the GA parameters such as initial and base population, mutation rate, elitism rate, crossover rate, and tournament size. This chapter also analyzes the process of convergence in great depth. Afterward, it digs into how applying the constraints affects the best solution.

Chapter eight summarizes this research project along with the recommendations and looks forward to future research in this area.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

A facility layout is a plan that defines the arrangement of all requirements to manufacture products or provide services. The term facility refers to an entity that assists the performance of an operation, such as a machine tool, a work center, a manufacturing unit, a machine shop, a department, a warehouse, etc. (Heragu, 1997).

2.2 Facility Layout Problem (FLP)

The facility arrangement in a plant, commonly referred to as the "Facility Layout Problem", is recognized to influence production costs, work in progress, lead times, and productivity substantially (Drira et al., 2007). A proper facility arrangement may enhance the overall operational efficiency and cut the total functional expenses to 50%. (Tompkins et al., 1996). Koopmans and Beckmann (1957), who were among the pioneers to explore this category of problems, characterized them as a frequent industrial problem to locate facilities in a plant aiming at minimizing the transporting costs within them. According to (Meller et al., 1998a), the facility layout problem entails discovering a non-overlapping planar orthogonal arrangement of specific numbers of, e.g., n, rectangular facilities inside a given rectangular plan site minimize the distance-based indicator. The facility layout is the allocation and the determination of the relative placements of a given number of facilities amongst the available area, as defined by (Azadivar & Wang, 2000). Another definition by Lee and Lee describes the Facility Layout Problem as the exploration of the placement of a certain number of facilities, say *n*, with distinct area sizes amongst a given total area, which may be restricted to the length or width of the plant area, with the objective of minimizing the sum of the material handling cost and the gap area costs. (Shayan & Chittilappilly, 2004) classify the facility layout problems under optimization problems, which attempt to enhance the efficiency of the layouts via focusing on numerous interactions between facilities, as well as material handling mechanisms while developing the layout.

2.3 What Makes the FLPs Distinguishable

(Drira et al., 2007) lists the factors and design criteria that characterize a facility layout problem as production system specifications, namely product portfolio and volumes, the variety of the parts flow, number of floors that the facility may be allocated to, the dimensions of the facilities, locations of receiving and dispatching, mechanism, and equipment used in material handling.

(Dilworth, 1996) asserts that layout plans rely on the variety and volume of the product and categorize it into four types: fixed product layout, process layout, product layout, and cellular layout. As described by(Drira et al., 2007), the cellular layout groups the facilities based on the families of parts they process. Finding the optimal layout of facilities within each cell is another FLP that should be addressed separately in this regard, according to (Proth, 1992) and (Hamann & Vernadat, 1993). The cellular layout looks to be a proper choice for representing an FLP in a hospital.

In general, two distinctive shapes consist of regular, namely typically rectangular, and irregular, namely typically polygons including at least a 270-degree angle, depict the facility shapes in an FLP, as claimed by (Kim, J.-G. & Kim, 2000) and (Lee, G.-C. & Kim, 2000) respectively. (Chwif et al., 1998) assert that a fixed or rigid block with specific dimensions, as characterized by a fixed-length (L_i) and a fixed-width (W_i), can portray a facility within an FLP. Plus, the same authors argue that a facility area can exhibit it in an FLP through its aspect ratio restricted by an upper and lower bound. Similarly, (Meller, Russell D. et al., 1998a) use the concept of representing a facility by its aspect ratio.

(Tompkins et al., 1996) argue that proper placement of handling devices can cut the materials handling costs by 10% to 30%, which, in turn, expectantly account for 20% to 50% of the production costs. The utilization of the vertical dimension for plants with restrictions in horizontal space, such as those found in the urban areas, is crucial and may make the distribution of the facilities over multiple floors reasonable, as reported in (Drira et al., 2007). As a result, the flow can occur not only between facilities on the same floor (horizontal movement) but also between floors (vertical movement). The handling mechanism requires certain types of transportation devices, such as elevators, conveyors, or escalators, to facilitate vertical movements. Elevators are the most commonly mentioned handling equipment in scholarly papers for multi-floor FLPs (Lee,

K.-Y. et al., 2005). FLPs involving vertical movements must discover the appropriate floor and its location on the floor for each facility; hence, these types of problems are renowned as multi-floor layout problems, as suggested by (Kochhar, 1998). (Johnson, R. V., 1982) appears to be within the leaders to explore multi-floor FLPs tasking with determining the relative positions of the facilities in a multi-floor structure, as reported by (Drira et al., 2007). The FLP with vertical movement thus grabs the attention of the other researchers in this field, such as (Bozer et al., 1994; Meller, Russell D. & Bozer, 1997; Meller, Russell D & Bozer, 1996). (Lee, K.-Y. et al., 2005) consider the number and placement of the elevators as given data, while (Matsuzaki et al., 1999) treat them as FLP decision variables and calculate them via an optimization process. Likewise, the number of floors can either be predetermined, as in (Lee, K.-Y. et al., 2005) or calculated based on the area provided by each floor and the space required by each facility, as in (Patsiatzis & Papageorgiou, 2002).

2.4 Static FLP vs. Dynamic FLP

The changes in demand, manufacturing volume, and products portfolio impact the material and parts flow, resulting in modifications in the layout plan (Drira et al., 2007). As reported in (Gupta & Seifoddini, 1990), one-third of manufacturing companies in the US experience a radical modification in their facility layout plan. The facts above have inspired researchers in FLP to introduce a new aspect in this field called Dynamic Facility Problems. This new approach divides the planning horizon into chunks such as weeks, months, or years. The flows between facilities probably change from a period to another; however, the flow within each different period stays unchanged. As a result, this new problem consists of a collection of FLPs, each of which ascribes a particular layout plan (Balakrishnan et al., 2003; Braglia et al., 2003; Kouvelis et al., 1992; Meng et al., 2004). Optimizing dynamic FLP pursues discovering a layout plan for each planning time unit, with the objective of minimization of the total material handling costs throughout planning duration plus the total costs of layout rearrangements that occurred within the periods, according to (Balakrishnan et al., 2003; Baykasoglu et al., 2006). (Baykasoğlu et al., 2001) also maintains that a dynamic FLP's objective function value must entail the costs incurred to all facility displacements that occur during layout rearrangements.

2.5 Different Approaches to the Mathematical Formulation of FLP

In order to mathematically describe the complicated relationships within the various existing components of a facility layout problem, there are numerous ways to model both static and dynamic FLP. Many studies on mathematical modeling of facility layout problems aim at proposing such formulations that can be adopted into optimization problems with single or multiple objective functions. The problem formulation most likely results in Quadratic Assignment Problem (QAP) or Mixed Integer Programming (MIP) depending on which approach is adopted in FLP mathematical modeling, i.e., discrete or continuous.

Newer methods such as graph theory (Kim, J. Y. & Kim, 1995; Leung, 1992; Proth, 1992) or neural networks (Tsuchiya et al., 1996) are adopted in FLP modeling and optimization to provide solutions to the layout problems.

Furthermore, in instances where the needed data is not readily accessible, the approach of Fuzzy is more or less recommended to be employed in FLP formulation.

2.5.1 Discrete Approach to the FLP Formulation

The discrete approach splits the plant area into identically sized and shaped rectangular pieces. Then, it assigns the blocks to the facilities in sequence until each facility's area is full (Fruggiero et al., 2006). It is possible to assign various numbers of blocks to the facilities that are uneven in size (Wang et al., 2005). Henceforth, the following formulation, which is renowned as a mathematical representation for the quadratic assignment problems, may be used to identify the relative placements of the facilities that can assure minimizing the overall cost of material handling (Balakrishnan et al., 2003):

$$\min \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N}\sum_{l=1}^{N} f_{ik} d_{jl} x_{ij} x_{kl} \qquad (2\text{-}1)$$

This mathematical expression represents the objective function to calculate the sum of the pairwise cost of flow within the facilities. *N* is the quantity of facilities in the FLP. $f_{ik}\ and\ c_{ik}$ are the flow and cost of flow between the facilities i and k respectively. Cost of flow is a weight factor and does

not necessarily associate with a monetary value. d_{jl}represents the distance within the positions *j* and *l*.

x_{ij} is 0 or 1 variable that indicates whether the facility i is assigned to location j, when equals one or otherwise when equals 0. This variable is also renowned as the FLP decision variable.

s.t.

$$\sum_{i=1}^{N} x_{ij} = 1 \qquad (2\text{-}2)$$

This equation assures the unique assignment of each facility to available locations.

$$\sum_{j=1}^{N} x_{ij} = 1 \qquad (2\text{-}3)$$

This equation assures that each location uniquely contains one facility.

Where we have n facility and m locations, n>m, a (Greenberg & Naval Postgraduate, 1969) suggest an updated version of The QAP named The Quadratic Semi Assignment Problem (QSAP), as pointed out by (Burkard et al., 1998). The only difference between QAP and QSAP is that the assignments of the facilities to the locations are not one-to-one mappings, allowing for the allocation of an arbitrary number of the facility to the same floor is permissible. Hence, the ascribed constraints, i.e. (2-3), which ensure one-to-one mapping of the facilities to the floors in the general formulation of QAP, are eliminated in QSAP formulation. Therefore, the QSAP may be a proper approach for the multi-stage FLPs, where the initial phase aims to allocate the facilities to floors with the objective of minimizing the overall handling costs within the facilities.

Although the FLP formulation based on the discrete approach is most likely used to minimize the total material handling costs, it may be adapted to other scenarios with some modifications in the definition of the decision variable and the parameters accordingly. For example, (Kouvelis & Chiang, 1992) and (Braglia, 1996) proposes a discrete FLP formulation in designing a single-row layout with the objective of minimizing backtracking parts. Additionally, (Afentakis, 1989) adopts the same approach to suggest a loop layout with minimum traffic congestion.

The discrete formation of FLP is the most frequently used approach in dynamic FLP (Drira et al., 2007). (Baykasoğlu et al., 2001; Lacksonen, T. A. & Enscore, 1993) uses the approach of discrete formulation in an FLP with the equal-size facilities to assure unique assignment of each facility to the available places in each period of the planning horizon, and then (Baykasoğlu et al., 2001; McKendall et al., 2006) enhances their FLP model to guarantee that each place uniquely accommodates one facility in each planning time unit with the benefit of this approach. Furthermore, (Balakrishnan et al., 1992; Baykasoglu et al., 2006) embeds the budget constraints ascribed to facility reconfigurations in dynamic FLP into their model using the discrete formulation of FLP.

2.5.2 Discrete Approach to the FLP Formulation

Many authors argue that using the continual approach to the formulation of the FLP (Drira et al., 2007), which is commonly renowned as Mix Integer Programming (MIP) (Das, 1993), can replace the discrete approach when the FLP is meant to propose the exact location of facilities and the discrete formulation seems to lose its appropriateness. The centroids or the bottom-left corner of the facilities define where to be located in the layout. In a manufacturing plant, for example, an FLP formulation is meant to precisely locate the facilities over the plant with the objective of minimizing the total cost of the flow of parts within the facilities. A straight line from the buffering point succeeding facility i to the feeding point preceding facility j defines the distance traveled by each piece within the two facilities, calculated using the following formula (Chwif et al., 1998).

$$d_{ij} = \left|x_j^B - x_i^F\right| + \left|y_j^B - y_i^F\right| \qquad (2\text{-}4)$$

Where x_i^B, y_i^B, x_j^F, and y_j^F are the coordination of buffering points and feeding points of the facilities i and j respectively.

The limitation of available area defines constraints to ensure that the space provided by the plant meets the entire area required by the facilities. The area needed by each facility must entail all the spaces necessary for its proper functionality, such as buffers, toolboxes, and (Lacksonen, T. A. J. I. J. o. P. R., 1997). Depending on the nature of the facility layout problem, the total required area may or may not include the empty space within the facilities (Braglia, 1996; Heragu & Kusiak, 1988). The adoption of the constraints into the FLP formulation that prohibit any possible

overlapping between the facilities is very crucial, which is stated by (Welgama & Gibson, 1993) as two criteria below:

$$(x_{jt} - x_{ib})(x_{jb} - x_{it}) \geq 0 \qquad (2\text{-}5)$$

$$(y_{jt} - y_{ib})(y_{jb} - y_{it}) \geq 0 \qquad (2\text{-}6)$$

t: top-left
b: bottom-left
$x_{it}, y_{it}, x_{ib}, y_{ib}, x_{jt}, y_{jt}, x_{jb},$ and y_{jb} are the coordination of the upper corners on the left and bottom corner on the right of the facilities i and j respectively.

Defining A_{ij} as empty space within the facilities (Mir and Imam, 2001) formulates the constraints to prevent facility overlap. The representation of the FLP, thus, is as below:

$$A_{ij} = \lambda_{ij}(\Delta X_{ij})(\Delta Y_{ij}); \qquad (2\text{-}7)$$

$$\Delta X_{ij} = \lambda_{ij}\left(\frac{L_i + L_j}{2}\right) - |x_i - x_j|;$$

$$\Delta Y_{ij} = \lambda_{ij}\left(\frac{W_i + W_j}{2}\right) - |y_i - y_j|;$$

$$f(x) = \begin{cases} -1, & for\ \Delta X_{ij} \leq 0\ and\ \Delta Y_{ij} \leq 0 \\ +1, & otherwise \end{cases}$$

$$\min C = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}\left(\left|x_j^B - x_i^F\right| + \left|y_j^B - y_i^F\right|\right) \qquad (2\text{-}8)$$

$L_i, L_j, W_i, and\ W_j$ are the facilities i and j's length and width, respectively. $x_i, x_j, x_j,$ $and\ y_j$ are the facilities i and j's coordination.

N is the number of facilities and $f_{ij}$is the flow within the facilities i and j. Depending on how a flow is defined, it may include staff, material, etc.

2.5.3 Fuzzy Approach to the FLP Formulation

The stochastic approaches, such as queuing networks (Meng et al., 2004), have been limitedly able to grab the consideration in real-size FLPs, which commonly suffer from ambiguous and insufficient data input. To deal with the FLPs that struggle with inaccuracy or incertitude (Evanst† et al., 1987; Grobelny, 1987; Raoot & Rakshit, 1991) suggest the concept of Fuzzy logic. (Evanst† et al., 1987) provides a fuzzy formulation for an FLP that aims at finding the arrangement of the

unequal-sized facilities on a site. This formulation defines Fuzzy pairwise relationships between the facilities and assigns a priority rating to each of them, indicating the importance of proximity within that pair of facilities. A heuristic method for solving this problem is then provided.

(Grobelny, 1987) introduces a model for an FLP that involves allocating n facilities to n defined places with the objective of minimizing the overall cost of handling. This model considers the pairwise proximity and flows between the facilities as the fuzzy linguistic variables. A heuristic algorithm is then developed based on 0-1 Fuzzy relationships that aim at finding the most proper assignment of the facilities to the available locations. (Raoot & Rakshit, 1991) use the same fuzzy logic aspects in modeling an FLP that entails discovering the best placements of the facilities on a site relying on their correlations, which have a Fuzzy implication. (Gen et al., 1995) come up with a model for a multi-row FLP with multiple objectives that include unequal-sized facilities. The model treats the empty spaces within the facilities as a fuzzy factor because exact pre-determination is impossible. Similarly, the FLP model (Dweiri‡ & Meier, 1996), which tackles a layout problem using a discrete approach, considers the flow of work-in-process between the facilities, the flow of information, and the handling equipment circulating within the facilities as fuzzy agents. The Activity Relationship Chart (ARC), which indicates the priority of the relationships between the facilities based on domain experts' viewpoints, is provided to the CORELAP heuristic method to propose the most appropriate facility arrangement.

2.5.4 Multi-Floor FLP Formulation

The concept of multi-floor relates to the layout problem of locating the facilities vertically on the floors and horizontally across each floor with the objective of minimizing total handling costs within the facilities. Due to land constraints and high prices, multi-floor FLP applications are more popular and realistic in constructing industrial plant projects or service-provider businesses in an urban area, such as hospitals or office buildings. (Bernardi & Anjos, 2013). The extra constraints in multi-floor FLP, i.e., traveling across both horizontal and vertical directions and the use of vertical handling equipment or stairwells, raise the level of problem complexity. Except in minor cases of multi-floor FLP, exact optimization methods prove limited capacities in providing a global optimum solution due to the higher complexity (Hahn et al., 2010). Heuristic algorithms may be a significant facilitator in real-size problems with high complexity (Bernardi & Anjos, 2013).

Solving methods suggested for multi-floor FLPs adopt a variety of approaches that involves one or more phases. Some of these approaches, such as SPACECRAFT (Johnson, R. V., 1982), launch with a basic solution and strive to improve iteratively through interchange the facilities on the same floor or between the floors. This method has several drawbacks, such as limitations in terms of swapping facilities that occur across the improvement process, as well as the possibility of dividing a particular department across multiple floors. The adverse effect on a department's functionality may prohibit it from being divided over different floors. MULTIPLE (Bozer et al., 1994), which employs the same concepts of continuous improvement as SPACECRAFT, attempts to solve the problem in a single stage. It does, however, outperform SPACECRAFT in terms of raising the number of facility swaps in each loop and preventing department division across the floors. The global optimum in the heuristics mentioned above is most likely compromised with the local optimum since their algorithms seek the best viable facility swap in each iteration. Thus, the process relies on the optimization trajectory.

(Liggett & Mitchell, 1981) and (Kaku et al., 1988) employ mathematical programming to formulate multi-floor FLPs, each of which bears certain conditions or drawbacks such as being limited to accommodating equal-sized facilities, providing a single lift position, and dividing the facilities between multiple floors, as also reported by (Bernardi & Anjos, 2013).

Various methods, such as the one based on genetic algorithm, exist that prove their capabilities to support other aspects of a real-sized FLPs, i.e., accountability for the need for lifts (Matsuzaki et al., 1999), considering the interior structures like walls and passages (Lee, K.-Y. et al., 2005), or corridors (Chang et al., 2006). Furthermore, (Goetschalckx & Irohara, 2007) use mathematical programming to formulate a multi-floor FLP with elevators as the handling equipment amongst the floors from two approaches of using either full-service elevators (MFFLPE-A) or elevators that serve a contiguous set of floors (MFFLPE-B). These formulations consider the number and locations of the elevators as the decision variables. (Hahn et al., 2010) propose a mathematical formulation using quadratic assignment problem (QAP) for the multi-floor FLP that entails minimization of both total handlings costs and the costs ascribed to the building evacuation plan, resulting in the proper balance between the cost factors mentioned above.

For instance, STAGES (Meller, Russell D. & Bozer, 1997) and FAF/TS (Abdinnour-Helm & Hadley, 2000) are amid the two-stage formulations that adopt mixed-integer linear programming

(MILP) to suggest the best distribution of the departments across the floors in their first phase, aiming at globally minimizing the total vertical travel costs within the departments. STAGES, afterward in its second phase, strive to discover the best placement of the departments individually on each floor using an updated version of SABLE, preventing the departments from being swapped between floors. FAF/TS, on the other hand, employs the Tabu Search algorithm in the second phase of the problem-solving process. The Tabu Search, like Simulating Annealing, attempts to outperform in terms of lowering the risk of slipping into a local optimum using the approach of placing the departments with a massive interaction on the same floor. In opposed to STAGES, other multi-stage methods, such as FLEX (Meller, Russell D. & Bozer, 1997), enable department swaps in the second phase (Bernardi & Anjos, 2013). The experimental data from benchmarking the FLEX over STAGES reveals that allowing the departments to be swapped over the second phase is not necessarily efficient since the rise in costs associated with vertical handling most likely outweigh the decrease in the costs ascribed with horizontal handling (Meller, Russell D. & Bozer, 1997).

Additionally, given the fact that STAGES outperforms the SABLE, (Bernardi) concludes that the approaches of multi-stage that do not allow departments swaps after assigning them to the floor in the first stage are a proper choice. Accordingly, the authors suggest a multi-stage methodology that employs the approach of mixed-integer linear programming for the first phase to allocate the departments to the floors with the aim of minimizing the vertical handling costs within the departments. Afterward, this method applies an approach that (Anjos & Vannelli, 2006) utilized for a single-floor FLP to tackle the problem of finding the optimal layout for each floor individually. (Ahmadi & Akbari Jokar, 2016) propose a multi-stage method to deal with multi-floor FLPs consisting of three stages, each of which is based on the approach of mathematical programming. The initial phase of this methodology is accountable for assigning the departments to the floors while minimizing the total costs of vertical flow. The second phase, afterward, establishes the departments' relative locations. Finally, the third phase finds a detailed layout for each floor, including fixed or non-fixed elevators.

2.5.5 Linearization of QAP

(Sherali & Alameddine, 1992) provide an approach of the reformulation for bilinear programming, which is claimed to be frequently applied in different fields such as location theory, dynamic assignment, economics and game theory, risk management, and so on. The purpose of this reformulation is to make relaxation in a problem, resulting in constructing tight linear programming that is imbeddable in a branch-and-bound algorithm. The new algorithm expectantly functions more efficiently in terms of computational complexity, while its convergence is theoretically proven as well. The optimum resulting from solving the linearized problem provides a rigid lower bound on the optimal value to the original problem.

(Adams, W. & Johnson, 1994) are the first, using the concept of linearized reformulation of (Adams, W. P. & Sherali, 1986, 1990), propose a Boolean mixed-integer linear reformulation for QAP to enable the computation of the bounds and construction of cutting planes. (Adams, W. & Johnson, 1994) argue that the absence of efficient QAP solving algorithms prevents it from attracting further academic attention. The combinatorial character of the problem is to blame for this inefficiency. The authors claim that their reformulation of QAP, which is based on the approach of mixed 0-1 linear programming, theoretically outperforms all other known linear reformulations concerning the strength of the continuous relaxations, dual space construction, and lower bound computation.

(Rostami & Malucelli, 2014) suggest a revised linearized reformulation for the QAP, which is claimed to function very successfully concerning the tighter lower bound it produces to the original problem. The authors theoretically prove that the linearized formulation of the QAP is equivalent to the original problem as well. In addition, their computational experiments also maintain that their linearized reformulation of the QAP demonstrates a more efficient performance in terms of CPU time.

(Billionnet & Elloumi, 2001) consider the following mixed integer linear reformulation of the QSAP as one of the strategies for establishing the dual space, computing QSAP lower bounds, and constructing cutting planes.

QSAP01:

$$\min\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{M} f_{ik}d_{jl}x_{ij}x_{kl} + \sum_{i=1}^{N}\sum_{k=1}^{M} a_{ik}x_{ik}\right) \quad (2\text{-}11) \quad x \in \{0,1\} \quad (i = 1,\dots,N, k = 1,\dots,M) \quad (j = 1,\dots,N, l = 1,\dots,M)$$

s.t.

$$\sum_{k=1}^{M} x_{ik} = 1 \quad (2\text{-}12)$$

With the definition of a new variable as: $y_{ikjl} = x_{ik}.x_{jl}$the linear reformulation is as below:

RLT01:

$$\min\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{M} f_{ik}d_{jl}y_{ikjl} + \sum_{i=1}^{N}\sum_{k=1}^{M} a_{ik}x_{ik}\right) \quad (2\text{-}9) \quad x, y \in \{0,1\} \quad (i = 1,\dots,N, k = 1,\dots,M) \quad (j = 1,\dots,N, l = 1,\dots,M)$$

s.t.

$$\sum_{k=1}^{M} x_{ik} = 1 \quad (2\text{-}10)$$

$$y_{ikjl} \leq x_{ik}$$

$$y_{ikjl} \leq x_{jl}$$

$$1 - x_{ik} - x_{jl} + y_{ikjl} \geq 0$$

$$y_{ikjl} \geq 0$$

2.5.6 Symmetry Breaking

Most of the various types of FLPs include symmetry effects in different forms, causing the optimization of the mixed-integer problem to be more complicated. The optimization of a 180-degree rotated FLP with pairwise isomorphic flows, for instance, obviously results in the same solution (Anjos & Vieira, 2017).

Various strategies are presented for breaking the symmetry effects and alleviating their impacts on the optimization complexity. (Meller, Russell D. et al., 1998b) use the technique *position q*, which is named by (Sherali et al., 2003), to fix departments to be located in a specific quarter of the facility using two constraints as $x_q \leq 0.5W_f$ and $y_q \leq 0.5L_f$ (where $x_q$and y_q are the coordination of the department's centroid, W_f and L_f are the facility's width and height, respectively, and the facility's corner at the left bottom is assumed as the origin point). In the initial phase of multi-stage FLP optimization methodologies, however, when the objective is to assign departments to the centroids of the floors, this technique loses its efficacy. Alternatively, (Sherali et al., 2003) propose *position p-q* as a more strict technique. This method chooses two prioritized departments, p, and q, either with the highest flow in between or based on area, and forces the p's centroid to be located south and west of the q's centroid. The strategy for breaking the symmetry can be imposed to the problem either implicitly or explicitly via adding constraints. The strategies for breaking the symmetry can be imposed to the problem either implicitly through reformulation of the model (adding conditions) or explicitly via introducing constraints (Anjos & Vieira, 2017). In allocating departments to floors in the initial step of multi-stage FLPs, we have two isomorphic permutations as x_{ikjl} and x_{jlik} when a pair of departments, i and j, are assigned to two different floors, k and i. The following

reformulation of the linearized QSAP, which is suggested as a reduced formulation of QSAP by (Billionnet & Elloumi, 2001), is expected to outperform in solving procedure.

RLT02:

$$\min\left(\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{k=1}^{M}\sum_{l=1}^{M} f_{ik}d_{jl}y_{ikjl} + \sum_{i=1}^{N}\sum_{k=1}^{M} a_{ik}x_{ik}\right) \quad (2\text{-}13)$$

s.t.

$$\sum_{k=1}^{M} x_{ik} = 1 \quad (2\text{-}14)$$

$$y_{ikjl} \leq x_{ik}$$

$$y_{ikjl} \leq x_{jl}$$

$$1 - x_{ik} - x_{jl} + y_{ikjl} \geq 0$$

$$y_{ikjl} \geq 0$$

$$x, y \in \{0, 1\}$$

$$(i = 1, \dots, N-1, k = 1, \dots, M)$$

$$(j = i+1, \dots, N, l = 1, \dots, M)$$

2.6 Different Approaches to Solving an FLP

The enhancement that facility layout planning can bring to an organization from different aspects has motivated the researchers to propose a myriad number of methodologies to solve a facility layout optimization problem. While these methodologies use diverse approaches, they all strive for a common goal of either discovering the optimum, whether global or local or finding the best solution that meets the given constraints. A solving method may fall into the categories of optimization algorithms (exact methods) or heuristic-based search algorithms (approximated

methods), as well as the new methodologies based on data science-related approaches such as machine learning or AI (Artificial Intelligence).

(Kouvelis & Kim, 1992) developed a branch and bound algorithm for working efficiently in medium-sized problems in designing mono-directional loop network layout problems in automated manufacturing systems, which is claimed to be enhanced by establishing dominance criteria for recognizing local optimum solutions. (Meller, Russell D. et al., 1998a) use the branch-and-bound technique offered by CPLEX to solve a reformulation of FLPs modeled after Montreuil's MIP approach. However, due to the complexity of the problem, this approach is accused of being inefficient in real-sized FLPs. (Kim, J. G. & Kim, 1999) suggest an algorithm with a branch-and-bound scheme to find the optimal location of the receiving and delivery points attributed to each department in a given layout with the objective of minimizing the total distance traveled by material flow within the receiving/delivery points. (Fischer et al., 2019) propose an enumerative method for solving the space-free multi-row FLP (MRFLP) that adopts a branch-and-cut scheme as an exact approach. This method is claimed to solve three situations in a reasonable computational cost: two rows each row 16 departments, three rows each row 15 departments, and four and five rows each row, 13 departments.

The fact that FLP's combinatorial nature causes the exact methods to lose their effectiveness in real-size FLPs has prompted a multitude of academic studies to lead to the development of heuristic and metaheuristic algorithms. (Kusiak & Heragu, 1987) name these approaches *suboptimal algorithms* and categorize them as construction algorithms, improvement algorithms, hybrid algorithms, and algorithms based on graph theory. The construction-based algorithms considered the oldest and the most straightforward algorithm in heuristic-based FLP solvers, begin generating a layout from scratch. Improvement type of approaches starts with an initial solution and progressively improves it by developing a new solution through a random or systematic swap of the facilities. The generated solutions are evaluated in each improvement iteration. The best one is maintained, and then the process continues until no further improvement can be made to the current best solution. The construction-based approaches are blamed for generating solutions lower than expected. The quality of solution produced by improvement-based approaches is very reliant on the initial solution (Singh & Sharma, 2006). Among the more prominent algorithms based on the construction approach are HC66 (Hillier and Connos, 1966), ALDEP (Seehof and Evans, 1967),

CORELAP (Lee and Moore, 1967), RMA Com I (Muther and McPherson, 1970), MAT (Edwards et al., 1970), PLANET (Deisenroth and Apple, 1972), Lsp (Zoller and Adendorff, 1972), Linear placement algorithm (Neghabat, 1974), FATE (Block, 1978), INLAYT (O'Brien and Abdel Barr, 1980), and FLAT (Heragu and Kusiak), as pointed out by (Kusiak & Heragu, 1987). On the other hand, the eight most well-known improvement-based algorithms, according to (Kusiak, 1997) are CRAFT (Buffa, 1963) and (Buffa et al., 1964), H63 (Hillier, 1963), H63-66 (Hillier, 1966), COL (Vollman et al., 1968), Sampling algorithms (Nugent et al., 1968), FRAT (Khalil, 1973), COFAD (Tomplins and Reed, 1976), and Revised Hillier algorithm (Picone and Wilhelm, 1984).

The exact approaches are unable to provide the optimal solutions for real-size FLPs in a polynomial time. On the other side, among the suboptimal methods, the solutions offered by construction-based approaches have lower quality than expected. Additionally, the improvement-based approaches are blamed for being overly reliant on the quality of the initial solution. Accordingly, several academic efforts have concentrated on developing the methods that combine schemes of the optimal and suboptimal approaches, dubbed Hybrid Algorithms by (Bazaraa and Kirca, 1983).

2.7 Optimization Metaheuristics

Optimal approaches have yet to make substantial achievements in dealing with real-size optimization issues, despite tremendous breakthroughs in digital computation, whether hardware or software. Furthermore, suboptimal approaches fail to retain reliability due to pitfalls such as creating lower-than-expected quality solutions and being overly reliant on the quality of the initial solution. While they offer a ray of optimism, hybrid approaches have not been able to fill the gaps as well adequately. Therefore, the high demand for reliable optimal or near-optimal solutions in the decision-making process has prompted academic studies to invent a new optimization era dubbed metaheuristics. The prefix “Meta,” which comes from the Greek language, implies that these approaches are beyond the former optimization algorithms from different aspects. Their logic is designed in such a way, enabling them to tackle the problems that are suffering from inadequate data inputs or limited computational resources. Most of the Metaheuristics commonly adopt some scheme of stochastic optimization that finding optimal or near-optimal solutions entails using random variables.

Massive breakthroughs in digital processing, whether hardware or software, have boosted metaheuristics' capacity to develop near-optimal solutions for exceedingly complex decision-making problems in a wide range of industries and service sectors, which had previously remained unsolvable.

A metaheuristic can provide a high-quality solution to a given optimization problem, provided that it can strike a balance between diversification and intensification. In concept, diversification means exploring the search area to discover the regions containing high-quality solutions effectively, and intensification represent exploiting the agglomerated search experience (Birattari et al., 2001).

2.7.1 Classification of Metaheuristics

(Birattari et al., 2001) classifies the metaheuristics based on the criteria consisting trajectory-based vs. discontinuous, population-based vs. single-point, memory usage vs. memoryless, single vs. multiple neighborhood structures, dynamic vs. static objective function, and nature-inspired vs. non-nature inspired.

Trajectory-based vs. discontinuous algorithms: Trajectory-based metaheuristics pursue one single solution instance that corresponds to a closed walk-on or larger jumps in the neighborhood graph. Typical examples of trajectory-based metaheuristics include Tabu Search (TS) and Simulating Annealing (SA). On the other hand, discontinuous methods entail a multitude of local searches, each of which begins with a starting point, i.e., solution construction, and continues with making modifications to formerly encountered local optimum and performing randomized greedy-construction heuristics. Ant Colony Optimization (ACO), Genetic Algorithm (GA), and GRASP fell into this class of metaheuristics.

Population-based vs. uni-point search: The significant difference between population-based and single-point metaheuristics is whether their rationale is based on adopting a large number of search points within a population or a single point. The optimization algorithm of single-point metaheuristics, such as SA, TS, ILS, and GRASP, processes just one instance of solution in each iteration. In population-based metaheuristics like ACO and GA, the search region is described by a population of generated solutions, whether called ant or genotype.

Memory use vs. memory-less: In the memory usage algorithm, the registered search history may impact the direction of future searches. Short-term memory is used to prevent cycling over previously encountered solutions, while long-term memory is used to diversify the searching. Population-based metaheuristics, such as ACO and GA, use the registered experience of formerly encountered solutions in constructing new solutions. SA and GRASP are two metaheuristics that do not employ memory in their algorithm logic.

Single vs. multiple neighborhoods: In ILS and ACO, for instance, a kick-move transfers the search to a secondary neighborhood when the search in the former neighborhood reaches optimal. The mutation as a GA operator mostly functions in a similar way. Another GA operator, crossover, may be explained as movements in hyper-neighborhoods (Vaessens et al. 1995), in which a bundle of solutions, in GA the size of this bundle has the size of two, is accountable for the construction of a new solution. The ACL and GRASP do not use a particular structure such as a neighborhood for constructing new solutions.

Static vs. dynamic objective function: While the majority of metaheuristics employ a single method of evaluating search state across the solving optimization problem, known as the static objective function, a handful of them use multiple methods, known as dynamic objective functions. The TS method may be classified as a dynamic objective function in which specific spots in the search area are avoided due to the extremely large value of the objective function value. The other metaheuristics, such as SA, ILS, GA, and GRASP, employ a static objective function.

2.7.2 Simulated Annealing (SA)

SA is a robust uni-solution metaheuristic approach proposed by (Kirkpatrick, 1984).and (Černý, 1985) to address complex optimization problems. It functions efficiently in avoiding local minima through consecutive changes of neighborhoods, i.e., transitioning to a state with higher energy until the best solution is reached in the search area (Kesavan et al., 2020). Emulating a thermodynamic phenomenon, i.e., simulated annealing approach starts at a very high temperature and attempts to find the minima at the lowest energy state at the end cooling process. Setting the initial temperature to a high value ensures a vast feasible area with a large number of solutions. The iterative algorithm then progressively decreases the temperature as it approaches the minima, forcing the process to converge (Gogna & Tayal, 2013). The impact of cooling schedules on the process entropy drives

the development of various strategies in SA, including exponential, linear, logarithmic, prolonged decrease, non-monotonic, and adaptive (Nourani et al., 1998).

(Gomes, 2000) propose an SA-based algorithm to address the facility layout problems in manufacturing plants that involve four necessary parameters: initial temperature, final temperature, the parameter of temperature, and cooling rate. The algorithm launches with generating a random solution that includes *n* facilities. The search comprises two loops: an inner loop that self-iterates at a set temperature and an outer loop that gradually decreases the temperature by the cooling rate until it reaches the final temperature. The inner loop manipulates the current solution through an exchange of facility i with j, when i, j ranging from 1 to $n-1$ and from i to n, respectively. The algorithm terminates when all the neighborhoods that existed in the neighboring set are traversed and analyzed by the aforementioned nested loops. In addition, the most recent best solution has to be maintained separately. The generated new solution, at the end of the inner loop, replaces the current best solution if it outperforms the best solution of the current neighborhood, as shown below:

if $f\left(R^{min}\right) < f(R^k)$ then

$$R^b = R^{min} \quad (2\text{-}15)$$

else

$$R^b = R^{max} \quad (2\text{-}16)$$

Where:

R^b: final best solution;

R^k: the most recent solution at the kth iteration

R^{min}: the solution with lowest cost in the neighborhood that includes that includes R^k;

R^{max}: the solution with highest cost in the neighborhood that includes that includes R^k;

R^k: the most recent solution at the kth iteration

$f\left(R^{min}\right)$: The lowest cost found in the neighborhood that includes R^k;

$f(R^{max})$: the highest cost found in the neighborhood that includes that includes R^k.

The expression (2-15) is to assure that the solution with minimum cost is retained. Alternatively, the expression (2-16) avoids premature convergence and slips into local minimal.

(Matai et al., 2013) argue that their enhanced version of SA is able to effectively solve multi-objective FLPs with more than 30 facilities. (Johnson, D. S. et al., 1989, 1991) investigate SA's performance on four NP-hard problems: the traveling salesman problem (TSP), graph partitioning problem (GPP), graph coloring problem (GCP), and number partitioning problem (NPP). According to the findings, judging the SA's performance is a mixed bag. In terms of computing time, the SA beat the other algorithms in TSP. However, in the other three examples, it was unable to demonstrate its superiority over the others due to higher computing costs or the provision of a low-quality solution. (Aarts & van Laarhoven, 1987) uses SA in graph partitioning problem with three different cooling rates and observe that the solution quality may differ by 10 percent. (Bertsimas & Tsitsiklis, 1993) report that literature accuses the SA approach of excessive computation time in some application cases and in general when the cooling rate is very small. (Bertsimas & Tsitsiklis, 1993) also maintains that SA is an easy-to-implement probabilistic approximation algorithm applicable to a wide range of problems and can propose proper solutions for complex optimization problems. (Gogna & Tayal, 2013) shed light on some considerations SA approach:

i. Setting the starting temperature to a sufficiently high degree, a too low temperature leads the algorithm to fall into local minima. At the same time, a too high temperature makes reaching the optimal solution exceedingly difficult for the algorithm.

ii. An appropriate cooling schedule, i.e., a sufficiently progressive decline in temperature, is required to provide a more stable mechanism.

iii. SA appears to be an appropriate approach for an optimization problem that has an oscillated path toward the global optimal, i.e., the presence of a large number of local minima. However, making the same choice for the problems with a steady path while approaching global minimum, i.e., the presence of a small number of local minimums, looks irrational and causes an excessively delayed process convergence.

2.7.3 Genetic Algorithm

GA approach is referred, to name a few: (Boussaïd et al., 2013; Dao et al., 2017; Kesavan et al., 2020; Nordin & Lee, 2016; Wong & Ming, 2019), to be the first nature-inspired heuristic

extensively applied for a wide range of purposes, such as industries, financial sectors, as so on, to provide as expected quality solutions to highly complicated decision-making problems in a reasonable time. (Misevičius & Verenė, 2021) propose a hybrid Genetic-Hierarchical algorithm for QAP, which is claimed to outperform the other metaheuristics in their literature using an improved crossover operator. The results are expected to be more promising if the algorithm's parameters are rigorously adjusted.

Despite all the advances made to the GA approach since its inception in 1975, it still consists of four core components: chromosomal encoding, selection, crossover, mutation, and fitness evaluation (Dao et al., 2017).

Fitness function is a set of performance measures ascribed to a particular chromosome, which are meant to be optimized. When the fitness function is evaluated, it provides a numerical value, also referred to as "fitness," or "the figure of merit," that is assumed to be proportionate to the "utility" or "ability" of the solution embedded into that chromosome (Beasley et al., 1993).

2.7.3.1 Selection

Among the literature, (Beasley et al., 1993) provide a very insightful and concise analysis of parent selection methods and classify them into two main classes of explicit or implicit fitness remapping. Fitness scaling, fitness windowing, and fitness ranking are the three subclasses of explicit fitness remapping. The optimal search may suffer from over-compression induced by a single extreme fitness value, either the best or worse, in both fitness scaling and fitness windowing techniques, especially in the face of unusually extreme fitness. Over-compression is also blamed on fitness rankings, even though distributing forcing power from a single instance to a group of best-fit individuals might relieve the problem.

The tournament selection method falls into the class of implicit parent selection methods. Due to proven efficiency, the binary tournament selection technique, known as the simplest and most frequently used method, randomly chooses two individuals and adds the better-fit one to the mating pool. This technique conceptually has an elitism effect on the optimization process. In general, although the tournament size is allowed to be set to more than two instances, the larger size reinforces the selection pressure, resulting in deterioration in the balance between intensification and diversification and higher risk for the search process to converge to a local-optima (Beasley et

al., 1993). The implementation of the tournament method is easy. It may need significantly less computational time since it is easy-to-implement in multi-threads and parallel processing development architecture (Goldberg & Deb, 1991).

2.7.3.2 Crossover

The algorithm adopted to implement a crossover operator undeniably impacts the process convergence and the GA efficacy in the search for optimality. The three primary crossover types are standard, binary, and real/tree crossovers. Crossover the parent genes to generate a better offspring (so-called a gene or a solution) is the keystone of the optimal searching in a GA. The algorithm adopted to implement a crossover operator has an undeniable impact on process convergence and the GA efficacy in the search for optimality. The application categorizes different crossovers into three main groups: standard, binary, and real/tree crossovers. Standard, binary, and real/tree (application dependent) crossings are the three primary categories, each of which is further broken into subcategories (for more information, refer to A). Depending on the application, the literature on GA mostly recommends using a crossover rate from 0.6 to 1.0. Although raising the crossover rate improves the possibility of crossing genes, it may reduce the likelihood of obtaining a favorable outcome.

While addressing any optimization problem using the GA approach, it is recommended to study the search space through the lens of modality extremes and analyze the stability in the performance of the existing crossover operator, subsequently investigating the need for selection or creation of a new crossover (via a combination of existed ones)

2.7.3.3 Mutation

If a GA solely uses a crossover operator as the new offspring generator, the good characteristics of the parents implicitly have more likelihood to be transmitted over the generations, resulting in deterioration in population diversity, a rise in the probability of premature convergence search process to a local-optima. (Hassanat et al., 2019). The literature recommends using mutation operators to alleviate this problem. Adopting mutation operators in a GA is an effective measure to prevent a wide range of genes from being either lost or unexplored over the generation transition, resulting in an enhancement or preservation of the GA search area and improved GA performance.

A mutation entails modifying the value of one or more bits of a single chromosome (Korejo et al., 2013). When a GA optimal search process becomes stuck in a local-optima, the more iterations the mutation has, the more likely it is to assist the process move away from the local-optima by reproducing in other neighborhoods and expanding the search area. Although a mutation delays the learning process, it compensates by preventing the GA search process from convergent to a local optimum. The implementation of the mutations has long time been in one of the forms as point, frameshift, translocation, ell of which entail exchanging, flipping, or switching binary bits when the chromosome is binary-coded. Furthermore, a random value from a range bounded to a lower and upper value replaces a gene in a chromosome in random mutations (Kanwal et al., 2013).

2.7.3.4 Elitism

Elitism in GA is a sub-function of the selection that aims to retain a diverse set of best-fit solutions, so-called high-quality individuals, from previous generations in a container known as elite pool and pass them to the next (Jaradat et al., 2016). In a GA mechanism, the elite pool functions as an adaptive memory, accumulating the data about the global optimum (Jaradat et al., 2016), boosting the learning process in GA optimal search to more effectively approach the global optimum. The adoption of the elite pool reinforces the aspect of exploitation in GA metaheuristics. Therefore, the size elite pool plays a crucial role in maintaining the balance between intensification and diversification, i.e. exploitation and exploration. Rising the elite pool size strengthens the power of exploitation, resulting in elite pressure, which may expedite the process convergence to a local-optima.

Conversely, decreasing the elite pool size deteriorates the learning process, leading to a delayed process convergence. Therefore, the size elite pool plays a crucial role in maintaining the balance between intensification and diversification, i.e., exploitation and exploration, respectively. Rising the elite pool size strengthens the power of exploitation, resulting in elite pressure, which may expedite convergence to a local-optima. Conversely, reducing the elite pool size degrades the learning process, leading to a delayed process convergence.

2.7.3.5 Microbial Crossover

The natural reproduction process of microbes inspired the development of this type of crossover approach in a GA heuristic. The GA procedure uses this notion to randomly choose two genotypes from the candidate pool and evaluate them based on their fitness values to identify the winner and loser. Afterward, the winner part infects the loser part by replacing some of the loser's genes with its genes. Finally, the winner part and infected genotype return to the candidate pool (Beasley et al., 1993). Although the crossover rate can be any value from 0 to 100%, a rate of 50% is recommended by the literature.

2.7.4 Genetic Algorithm vs. Simulated Annealing

Simulated annealing, tabu search, and genetic algorithms are, without a doubt, amongst the most reputable heuristic methods (Reeves, 1996). The Genetic Algorithm is a population-based metaheuristic approach that has grabbed researchers' interest owing to its remarkable ability to provide near-optimal solutions to even the most complicated problems. Amongst the single-point search heuristic methods, Simulated Annealing has gained the attention of academics in solving optimization problems. GA and Genetic Algorithm appear 147 times among 1662 titles in a search for the most popular subject in the field of OR (Operations Research), compared to 11 times for simulated annealing (Chambers, 1998).

Population-based metaheuristics are robust searching algorithms because they can reproduce a new solution, perhaps a better-fit, through recombining the existing solutions (Jaradat et al., 2016). This is the characteristic that mainly contributes to exploitation, resulting in intensifying the optimal search. The use of a diverse elite pool as a memory to retain valuable data when searching for global optima provides the search process with the capability to more efficiently exploit the information in optimal search (Jaradat et al., 2016). Compared with single-point search heuristics, population-based heuristics, which use sets of solutions rather than a single solution to reproduce a better-fit solution, offer a significant advantage in dealing with QAP (Misevičius & Verenė, 2021). Furthermore, (Misevičius & Verenė, 2021) consider genetic algorithms the most potent population-based metaheuristics in providing the solution to QAP and other related problems.

SA is a single-point neighborhood search heuristic, which adopts a memory-less algorithm with the capability to escape local optima and hence avoid premature convergence (Gogna & Tayal, 2013). (Goldberg & Deb, 1991) addresses two problems linking to simulated annealing: lack of population and inability to reproduce new solutions through recombining exited ones. These problems are likely to lead to weakness in exploitation.

CHAPTER 3 CONTEXT OF THE STUDY AND THE METHODOLOGY

The main rationale for defining this project is to offer an optimal facility layout plan for a future hospital near Montreal, QC. The initial studies in the project led to generalizing the main idea into developing an optimization framework specially designed for providing the optimum solutions for the problems in facility layout planning. This framework assists decision-makers in finding the best distribution of departments among floors and buildings, minimizing the travel costs under different conditions and configurations for future projects or existing facilities (for the purpose of continuous improvement) in a reasonable time.

The following sections shed light on the facility layout optimization problem in a hospital, which is the focus of this research project, from several perspectives including, client expectations and requirements, the client data, and an analysis of the adopted methodology and technologies to develop an optimization framework.

3.1 Client’s Requirements and Expectations

The project's stakeholders expect a solution that adopts quantitative optimization methods to offer an optimal Facility Layout Plan for a future hospital, which promise an improvement of the efficiency in service delivery (as the main KPI of the optimization problem) considering the following constraints:

i. Bounding a specific department to one or more floors.

ii. Preventing a specific department from being allocated to one or more floors.

iii. Increasing or forcing the adjacencies among the departments with interdependencies, such as functional and shared resource dependencies.

iv. Decreasing or avoiding the adjacencies between the departments with deleterious or unfavorable impacts on each other's performance.

3.2 Client Data

Even a very sophisticated quantitative model must be backed by relevant data in order to prove its efficacy. Therefore, modeling a problem mainly relies on the existing data, or their availability is

manageable, forasmuch as data provisioning is time-consuming and requires considerable infrastructure. An initial architectural plan has already been drafted, provides the geospatial data and the area available per floor for this project. Additionally, an analysis of facility planning using a qualitative methodology (namely, the Systematic Layout Planning approach - SLP) reflects a relational matrix containing the pairwise flow of patients, personnel, and materials between departments or service points, as well as weight factors that prioritize the flows.

The optimization framework was mocked up using a model based on a CAD file pertinent to an emergency hospital from the internet to avoid the expected challenges in data gathering and accelerate the development process. The flow between service points and the weight factors are estimated accordingly. The friendly user design of the framework's data feeding allows the users to feed their data into the FLP model afterward and obtain their expected solutions.

The following sections will render the primary physical mock-up model data in detail, i.e., the general specifications and geospatial data extracted from the CAD file found on the internet and the estimated data for the flow and weight factors.

3.2.1 General Specifications

The physical base model for FLP modeling consists of two principal buildings located in north and south wings. The two buildings are linked through a functional area on the ground floor, and, likewise, a bridge interconnects two wings on the first floor. Inside the bridge is a non-functional

Table 3-1 General physical specifications of the Emergency Hospital

Buildings	Floors	Area (m^2)	Available Space (m^2)
North Building	Basement	680	550
	Ground Floor	680	535
	First Floor	900	780
South Building	Basement	670	650
	Ground Floor	1024	980
	First Floor	900	760
Connecting area on the ground floor	Ground Floor	600	550
The area inside the bridge	First Floor	150	110

area that only accommodates the elevators and stairs. Table 3-1 shows the general specifications of this hospital. In addition to this, Appendix 1 provides drawings of each side of the floors in the buildings.

3.2.2 Physical Model's Geospatial Data

The Emergency Hospital's CAD file analysis using AutoCAD software allowed us to extract all detailed geospatial data of the buildings, floors, departments, and service points. Table 3-2 contains the coordinates of the floors' centroids.

Table A- 1, list of departments and service points and their space requirements

Table 3-2, the coordinates of the floors' centroids.

Building	Floor	Centroid's Coordinates		
		X (m)	Y (m)	Z (m)
North Building	Basement	-460.5985	60.2028	-3.0808
	Ground Floor	-570.5838	59.8999	0.0000
	First Floor	-561.3982	-5.8805	4.4865
South Building	Basement	-468.9259	43.4499	-4.0808
	Ground Floor	-578.3015	33.8161	0.0000
	First Floor	-574.6514	-29.8383	4.4865
Connecting area on the ground floor	Ground Floor	-569.2917	47.7948	0.0000
The area inside the bridge	First Floor	-564.3508	-14.495	4.4865

Departments			Service Points				
Description	Code	ID	Description	Code	ID	Area (Min)	Area (Max)
Surgery Clinic	D00117	SG011	Operating room for major surgeries	D00021	SG001	34	37
			Operating room for minor surgeries	D00022	SG002	34	36
			Preparation & anesthesia room for major surgery	D00023	SG003	21	23
			Preparation & anesthesia room for minor surgery	D00024	SG004	21	23
			Sterilization room for operating room	D00025	SG005	17	20
			Recovery room	D00026	SG006	17	20
			Post-anesthesia care unit 1	D00027	SG007	12	14.5
			Post-anesthesia care unit 2	D00028	SG008	12	14.5

Departments			Service Points				
Description	**Code**	**ID**	**Description**	**Code**	**ID**	**Area (Min)**	**Area (Max)**
			Post-anesthesia care unit 3	D00029	SG009	12	14.5
			Post-anesthesia care unit 4	D00030	SG010	12	14.5
Outpatient Clinic	D00116	OC007	Doctors' office 1	D00036	AD004	20	22
			Waiting room	D00037	GS015	10	12
			Staff lounge in the outpatient clinic	D00038	AD005	12	14
			Inpatient emergency caring room	D00039	OC001	45	49
			Outpatient clinic room 1	D00040	OC002	14	16
			Outpatient clinic room 2	D00041	OC003	14	16
			Admission office	D00043	OC005`	7	9.5
			Waiting room (outpatient clinic)	D00044	GS016	10	11
			Stretchers room	D00045	OC006	10	11
Medical Imaging Test clinic	D00118	DC009	Echocardiography	D00046	MI001	20	23
			X-Ray	D00047	MI002	25	31
			The waiting room for the MI department	D00048	MI003	35	37
			Doctor's office in MI department	D00049	MI004	15	18
			Staff lounge	D00050	MI005	10	11
			Reception office in MI department	D00051	MI006	20	23
			MRI	D00145	MI007	30	33.5
Specialized Clinic	D00119	SC012	Registration & admission office	D00058	AD010	12	15
			Outpatient surgery room 1	D00059	SC001	16	19
			Outpatient surgery room 2	D00060	SC002	16	19
			Waiting room 1 at SC	D00061	SC003	15	17
			Waiting room 2 at SC	D00062	SC004	15	17
			Gynecology & obstetrics doctor office 1	D00063	SC005	15	17
			Gynecology & obstetrics doctor office 2	D00064	SC006	15	17
			Orthopedic doctor office	D00065	SC007	15	17
			Neurology doctor office	D00066	SC008	15	17
			Cardiology doctor office	D00067	SC009	15	17
			Hematology doctor office	D00068	SC010	15	17

Continued next page

Table A- 2 in Appendix A includes a list of departments and service points and their space requirements. The following two chapters, Mathematical Modeling and Development of a GA Based Heuristic Optimizer, explain how this optimization tool uses the extracted data to instantiate an instant of the model to pass into the optimizer.

3.2.3 Flow Matrix

As indicated in Client Data, the primary key indicator in searching for an optimum solution in this optimization framework, based on the client's expectations, is the improvement in total service

delivery efficiency obtained by a solution (3.2). Minimizing the non-value-added activities in the service delivery process, in some aspects, may reflect improvement in total service delivery efficiency. The displacements between service points by the patients, personnel, and materials are an explicit example of non-value-added activities. Hence, minimizing the total distance traveled by the agents mentioned above leads to improving the efficiency in service delivery time. In facility layout planning methodologies, the flow between two service points refers to the frequency of trips performed by each of the agents described above in between in a given period of time. Thus, the total distance traveled is calculated by summing the product of flow between a pair of service points and the corresponding distance as explained by the following formula:

$$\sum_{i,j} f_{ij}.d_{ij} \qquad (3\text{-}1) \qquad i \neq j$$

Where i and j each of which refers to a service point.

This optimization framework assumes that the quantitative analysis phase of facility layout planning establishes how to aggregate three types of flows that correspond to patients, personnel, and materials. The optimization algorithm uses the aggregated flows value in a matrix structure as an input. The flow matrix is a square matrix that maintains all of the facilities in rows and columns correspondingly. It contains the flow between each pair of facilities at their cross point in the form of from origin to destination.

The estimation of flow matrix bases its calculations on an emergency hospital with 1000 visits a day which is distributed amongst the departments as in Table 3-3

Table 3-3, the admission capacity of the functional divisions based on 1000 visits a day.

Departments	Admission Capacity (%)	Daily Visits
Outpatient clinic	20%	200
Surgery Clinic	5%	50
Medical Imaging Test Center	15%	150
Specialized Clinic	10%	100
Physiotherapy Clinic	10%	100
Nephrology & Hemodialysis Clinic	5%	50
Medical Test Lab (Blood & Pathology)	15%	150
General Clinic	20%	200

In addition, visitors are expected to select one of the three hospital entrances on the basis of the probability, as seen in Table 3-4. Forty percent of visitors refer to the operating division via the information service desk.

Table 3-4, the likelihoods for visitors to enter the hospital from each entrance.

Hospital Entrances	Probability of Use
Main Entrance	60%
North Entrance	30%
East Entrance	10%

Since the navigation network in this project is undirected, the flow between each pair of service points in both directions is integrated into one value. Therefore, the flow matrix is an upper triangular matrix, as seen in Table A- 4 in Appendix A (Client Data).

3.2.4 Travel Cost Unit

Displacements in both horizontal and vertical directions are not equally desired, forasmuch as particular bottlenecks along the routes, such as elevators, significantly increase vertical travel time, thus leading to a delay in service delivery time. Adopting a weight factor aids in the development

of a notion of travel costs, which individually correspond to each direction of travel. This weight factor, or more exact, travel cost unit, may not necessarily refer to any monetary value.

3.2.5 Relationship Chart

In order to give whether more or less importance to a flow between a pair of service points, e.g., to apply proximity preferences, a weight factor is defined to act as coefficients for the corresponding flow. The qualitative facility layout planning methodologies, such as SLP, have specific tools to prioritize the flows between service points as a relationship chart. Chapter 5, on page 105, elaborates how this optimization framework may use a relationship chart to apply FUZZY adjacency constraints.

3.3 Methodology

This optimization framework enjoys a multi-layer architecture design as below:

i. Front-End

ii. Model Constructor

iii. Optimizer

The following sections explain each layer in detail.

3.3.1 Front-End Layer

The frontend layer, including GUIs and data tables, is implemented in Microsoft Excel using the VBA programming environment. This layer is responsible for gathering data from the client-side, managing data (i.e., CRUD and data validation), translating data into a data structure that can be interpretable by the model construction layer in Julia, and retrieving post-execution data. Forasmuch as Microsoft Excel is the most widely used spreadsheet software with a very easy-to-use built-in programming environment, i.e.,VBA, is the main reason for choosing the software as the platform for data manipulation. Besides, a mid-level Excel user can draw up different reports from output data in the desirable formats using existing powerful features.

3.3.2 Model Constructor Layer:

This layer is in charge of constructing the FLP model based on either MILP or GA. The result of optimizing the generated model in this phase may be an optimal or near-optimal solution, which contains the best possible distribution of the service points among the buildings and floors. The optimization framework injects the GA model into the embedded GA-based optimizer as it injects the MILP into the external solver, e.g., Gurobi or CPLEX, via a plugin in Julia. The following sections elaborate the procedure of generating the FLP model in both MILP and GA aspects, which basically includes common steps with specific differences according to the concept of the methodology.

i. **Data streaming**: Both aspects of model construction begin with this step, allowing for the transfer of user data from the front-end layer into the constructor layer and the return of the optimization result and log data from the back-end to the front at the conclusion of the process.

ii. **Flow Matrix**: This step, which establishes the flow matrix between the service points, is shared by both FLP modeling approaches. Later on, the flow matrix contributes to the development of the model's objective function.

iii. **Navigation Network**: Both aspects of FLP modeling adopted by this optimization framework require establishing the navigation network that contains the routes between the centers of the floors. The navigation network plays a crucial role in developing the model's objective function by calculating the shortest path between the centroids. Chapter 4, on page 49, describes how the optimization framework builds the navigation network in detail.

iv. **Development of the objective function**: The objective function represents the goal of the optimization, which conducts the optimum search toward the optimality. Hence, the existence of this step is necessary for each of the approaches described above, albeit with distinct implementations. In MILP, the model constructor writes the objective function as a mathematical expression and embeds it into the model object. Alternatively, in GA-based FLP modeling, the objective function is a function that returns a single scalar value as it is invoked. The model constructor incorporates a reference of this function into the GA model.

v. **Embedding the Constraints**: This FLP optimization framework may enact certain constraints in the model construction stage, resulting in a definite effect on the solutions. These types of constraints are called construction level constraints or binary constraints. The Unique Assignment Constraint (UAC) and Restricted Assignment Constraint (RAC) are from these types of constraints. Alternatively, the framework can apply various types of constraints throughout the optimization process via the concept of penalizing the objective function value, with varying effects on the solutions. The Floor's Area Availability Constraints and Adjacency Constraints can be implemented at the construction level or the process level. The section Enactment of Constraints (5.3.2 on page 60) explains this concept in more detail.

   Given the fact mentioned above, like the objective function step, both FLP modeling aspects include this step but with distinct implementations. In MILP, the model constructor enacts the constraints in the form of certain mathematical expressions and wraps them up in the model object. In place, in GA modeling, the constructor applies the construction level constraints to the base chromosome. (For definition of the chromosome, refer to 5.1)

vi. **Development of the base chromosome**: This step exclusively exists in GA modeling. The model constructor generates a base chromosome, which represents the assignment pattern, to determine how the service points relate to the floors. Base chromosome also includes the construction level constraints, such as UACs, and serves as a template for solution reproduction across the optimization process.

3.3.3 Optimizer Layer

This layer encompasses two different concepts of implementations, depending on the type of model it receives. When the model is a mathematical model constructed by the JuMP library, this layer is responsible for passing the model to an external solver by injecting it into that solver's plugin. Conversely, if the model is a GA-type object, the solver layer will pass it to the built-in GA solver. Chapter 6, Experimentation and Analysis of Results, represents all post-execution data and analysis.

3.3.3.1 MILP Solver

The optimizer layer may use specific external solvers that are well-known to be leaders in the context of mathematical optimization, namely: Gurobi and CPLEX. Indeed, this has been possible thanks to the dedication from their developer companies to the academic community via granting their academic licenses to the research activities. Chapter 4, Mathematical Optimization, and chapter 6, Experimentation and Analysis; demonstrate the developed mathematical models, the solution analysis for developed models under some defined conditions.

3.3.3.2 GA-Based Optimizer

This project has developed a heuristic optimizer based on the Genetic Algorithm to find the optimal or near-optimal solution for optimization problems in facility planning with the non-linear objective function. Optimization based on a genetic concept uses an iterative, evolutionary metaheuristic. The GA optimizer embedded in this optimization framework begins with receiving the GA model generated by the model constructor layer and establishes an initial population. Then, it approaches the optimality through generation transition using the operators, namely selection, crossover, and mutation, taking the concept of elitism. Chapter 5, Development of a Genetic Algorithm Based Heuristic Optimizer, elaborates on implementing the built-in GA optimizer in detail. Furthermore, Chapter 6, Experimentations and Analysis of Results, analyzes the GA optimizer's performance and process using a descriptive statistics approach to demonstrate its effectiveness. Further, it evaluates the efficacy of the optimization framework in enacting the constraints in practice.

3.4 Development Environment and Programming Technology

The use of Julia programming technology and ecosystem is one of the distinctive aspects of this optimization tool. The main reasons for choosing Julia programming technology in the development of this optimization tool are as follows:

i. A high-level programming language with lightning-fast speed

ii. A feature-rich programming language in mathematical programming and graph theory

iii. Feature-packed in linear algebra operations

iv. One-of-the-kind in data manipulation and data science

v. Powerful in functional programming with the capability to implement the aspects of object-oriented programming (OOP)

CHAPTER 4 MATHEMATICAL OPTIMIZATION IN FLP

This chapter offers a detailed explanation of how the designed framework adopts mathematical optimization methodologies, also referred to as the quantitative optimization methods, in developing an FLP optimization model.

4.1 The Approach of Integrating Qualitative Methods into the Quantitative Methodology

Although quantitative methods presumably prove to be more accurate in searching for optimal in an optimization problem, their need for a mathematical model that accurately represents all the hypotheses and considerations of the problem should not be overlooked. Even though a model elaborately developed using the most sophisticated techniques has been adopted, it could still have difficulties grasping all aspects of a real-world problem. This chapter, in what comes next, describes how this FLP optimization framework uses the outcomes of the FLP qualitative analysis, i.e., the matrix of relations and the matrix of flows between service points (as seen in Table A- 4) which contain weight factor and flow of patients/personnel/materials respectively, as the translator of the requirements of a real problem and map them into the parameters of the optimization model.

4.2 Mathematical Modeling

As previously stated, using the mathematical methods in optimizing facility layout was determined as the best fit for the FLP in the current project. Mathematical optimization has broadly been applied to facility layout planning for a long time. The very first step in mathematical optimization is modeling, but beforehand, the following section focuses on defining terms and concepts we need to select and develop an appropriate model.

4.2.1 Definitions

Service point: The smallest indivisible business unit provides a value-added service as part of the whole process in a department.

Department: An integration of the service points which are grouped based upon providing a unique value-added service. A department may be physically divisible amongst different locations in the plant.

Horizontal Distance: The distance horizontal distance is all distances traveled by patients, staff, or materials along the X or Y Cartesian axes within the origin and the destination.

Vertical Distance: The distance horizontal distance is all distances traveled by patients, staff, or materials along the Z Cartesian axis within the origin and the destination.

Travel Cost: The travel cost is a metric that aims to prioritize travel in one direction over another. This factor does not necessarily associate with any monetary value. In this project, transport cost includes horizontal and vertical transport costs that, as their names imply, are related to horizontal and vertical travel.

Architectural Consideration: The requirements and conditions are taken into effect in the facility layout based upon the architectural or applicable standards and are not influenced by the optimization process.

4.2.2 Analysis of Hypotheses

1) During the first stage of the decision process, the locations of the service points on each floor are not decision variables.
2) The location of elevators is a design consideration and is determined in the architectural plan.
3) Negative rejection relationships between service points and departments are not allowed. If rejection or incompatibility is present, it will be modeled as a constraint.
4) Building floors do not necessarily need to be equal, but the projection of higher floors on the ground falls inside or on top of the projection of lower floors.
5) During the 1st stage of the decision process, the travel distance is computed as the weighted centroid-to-centroid between the floors in both horizontal and vertical directions.
6) Travel in the vertical direction between the floors can only occur through elevators. All elevators are full-service and move bi-directionally. The modeling does not involve the capacity of the elevators.

7) The architectural plan gives the shape and area for each floor in each building.
8) The architectural plan gives the maximum number of buildings and the maximum number of floors in each building.

4.2.3 Representation of the Mathematical Model

Chapter 2 sufficiently explains that the Quadratic Assignment Problem is the longstanding model employed in FLP aiming at finding the optimal allocation of n service points to n distinct places while minimizing the total travel cost. Given that the first stage of FLP in this project would account for assigning N service points to the centroid of K floors, the following QSAP (Quadratic Semi-Assignment Problem) model, i.e., the equation 4-1 to 4-3, is the best fit formulation this project's FLP. In what comes next, this section explains how this model represents all aspects of the FLP in detail.

$$min \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{M} \sum_{l=1}^{M} w_{ij}.f_{ij}.(c^h.d_{kl}^h + c^v.d_{kl}^v).x_{ik}.x_{jl} \qquad 4\text{-}1$$

$s.t.$

$$\sum_{k=1}^{M} x_{ik} = 1 \qquad 4\text{-}2 \qquad i = 1, \dots, N$$

$$\sum_{i=1}^{N} a_i x_{ik} \leq A_k \qquad 4\text{-}3 \qquad k = 1, \dots, M$$

$x_{ik}, x_{jl}, z_{ikjl} \in \{0,1\}, \quad i,j = 1, \dots, N, \qquad k,l = 1, \dots, M$

$\boldsymbol{f_{ij}}$: flow between service points i and j

$\boldsymbol{w_{ij}}$: weight factor to prioritize the relation between service points i and j

$\boldsymbol{c^h}, \boldsymbol{c^v}$: horizontal and vertical travel cost correspondingly

$\boldsymbol{d_{kl}^h}, \boldsymbol{d_{kl}^v}$: horinzontal and vertical distance between centroids of the floors k and l.

$\boldsymbol{x_{ik}}$: Boolean decision variable and equals to either 1 when the service points i is assigned to floors k or 0 otherwise.

$\boldsymbol{a_i}$: the area required by the service point i.

$\boldsymbol{A_k}$: available area provided by the floor k

4.2.4 Decision Variable

The objective function evaluation (4-1) must involve the service points effectively assigned to a floor rather than every presumable permutation of them. An implementation of this concept in a mathematical model involves the product of pairwise permutation of the binary decision variables, i.e. $x_{ik}.x_{jl}$, by which each variable determines the floor accommodates the corresponding service point.

4.2.5 Objectives Analysis

This section explains how the optimization model aims to achieve the project's general objective via translating them into the components that can be embedded into the mathematical model.

i. Improving the efficiency of service delivery

The main objective is to have an optimum layout plan that can provide more efficient service delivery. Given that service efficiency connects explicitly with the efficiency of the delivery process, reduction in non-value-adding activities in the process leads to the enhancement in process efficiency. Hence, FLP optimization mainly focuses on minimizing total travel distance between service points since material handling and the transportation of patients and hospital personnel within a hospital are obvious examples of non-value-added activities. Additionally, it should be noted that the types of transports within the hospital are primarily identified as horizontal and vertical. Provided limit capacities of the lifts, commonly used to handle vertical travel, may prompt an incentive to avoid vertical travel as much as possible.

In order to explicitly address this purpose in the model, distinct weighting factors, namely c^h and c^v, are devised to act as coefficients for the horizontal and vertical distances to calculate the travel cost associated with each transportation type accordingly. For each pair of service points, the pairwise travel costs should be multiplied by the flow between those points. The flow between a pair of service points is quantified by the number of trips taken by patients, staff, or material between them within a given period, such as a day, week, month, or year.

ii. Pairwise Prioritization of the Adjacencies between Service Points

Functional dependencies or resource-shared dependencies can stimulate the tendency to prioritize the adjacency between two service points via reducing the distance between them as much as possible. The optimization model can incorporate this strategy through more emphasizing the role of flow coefficient in the model. This means that the product of the matrix of relations, which is the outcome of the qualitative methods and includes weighting factors for rank assessment of the links between the service points, with flow matrix will yield a new flow matrix that more effectively represents the importance of interrelations between service points. The following mathematical expression states the new strategy:

The facts described above reveal that, from minimizing the travel cost viewpoint, the optimization process must seek the minimum value of total travel cost, which is evaluated by the function below to reach the optimal solution:

$$\sum_{i=1}^{N-1}\sum_{j=1}^{N} w_{ij} \times f_{ij} \times (c^h \times d_{ij}^h + c^v \times d_{ij}^v) \qquad (4\text{-}4)$$

4.2.6 Constraints Analysis

Four categories of constraints may be incorporated into the mathematical model described in 4.2.3 (page42) for this project's FLP optimization framework. This section explains each type in detail.

i. Unique Assignment Constraints (UQAC)

As a result of the definition of the service point and its distinction from a department, a service point locates exclusively on a floor, whereas a department can be divisible within several floors. In order to enforce all the feasible solutions to commit to respecting this policy, the model embeds equation 4-2.

ii. Floor Area Availability Constraints (FAAC)

In the FLP, the floor areas are the Capacity Constrained Resources (CCR), meaning that the service points located on a specific floor are constrained by the available space associated with that floor.

Adopting inequality 4-3 into the model ensures that the viable solutions fulfill the floor area restrictions.

iii. Restricted Assignment Constraints (RAC)

An elaborately developed FLP needs to exert restrictions on assigning service points to floors depending upon the functional requirements or architectural considerations. Integration of the following equation to the model applies restricted allocation policy to the problem's feasible area:

Likewise, equation (4-5) can formulate the rule when the model must prevent a service point from

$$\sum_{k \in \mathbb{D}} x_{ik} = 1 \qquad (4\text{-}5)$$

- $\mathbb{D} = \{\forall\, f_i \in \mathbb{D},\ i \in \mathbb{N}, 1 \leq i \leq N | d_1, d_2, \dots, d_{10}, \dots, d_n\}$
- $\mathbb{D}$ represents the set including the available floors to which a certain service point can be assigned.

locating on a particular floor by providing the complement set comprising the allowed floors assigned.

iv. Adjacency Constraints

The 1st-stage model (i.e., assignment model) adopts this category of constraints either to force the placement of multiple specific service points on the common floor or to prevent them from sharing the same floor. The adjacency policies are integrable in the FLP model using the following equations:

a) Shared floor adjacency constraints

$$\sum x_{ik} = m\nu \qquad (4\text{-}6)$$

- $\mathbb{S} = \{\forall\, s_i \in \mathbb{S},\ i \in \mathbb{N}, 1 \leq i \leq N | s_1, s_2, \dots, s_{10}, \dots, s_n\}$

$$\sum_{i \in \mathbb{S}} x_{ik} \leq 1 \qquad (4\text{-}7)$$

- $\mathbb{S} = \{\forall\, s_i \in \mathbb{S},\ i \in \mathbb{N}, 1 \leq i \leq N | s_1, s_2, \dots, s_{10}, \dots, s_n\}$
- $\mathbb{S}$ represents the set including the service points to share the floor.
- k is the index number of the floor to be shared

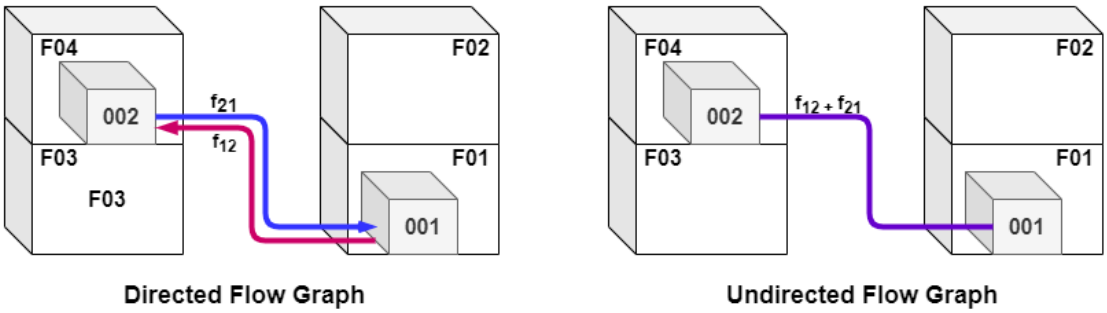
- k is the index number of the floor to be shared

b) Distinct-Floors Adjacency Constraints

4.2.7 Integer Cut

Flow network graph in an FLP is undirected when the flow directions within the service points are not meaningful in analyzing the network graph. In this case, an aggregated undirected flow represents the flows between two service points, and the flow amount is the accumulation of the flow values in both directions. Figure 4-1 illustrates this concept intuitively.

Figure 4-1, Directed and undirected flow network graph.



In an undirected flow network graph, the flow matrix is an upper triangular matrix. On the other hand, the affinity between the service points i and j, i.e., products of permutation of i and j that are denoted as $(f_{ij} + f_{ji}).x_{io}.x_{jd}$ and $(f_{ij} + f_{ji}).x_{jd}.x_{io}$, are identical. The phenomenon mentioned above causes a symmetric effect in the objective function, resulting in more nodes in the solving algorithm and a longer solution time. (Billionnet & Elloumi, 2001) suggest including the condition of $i < j$ in a QSAP model, which is represented in 4.2.3, to alleviate the symmetric effect.

4.2.8 Adding Valid Inequalities

The integration of valid inequalities into mathematical optimization models boosts their performance up in terms of solving time, according to (Anjos & Vieira, 2017). Devising an

Figure 4-2, Valid Inequality.

$$x_{ik} + x_{jl} - z_{ikjl} \leq 1 \qquad (4\text{-}8)$$

$$x_{ik}, x_{jl}, z_{ikjl} \in \{0,1\}, \quad i,j = 1,\dots,N., \qquad k,l = 1,\dots,M$$

initiative to reduce the number of branches while the optimization process runs can yield a dramatic enhancement in total performance. A study (Ejeh et al., 2018) suggests defining a binary variable, namely z_{ikjk}, to flag whether or not each pair of service points, namely: i and j, share the same floor. Then, the rule to assign the two service points to distinct floors when they are not located at the same floors, must forced through embedding the following inequality into the model:

Section 0 also uses this technic as a part of the QSAP model linearization.

4.2.9 Linearization of the QSAP Model

The QSAP model falls into the Mixed Integer Non-Linear Programming (MINLP) category due to its non-linear objective function. Even though some of the available solvers can find global optimal in a less-than real-size MINLP, a scrutiny of the solving algorithm reveals that they rely on piecewise linearization to use the known MILP algorithms. Therefore a linearized reformulation of the QSAP model, suggested by (Billionnet & Elloumi, 2001), is used to linearize the QSAP model in the first phase of FLP in this project. The new reformulation, from (4-9) to (4-14), represents a MILP model, which is now applicable to MILP optimization algorithms via different available solvers such as Gurobi and IBM CIPLEX. When needed, the restricted assignment constraints or adjacency constraints may be added to this model.

MILP Model for the 1st Stage of the FLP Model.

$$min \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{M} \sum_{l=1}^{M} f_{ij}^{w} . (c^{h} . d_{kl}^{h} + c^{v} . d_{kl}^{v}) . z_{ikjl} \qquad (4\text{-}9)$$

$s.t.$

$$\sum_{k=1}^{M} x_{ik} = 1 \qquad (4\text{-}10) \qquad i = 1, \dots, N$$

$$\sum_{i=1}^{N} a_i x_{ik} \leq A_k \qquad (4\text{-}11) \qquad k = 1, \dots, M$$

$$z_{ikjl} \leq x_{ik} \qquad (4\text{-}12)$$

$$z_{ikjl} \leq x_{jl} \qquad (4\text{-}13)$$

$$z_{ikjl} \geq x_{ik} + x_{jl} - 1 \qquad (4\text{-}14)$$

$$x_{ik}, x_{jl}, z_{ikjl} \in \{0, 1\}, \quad i, j = 1, \dots, N, \qquad k, l = 1, \dots, M$$

$\boldsymbol{f_{ij}^{w}}$: prioritized flow between service points

$\boldsymbol{c^{h}}, \boldsymbol{c^{v}}$: horizontal and vertical transport const correspondingly

$\boldsymbol{d_{kl}^{h}}, \boldsymbol{d_{kl}^{v}}$: horinzontal and vertical distance between centroids of the floors k and l.

$\boldsymbol{z_{ikjl}}$: Boolean integration variable equivalent to $x_{ik} . x_{jl}$ and equals to either 1 when the service points i and j are assigned to floors k, l correspondingly or 0 otherwise.

$\boldsymbol{x_{ik}}$: Boolean decision variable and equals to either 1 when the service points i is assigned to floors k or 0 otherwise.

$\boldsymbol{a_i}$: the area required by the service point i.

$\boldsymbol{A_k}$: available area provided by the floor k

4.2.10 Analysis of Solving Method

4.2.10.1 Implementation of the FLP Model

4.2.10.1.1 The flow of Patients, Staff, and Material within the Service Points

The matrix of the flow of patients, staff, and material is one of the crucial parameters involved in evaluating the objective function. The FLP qualitative analysis methods yield flow matrix and often relations weights matrix. Users enter the flow data via the GUIs developed in the Excel environment. The XLSX library in Julia facilitates streaming flow data from Excel to the model layer. The model layer constructs a flow matrix that contributes to the construction of the navigation network graph, as well as the objective function. Next section deals with the navigation network graph in detail.

As discussed in chapter 3, an estimated flow matrix is drawn up for this project, as shown in Table A- 4 in Appendix A.

4.2.10.1.2 The Internal Navigation Network Graph

The distance, the direct path between the service points that the flows must navigate in each journey, is another key parameter to consider when evaluating the objective function. In order to calculate the distances, this framework establishes a navigation network graph comprising origin and destination points and navigation routes in between. A set of connection points and connecting lines, which link the points together, made up a navigation path. Since in this phase, the service points are generally assigned to the floors centroids, in each navigation path, a pair of floor centroids take the place of origin and destination points correspondingly. The navigation routes conduct the flow from an origin to the destination via the connecting lines that straightly extend along with one of the Cartesian axes and join consecutively together at connecting points.

Likewise, the flow data, the adopted GUIs using VBA in Excel enable the users to enter the coordinates of the centroids and connecting points and the configuration of the navigation network. The network configuration deals with defining the routes by determining the order of connecting points included. The model layer receives the data by streaming it from the Excel workbook and passing it to the graph builder function.

This framework uses the libraries $LightGraphs$ and $MetaGraphs$ from the Julia ecosystem to implement the navigation described above network. Firstly a simple graph is created that accounts for providing interrelations within the navigation routes and the composition of connecting points in each one. Each graph encompasses nodes and edges to represent the connection points and connecting lines correspondingly. Then the graph builder function calls the $MetaGraphs$ to enrich the created light graph type object with metadata that includes attaching the coordination of the connection points to the nodes and adding the weights, notably the flow amount and the travel costs, to the edges. The edges extended along either the X or Y axis receive the horizontal travel cost as opposed to the edges extending along the Z-axis that incur the vertical travel cost. The graph builder function returns a $MetaGraphs$ type objects that are responsible for calculating the total travel costs associated with each pair of service points as shown below:

$$f_{ij}^{w}.(c^{h}.d_{ij}^{h}+c^{v}.d_{ij}^{v}) \qquad (4\text{-}15)$$

Supposing that more than one route is possible between a pair of service points, the graph object applies the shortest path algorithm to find an optimal route and calculate the pertinent travel cost to return.

4.2.10.1.3 FLP Model

As discussed in chapter 3, a library from the Julia ecosystem, namely JuMP, allows us to establish a MILP model and inject it into several solvers as easily as possible. The key advantage of using JuMP, aside from its robust features, is integrating the Julia's strength points, notably in working with matrix and linear algebra operations and the exceptional performance, with the versatility of choosing from a vast range of available MLIP and MNLIP solvers.

After creating the data model, the Model Layer instantiates a JuMP type model object and proceeds with adding the variables and constraints, which have been encoded in interpretable scripts, to the model. Then, it creates the objective function and embeds it into the model object. It finally injects the model object to the solver using its plugin in Julia and calls the optimization procedure to search for the optimal solution.

4.2.10.1.4 Optimal Solution

When the optimization process concludes with an optimal solution, apart from the report provided by the solver, which is interpreted according to the guidance can be found at the developer's website, such as Gurobi or IBM CIPLEX, all post-execution results are accessible via the model object. The result data will be streamed back to the Excel environment by writing a workbook using the XLSX library. Microsoft Excel provides appropriate tools to analyze and visualize the data. Chapter 6 provides examples of the analysis and visualization of the results.

4.2.11 Analysis of Result

As previously mentioned, the FLP in this project is split into two stages, the first of which aims to find an optimal solution for the assignment of the service points to the floors. This stage, therefore, delivers an assignment plan that promises to minimize the total cost of transport within the service points while meeting the constraints.

CHAPTER 5 DEVELOPMENT OF A GENETIC ALGORITHM BASED HEURISTIC OPTIMIZER

As mentioned in chapter 3, due to the challenges in solving the QSAP model developed for the first stage of the problem in real scale on account of being a case of NP-hard problem, project's B.O.K decided to place the development of a GA-based heuristic solver in project's plan. This chapter unfolds the detail of the development process of this heuristic solver. Figure 5-1 illustrates the process roadmap of the optimizer development.

Figure 5-1. GA-based heuristic optimizer development process roadmap.

And subsequent sections explain the concept design and detailed design of the development process in more detail.

5.1 Definitions

This section provides detailed and accurate definitions for all GA-related terms and structural components necessary to understand the developed algorithms for this optimization framework deeply.

1. Bitflag

A data structure that refers to one bit and maintains a Boolean value of TRUE/FALSE. This solution architecture uses a bitflag to address a decision variable (x_{ij}), which determines whether or not a specific service point is on a particular floor.

2. Bitarray

An array type data structure that comprises a set of bitflag. The array indices, namely n and m, refer to the number of floors and service points, respectively. This solution architecture uses a bitarray of $n \times 1$ to address the location of a specific service point on the floors and a bitarray of $n \times m$ to represent a complete layout solution.

3. Phenotype

In biological definition, Phenotype is a realistic and tangible set of visible traits that are uniquely associated with an individual, such as eye color, hair color, physical shape, and the like. Each feasible layout assignment plan corresponds to an individual if we intend to define such an architecture in our solution architecture. Subsequently, a phenotype would represent an arrangement of departments on floors and in buildings, subject to architectural essentials and functional requirements. Phenotype is a structure out of the scope of this solution architecture, and the reason for defining it is for information and understanding how genotype links with phenotype.

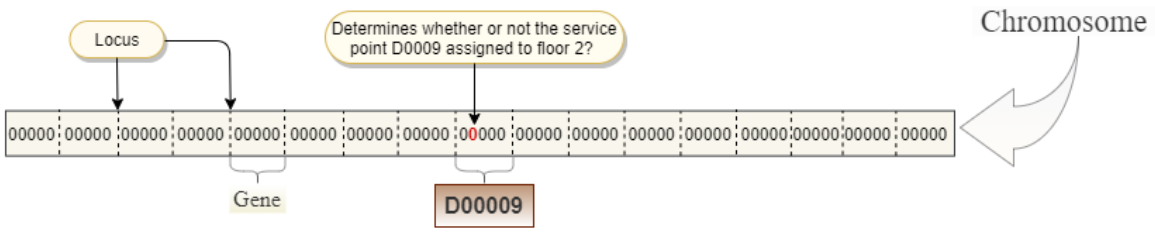
4. DNA

DNA is the smallest and most basic data element (a bit size) to encode a trait in heredity. In this solution architecture, the Boolean data type of True or False contained in a DNA specifies whether or not a particular service point has located on a specific floor. The order of bits in DNA is essential since the index number of each bit in the gene vector, which is equivalent to the row index in the matrix structure, correlates to a floor number.

5. Gene

In biological concepts, a gene is a segment composed of a set of DNA that encodes a trait of an individual. Maintaining the same approach in this architecture, a gene is a set of DNA, i.e., a vector of *n* (i.e., the number of floors) bits, depicts the unique assignment (given that multiple-allocations of a service point is not allowed in this architecture) of a particular service point to a floor. This definition confirms that all the bit values contained in the gene vector are allowed to store Boolean value False except for one of them at a time. This would sufficiently satisfy the constraint of the

Figure 5-2, the structure of a chromosome in GA representation of FLP.



unique assignment of service points to floors. A gene has a one-to-one relationship with a service point. Figure 5-2 shows the gene structure and its relationship with the chromosome and the service points.

6. Allele

An allele is multiple forms of a particular gene. Each gene has two alleles passed down from the offspring's parents. Depending upon the crossover results and mutation on these alleles, a new version of the gene is replicated to locate the offspring's genotype. The existence of alleles causes diversity in the genotype. An example of an evolutionary optimization algorithm based on biological concepts may help to clarify the definition. While reproducing a new genotype from parents A and B, given that a particular gene may determine which floor a specific service point is allocated to, there are two alleles from offspring's parents to reproduce a new version of the gene that performs the same function in the new genotype. Indeed, these two alleles are two distinct versions of the same gene, each indicating that the same service point is located on a different floor. The crossover and mutation algorithms will now be in charge of determining how a new gene may be generated from alleles A and B.

From the development point of view, the allele is a vector of *n* bits (i.e., a collection of DNA) integrated into a gene to allow a service point to be allocated to a floor (as illustrated in Figure 5-3). The DNA number in an allele is equal to the numbers of DNA in a gene, which is a fixed integer depending on the floor count of the FLP. This explanation suggests that, in order to shift a service point from one floor to another, the affiliated gene must swap its contents with another different allele.

7. Chromosome

There are two approaches to identifying a chromosome: basically, a chromosome is a sequence of DNA, and from another viewpoint, a chromosome is an encompassing collection of genes. In other words, the meanings above convey that a chromosome is a segmented form of DNA, that each segment is a specific gene (as seen in Figure 5-2).

In the data model, the chromosome is a data type, composed of a bitarray of order $n \times m$ (which n and m are the number of floors and service points correspondingly and represent a segmented form of DNA by rows and columns) and some information fields which indicate the generation

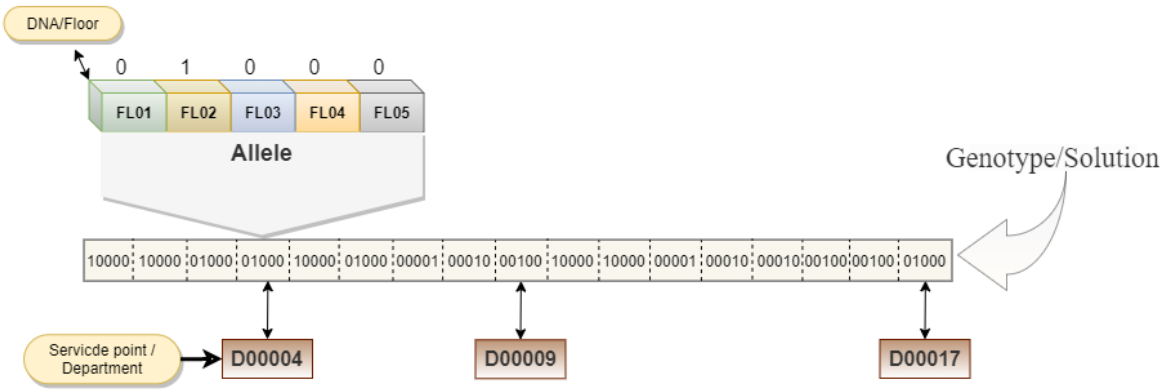
number, corresponding fitness value, a numerical field that indicates whether the chromosome generated by a crossover or a mutation, and the number of iterations used to reproduce the chromosome. Each bit in the bitarray specifies whether or not a specific floor accommodates a particular service point by its row and column indices.

The sequence of genes in a chromosome is meaningful. The index of a gene corresponding to the column number in the matrix structure is the primary key to interrelate it to its affiliated service point.

8. Genotype

A genotype is the ordered sequence of alleles. One may say, from another perspective, that a genotype is an instance of a chromosome filled in by alleles. Thus, each genotype suggests an FLP feasible solution in this architecture, including a service point assignment plan throughout the floors.

Figure 5-3, the schema that describes how the genotypes are constructed.



9. Chromosome vs. Genotype

Both are a representation of a set of DNAs from two aspects. A chromosome is a structural rendering for DNA, i.e., a segmented form of DNA, to depict which gene is accountable for what trait. On the other side, a genotype is an instant of DNA structurally instantiated based on the chromosome and filled in with alleles. Therefore, one can say that the chromosome functions as a

template for the instantiation of the genotype. Figure 5-2 and Figure 5-3 have an intuitive explanation for their difference in definition and the relationship in between.

Figure 5-4, schematic view from a Bitarray that is embedded in a genotype and contains an FLP solution.

Floor ($k = 1 : K$)	Service Points ($i = 1 : N$)																
	1	1	0	0	1	0	0	$x_{18}=0$	0	1	1	0	0	0	0	0	0
	$x_{21}=0$	$x_{22}=0$	$x_{23}=1$	$x_{24}=1$	$x_{25}=0$	$x_{26}=1$	$x_{27}=0$	$x_{28}=0$	$x_{29}=0$	$x_{210}=0$	$x_{211}=0$	$x_{212}=0$	$x_{213}=0$	$x_{214}=0$	$x_{215}=0$	$x_{216}=0$	$x_{217}=1$
	0	0	0	0	0	0	0	$x_{38}=0$	1	0	0	0	0	0	1	1	0
	0	0	0	0	0	0	0	$x_{48}=1$	0	0	0	0	1	1	0	0	0
	0	0	0	0	0	0	1	$x_{58}=0$	0	0	0	1	0	0	0	0	0

10. Diploid

An object type that contains a couple of genotypes.

11. Genome

An object type that contains a collection of *n* genes. This solution architecture uses the genomes to represent the population or the yields of crossover or mutation operations.

12. Fitness Value

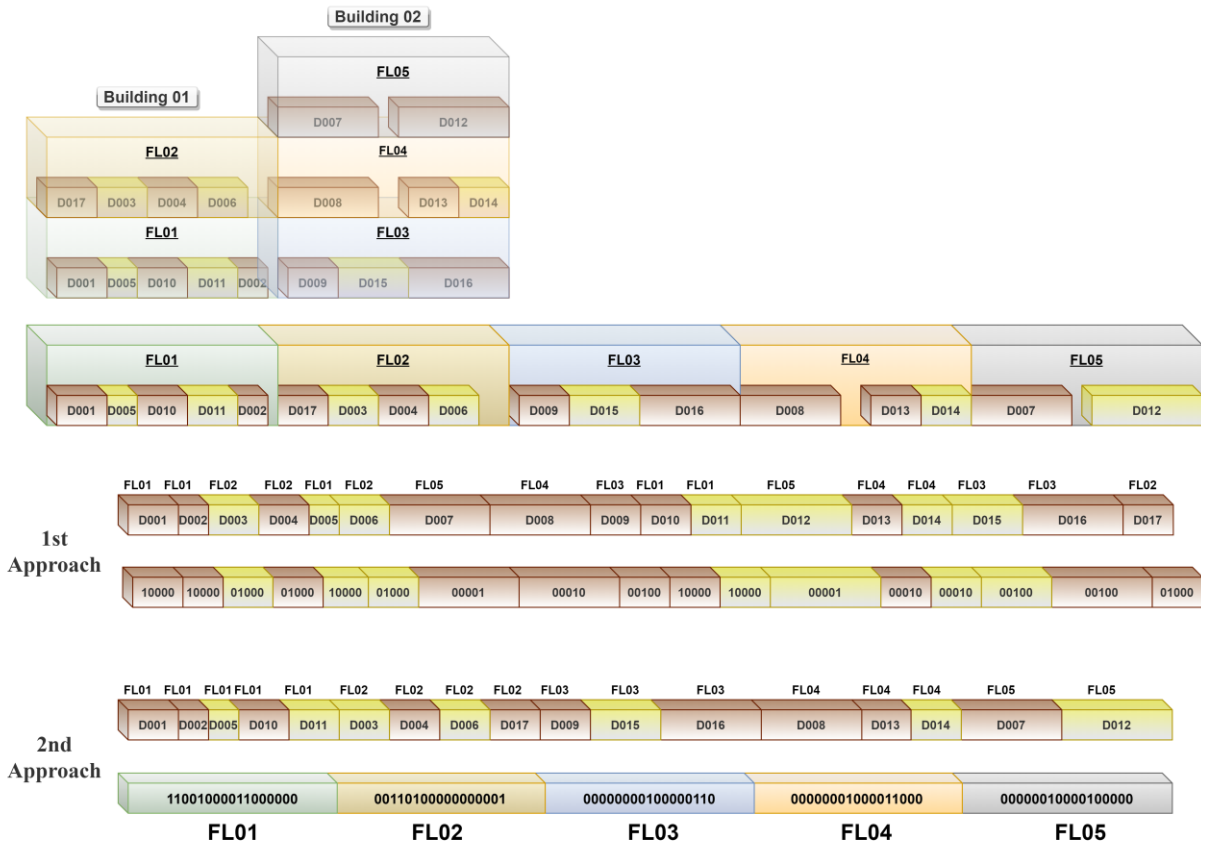
The optimization process resembles a race in which genotypes compete for survival over generations. The fitness value is the key indicator for each genotype to prove its competence to win the competition against the others, whether selected randomly or rule-based. The fitness value for each genotype is the outcome of evaluating the objective function of the FLP model with the solution contained in the genotype. The genotype embeds a numerical field to maintain the fitness value, calculated at the instantiation time in the data model.

5.2 GA Representation of the FLP Problem

As explained in chapter 4, the decision variable in the FLP mathematical model formulation has two dimensions of i and j that correspondingly associate with the service points and the floors. Given that we have decided to develop a heuristic optimizer based on a Genetic Algorithm, the first step is to reformulate the FLP model as though being applicable in the GA algorithm. Depending on how the decision variable is associated with the DNA and how the segments of DNA make up the gene structure, there are two general approaches to formulate the FLP model, as illustrated in Figure 5-5.

Development considerations have motivated the implementation phase to use the first approach of representation of the FLP model. (as seen in Figure 5-5). Except for one, all of the bitflags in a

Figure 5-5 The approaches to GA representation of the FLP model.

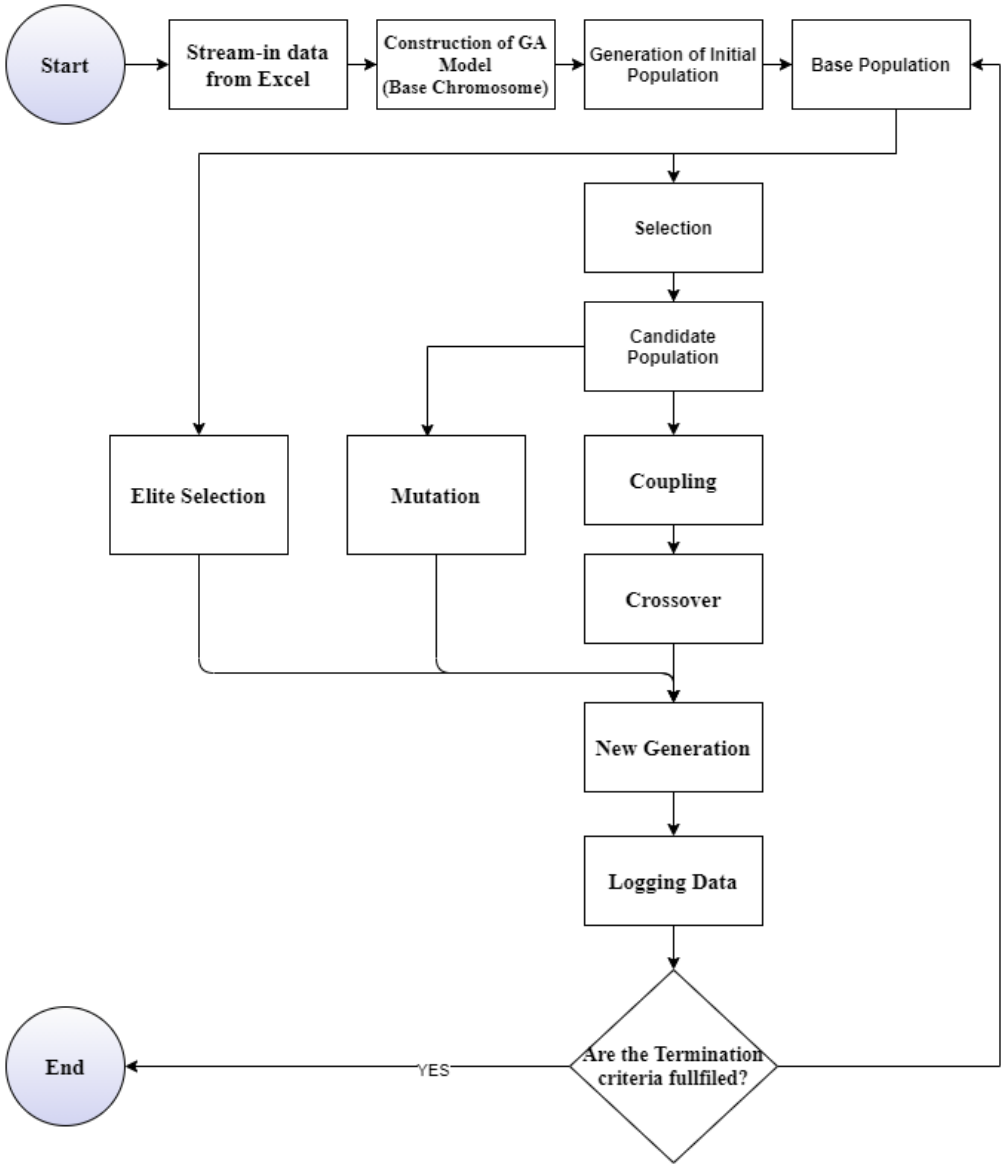


gene vector would be set to False to satisfy the constraint of the unique assignment of the service points to the floors. This feature of the first representation approach allows the implementation phase to integrate the evaluation of unique assignment constraints with the instantiation of Allele. Using this initiative improves the efficiency optimization process during the execution.

5.3 Detail Design

This optimization process is an evolutionary mechanism that approaches the optimal solution by adopting a biologically inspired logic, namely genetic inheritance, via generation transition. The process begins with the initial population, conducts an iterative optimum searching process, and meets the termination criteria. The application of developed GA operators, including *Selection*, *Elitism*, Coupling, *Crossover*, and *Mutation*, contributes to maintaining convergence and evolutionary amelioration during the searching process. Figure 5-6 illustrates a GA-based procedure adopted by this optimization framework to find the closest best-fit solution to the optimum. This routine can undergo modifications during the deployment step to enhance process efficiency by adding functionality to the operators, adding new operators, or using interoperability between the operators. This section describes the process in detail.

Figure 5-6, GA flow process chart.



5.3.1 Initial Population

This is the starting point of the GA optimization process that launches with streaming the data from the Excel workbook and establishing the data structure in the model layer after validating data coherence, i.e., the consistency among the constraints and just-in-case data type conversion in case.

It continues with iteratively producing feasible genotypes on a random-based algorithm. This module delivers the initial population in a genome, a data collection type containing N (number of initial population) feasible solution to the optimizer layer to lunch the iterative optimization sub-procedure. Figure B- 1, in Appendix B, illustrates the initial population process intuitively in detail. The following section explains the detail of how the initial population module validates the feasibility of the genotypes.

5.3.2 Enactment of Constraints

In the optimization framework, although the GA-based optimizer adopts and evaluates the constraints that differ from the mathematical programming, both approaches support the same types of constraints and provide a common definition.

This optimization framework consistently evaluates the constraints adopted to the GA model to verify how the solutions embedded in the genotypes respect the constraints. There are two distinct approaches to the constraints evaluation: explicit or implicit, depending upon when, construction level or process-oriented, and how, definitive or partially, the framework conducts the evaluations. The following subsections explain these two approaches in detail.

5.3.2.1 Explicit Evaluation of Constraints:

In this approach, the optimization framework strictly validates DNA, corresponding to an FLP solution, against all constraints set as explicitly evaluated at the construction step before incorporating it into genotype instantiation. This aspect of constraint evaluation is also referred to as binary evaluated constraint since it has a double-edged definite outcome, indicating whether or not the examined solution violates any of the constraints. When the explicit evaluation finds no evidence of a violation of any of the explicitly evaluated constraints by the examined solution, it returns it as a feasible solution to be contained by a genotype object. Otherwise, the function discards the evaluated solution returns an empty object (i.e., empty value).

This architecture employs this approach to evaluate the constraint types, including the unique assignment of service points to the floors (UQAS), the assignment of the service points to a limited number of floors (RAC), and the enforcement adjacency of the service points (AJC).

5.3.2.2 Implicit Evaluation of Constraints:

In some cases of FLP problems, the constraints narrow the feasible area to the extent that the optimization run-time extends extremely owing to the extensive number of iterations needed to instantiate a genotype with an assured feasible solution. This optimization architecture introduces the approach of implicit evaluation of constraints to alleviate this problem. In this way, the constraints evaluation function relaxes any constraint set as implicitly evaluated from the GA model and overlooks the construction level evaluation for that constraint. In place of construction level evaluation, the optimization framework penalizes the solution's fitness value for each constraint violation via the fitness function. Given that the elitism aspect of the GA-based optimization algorithm consistently maintains the better-fit solutions in each optimal searching iteration to suggest at optimization completion, the best-fit solutions with a lower penalty have a better probability of becoming the candidate for the closest-to-optimal solution. This aspect of the constraints evaluation implies that evaluating a solution's accordance with an implicitly evaluated constraint is process-oriented. The optimization process suggests the best-fit solution with the minor penalty ascribed to constraint violations at the end of the optimum searching process.

Additionally, the evaluation result of a solution against an implicitly evaluated constraint is non-definite. When the penalty corresponding to the violation of a constraint in the solution's fitness value is zero, the solution maintains maximum respect to the constraint, which is progressively and differentially attained throughout the optimization process. This framework also refers to these restrictions as FUZZY-evaluated constraints from this standpoint. The floor area constraints and the adjacency constraints can be either set as implicitly or explicitly evaluated constraints, and the result of the optimization, obviously, would not be the same. Chapter 6 intuitively demonstrates this fact using an optimization designed experiment.

This measure gives the initiative to the decision-maker to determine an over-assignment allowance rate for each floor to let the optimizer assign the service points to a floor more than its accessible area. As you change this strategy to explicit feasibility validation by letting the over-assignment allowance rate to zero, you can relax a constraint of this type for a floor by setting the over-assignment allowance rate to a relatively big value. One may perhaps encounter the ambiguity of whether the presence of infeasible solutions among the feasible area would not be harmful to the optimization process. The solution architecture responds to this question by embedding the over-

assignment cost to the fitness function and letting the user set the over-assignment cost per unit. In this way, adding the over-assignment cost to the fitness value of an over-assigned solution increases its chance of being discarded by the selection operator at the beginning of each round of optimization.

5.3.3 Explicitly Evaluation versus Implicitly Evaluation

Explicit or implicit evaluation: which is better? When both are viable while adopting the constraints into a GA model, this is an essential topic to consider. To provide a compelling answer to this question, this section compares these two approaches from different angles as below:

i. In explicit evaluation, the framework assures the effectiveness of a constraint over the feasible area via validating all solutions against it. In contrast, in implicit evaluation, the optimization process is responsible for maximizing the adherence of the best-fit solution to an explicitly evaluated constraint (FUZZY-evaluated constraint) through minimizing the corresponding penalty as a part of minimization of the solution's fitness value. Therefore, in this approach, the framework exempts every generated solution from being evaluated against the explicitly evaluated constraints, resulting in a significant decline in computational time.

ii. GA optimum searching algorithm is an evolutionary optimization based on minimizing pairwise travel costs between service points. In such a paradigm, the more distinct permutations of service points exposed to the GA operators in each search iteration, such as crossover and mutation, the higher the optimization efficacy may be expected. There is a considerable risk of overlooking some of the permutations of service points in each optimization iteration in explicit evaluation. The feasible area is restricted to the solutions with no clue of any constraint violation, resulting in an adverse effect on the optimization efficiency.

iii. In some GA optimization experiments, the feasible area is insufficient to provide any feasible solution, e.g., a lack of convergence. With some constraints set to implicit evaluation, when possible, the optimization mechanism can provide the best-fit solution with the maximum adherence to the constraints, i.e., the minor penalty for violation of the constraints in fitness value.

iv. The optimization process efficacy may directly impact the extent to which the best-fit solution adheres to an implicitly assessed constraint, given that the implicit evaluation is process-oriented. When, for example, the optimal searching method fails to introduce a global best-fit solution, e.g., due to being trapped in an immature convergence, the confidence in the best-fit solution's maximum adherence to the constraint is likely jeopardized.

5.3.4 GA Operator: Selection

Even though the selection step seems to be undervalued in the evolutionary GA-based optimization process, it has been demonstrated to be crucial to the convergence time and the efficiency of the optimization process. Adopting an appropriate selection algorithm can ensure that the optimization process avoids premature convergence. Numerous studies characterize the tournament selection method by its merits over other selection strategies, most notably, low complexity in computational time, the potentiality of being multi-threadable and parallelizable in implementation, more resistance to the predominance of dominant individuals, no need for integrating computationally-intensive or resource-demanding algorithms for sorting or scaling the fitness values(Shukla et al., 2015). The abovementioned advantages are sufficiently persuasive to opt for the tournament selection method in implementing the GA selection operator.

Although binary tournament selection can minimize the selective pressure as an effective initiative to prevent early convergence, the tournament size in the developed selection operator is configurable, which allows the user to conduct an n-ary tournament selection in the optimization process.

5.3.5 GA Operator: Elitism

Due to the random and stochastic nature of GA-based heuristic optimization, there would be a substantial probability of losing the best solutions if the optimization algorithm did not include any reassurances to maintain the best solutions in successive generation transitions. The Elitism operator in this architecture secures the presence of the best solutions from former generations in each optimization rounds. The Elitism Rate parameter of this operator sets the percentage of best-fit genotypes to survive in the next generation. Finding optimal values for this parameter is an empirical task requiring close attention. A high rate of elitism presumably leads the optimization

track to an immature convergence. Chapter 6, Experimentations and Analysis of Results, investigates the effect of this parameter on a GA-based heuristic optimization in the FLP problem.

5.3.6 GA Operator: Coupling

Once the selection operator provides a population to participate in the GA-based optimization process, the optimizer needs instruction on forming the mating pool. The operator Coupling dictates your strategies for coupling the parents. Although the most basic one is random coupling, the optimizer can feature one or more rule-based coupling strategies that are well-fitted to the problem and make the optimization process functions more efficiently according to your experiments, too.

This optimization framework provides two operators comprising one random-based and one rule-based named Best-Worst. The random-based coupling operator pairs the parent genotypes randomly with no replacement. Meanwhile, as its name suggests, the Best-Worst operator is responsible for no-replacement coupling the best-fit genotypes sequentially from the top with the worst ones from an ordered population based on their fitness values.

This operator receives the population in a Genome data type and returns the mating pool in a Diploids data type object.

5.3.7 GA Operator: Crossover

This is the main GA operator that enables the inheritance of traits from parents to their children by selecting and dispatching a particular number of genes from each parent side to instantiate offspring's genotype. Like the other operator, the design and development of crossover operators can follow two different approaches: random-based and rule-based operators. This optimization architecture incorporates a configurable random-based Crossover operator that receives a Diploid as input and returns a Genome containing the parents (at least one of the parents) and their offspring. The concept used in the development of this operator combines the impression of partially-mapped crossover and cycle crossover, as outlined underneath. In Appendix B, Figure B-2 illustrates the crossover flow process.

5.3.7.1 Design of concept

A blend of ideas that shape up the partially-mapped and cycle-mapped crossovers, which (Umbarkar & Sheth, 2015) has briefly explained in their review on the different types of crossovers, was a reasonable ground to base the development of a new crossover operator that well-fitted to the FLP problem in this project. Initially, this Crossover operator receives a Diploid containing two genotypes, e.g., F and M, and the Crossover Rate, i.e., the percentage of the genes available to take part in the crossover, say R, and randomly marks a locus on the genotype F as a start point (i.e., P1 in Figure 5-7). According to the logic of partially-mapped crossover, this operator has to count $\lfloor R \times ng \rfloor$, ng is the total number of genes in base chromosome, genes from the start point and marks there a locus as the endpoint (i.e.: P2 in Figure 5-7), and then maps the alleles contained in this range of genes to a new instance of DNA (i.e., a base chromosome that contains a solution template). Forasmuch as the evaluation of constraints probably prevents mapping some of the alleles to a new base chromosome, determination of endpoint is impossible. Hence, alternatively,

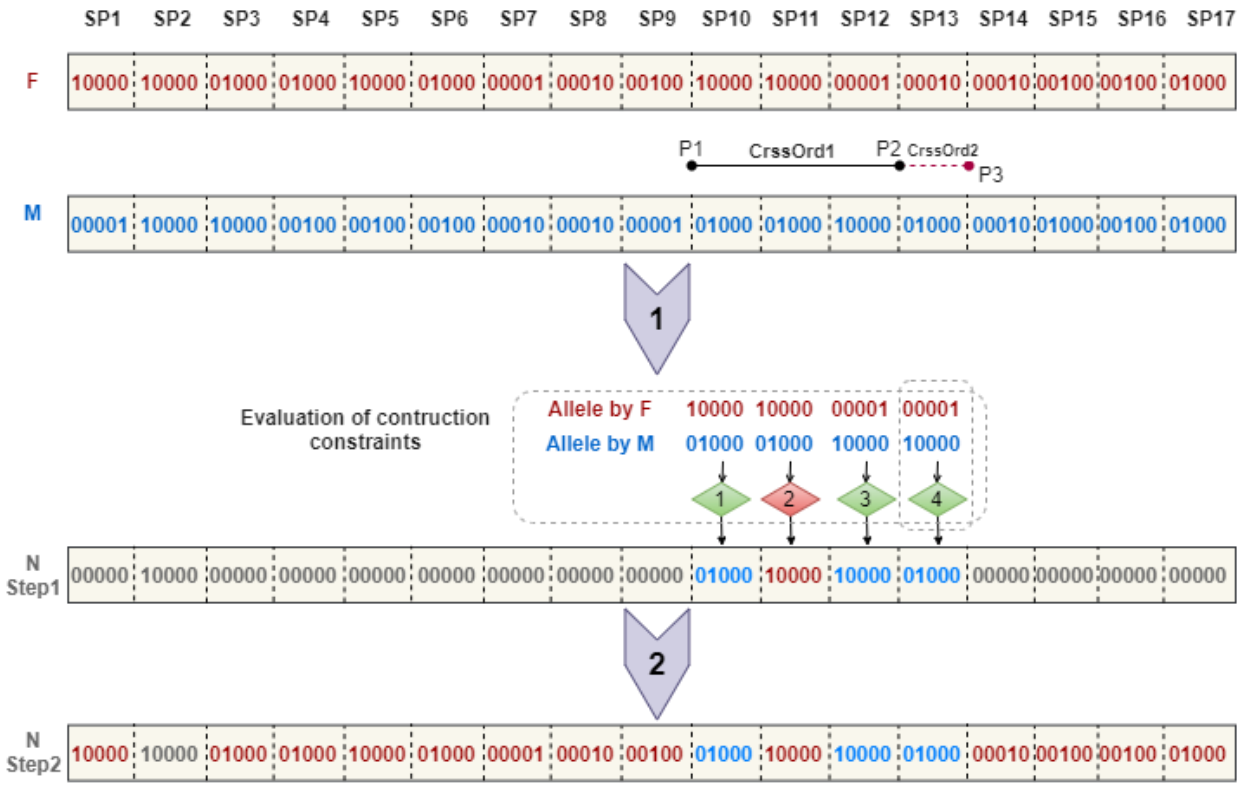


Figure 5-7, schematic view of the Crossover procedure.

the mapping of alleles has to trigger from the start point and iteratively continue till the N is reached. Figure 5-7 illustrates a crossover operation with ⌊R×ng⌋=3 as an example. The schematic view elaborates on mapping the alleles from chromosome M to the base chromosome, i.e., N; the second iteration fails due to a violation in the evaluation of construction constraints. Hence, the mapping alleles continue for one more iteration, and the endpoint advances to the P3.

In some cases, the start point is located somewhere in the genotype where the endpoint is reached before letting the mapping process fulfill the N iterations. To handle such situations, the crossover logic supposes the DNA as a circled sequence of genes that mapping process can cycle back to the beginning of the DNA to complete the N iterations. Once the mapping of alleles from genotype F to the new solution is over, the operator puts the seal on the new DNA by filling the remaining empty genes with the alleles contained in the corresponding genes in genotype M and dispatches it to the instantiation of a new genotype, namely child's genotype, after being validated against the capacity constraint, i.e., the constraint of maximum available area for each floor. Should the DNA fail to prove its compliance with the constraint, the operator discards it as flawed DNA and retries to randomly reproduce flawless DNA. The iterations for reproducing flawless DNA continue until the parameter of *MAX_ATTEMPTS_TO_FLOWLESS_DNA* is reached. This reproduction process loops again by swapping the parent genotypes at the positions F and M if the value of parameter Mode would be *dbl*.

5.3.7.2 Microbial Crossover

This operator features a crossover based on the Microbial Crossover concept, which (Harvey, 2011) claims for its better efficiency in GA-based optimization. Microbial Crossover receives a diploid, ranks the contained genotypes in the order of Winner and Loser according to their fitness values. The winner genotype has the better fitness value, given whether the fitness function objective is minimization or maximization (in this project the .less fitness value due to the minimization fitness function). After ranking the parent genotypes, the crossover algorithm proceeds with reproduction offspring complying with the routine described in except two latest steps as firstly, in microbial crossover logic, looping again with swapping the winner and loser genotypes is not rational, and secondly, the operator nullifies the loser genotype rather than embedding it in the returning Genome. Provided that microbial crossover could prove its competencies to improve the

performance of the GA-based optimization in total, it can justify the claim for being more efficient owing to two coherent reasons as firstly, having less iteration in its routine, and secondly, combining the concept of elitism in the crossover algorithm, that reduce noticeably the execution load in each optimization round.

5.3.7.3 Random Crossover Rate

To generate a new DNA, as mentioned in 5.3.7.1, the crossover operator needs a required parameter, namely *Crossover Rate* that is limited to the interval bounded between zero and one, to calculate the number of candidate service points that must be involved in the mapping iterations. Having Crossover Rate set to a single value, e.g., 0.7, probably demonstrates an acceptable efficiency during the starting optimization rounds. The solutions need more changes by making genes crossover among the genotypes. In the meantime, the more the optimization process progresses towards the optima, the fewer changes the solutions need to occur in the assignment plan. Indeed, while the optimum searching process gets closer to the optimal point, a high crossover rate gradually may lose its efficiency. To mitigate this inefficiency, the crossover operator is geared to accept an interval of 0 to 1 as the crossover rate and picks a random value per each function call.

Chapter 6, Experimentations and Analysis of Results, presents an analysis of the GA-based optimization for the FLP problem in this project under the conditions described in the sections above.

5.3.8 GA Operator: Mutation

Unlike the crossover operator that attempts to propagate the well-fitted traits among the population through heredity, the mutation operator is mainly responsible for maintaining the population diversity via modification of one genotype. Nevertheless, the mutation operator can randomly reproduce a better-fitted genotype, which implies a better solution. A well-designed mutation algorithm that functions efficiently could diminish the risk of immature convergence and prevent the optimization process from being trapped in a local optimum. The following sections explore the detail of the developed mutation operators and their algorithms.

1. Mutation Operator: Floor Swap

This operator receives a genotype and randomly picks two genes, i.e., two service points, up from the ones available to mutation. Then, it simply checks out the possibility of swapping the floors that both service points are currently located on via evaluating the validity of the constraints against this mutation. If the mutation is valid, it instantiates a DNA type based on the initial genotype's DNA and swaps the alleles in those genes to put the floors swap in practice. Otherwise, the program execution control keeps the first chosen gene, randomly picks another gene amongst the remaining genes in the genotype, and repeats the same process described above. Finally, supposing the operator could generate a flawless instance of DNA, it dispatches it to the instantiation of a genotype and returns the new genotype to the new generation pool. Otherwise, the operator returns an empty object type (i.e., empty value).

2. Mutation Operator: Near-or-Away

The steps that make up the procedure of this operator are exactly in common with the previous operator, as described above, except for the mutation logic. When the operator receives a genotype, there are two scenarios, depending on which floors the randomly chosen service points are located on, as:

i. , if both service points are on the same floor, the operator attempts to reallocate the second service point randomly to another floor via changing the allele contained in the corresponding gene in the genotype;

ii. Otherwise, the operator tries to reassign the second service point to the same floor as the first service point is located.

5.3.9 GA Operator: Screening

In practice, there is a significant risk that the optimization trajectory encounters a situation in which the population is classified between two groups of elite individuals and a massive population with fitness value very far from the optimum point and the vast gap in between. In such circumstances, the most efficient decision would be to terminate optimization iterations even when the optimal has not yet been achieved. This heuristic optimization tool incorporates a measure referred to as population screening makes the algorithm less vulnerable to the aforesaid unintended condition.

By first applying unity-based normalization, also termed Min-Max Feature Scaling, the screening operator maps the fitness values of the population into the range of [0,1] using the following function (where $FV_{\min}$ and FV_{max} are the minimum fitness value and maximum fitness values of the population):

$$FV_s' = \frac{FV_s - FV_{\min}}{FV_{\max} - FV_{min}}$$

Then, the screening routine proceeds with classifying the normalized range of values based on the range of [0.0, 0.25, 0.5, 0.75, 1.0]. The final step involves conducting random filtering on each class according to the filtering rate for each class and provided by the user. Integrating this operator in the optimization process promises a population with a more balanced distribution while maintaining diversity.

5.3.10 Optimization Process Termination Criteria

Figure 5-6 shows that the GA-based optimization mechanism needs criteria to terminate the iterative optimal searching when it reaches the closest best-fit solution to the optimum and cannot introduce any better solution. According to this optimization framework, for termination to occur, the following must be true:

i. $\lfloor UQR_{elites} \times P_{elites} \rfloor = 1$, where $UQR_{elites}$is uniqueness rate of elites population, and P_{elite} is the population of elites

ii. $UQR_p \leq 10\%$, where $UQR_p$is diversity percentage of candidate pool's population

iii. No better-fit solution after five consecutive iterations.

The logic behind the criteria above suggests that when the first condition occurs, convergence has taken place, while the diversity of the pool of candidates, according to the second condition, is insufficient to introduce a better solution. After five straight optimization rounds that satisfy these two conditions, the termination occurs.

5.3.11 Optimization Round Report

The GA-based heuristic optimizer provides a detailed and informative report for each optimization round, which helps track the optimization path in each iteration. This report, as illustrated in Figure 5-8, focuses on four main sections as below:

1) This section provides statistics about each population category, including base population, elites, selected population, candidates pool, and Diploids at the beginning of each optimization round. The $FTV\ Uniqueness\ Rate$ and the $Div.Rate$ of each category are the crucial data elements in this section since they provide an estimation of how diversified is the population participated in the optimization with a reasonably light computational burden. The computational complexity would make it too time-consuming to analyze the diversity of the population in depth. The FTV Uniqueness Rate explains how the fitness values that appear in the population are varied, whereas the $Div.Rate$ indicates how many distinct genotypes existed since they have been reproduced in various rounds and by different operators. These measures can also indicate how efficient the GA operators have been, especially when it comes to Initial Population, Selection, and Screening.

2) This is the most intuitive visualization provided for assessing the balanced distribution of individuals within a population. It uses the Min-Max Feature Scaling, explained in 5.3.9, to depict the classified scattering of the fitness values that appear in the population.

3) Statistical information in this part of the report and the lines under the Stat. info are other aspects of monitoring the progress of optimization processes via quantitative indicators. As seen in Figure 5-8, the report segments include the data related to the top three best-fit individuals, two worst cases, and statistics for the whole population to demonstrate local improvement within a round and to track the global improvement compared with the initial round. The fitness value is also categorized into travel cost and over-assignment cost. The presence of a data field named $Attempt.No.$ (i.e. attempt number) would be significant when you figure out which operators could confirm their effectiveness by reproducing the best-fit genotypes. In practice, $Attempt.No$ for a genotype is zero when the genotype is reproduced exclusively by mutations but otherwise by crossovers. When the $Attempt.No.$ is non-zero, a greater $Attempt.No.$ denotes that more iterations have been required to

generate the genotype. This could indicate constraints pressure on the feasible area or inconsistencies among the adjacency constraints. Reconsideration of the constraints is advised in cases of extremely frequent genotypes with high Attempt No. values.

4) This segment of the report, likewise the second part, visualizes the balance of population distribution at the end of each optimization round.

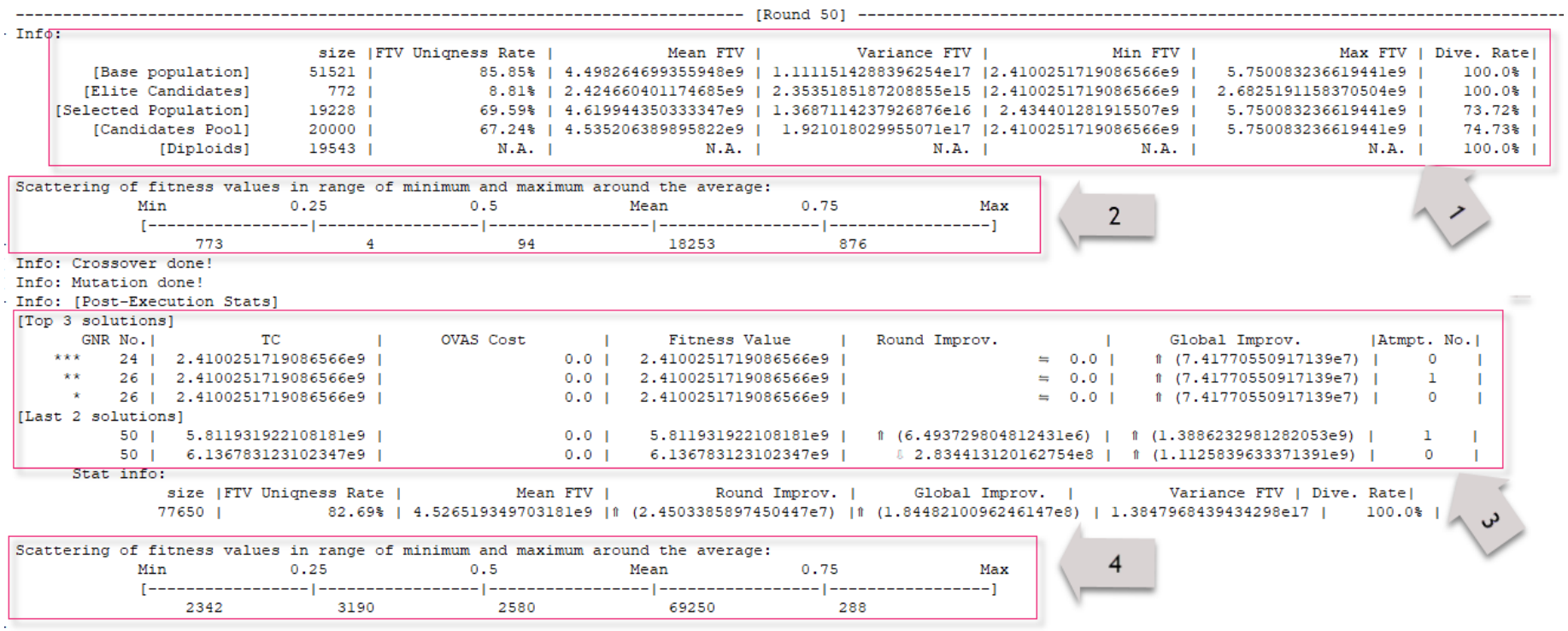
5.3.11.1 The History Log

The statistical and performance information presented in the optimization round report, which is sufficiently explained in the previous section, is registered in a collection datatype and would be streamed back to the Excel environment for more detailed post-execution analysis.

5.3.11.2 Optimization Outcome

The solution embedded in the best-fit genotype found will be streamed back to Excel for more analysis at the optimization process conclusion. Chapter 6, Experimentations and Analysis of Results, shows the analysis of results in several experimentation

Figure 5-8, Optimization round report.



```
---------------------------------------------------------------------------- [Round 50] ----------------------------------------------------------------------------
Info:
                          size |FTV Uniqness Rate |           Mean FTV |          Variance FTV |             Min FTV |             Max FTV | Dive. Rate|
      [Base population]  51521 |           85.85% | 4.498264699355948e9 | 1.1111514288396254e17 |2.4100251719086566e9 |  5.750083236619441e9 |    100.0% |
     [Elite Candidates]    772 |            8.81% | 2.424660401174685e9 | 2.3535185187208855e15 |2.4100251719086566e9 | 2.6825191158370504e9 |    100.0% |
[Selected Population]  19228 |           69.59% | 4.619944350333347e9 | 1.3687114237926876e16 | 2.434401281915507e9 |  5.750083236619441e9 |    73.72% |
     [Candidates Pool]  20000 |           67.24% | 4.535206389895822e9 |  1.921018029955071e17 |2.4100251719086566e9 |  5.750083236619441e9 |    74.73% |
             [Diploids]  19543 |             N.A. |               N.A. |                  N.A. |                N.A. |                 N.A. |    100.0% |
Scattering of fitness values in range of minimum and maximum around the average:
            Min              0.25              0.5              Mean              0.75              Max
            [-----------------|-----------------|-----------------|-----------------|-----------------]
                773               4                94               18253             876
Info: Crossover done!
Info: Mutation done!
Info: [Post-Execution Stats]
[Top 3 solutions]
      GNR No.|           TC          |      OVAS Cost       |      Fitness Value     |   Round Improv.          |      Global Improv.       |Atmpt. No.|
    ***    24 |  2.4100251719086566e9 |                  0.0 |  2.4100251719086566e9 |                   = 0.0 |    ⇑ (7.41770550917139e7) |     0    |
     **    26 |  2.4100251719086566e9 |                  0.0 |  2.4100251719086566e9 |                   = 0.0 |    ⇑ (7.41770550917139e7) |     1    |
      *    26 |  2.4100251719086566e9 |                  0.0 |  2.4100251719086566e9 |                   = 0.0 |    ⇑ (7.41770550917139e7) |     0    |
[Last 2 solutions]
          50 |   5.811931922108181e9 |                  0.0 |   5.811931922108181e9 |  ⇑ (6.493729804812431e6) |  ⇑ (1.3886232981282053e9) |     1    |
          50 |   6.136783123102347e9 |                  0.0 |   6.136783123102347e9 |  ⇓ 2.834413120162754e8 |  ⇑ (1.1125839633371391e9) |     0    |
     Stat info:
              size |FTV Uniqness Rate |           Mean FTV |          Round Improv. |      Global Improv.  |          Variance FTV | Dive. Rate|
             77650 |           82.69% | 4.526519349703181e9 |⇑ (2.4503385897450447e7) |⇑ (1.8448210096246147e8) | 1.3847968439434298e17 |    100.0% |
Scattering of fitness values in range of minimum and maximum around the average:
            Min              0.25              0.5              Mean              0.75              Max
            [-----------------|-----------------|-----------------|-----------------|-----------------]
                2342              3190             2580              69250             288
```

CHAPTER 6 EXPERIMENTATIONS AND ANALYSIS OF RESULTS

This chapter presents the result of the FLP optimization experiments designed using the models provided in Chapters 4 and 5 in two main parts: The first part conducts a preliminary study on solving the FLP MILP model using a MIP solver, namely Gurobi, and shows its performance in a real-size problem; the second part addresses a thorough analysis of the developed GA-based optimizer's performance from three perspectives:

i. Analysis of convergence in optimization process using descriptive statistical approach.

ii. Analysis of sensitivity of the GA optimization process to changes in the GA parameters such as initial and base population size, mutation rate, crossover rate, elite pool size, tournament size.

iii. Applying different types of constraints to investigate how they affect the best-fit solutions.

6.1 The Objective of the Experiments and Methodology

6.1.1 Objectives

As mentioned in chapter 2, GA-based optimization heuristics begins with a randomly constructed population and approaches the optimality through an iterative search process. The intensification of the search and diversification of the searching area plays a crucial role in GA optimization. Hence, striking a balance between intensification and diversification, or equivalently exploitation and exploration, is critical to boosting the search process towards the global optimum. When search intensification surpasses diversification, the optimization is compromised, resulting in a premature convergence to a local-optima.

In GA, the selection operator forms the new generation amongst the survivors from the previous generation. Since it provides the search process with the opportunities to revisit the search spots of earlier iterations, it contributes to the exploitation. Likewise, the elitism operator, which functions such as an adaptive memory, retains the best-fit genotypes in generation transition. Therefore, a diverse elite pool facilitates more effective exploitation. Furthermore, the crossover rate, which represents the likelihood of the genes crossing in genotypes mating, impacts the exploitation or intensification process. On the other hand, the number of mutation operators and mutation rate

contribute to process diversification by widening and breeding the search area, resulting in more efficient exploration.

The facts mentioned above show the importance of balance between intensification and diversification and how GA operators can affect this balance. Therefore, conducting an insightful sensitivity analysis that can rationalize the impacts of GA operators on the optimization process is vital first to ensure the effectiveness of the devised algorithm and second, to find out the most effective parameter configuration to obtain the best result from the optimizer.

Furthermore, several experiments are designed to ensure the optimizer's performance in applying the constraints to the FLP and analyze its effects on the best-fit solution.

6.1.2 Methodology

A GA metaheuristic optimization algorithm's random intrinsic basis provides unique experiences for each optimization. Therefore, analyzing the empirical data at a micro-level is complicated. In place, adopting a macro-perspective approach using descriptive statistical metrics can facilitate having a holistic sensitivity analysis.

i. Demographic diversity analysis chart

This is a chart with three-axis: the primary vertical axis represents the fitness value, the secondary vertical axis represents the population diversity, and the horizontal axis represents the optimization process timeline. This chart studies the trends in the best-fit genotype’s fitness value and concerning population diversity.

ii. Process convergence analysis

This is a chart with three-axis: the primary vertical axis represents fitness value, the secondary vertical axis represents the variance of fitness values in each generation, and the horizontal axis represents the optimization process timeline. This chart studies the trends in the fitness value for best-fit genotype, worst-fit genotype, and mean of the fitness values along with the variance of fitness values. Observing this chart combined with the demographic diversity analysis chart can illustrate how the GA optimization converges to either local-optima or a global near-optimum solution.

iii. Demographic composition heat map table

This is an 8-column table. The first column shows the iteration number of optimization. Columns 2-9 shows the distribution of genotypes in the ranges of $\mu\pm\sigma$, $\mu\pm2\sigma$, $\mu\pm3\sigma$, greater than $\mu+3\sigma$, and less than $\mu-3\sigma$, based on their fitness values. These columns have meshed with yellow to red gradient colors, in which light yellow represents a low-dense area, and the solid red represents a high-dense area. The 10^{th} and 11^{th} columns contain the excess kurtosis and skewness for the corresponding iteration to their row. This is an efficient statistical analysis tool to monitor population diversification and assess the process exploration effectiveness. This tool can also show deterioration in demographic composition affects optimization effectiveness in approaching optimality.

iv. Analysis of demographic fitness to normal distribution

This analysis tool contains ten charts representing the fitness of genotypes fitness values corresponding to one specific iteration to the normal distribution curve across the optimization with the interval of ten. The fitness values in each chart are fitted to a standard normal curve using the mean value and standard deviation of the base population pertinent to one specific optimization iteration. This analysis tool intuitively illustrates that base population fitness may relate to process intensification, diversification, and convergence.

6.1.3 Design of Experiment

6.1.3.1 The creation of an instance of FLP

A real-life project needs actual and up-to-date data to provide a practical solution. Real data collection is a time-consuming process requiring sufficient infrastructure, and it does not fit into the project's timeline. The FLP for conducting the experiments in this section was mocked up using an architectural plan pertinent to an emergency hospital to avoid the data collection challenge. The file containing the CAD format of this architectural plan is obtained from the internet. Despite the size of this project falling into the small to medium FLPs, it adequately meets the optimizer's validation requirements. Furthermore, the computation efforts to optimize a project of this size allow us to carry out a planned number of experiments in the project's time frame. Table 3-1 presents the general specifications extracted from the CAD file mentioned

above, as well as Table 3-2 and Table A- 1, list of departments and service points and their space requirements

Departments			Service Points				
Description	**Code**	**ID**	**Description**	**Code**	**ID**	**Area (Min)**	**Area (Max)**
Surgery Clinic	D00117	SG011	Operating room for major surgeries	D00021	SG001	34	37
			Operating room for minor surgeries	D00022	SG002	34	36
			Preparation & anesthesia room for major surgery	D00023	SG003	21	23
			Preparation & anesthesia room for minor surgery	D00024	SG004	21	23
			Sterilization room for operating room	D00025	SG005	17	20
			Recovery room	D00026	SG006	17	20
			Post-anesthesia care unit 1	D00027	SG007	12	14.5
			Post-anesthesia care unit 2	D00028	SG008	12	14.5
			Post-anesthesia care unit 3	D00029	SG009	12	14.5
			Post-anesthesia care unit 4	D00030	SG010	12	14.5
Outpatient Clinic	D00116	OC007	Doctors' office 1	D00036	AD004	20	22
			Waiting room	D00037	GS015	10	12
			Staff lounge in the outpatient clinic	D00038	AD005	12	14
			Inpatient emergency caring room	D00039	OC001	45	49
			Outpatient clinic room 1	D00040	OC002	14	16
			Outpatient clinic room 2	D00041	OC003	14	16
			Admission office	D00043	OC005`	7	9.5
			Waiting room (outpatient clinic)	D00044	GS016	10	11
			Stretchers room	D00045	OC006	10	11
Medical Imaging Test clinic	D00118	DC009	Echocardiography	D00046	MI001	20	23
			X-Ray	D00047	MI002	25	31
			The waiting room for the MI department	D00048	MI003	35	37
			Doctor's office in MI department	D00049	MI004	15	18
			Staff lounge	D00050	MI005	10	11
			Reception office in MI department	D00051	MI006	20	23
			MRI	D00145	MI007	30	33.5
Specialized Clinic	D00119	SC012	Registration & admission office	D00058	AD010	12	15
			Outpatient surgery room 1	D00059	SC001	16	19
			Outpatient surgery room 2	D00060	SC002	16	19
			Waiting room 1 at SC	D00061	SC003	15	17
			Waiting room 2 at SC	D00062	SC004	15	17
			Gynecology & obstetrics doctor office 1	D00063	SC005	15	17
			Gynecology & obstetrics doctor office 2	D00064	SC006	15	17
			Orthopedic doctor office	D00065	SC007	15	17
			Neurology doctor office	D00066	SC008	15	17
			Cardiology doctor office	D00067	SC009	15	17
			Hematology doctor office	D00068	SC010	15	17

Continued next page

Table A- 2, which provide geospatial data for the FLP model.

6.1.3.2 Design of experiments planning and implementation

This phase of the project, namely the design of experiments, plans 17 different experiments to conduct a thorough sensitivity of analysis of GA-based optimization and investigate whether the optimizer effectively applies all type constraints to the best-fit solutions. Table 6-2 presents a summary of planned experiments and the results in brief. The following sections in this chapter provide a detailed analysis of each experiment.

In a nutshell, the analysis of results of the experiments reveals the following facts about the developed GA-based optimizer, which all of them are in accordance with the literature:

i. The GA operators, including selection, elitism, and crossover, contribute to the exploitation aspect of the GA-based optimal search process, which boosts search intensification.

ii. The mutation operator contributes to the exploration aspect of the GA-based optimal search process, which impacts the diversification of the search area.

iii. The balance between search intensification and diversification of the search area is crucial in the optimizer’s performance.

iv. The developed GA-based optimization algorithm's performance may not be adequate to balance the intensification and diversification throughout the optimization process, resulting in a tendency to prematurely converge to a local-optima.

v. The optimizer successfully applies all types of constraints to the best-fit solution.

This study begins carrying out the first experiment, i.e., BSE-3000-2000, with the following conditions and parameters configuration:

i. The unique assignment constraints and the floor area availability constraints are the only constraints used in the GA optimization. The remaining constraints, including the restricted assignment and adjacency constraints, are relaxed for this step.

The service points involved in this optimum search are as listed in Table A- 1, list of departments and service points and their space requirements

Departments			Service Points				
Description	**Code**	**ID**	**Description**	**Code**	**ID**	**Area (Min)**	**Area (Max)**
Surgery Clinic	D00117	SG011	Operating room for major surgeries	D00021	SG001	34	37
			Operating room for minor surgeries	D00022	SG002	34	36
			Preparation & anesthesia room for major surgery	D00023	SG003	21	23
			Preparation & anesthesia room for minor surgery	D00024	SG004	21	23
			Sterilization room for operating room	D00025	SG005	17	20
			Recovery room	D00026	SG006	17	20
			Post-anesthesia care unit 1	D00027	SG007	12	14.5
			Post-anesthesia care unit 2	D00028	SG008	12	14.5
			Post-anesthesia care unit 3	D00029	SG009	12	14.5
			Post-anesthesia care unit 4	D00030	SG010	12	14.5
Outpatient Clinic	D00116	OC007	Doctors' office 1	D00036	AD004	20	22
			Waiting room	D00037	GS015	10	12
			Staff lounge in the outpatient clinic	D00038	AD005	12	14
			Inpatient emergency caring room	D00039	OC001	45	49
			Outpatient clinic room 1	D00040	OC002	14	16
			Outpatient clinic room 2	D00041	OC003	14	16
			Admission office	D00043	OC005`	7	9.5
			Waiting room (outpatient clinic)	D00044	GS016	10	11
			Stretchers room	D00045	OC006	10	11
Medical Imaging Test clinic	D00118	DC009	Echocardiography	D00046	MI001	20	23
			X-Ray	D00047	MI002	25	31
			The waiting room for the MI department	D00048	MI003	35	37
			Doctor's office in MI department	D00049	MI004	15	18
			Staff lounge	D00050	MI005	10	11
			Reception office in MI department	D00051	MI006	20	23
			MRI	D00145	MI007	30	33.5
Specialized Clinic	D00119	SC012	Registration & admission office	D00058	AD010	12	15
			Outpatient surgery room 1	D00059	SC001	16	19
			Outpatient surgery room 2	D00060	SC002	16	19
			Waiting room 1 at SC	D00061	SC003	15	17
			Waiting room 2 at SC	D00062	SC004	15	17
			Gynecology & obstetrics doctor office 1	D00063	SC005	15	17
			Gynecology & obstetrics doctor office 2	D00064	SC006	15	17
			Orthopedic doctor office	D00065	SC007	15	17
			Neurology doctor office	D00066	SC008	15	17
			Cardiology doctor office	D00067	SC009	15	17
			Hematology doctor office	D00068	SC010	15	17

Continued next page

Table 6-1, optimizer parameters configuration table.

Parameter Name	Value	Description
Experiment ID=	BSE-3000-2000	
Number of service points	60	
Number of floors	8	
Over-assignment allowed	FALSE	
Strategy for departments area	Min Area	
Strategy for floor area availability	Min Area	
Horizontal transport cost	1 unit	
Vertical transport cost	10 unit	
Initial population	3000	
Genotype max attempts	5	Maximum attempts to generate a feasible genotype.
Selected population in each round	2000	
Max. round to create initial population	5000	Maximum round to generate the initial population
Tournament size	2	
Elite rate	0.01	1% of base population in each round
Crossover rate	0.6	60% of first parent DNA with 40% of second parent
Mutation rate	0.3	30% of candidate population are mutated each round
Crossover mode	Double	
Population screening rates	N.A.	The population screening operator does not apply to the optimization process.
Floor over-assignment rates	N.A.	

ii. Table A- 2and the flows between the service points are as demonstrated in Table A- 4.

iii. Table 6-1 shows the configuration of the GA parameters for the optimizer in this optimization performance:

iv. Figure 5-6 illustrates the GA-based algorithm adopted by this optimization framework to use in the experiments designed for this chapter.

After termination of the optimization process, the validity of the result is verified by evaluating the optimal or near-optimal solution against the constraints. Then options to enhance the process efficiency are explored via analyzing the log data, reconfiguring the GA parameters, or modifying the algorithm by adding GA operators or changing the order in which the GA operators are applied. In each experiment the changes according to the experiment plan are made to the conditions and configuration described above, considered as the base experiment, and is applied to optimization case accordingly.

6.1.3.3 Definitions

- **Initial population**: the set of individuals randomly generated by population operator at the beginning of the GA optimization. Initial population occurs exclusively in first iteration of optimization.
- **Base population**: a set of individual, i.e., genotypes, which are selected by select operator and are maintained in candidate pool.
- **Optimization round/optimization iteration**: each search iteration that leads to forming a new generation that acts as initial population for next optimization iteration.
- **Empirical rule (68,-95-99 rule)**: this statistical rule states that: 99.7% of the observations obtained from an experiment that follows normal distribution fall into the area of $\mu \pm 3\sigma$; 95% of the observations are distributed across the area within $\mu \pm 2\sigma$; and 68% of the observations fall into the area within $\mu \pm \sigma$. (source)

Table 6-2, design of experimentations summary report.

No.	Experiment ID	Objective	Change	Num. of Itr.
1	PPL-3000-2000	• Ensure effective applying of constraints • Analysis of convergence • Analysis of GA operators effectiveness	Base configuration	100
	Results : • Optimizer effective apply the constraints UQAC and FAAC to the solution • The search process converges to a near-optimum solution • Crossover generates 58 better-fit genotype • Mutation generates four better-fit genotype • Optimization suffers from a lack of population diversity and intensification seems to surpass diversification			
2	PPL-1000-6000	Analysis of increase in the initial and base population	Initial ppl = 10000 Base ppl = 6000	100
	Results : • Proposes a new better-fit genotype with 7.61% improvement in fitness value • No observations of any notable improvement search process behaviors			
3	MTN60-3000-2000	Analysis of increase in mutation rate	Mtn rate = 0.60	100
	Results : • Proposes a new better-fit genotype with 15,23% improvement in fitness value • Improves the population diversity			
4	MTN100-3000-2000	Analysis of increase in mutation rate	Mtn rate = 1.00	100
	Results : • Proposes a new better-fit genotype with 18,20% improvement in fitness value • Improves the population diversity compared to both base experiment and MTN60-3000-2000			
5	ELT05-3000-2000	Analysis of increase in elitism rate	Elite rate=0,005	100
	Results : • Proposes a new better-fit genotype with 5.55% improvement in fitness value • The optimization algorithm is more resistant to population diversity degradation			
6	ELT20-3000-2000	Analysis of increase in elitism rate	Elite rate=0,02	100
	Results : • Proposes a new better-fit genotype with 12.31% improvement in fitness value • The optimization algorithm is more resistant to population diversity degradation compared to the base experiment			
7	ELT30-3000-2000	Analysis of increase in elitism rate	Elite rate=0,03	100
	Results : • Proposes a new better-fit genotype with 3.75% improvement in fitness value compared to the base experiment, but worse than two experiments: ELT05-3000-2000 and ELT30-3000-2000. • The worst population diversity compared to all previous experiments and the optimization most likely converges to local optima.			
8	CSO50-3000-2000	Analysis of decrease in crossover rate	Crossover rate=0.5	100
	Results : • Proposes a new better-fit genotype with a 7.00% improvement in fitness value compared to the base experiment • The optimization becomes very resistant to population diversity degradation • Improvement in the balance between intensification and diversification			

Continued on next page.

Continued from previous page.

No.	Experiment ID	Objective	Change	Num. of Itr.
9	CSO70-3000-2000	Analysis of increase in crossover rate	Crossover rate=0.7	100
	Results : • Proposes a new better-fit genotype with a 10.29% improvement in fitness value compared to the base experiment • No witness of improved population diversity			
10	CSORDM-3000-2000	Analysis of using random crossover	Crossover rate=[0.5-0.7]	100
	Results : • Proposes a new better-fit genotype with a 2,68% improvement in fitness value compared to the base experiment • The optimization becomes more resistant to population diversity compared to base experiment and CSO70-3000-2000, but less than CSO50-3000-2000.			
11	TNS01-3000-2000	Analysis of decrease in tournament size	TRN size=1	100
	Results : • Proposes a best-fit solution with a fitness value 32.42% worse than the base experiment • More diverse population • The search process approaches optimality at a significantly slower pace			
12	TNS03-3000-2000	Analysis of increase in tournament size	TRN size=3	100
	Results : • Proposes a best-fit solution with a fitness value 7.68% better than the base experiment • Slightly more resistant to population diversity degradation • The search process approaches optimality at a significantly faster pace			
13	RAC-3000-2000	Effectively applying RAC constraints	--	50
	Results : • Investigating the assignment table resulted by best-fit solution shows that RACs effectively applied.			
14	AJC01-3000-2000	Effectively applying binary adjacency constraints	--	
	Results : • Investigating the assignment table resulted by best-fit solution shows that defined binary adjacency constraints are effectively applied.			
15	AJC02-3000-2000	Effectively applying fuzzy adjacency constraints to enforce the proximities.	--	
	Results : • Investigating the assignment table resulted by best-fit solution shows that defined fuzzy adjacency constraints are effectively applied.			
16	OVAS01-3000-2000	Effectively applying the relaxation of LAC constraints.	--	
	Results : • Investigating the assignment table resulted by best-fit solution shows that the relaxation of floors' limited available area constraints are effectively applied.			
17	OVAS02-3000-2000	To demonstrate the improvement in optimization performance caused by relaxation of LACs.	--	
	Results : • The relaxation of the LACs are effectively applied to optimization and make the initial population generation 28 times faster, as well as 201 sec reduction in each optimization iteration.			

6.2 Mixed Integer Linear Programming

This section presents the result of solving the linearized FLP model for the current project demonstrated in 4.2.9 using Gurobi solver. In this first experience with the MILP optimization, the model is not subjected to any supplementary constraints, including restricted assignment constraints and adjacency constraints. The optimization run was intentionally interrupted after 48 hours, and the gap size of 68.9. It is important to note that the subject of this experiment is an FLP with 60 service points and eight floors. In this problem, the run-time exponentially escalates with every slight increase in the number of service points or floors. The result confirms that an FLP QSAP model is an np-hard problem and consequently does not yield an optimal solution in polynomial time despite linearization and the use of a MILP solver.

6.3 Genetic Evolutionary Algorithm-Based Heuristic Optimization

6.3.1 Verification of the Effectiveness of the Constraints

The efficacy of the optimizer in maintaining the constraints is verified through evaluating them against the best solutions exported to an Excel workbook at the optimization process termination time. Table 6-3 explains that the floor available area constraints are well-respected, and none of the floors are over-assigned. The analysis of the best-fit assignment table demonstrates that the unique assignment constraints have been effectively applied once 100 rounds of optimization have been completed. In each of the two best solutions, the optimizer has successfully assigned each service points to only one floor at a time, as seen in the last column of each table accordingly.

Table 6-3, Unique assignment constraints verification for the first best solution after 120 rounds of optimization

First best solution after 100th optimization round.					
Code	**ID**	**Description**	**Min Area**	**Min Area Assigned**	**Over-Assigned?**
F01	NBBS	Basement, the north building	550	230	FALSE
F02	NBGF	Ground floor, north building	535	530	FALSE
F06	SBFF	First floor, south building	760	650	FALSE
F03	NBFF	First floor, north building	780	700	FALSE
F04	SBBS	Basement, south building	600	532	FALSE
F05	SBGF	Ground floor, south building	980	966	FALSE
F07	MBGF	Ground floor, middle building	550	543	FALSE
F08	MBFF	First floor, the middle building	110	103	FALSE

6.3.2 Analysis of Optimization Report

Figure C- 1 and Figure C- 2 depict the round summary report (RSR) for the first and last rounds of the experiment BSE-3000-2000 with the data and parameter configuration described in the previous section. The optimizer spent about 688 minutes completing 100 rounds of optimization. The first RSR (Figure C- 1) indicates that the initial population operator has successfully established a base population. According to the pre-execution statistical data in this report, other GA operators, namely Elite, Selection, and Coupling operators, proved their expected performance in terms of generating elite class population, candidate pool, and diploids, respectively. The scattering chart also illustrates a relative balance within the fitness values in the candidate pool. Therefore everything appears to be in place to start the optimization process.

The last round's RSR (Figure C- 2) included in this analysis suggests that the termination criteria has come true because:

i. $Div_{elites} \times p_{elites} = 59 \times 0.0169 \cong 1$, means that the convergence has occurred;

ii. $Div_p \leq 10\%$

iii. More than five rounds without a better-fit solution, which, along with the previous condition, indicates that the diversity of the candidate pool is inadequate to propose a new better-fit solution.

The scatter chart of the base population demonstrates that a significant number of 1121 solutions maintains the same fitness, value equal to the minimum fitness value, which signifies the convergence in the optimization process.

6.3.3 Analysis of the Efficacy of the GA Operators

The purpose of this subsection is to analyze the GA operators as their performance data in the optimization experiment BSE-3000-2000 is illustrated in Figure 6-1. As previously mentioned, the optimizer adopts two types of GA operators, namely Crossover and Mutation operators, to generate new genotypes to form a new generation. Even though randomness is an intrinsic part of the algorithm of these operators and therefore would cause them not to behave the same way in each optimization experiment, the algorithms adopted into these operators and the parameter configuration have a significant impact on their efficacy. This analysis of the results indicates that the optimization procedure has yielded a better-fitting solution in 64 rounds out of 100 rounds across the optimization. The crossover operator generated 58 better-fit solutions with minimum fitness values compared to the six best solutions from the mutation operator. These data support the expected roles of crossover and mutation in the GA optimization framework, which address the intensification and diversification, respectively.

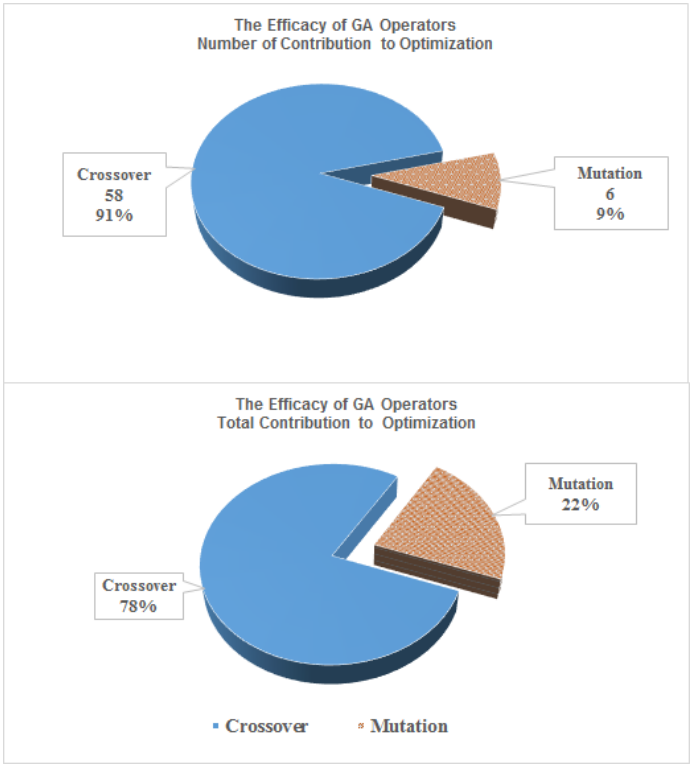


Figure 6-1, Analysis of efficacy of GA operators (BSE-3000-2000).

6.3.4 Convergence Analysis of Optimization Process

Figure 6-2 shows the correlation between demographic diversity and optimization effectiveness in finding better-fit genotypes, splits the optimization path into two different zones: effective and ineffective. The effective zone, which begins with the 10th iteration and bounds to the 60th, is the range of path that the optimizer aggressively approaches the optimality via demonstrating its capability in proposing better-fit solutions with lower fitness values. From the 5th iteration onwards, the demographic diversity, namely uniqueness in fitness values, increases and peaks at the end of the effective zone. In contrast, the optimizer appears to lose its capability to bring a significant improvement in fitness value in the ineffective zone, which is defined as the range of iterations from 61 to the last. Across the ineffective zone, demographic diversity deteriorates with a steep and abrupt drop that continues until the termination of optimization, which seems to correspond to the inefficacy of the optimizer.

Figure 6-3 illustrates that the statistical measures, i.e., minimum, mean, maximum, and variance, decline at a constant proportional rate until the 40th iteration when the mean value and minimum value begin to converge, resulting in a confluence in the 60th optimization round. The minimum and mean values confluence indicates a high density of the fitness values around the minimum point, which happens at the beginning of the ineffective zone. This is an implication of convergence occurrence in the optimization process. Meanwhile, the variance value, along with the maximum value, ends declination at around the 45th round and begins a steady transient until the termination of the 100th round, which may denote the presence of a sparse number of scattered points far from the mean value.

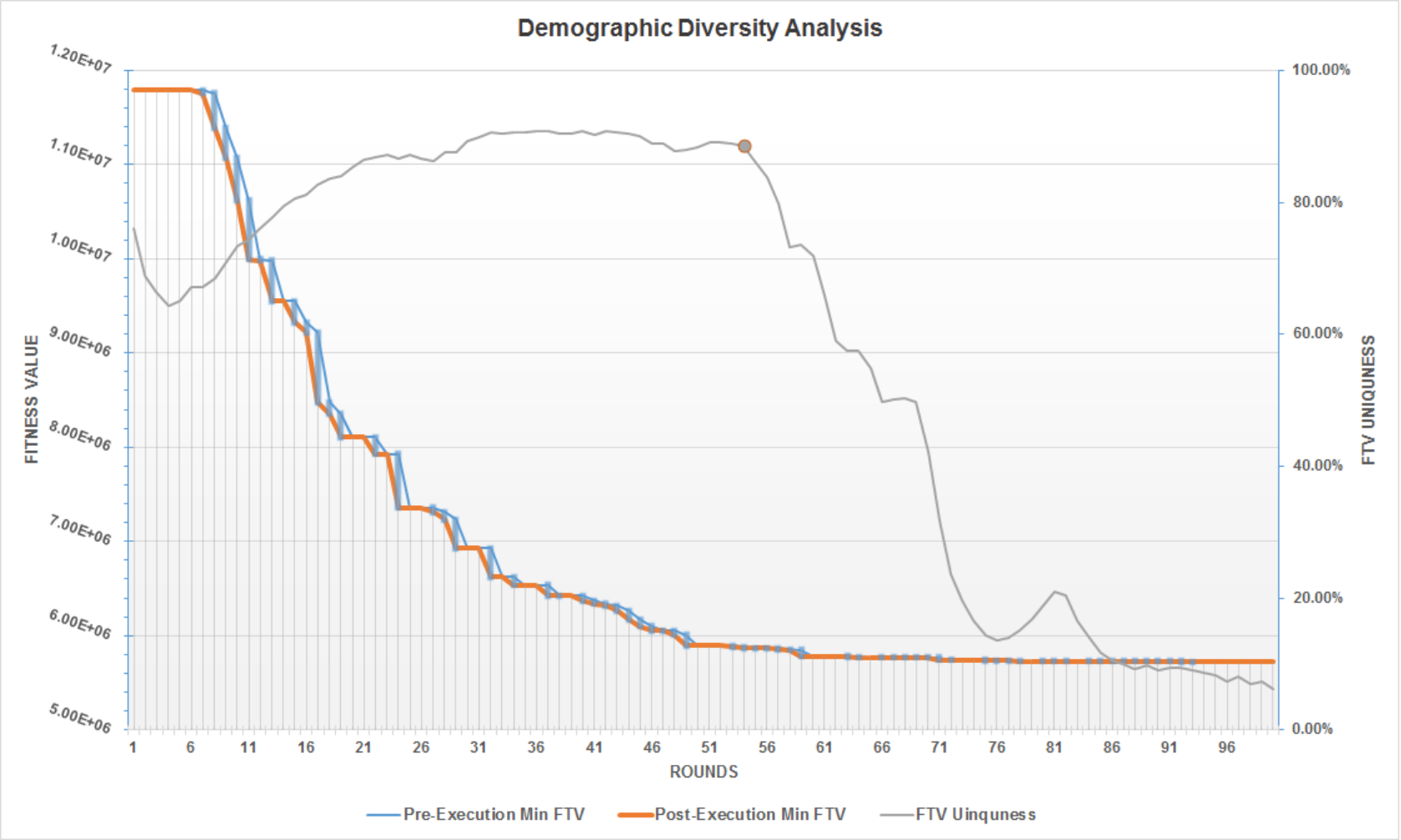


Figure 6-2, analysis of correlation between demographic diversity and optimization efficacy.

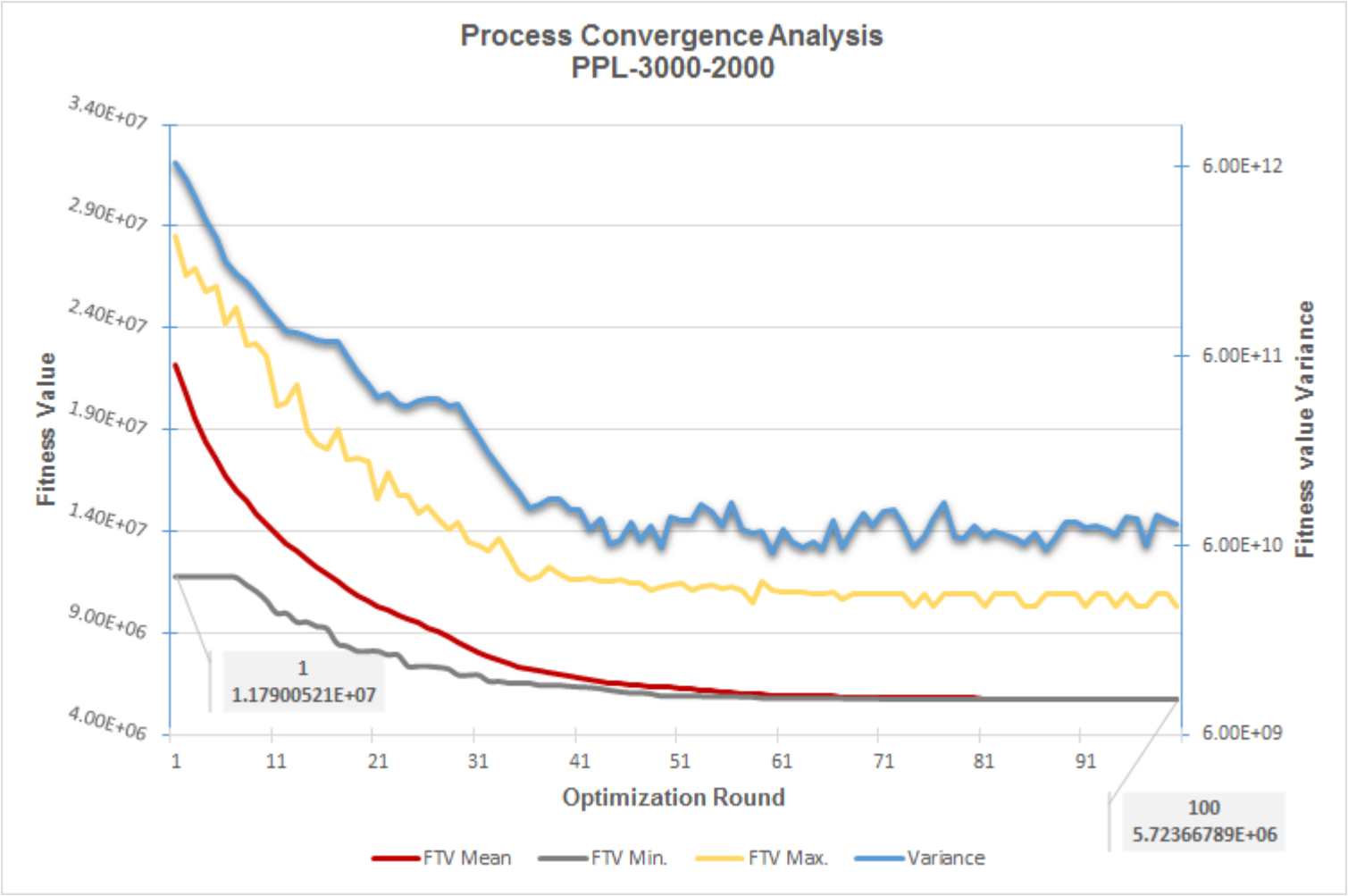


Figure 6-3, process convergence analysis.

Table 6-4 depicts the demographic distribution of the candidate pool across the optimization experiment (BSE-3000-2000) with an interval of ten rounds. The first 50 cycles suggest a normal distribution with a high population density around the mean value, corresponding to the empirical rule (68-95-99 rule). This optimization experience refers to this time slice as the "effective zone" given its mass contribution to most of the gains made while identifying the current best-fit genotype. During the 60s rounds, the population distribution seems to maintain fitting to the normal distribution skewed to the right half of the bell. Henceforth it departs this fitness and roughly resembles a Skew-t distribution. This heat map analysis unveils that the optimal searching algorithm fails to retain the demographic composition and reaches an over-density in the ($\mu \pm \sigma$) region from the 70s round to the last one, which is deemed an ineffective time zone this experience case. The transition in demographic composition as demonstrated in Table 6-4 highlights the idea the crossover operator may increase the risk of immature convergence occurrence when receiving a couple of genotypes from the same class, i.e., the demographic class attributed to each of the distances as $\mu\pm\sigma$, $\mu\pm2\sigma$, and $\mu\pm3\sigma$. In such a case, the alleles contained by both genotypes are not sufficiently distinctive, and the offspring would most likely be identical.

Figure 6-4 and Figure 6-5, which plot the fitness values collected from the candidate pool in the iterations indicated in Table 6-4, provide a more intuitive understanding of the cause for the emergence of the ineffective zone from the other approach. (For better scalability, the data is visualized in two graphs). Excess Kurtosis values calculated for the data in each round, as shown in Table 6-4, allow for the division of entire of the optimization trajectory into three phases, including the rounds with Excess Kurtosis as $Exc.Kurt. < 0$, $0 \leq Exc.Kurt. \leq 1$, and $Exc.Kurt. > 1$.

The first phase, when excess kurtosis is less than zero, shows a $Platykurtic$ type of distribution, suggesting a scarcity of data on the tail and thus a low probability of identifying extreme values. This appears to be the primary reason why the first ten iterations of the searching algorithm fell short of producing any better-fit value (according to Figure 6-2 and Figure 6-3), which is expected in the area smaller than (μ - 3σ). All observations, provided by Figure 6-2 to 6-5, and Table 6-4, supports the fact that the *selection* operator carries out its part by densifying the candidate pool's population around the mean value, resulting in a decrease in the value of σ and, as well as in the narrowing the distances of $\mu\pm\sigma$, $\mu\pm2\sigma$, and $\mu\pm3\sigma$. This may intensify optimal search by giving the

crossover operator more probability to reproduce better-fit genotypes given that the extreme value in μ-3σ is now more accessible

In the 2nd phase, beginning with the 10th round, the data distribution tends to follow the *Mesokurtic* kind of distribution closest to the normal distribution, indicating a higher level of data density in the curve's tail and the region of ($\mu \pm 2\sigma$). This observation also reinforces the concept that the *selection* operator has been fulfilling out its part by densifying the candidate pool's population around the mean value, resulting in a steep decline in the population candidate pool's variance as demonstrated by Figure 6-3, as well as in the narrowing the distances of $\mu\pm\sigma$, $\mu\pm2\sigma$, and $\mu\pm3\sigma$. The *selection* operator's proper performance also contributes to optimal search intensification via enabling the *crossover* operator to reproduce better-fit genotypes holding fitness values between the left extreme values, i.e., μ-3σ, which is now more accessible than the first phase. According to Figure 6-2 the optimization algorithm was able to diversify the population in the candidate pool. Additionally, the skewness of the data in Table 6-4 explains that data distributions on both sides of the bell curve are reasonably symmetrical due to the skewness of the data. The confluence of the factors cited seems to be the driving force behind the GA searching algorithm's high effectiveness in approaching the optimum with a relatively fast pace throughout this phase bound to the 40th round.

As shown in Table 6-4, the Excess Kurtosis value exceeds one after the 40th round and thus denotes that the data distribution, henceforth, conforms to a Leptokurtic distribution. As a result, the tail spikes to an extreme peak, indicating a massive number of outliers and anomalous data points. During this stage, while Figure 6-2 depicts the population diversity is suffering an outsized decline, variance value falls to its lowest point at the 51st round and after that begins to follow a roughly steady trend until the completion of the 100th round (as can be seen in Figure 6-3). The concurrence of these two observations implies that the data agglomeration occurs at numerous points in the tail portion of the curve rather than expected convergence exclusively around the optimum or mean value. Additionally, the considerable abrupt increase in the skewness beginning with the 60th round, as shown in Table 6-4, denotes a deterioration in the demographic composition and tendency of right-side skewness in the curve. Combining the observations mentioned above leads to a reduction in the probability of detection nearer-to-optimal point in the area of less than (μ - 2σ). Consequently, it underlies the emergence of the ineffective zone.

In a nutshell, the preceding analysis supports the conclusion that although the optimal searching process proves the convergence in its evolutionary transition approaching optimality, it has a great tendency to slip into a premature convergence and lose the efficacy in searching for better-fit genotypes (solutions). It also introduces the sudden steep declination in demographic diversity of the candidate pool and the deterioration of its composition as the primary driving force behind this tendency. The following sections explore remedial measures to apply to the algorithm to alleviate the drag of premature convergence: 1) increasing the size of the initial and the base population and 2) manipulating the parameters of the *crossover*, *mutation, elitism,* and *selection* operators. Results of each applied measure would be analyzed to validate their effectiveness. The *coupling* operator may adopt an enhanced algorithm that facilitates the coupling of genotypes from the distinct demographic classes to slow down the pace of convergence occurrence as a proposal for future development in the optimization framework.

Table 6-4, heat map to illustrate the demographic distribution of the candidate pool (BSE-3000-2000).

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	4	62	264	606	763	283	18	0	-0.13161	-0.39012
10	7	51	233	633	796	250	22	8	0.43490	-0.18946
20	8	58	216	714	724	234	39	7	1.07785	-0.16704
30	1	66	250	623	784	238	31	7	0.25111	-0.12833
40	0	43	272	653	734	259	33	6	1.49823	0.25899
50	9	68	227	589	916	173	12	6	6.20983	0.19239
60	0	1	198	767	991	31	2	10	83.48312	5.33632
70	0	0	0	1232	219	532	13	4	78.01099	4.56337
80	0	0	462	2	1526	2	2	6	107.82690	5.69669
90	0	0	0	1976	16	2	0	6	632.74646	24.31242
100	0	0	0	1775	219	1	1	4	1601.88815	38.66706

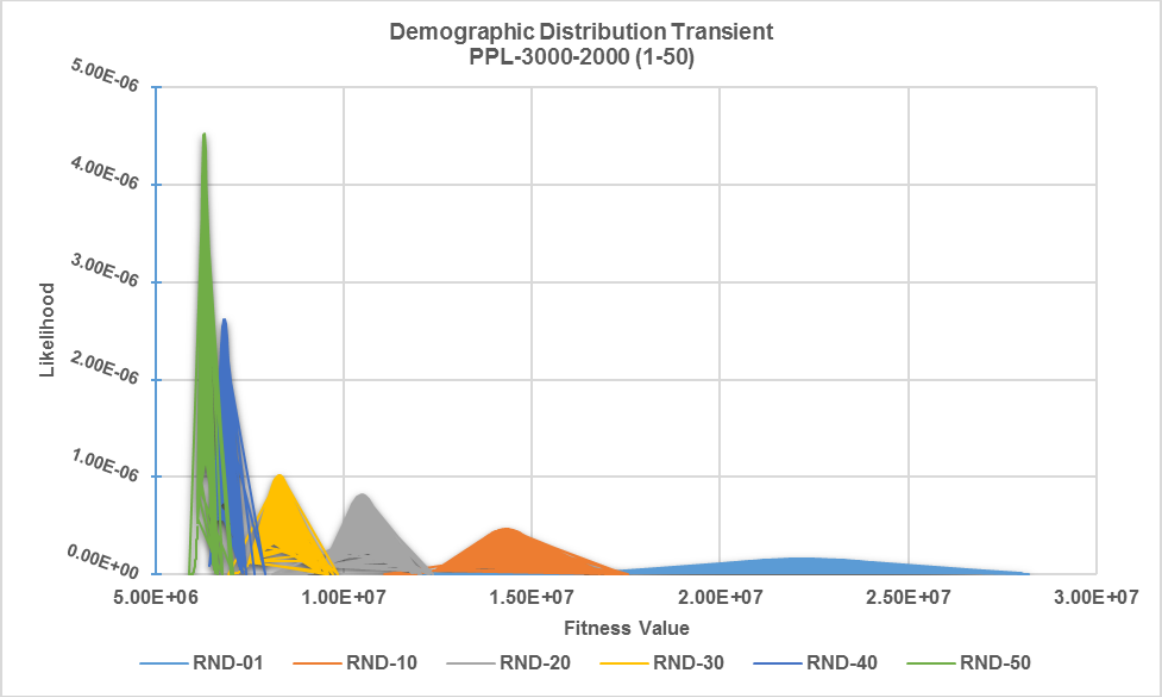


Figure 6-4, the transition in demographic composition across the rounds 1 to 50.

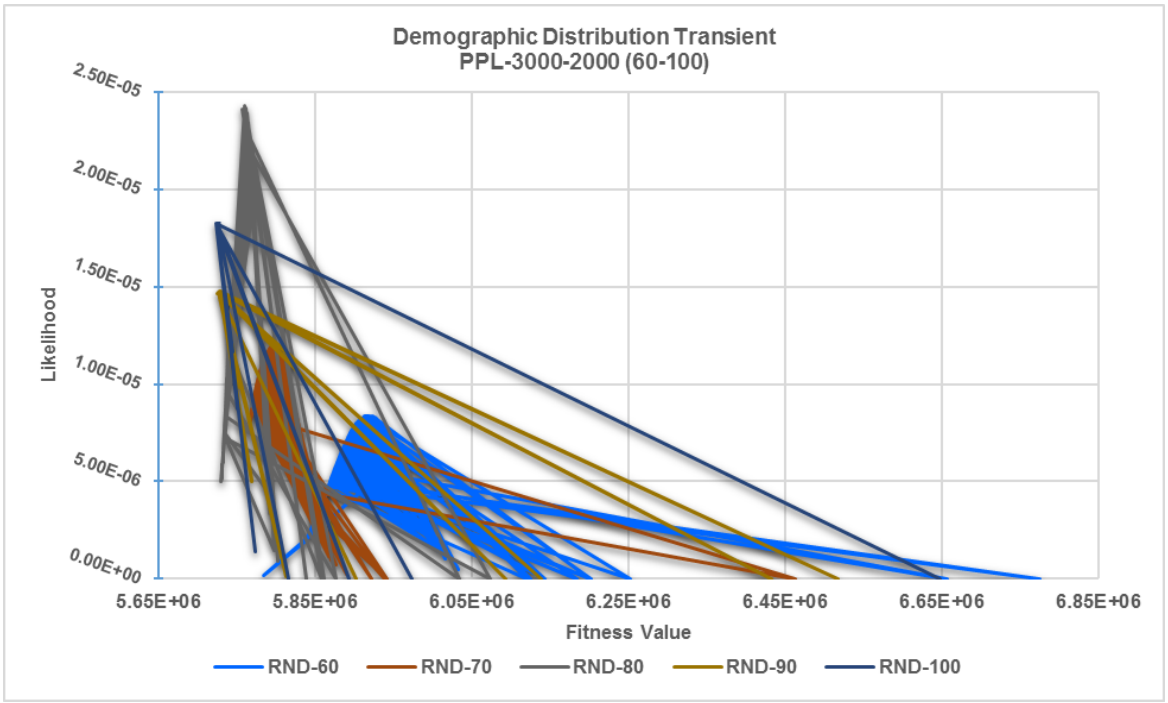


Figure 6-5, the transition in demographic composition across the rounds 60 to 100.

6.3.5 Analysis of Increase in Initial and Base Populations

This section presents an analysis of the findings from a new optimization experiment conducted with the following parameter setups to study the effect of an increase in the initial and the base populations on the convergence of the optimization process:

Table 6-5, the parameter setups for experiments PPL 10000-6000.

Parameter Name	Value
Experiment ID	PPL 10000-6000
Base population	10000
Selected population in each round	6000

- The rest of parameters are the same as PPL 2500-1500 mentioned in Table 6-1

Figure C- 3 shows the fit of the candidate pool's population distribution with the normal curves for the ten selected rounds as specified in the heading of each graph. This collection of graphs, along with the demographic heat map, as shown in Table C- 1, exhibit the same behavior in the search process as seen in the base experiment regarding the deterioration in the population diversity and process convergence to a local-optima. In other words, a larger base population (candidate pool) does not address any noteworthy improvement in population diversity, which could promise a better balance between intensification and diversification, according to Figure C- 4 and Table C- 2.

6.3.6 Analysis of the Impact of Increasing the Mutation Rate

According to the convergence study (6.3.4), the degradation of demographic diversity is one contributing factor to premature convergence in the optimization process and, as a result,

Table 6-6, the parameter setups for the experiments used in the mutation rate analysis.

Parameter Name	Value	
Experiment ID	MTN60 3000-2000	MTN100 3000-2000
Mutation rate	0.60	1.00

- The rest of parameters are the same as PPL 3000-2000 mentioned in Table 6-1

inefficiency in the optimal searching algorithm. Essentially, this assertion emphasizes the role of the mutation operator, whose primary job in a GA algorithm is to preserve population diversity. Thus, this section investigates whether raising the mutation rate can better balance intensification and diversification. In other words, whether more mutations through raising the mutation rate could prevent the candidate pool population's diversity from being degraded. To this end, the optimizer runs two new experiment cases with the following settings and returns the results for detailed examination.

Results reveal that raising the mutation rate impacts the optimal searching trajectory from different viewpoints. Overall, it significantly enhances the algorithm's ability to find better-fit genotypes. Figure C- 6, a series of graphs illustrating the candidate pool's population's good fit into the normal bell curve, explains how the demographic composition transforms across the searching iterations. Both experience instances begin with a roughly same $Mesokurtic$ distribution (or more precise to say $Platykurtic$ with a slightly negative excess kurtosis in the first round) and retain this shape until the 40th iterations. The case with a mutation rate of 1.0 has a stronger tendency to transform into a Leptokurtic distribution type. This tendency may be related to the fact that the mutation operator with the rate of 1.0 functions like seeding pointlessly rather than fertilizing the desired region of $(\mu \pm 2\sigma)$. The higher the mutation rate, the more resistant the optimization algorithm is to population diversity degradation, based on Table C- 5. More specifically, increasing the mutation rate can contribute to a better balance between intensification and diversification.

6.3.7 Analysis of the Impact of Modification of the Elitism Rate

Elitism is primarily concerned with conducting evolutionary searches toward optimality, but it also works in tandem with the crossover and select operators to achieve the process convergence.

Table 6-7, the parameter setups for the experiments used in the elitism rate analysis.

Parameter Name	Value		
Experiment ID	ELT05 3000-2000	ELT20 3000-2000	ELT30 3000-2000
Elitism rate	0.005	0.02	0.03

- The rest of parameters are the same as BSE 3000-2000 mentioned in Table 6-1

Elite pool in GA optimization acts as an adaptive memory to retain the best-fit genotypes over generation transition. The larger the size of the elite population set in the optimization configuration, the more substantial elite pressure drives the searching process forward. Finding an efficient elitism rate in the optimization process is critical since an excessively large elite population size increases the risk of being trapped in a premature convergence. In contrast, an insufficient elite population size helps to reduce the effectiveness of the optimum searching procedure. This section studies the effect of changing elitism on the optimal searching procedure through analyzing the findings of running three instances of the FLP with the configuration as shown in Table 6-7 in the optimizer:

As shown in s Figure C- 8, Figure C- 9, and Figure C- 10 and Table C- 6, Table C- 7, and Table C- 8, findings reveal that the experiment with an elitism rate of 0.005 (equivalent to 0.5% of the base population) shows a slight performance improvement compared to BSE-3000-2000. Still, both of them have roughly the same behaviors regarding the deterioration in the demographic composition. On the plus side, the experiment ELT20-3000-2000 with an elitism rate of 0.02 (equivalent to 2% of the base population) proves a noteworthy enhancement in maintaining its relatively good demographic composition until the 50th iteration and henceforth, resists against the deterioration in the composition to the 70th round. Afterward, it undergoes distortion in population composition. As seen in Table C- 9, the performance benchmark also asserts the same fact with reflecting a 12.31% reduction in fitness value of the best-fit found solution and a 14.46% increase in the overall improvement across the 100 optimization iterations. The third experiment (ELT30-3000-2000 with elitism rate of 0.03), on the other hand, may preserve acceptable demographic composition for no more than 30 iterations of the optimal searching procedure before the candidate population deforms dramatically. In other words, the synergy of elitism pressure with the selection pressure and the process intensification becomes so powerful that the resistance to deformation is ineffective, causing the search process to converge to a local-optima. Figure C- 11 illustrates the influence of population diversity on the mechanism’s efficacy, supports the conclusions mentioned above.

6.3.8 Analysis of the Effects of Changes in the Crossover Rate

The crossover operator is the main GA component contributing to process convergence. Essentially, the higher the crossover rate provided in the configuration of an instance of the GA optimization model, the faster the predicted convergence occurs, whether immature or mature, throughout the searching iterations. The purpose of this section is to investigate the impacts of modifying the crossover rate through establishing three optimization instances with the setups seen in Table 6-8. The findings of this study assist in choosing an effective crossover rate to optimize the deployed GA-based optimization algorithm to its maximum potential.

Table 6-8, the parameter setups for the experiments used in the crossover rate analysis.

Parameter Name	Value		
Experiment ID	CSO50 3000-2000	CSO70 3000-2000	CSORDM 3000-2000
Crossover rate	0.5	0.7	Randomly from the range of (0.5, 0.7)

- The rest of parameters are the same as BSE 3000-2000 mentioned in Table 6-1

According to Figure C- 12 the population scattering of the candidate pool conforms to a Platykertic type of distribution with a very modest negative excess Kurtosis, according to Table C- 10, throughout the early rounds of the experiment CSO50-3000-2000 (means very close to Mesokutic distribution). The excess kurtosis values in Table C- 10 indicate that the candidate pool demographics maintain to fit a Mesokurtic distribution until the rounds 50s. Furthermore, the optimal searching procedure provides considerable resistance to transforming to a Leptokurtic distribution by drawing back from a peak in rounds 40 and 60. Additionally, the values listed in the last column of Table C- 10 explain that the demographics' skewness, either to the right or left of the curve, is relatively ignorable until the 70th round, and the population distribution is nearly symmetric. More specifically, in Figure C- 15, the upper red line signifies that the population diversity remains relatively stable until the 80th iteration and then begins to decline progressively to a value of 40% at the termination round. The lower redline asserts that the effective zone of the better-fit searching mechanism extends from the very initial iteration to approximately the final rounds. The aforementioned shear of evidence supports the fact that setting the crossover rate to a

value of 0.5 allows the searching algorithm to retain the population diversity at an effective level via applying a more balanced force behind the process convergence. Figure C- 12, on the other hand, unveils the emergence of outliers from the 20th round that extensively increases up to the ending of the searching iterations. From the 80th round on, the population largely agglomerates at the curve's tail and around the mean value, with the rest appearing as outliers. This phenomenon indicates the deterioration of the demographic composition to such an extension that it impacts the process efficacy.

The same lines of reasoning also conclude that the population's good fitness into the normal distribution in the experiment CSO70-3000-2000 exhibits the same behavior as in the former experiment over the first to 50th iterations of the optimal search (according to Figure C- 13). However, there are some noteworthy differences in the successive iterations that will be discussed in detail later. Compared to the preceding experiment, CSO70-3000-2000 appears less resilient to demographic deformation, as evidenced by the rapid transformation to Leptokurtic distribution from the 60s round. The cause of this may be the raising of the crossover rate, which reinforces process convergence by increasing the crossover operator's involvement. As shown in Table C- 11, the demographic composition deteriorates, with the region μ-σ being empties, the population mostly agglomerating in the region of μ-σ, and a few individuals appearing in the area greater than $\mu+3\sigma$ as outliers, along with a notable increase in excess kurtosis and skewness values. In Figure C- 15, the upper blue line depicts that after the 50th iteration of the case CSO70-3000-2000, the candidate pool's population diversity begins to degrade. At the same time, the blue bottom line indicates that the optimal searching mechanism concurrently enters the non-effective zone by falling short of efficacy and failing to find a better-fit solution. The shreds of evidence mentioned above support the fact that increasing the crossover rate causes this operator to contribute more to the intensification in the optimal search process, forcing the process to converge to a local-optima. This premature convergence occurs due to an immense agglomeration population around the mean value and over the regions of $\mu \pm \sigma$.

In the experiment case of CSORDM-3000-2000, the good fit of the population into the normal curve, as depicted in Figure C- 14, across the first 40 iterations looks much like preceding trials. From the 50th round onwards, the demographic shifts to Leptokurtic skewed to the right. As illustrated via the upper green line in Figure C- 15, the demographic deformation pace and the

population diversity degradation trend indicate that the driving force of process convergence is more than CSO50-3000-2000 but less than CSO70-3000-2000. A noteworthy fact, in this case, is that the optimum searching cycles enter the ineffective zone sooner than both precedents as a result of a lack of efficacy in generating any better-fit genotype.

6.3.9 Analysis of the Effects of Changes in the Tournament Size

According to the academic review performed in chapter two, several algorithms have been proposed to adopt the Selection operator. One of them may be picked depending on the optimization problem circumstances. The GA-based algorithm developed here uses the Tournament Selection algorithm as the core mechanism to establish the candidate pool population. The tournament selection algorithm has a parameter, namely tournament size, affecting the convergence's pace. This section aims to examine how the optimization process is sensitive to the change in the tournament size. The literature suggests that a larger tournament size may lead to a premature convergence in the optimal search process. The following reasoning explains the rationale that supports this claim. Tournament selection integrates the concept of elitism into its algorithm. This indicates that a larger tournament size boosts the selection pressure, causing the balance between intensification and diversification, or equivalently exploitation and exploration, to deteriorate. Consequently, this may cause the optimal search to converge to a local-optima, also known as premature convergence. The analysis of the experiments aims to validate this theatrical hypothesis.

To this end, it provides a detailed analysis of the results of two new instances of the GA algorithm executed by the optimizer with the parameters listed in Table 6-9 and benchmark them against the base experiment of PPL-2000-3000, which holds a tournament size of 2.

Table 6-9, the parameter setups for the experiments used in the tournament size analysis.

Parameter Name	Value	
Experiment ID	TNS01 3000-2000	TNS03 3000-2000
Tournament Size	1	3

- The rest of parameters are the same as BSE-3000-2000 mentioned in Table 6-1

Theoretically, the larger the tournament size, the more likely the better-fit, more frequent genotypes being a member of the candidate pool. Therefore, a larger tournament size reinforces driving forces behind the process convergence, perhaps leading to an immature convergence. Conversely, the optimum search algorithm is less aggressive with a smaller tournament, resulting in a possible delayed convergence. Following is a detailed analysis of how the execution outputs support the abovementioned facts.

The data obtained from the observations from running the two optimization instances defined for this section confirm the theoretical hypothesis. The good fit of the candidate pool's population into the normal curve, as illustrated in Figure C- 16, shows that the fit distribution has been transformed from a Mesokutic to a Leptokurtic at a slower pace than the base experiment case throughout the first 100 optimal searching iterations. The heat map table, Table C- 14 illustrates that the demographic composition has degraded less than the basic optimization scenario during the same number of search cycles, supporting the theoretical hypothesis. The excess kurtosis values also imply that the fit distribution is Mesokurtic (or, more precisely, Leptokurtic with very low excess kurtosis), which is approximate. The population diversity of the candidate pool remains steady at the peak during the optimal searching trajectory, as shown in Figure C- 18. While all of the evidence suggests that finding the best-fit genotype will be highly effective, the blue bottom line in Figure C- 18 shows a discouraging trend in finding the best-fit genotype in each iteration until the last round, which is equivalent to the fitness value achieved in the base experiment's 25^{th} round. The sparsely populated regions of μ-3σ and μ-2σ are most likely the main reason for this lack of effectiveness, which is the result of smaller tournament size and consequently reduced likelihood of less frequent better-fit genotypes being included in the candidate pool. Degradation of the population dispersed to the regions of μ-3σ and μ-2σ makes the better-fit genotypes generation

process less fertile. A sharp drop in the efficacy of the crossover operator also supports this viewpoint, according to Table C- 16.

In the experiment case of TNS03-3000-2000 with a tournament size of 3, the optimization trajectory is approximately analogous to the base experiment during the first 40 optimal search iterations; subsequently, the good-fit distribution transformation (from Mesokurtic to Leptokurtic), as well as the deterioration of the demographic composition and a subsequent immature convergence occurs rapidly. The degradation of population diversity of the candidate pool looks roughly like the base experiment with a partial improvement in the searching process efficacy, according to the red bottom line in Figure C- 18.

The detailed analysis above maintains the theoretical hypothesis that increasing the tournament size raises the probability in the selection operator to include more frequent better-fit genotypes in the candidate pool and have more population in the area to the left of the μ-σ. In this way, the crossover operator is expected to show a better performance through reproducing better-fit genotypes by amalgamating a genotype with fitness value less than μ-σ with one that holds fitness value greater than μ-σ. This partially improves optimal search process efficacy, but it also speeds up premature convergence.

6.3.10 Representation of the Best Found Solution

The optimization framework represents the best solution embedded in the best-fit genotype in an assignment table once the optimal search algorithm fulfills the termination criteria. The assignment table determines the service points each floor accommodates.

Table C- 17 demonstrates the best solution discovered in the optimization experiment case MTN100-3000-2000 in an assignment table. There are many illogical assignments in this solution, e.g., the assignment of the indoor parking to the first floor and the like. This is because this optimization experiment has assigned the service points to the floors exclusively based on the pairwise flow between the service points and has not taken the constraints, such as functionality constraints, limitation in shared resources, architectural considerations, and so on, into account. The following sections will showcase the optimizer features to define the constraints in the GA

optimization problem and its capabilities in finding the optimal solution that respects all the adopted constraints.

6.3.11 Applying Restricted Assignment Constraints (RACs)

During the model's definition stage, the implemented evolutionary heuristic optimization framework provides a straightforward definition of restricted assignment constraints in an optimization model through a specially designed GUI that allows storing the RACs in an Excel spreadsheet template file.

Table C- 18 presents a list of the RAC constraints established for the problem instance considered here project, representing the structure most similar to how the framework stores the constraints data in an Excel file. The first column of the list contains the identification number of the constraints, followed by the second column, which holds a list of IDs of the constraints involved in the corresponding constraint. When TRUE, the Boolean value placed in the third column indicates that the department implicated in the constraint must be allocated exclusively to the floors mentioned in Floor(s) ID column or must be prevented from being assigned to other floors. The Boolean value in the last column implies whether or not the relevant constraint should be included in the GA model when constructed in the Julia environment. The optimizer executes the GA-based optimization model using the configuration shown in Table C- 19 and finds the best feasible solution, presented in Table C- 20. Figure C- 19 illustrates the summary reports corresponding to optimal search iterations' first and last labs.

Figure C- 19 depicts the starting point of the optimization path as well as the destination. As the termination criteria have been met, i.e., the uniqueness of the fitness values corresponding to the elites and the candidate pool has fallen below 1.0 and 10.0, respectively. The optimum search algorithm has been unable to introduce any better-fit solution for more than the last five iterations; the best feasible solution discovered in the 50th round is assumed to be the closest to optimal.

The assignment table, which represents the best feasible solution in this optimization experiment, ensures that all the RACs adopted to the GA model have been strictly respected. It also projects how the pairwise flow between the service points affects the optimization algorithm in allocating the service points to the floors. For instance, forasmuch as the GA model adopts a RAC to restrict

the allocation of Warehouse Office to one of the floors F01 and F04 when the optimizer determines that the assignment of Warehouse Office to F01 is the most optimum choice for the best-fit solution of the round, it assigns the warehouses, i.e., the service points WH001 to WH008, to the same floor as well, as a consequence of minimizing the objective function value. The flow between the service points contributes to the value of the objective function as a coefficient in the calculation of the transportation cost. Even though the value of flow between two service points may encourage the optimization algorithm to locate the two service points reasonably near each other, it may not be sufficient to allow them to share the floor when it is strictly required. In such cases, the optimizer provides a feature to define the adjacency constraints. The following section explains how to define these constraint types their effects on the final best solution discovered by the optimum search algorithm.

6.3.12 Applying Binary Adjacency Constraints

Some facility layout model needs to locate some of the service points or departments on the same floor due to several reasons such as functional dependencies, the use of shared resources, and so on. On the contrary, some service points, in some cases, may adversely affect each other's proper functionality that stimulating the tendency to prevent them from locating on the same floor. The developed GA-based optimizer allows the users to embed these types of constraints in the model by defining the Adjacency Constraints (AJCs) via a particular GUI in the Excel Spreadsheet template file. This feature enables the user to adopt the rules into the model in order to either enforce or prevent the allocation of multiple service points to the same floor. This section demonstrates how applying the AJCs on the FLP model affects the best solution discovered by the optimizer via defining a new optimization experiment labeled as AJC01-3000-2000 with the same parameter configuration detailed in Table C- 19 and adopting the AJCs listed in Table C- 21.

The result of the optimization process on the case AJC01-3000-2000 is as illustrated in Table C-22. This table maintains that all types of adopted constraints to the model, including UQAC, FAAC, RAC, and AJC, have been fully respected.

A thorough investigation of the above assignment table reveals the need for additional constraints, as explained in the following. For Warehouse 7, D00020, the flow effect seems insufficient to group it into the warehouses in the Basement of the North Building. Furthermore, due to the same

reason, the current solution locates Warehouse 8 on floor F08, whereas the desired location would be next to the Pharmacy. On the other hand, we may prefer not to locate the Surgery Clinic, D00117, and Medical Test Lab, D00122, on the same floor due to toxic interactions they predictably have on their proper functioning. In order to resolve the issues mentioned above, the constraints outlined in Table C- 23 are added to the model.

The result of the optimization process on the case AJC02-3000-2000 is as illustrated in Table C-24. The Warehouse Office and Warehouses 1-7 are visibly clustered together in the Basement of the North Building, as seen in this table. Warehouse 8 is on the ground floor of the South Building next to the Pharmacy. In contrast, the Medical Test Lab (Blood & Pathology) does not share the same floor as the Surgical Clinic to fulfill the negative adjacency constraint and avoid the unfavorable interplay. It is noteworthy that the effect of travel cost ascribed to the interdepartmental interactions drives the optimization process to maintain these two departments nearby as much as possible by placing them on the same floor but in two separate buildings. Given indoor connecting ways between the same floors in different buildings, this new assignment ensures the shortest feasible distance while avoiding significant vertical travel costs.

6.3.13 Applying FUZZY Adjacency Constraints

FUZZY adjacency constraints (FAC) engage in model construction when the GA model aims to decide the proximity or distance of two or more service points based on the significance of their mutual connections and correlations with one another. Unlike the binary adjacency constraints, which intervene when the intention is to adopt a rule of whether or not two or more service points must share a particular floor, FACs integrate a measure into the model to determine how nearby or distant two services have to be. In order to demonstrate the result of the application of FACs in the FLP best solution, a new optimization case is designed where a weight factor comes into play as a coefficient of the travel cost between two service points in the objective function to prioritize their mutual interactions.

This section presents the results of the optimization experiment AJC02-3000-2000, which shows the impact of FACs on the FLP best solution. In this scenario, two service points, namely the General Clinic and the Specialized Clinic, are targeted to prioritize contiguity. Conversely, two other service points, namely the Surgical Clinic and the Outpatient Clinic, are intended to have a

minimum priority of contiguity. Therefore, weighting factors of 10 and 0.1 are assigned to the interactions between them, respectively. The priority of pairwise interactions between the rest of the service points remains unchanged and is equal to one.

Table C- 25 shows the assignment table drawn up based on the optimization scenario AJC02-3000-2000. The result shows that the algorithm maintains a reasonable performance when enacting the FUZZY constraints. The proximity of the General Clinic and the Specialized Clinic on the Ground Floor of two buildings, i.e., the South Building and the Middle Building respectively, given that a passageway exists between the buildings, seems logical. It implies the optimizer has made a near-optimal trade-off decision in terms of correlations between these departments and others to maximize their continuity while avoiding the vertical distance. On the other hand, the allocation of the Outpatients Clinic and the Surgery Division to different floors in separate buildings, i.e., the Ground Floor in the North Building and the First Floor in the South Building, respectively, fulfills the intention of distancing them by increasing the horizontal and vertical distance between them.

6.3.14 Relaxation of Limited Available Area Constraints

As explained in Enactment of Constraints (5.3.2), this optimization framework facilitates defining specific types of constraints, such as FAACs, either as Binary Constrains (construction level constraints) or FUZZY constraints (process-oriented constraints). This section discusses the effect of defining the FAACs as FUZZY constraints on the result of the optimization case OVAS01-3000-2000.

The relaxation of a FAAC, either partially or entirely, transfers the load of evaluation of the constraint from the construction level to the process level. For this purpose, the user allows the floors individually to be over-allocated by setting the ascribed over-assignment rate to an arbitrary non-zero percentage. Additionally, selecting the option *With-OVAS* in framework configuration to TRUE is necessary to activate over-assignment in the optimization process. The over-assignment rates set for the floors in OVAS01-3000-2000 are as indicated in Table 6-10. The rest of the configurations and the defined constraints are as in the optimization experiment AJC02-3000-2000.

Table 6-10, the over-assignment rates of the floors in optimization case OVAS01-3000-2000.

Building	Floor	Available Area	OVAS Rate
North Building	Basement	550	20%
	Ground Floor	535	20%
	First Floor	780	10%
South Building	Basement	650	20%
	Ground Floor	980	20%
	First Floor	760	10%
Middle Building	Ground Floor	550	20%
	First Floor	110	10%

Table C- 26 shows the assignment table based on the best-fit solution found by OVAS01-3000-2000. The assignment table suggests that the best-found solution respects all the adopted constraints in the GA model. The optimizer proves the ability to conduct the optimal search at boundary conditions. The final best-fit state does not suffer from over-allocation since the available floor area meets the needs of the service points. Compared to the last experiment, i.e., AJC02-3000-2000, Table 6-11 shows the additional efficiencies this approach provides to the optimization process.

Table 6-11, advantages of the approach of relaxation of FAACs optimization performance.

Discription	OVAS01-3000-2000	AJC02-3000-2000	Result
Time elapsed for initial population 3000 instances of genotype (hh:mm:ss)	11:20	5:14:36	about 28 times faster
Time elapsed in each iteration	15:29	18:50	about 201 seconds reduced
Best-found solution fitness value	1.173E+07	1.190E+07	1.404% improved

Another optimization experiment, designed for when the available floor area does not fulfill the demands of the service points, demonstrates the capacity of this approach to enable the framework to find the best-fit solution with the least over-assignment for each of the floors. The floor available

space is reduced by 30% in this optimization scenario, OVAS02-3000-2000. Alternatively, an over-assignment rate has been determined separately for each floor to compensate for the reduction in area availability and the defined over-assignment allowance as indicated in Table 6-10. As a result, Table 6-12 shows the revised area availability and over-assignment budgets for floors.

Table 6-12, the revised space availability and over-assignment rates of the floors in optimization case OVAS02-3000-2000. (OVAS: Over-Assignment)

Building	Floor	Available Area (m^2)	OVAS Rate
North Building	Basement	385	71.5%
	Ground Floor	374.5	71.5%
	First Floor	546	57.0%
South Building	Basement	455	71.5%
	Ground Floor	686	71.5%
	First Floor	532	57.0%
Middle Building	Ground Floor	385	71.5%
	First Floor	77	57.0%

The analysis of the results suggests that the optimization algorithm proves a notable performance via a search for optimum at boundaries. In detail, Table C- 27 shows the assignment table, which represents the best-found solution in optimization scenario OVAS02-3000-2000. Table 6-13, which consolidates the data regarding the area availability of the floors, offers a comprehensive view to evaluate the performance of the optimization algorithm under the conditions of relaxing the FAACs. The analysis of the results suggests that the optimization algorithm proves a notable performance via a search for optimum at boundaries. The best-found solution's objective function value bears a total travel cost of 1.185602674165823e7, according to the summary report (Figure C- 20, which is reasonably comparable with the total travel costs in the optimization without relaxation of FAACs, i.e., 1.1901456803617027e7 in AJC02-3000-2000.

Meanwhile, Table 6-13 supports the fact that the optimum searching algorithm is highly effective in avoiding floor over-assignment via preventing the over-allocation of the service points to the floors close to the navigation network's center, where travel costs are presumably the lowest. In contrast, it does not encourage the excessive dispersing of service points over more distant floors,

i.e., basements and first floors. However, it is allowed to the extent of their space over-assignment budget, as indicated in Table 6-13.

Table 6-13, the revised space availability and over-assignment rates of the floors in optimization case OVAS02-3000-2000. (OVAS: Over-Assignment)

Building	Floor	Available Area (m^2)	Reduced Available Area (m^2)	Available Area with OVAS (m^2)	Assigned Area (m^2)
North Building	Basement	550	385	660	434
	Ground Floor	535	375	642	517
	First Floor	780	546	857	590
South Building	Basement	650	455	780	640
	Ground Floor	980	686	1176	987
	First Floor	760	532	835	537
Middle Building	Ground Floor	550	385	660	423
	First Floor	110	77	121	116

CHAPTER 7 CONCLUSIONAND RECOMMENDATIONS

7.1 Review of Project's Objectives

The main objective of this research project is to design and implement a decision-making support tool that assists facility layout planners in the field of healthcare to:

- Find the best assignment of the departments or service points to buildings and floors in a future or existing hospital while ensuring the most efficient use of limited resources such as land and capital; it also meets the constraints: unique assignment of the service points to the floors, floor's area availability constraint, restricted assignment of the departments or service points to a floor or floors and adjacency constraints, and adjacency constraints.
- Modify the existing layout plans to improve and be adaptive to the changes in services continuously.

7.2 Conclusions of the Research Project

7.2.1 Mathematical Optimization Approach to FLP

A mathematical model based on the QSAP approach has been developed that accounts for the allocation of all the service points to the buildings and floors with the objective of minimizing the costs of traveling and material handling within the service points. This model ensures meeting the four types of constraints mentioned in 7.1 as well. In order to use the existed MILP solvers, such as CPLEX and Gurobi, the QSAP model has been linearized using the best technic found in the literature. Then, the linearized model was implemented in the Julia programming ecosystem using JuMP and JuliaOpt libraries

7.2.2 Metaheuristics Approach to FLP

An optimization framework with a metaheuristic approach based on the genetic algorithm was devised and implemented to make this FLP optimizer a tool that works effectively in a real-size facility planning project. This framework has been implemented in three layers: the frontend, data layer, and optimizer layer. In order to receive user data in a standard format, an Excel file

containing VBA-developed GUIs has been established. The optimizer layer streams, and after constructing the data layer, provides the best solution, i.e., near-optimal solution, via running a GA-based optimization algorithm. The GA optimizer encompasses six configurable operators, namely *population, selection, elitism, coupling, crossover,* and *mutation,* developed in the Julia development ecosystem. The GA optimizer contains five configurable operators, namely *population, selection, elitism, diploid, crossover,* and *mutation,* developed in the Julia development ecosystem. The population operator uses a random-based technique to build the initial population, which is only used before going through the optimization rounds, then delivers it to the selection operator to construct the base population. The optimizer provides two types of Crossover operators that use the stochastic regular double-side and microbial algorithms, respectively. It also supplies two types of Mutation operators, which are established based on two point–to–point stochastic algorithms. Once meeting the termination criteria, the optimization process terminates and delivers the three best solutions.

The optimizer delivers the best solution back to the client-side in an Excel workbook.

7.3 Experimentations and Sensitivity Analysis

As the optimization process terminates, the optimizer delivers the best solutions along with the metadata back to the client-side in an Excel workbook. The analysis of the solutions proves that all of them effectively meet all the applied constraints. Furthermore, the result of analyzing the sensitivity of the optimization process to changes in the parameters using descriptive statistical analysis in PowerBI and Excel are as follow:

- The analysis of the demographic composition of the base population proves its inextricable link with optimization efficacy. The optimizer is projected to perform well when the base population is well fitted to a normal distribution and a type of Mesokurtic (6.3.3).
- Although the experiment case of an increase in the initial and base population shows a slight improvement in finding a better-fit solution, the statistical analysis of the process does not observe a substantial improvement in the optimization effectiveness zone (6.3.5).
- An increase in mutation rate addresses a more resistant optimization to degradation of diversity in the base population, according to the analysis of change in mutation rate (6.3.6).

- The statistical analysis does not address any noteworthy changes in process behavior in case of a decrease in elitism rate, i.e., from 0.1 to 0.05. Meanwhile, a rise in the elitism rate affects the optimization process in two different ways. Increasing the elitism rate from 0.01 to 0.02 improves optimization performance significantly, while the increase to 0.03 makes the optimal search too greedy, leading to the risk of falling into an immature convergence (6.3.7).
- Changes in crossover rate also affect the optimization process in different ways. The optimum search algorithm can more efficiently preserve population diversity by lowering the crossover rate from 0.6 to 0.5, resulting in an enhancement in the effectiveness zone. This fact supports a better balance within the search intensification and population diversification. Raising the crossover to 0.7, on the other hand, boosts the driving force behind the process convergence, resulting in deterioration in the balance between the search intensification and diversification and the occurrence of immature convergence. Meanwhile, in the case of using a crossover operator with a random rate ranging from 0.5 to 0.7, the analysis reveals that the pace of process convergence expectantly is more than the rate of 0.5 and less than 0.7. Applying a random crossover rate narrows the effectiveness zone (6.3.8).
- Setting the tournament size to 1 has an adverse effect on optimization efficacy, compared to tournament size 2 in the base experiment, due to a deterioration in search intensification. However, the tournament size of 3expedite significantly the occurrence of convergence (6.3.9).
- The optimizer effectively applies all types of constraints, including limited assignment constraints, binary adjacency constraints, and FUZZY adjacency constraints, using both implicit and explicit approaches, according to the analysis of constraints (6.3.11, 6.3.12, and 6.3.13).

7.4 Recommendations

The analysis of experimentations supports the fact that striking a balance between search intensification and population diversification plays a crucial role in optimization efficacy.

Therefore, to have the best performance of the optimizer in proposing the nearest-optimal solution and lowering the risk of premature convergence, the same sensitivity analysis using descriptive statistical analysis can help determine the most efficient configuration of the parameters.

7.5 Future Research

Further research is recommended in the following four areas:

1. The current project is the first stage of developing an optimizer framework to provide the best solution to an FLP of a future hospital. The second stage, which is to find the optimum or near optimum facility layout for each floor, is an interesting subject for further research. The QAP mathematical approach is suggested in the literature for small-size single floor FLP when the number of service points is less than 30 on each floor. Adopting the same genetic algorithm approach for a large number of service points on each floor is likely to be a more feasible proposal.

2. Literature suggests using adaptive GA improves the optimization performance in providing higher quality solutions. Adaptive GA is able to re-launch its search process based on adaptive conditions to escape out of local-optima when it is trapped by a premature convergence (Dao et al., 2017). Using an adaptive crossover assists the optimization process in balancing exploitation and exploration, improving the optimization performance (Neri & Tirronen, 2010). Devising an adaptive GA using the statistical technique used in the sensitivity analysis, explained in detail in chapter 6, to the current framework could be a fascinating proposal for further research.

3. Hybridization of an evolutionary algorithm, such as GA, with local search heuristics, such as simulated annealing or hill climbing, may improve the performance of the GA optimization (Kesavan et al., 2020). Therefore, hybridization of the current developed GA-based optimizer or devising a hybrid crossover operator is also another exciting ground for future research.

4. (Cappart et al., 2020) opens a new era to combinatorial optimization combining a data science-related approach, namely reinforcement learning, with constraints programming. In this regard, (Chalumeau et al., 2021) introduce a Julia-based library for constraints

programming solvers using the concept of reinforcement learning. Since the FLP in the current project is an instance of combinatorial optimization, conducting research on using this breakthrough in solving FLP could be intriguing for future research.

REFERENCES

Aarts, E. H. L., & van Laarhoven, P. J. M. (1987, 1987//). *Simulated Annealing: A Pedestrian Review of the Theory and Some Applications.* Pattern Recognition Theory and Applications, Berlin, Heidelberg (pp. 179-192).

Abdinnour-Helm, S., & Hadley, S. W. (2000). Tabu search based heuristics for multi-floor facility layout. *International Journal of Production Research, 38*(2), 365-383. https://doi.org/10.1080/002075400189464

Adams, W., & Johnson, T. (1994). Improved linear programming-based lower bounds for the quadratic assignment problem. In *Quadratic Assignment and Related Problems* (pp. 43-75). American Mathematical Society : Providence, Rhode Island.

Adams, W. P., & Sherali, H. D. (1986). A Tight Linearization and an Algorithm for Zero-One Quadratic Programming Problems. *Management Science, 32*(10), 1274-1290. https://doi.org/10.1287/mnsc.32.10.1274

Adams, W. P., & Sherali, H. D. (1990). Linearization Strategies for a Class of Zero-One Mixed Integer Programming Problems. *Operations Research, 38*(2), 217-226. https://doi.org/10.1287/opre.38.2.217

Afentakis, P. J. I. J. o. F. M. S. (1989). A loop layout design problem for flexible manufacturing systems. *1*(2), 175-196.

Ahmadi, A., & Akbari Jokar, M. R. (2016). An efficient multiple-stage mathematical programming method for advanced single and multi-floor facility layout problems. *Applied Mathematical Modelling, 40*(9-10), 5605-5620. https://doi.org/10.1016/j.apm.2016.01.014

Anjos, M. F., & Vannelli, A. J. I. J. C. (2006). A New Mathematical-Programming Framework for Facility-Layout Design. *18*, 111-118.

Anjos, M. F., & Vieira, M. V. C. (2017). Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions. *European Journal of Operational Research, 261*(1), 1-16. https://doi.org/https://doi.org/10.1016/j.ejor.2017.01.049

Azadivar, F., & Wang, J. J. I. J. o. P. R. (2000). Facility layout optimization using simulation and genetic algorithms. *38*(17), 4369-4383.

Balakrishnan, J., Cheng, C. H., Conway, D. G., & Lau, C. M. J. I. J. o. P. E. (2003). A hybrid genetic algorithm for the dynamic plant layout problem. *86*(2), 107-120.

Balakrishnan, J., Jacobs, F. R., & Venkataramanan, M. A. (1992). Solutions for the constrained dynamic facility layout problem. *European Journal of Operational Research, 57*(2), 280-286. https://doi.org/https://doi.org/10.1016/0377-2217(92)90049-F

Baykasoglu, A., Dereli, T., & Sabuncu, I. (2006). An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems. *Omega, 34*(4), 385-396. https://doi.org/https://doi.org/10.1016/j.omega.2004.12.001

Baykasoğlu, A., Gindy, N. N. J. C., & Research, O. (2001). A simulated annealing algorithm for dynamic layout problem. *28*(14), 1403-1426.

Beasley, D., Bull, D. R., & Martin, R. R. J. U. c. (1993). An overview of genetic algorithms: Part 1, fundamentals. *15*(2), 56-69.

Bernardi, S., & Anjos, M. F. (2013). A two-stage mathematical-programming method for the multi-floor facility layout problem. *Journal of the Operational Research Society, 64*(3), 352-364. https://doi.org/10.1057/jors.2012.49

Bertsimas, D., & Tsitsiklis, J. (1993). Simulated Annealing. *Statist. Sci., 8*(1), 10-15. https://doi.org/10.1214/ss/1177011077

Billionnet, A., & Elloumi, S. (2001). Best reduction of the quadratic semi-assignment problem. *Discrete Applied Mathematics, 109*(3), 197-213. https://doi.org/https://doi.org/10.1016/S0166-218X(00)00257-2

[Record #65 is using a reference type undefined in this output style.]

Boussaïd, I., Lepagnot, J., & Siarry, P. J. I. s. (2013). A survey on optimization metaheuristics. *237*, 82-117.

Bozer, Y. A., Meller, R. D., & Erlebacher, S. J. J. M. S. (1994). An improvement-type layout algorithm for single and multiple-floor facilities. *40*(7), 918-932.

Braglia, M. (1996). Optimisation of a simulated-annealing-based heuristic for single row machine layout problem by genetic algorithm. *International Transactions in Operational Research, 3*(1), 37-49. https://doi.org/https://doi.org/10.1016/0969-6016(96)00006-8

Braglia, M., Zanoni, S., & Zavanella, L. J. I. J. o. P. R. (2003). Layout design in dynamic environments: Strategies and quantitative indices. *41*(5), 995-1016.

Burkard, R. E., Çela, E., Pardalos, P. M., & Pitsoulis, L. S. (1998). The Quadratic Assignment Problem. In *Handbook of Combinatorial Optimization* (pp. 1713-1809). Springer US : Boston, MA.

Cappart, Q., Moisan, T., Rousseau, L.-M., Prémont-Schwarz, I., & Cire, A. (2020). Combining Reinforcement Learning and Constraint Programming for Combinatorial Optimization.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications, 45*(1), 41-51. https://doi.org/10.1007/BF00940812

[Record #148 is using a reference type undefined in this output style.]

Chambers, L. D. (1998). *An Indexed Bibliography of Genetic Algorithm.*

Chang, C.-H., Lin, J.-L., Lin, H.-J. J. I. E., & Systems, M. (2006). Multiple-Floor Facility Layout Design with Aisle Construction. *5*, 1-10.

Chwif, L., Barretto, M. R. P., & Moscato, L. A. J. C. i. i. (1998). A solution to the facility layout problem using simulated annealing. *36*(1-2), 125-132.

Dao, S. D., Abhary, K., & Marian, R. J. E. S. w. A. (2017). An innovative framework for designing genetic algorithm structures. *90*, 196-208.

Das, S. K. (1993). A facility layout method for flexible manufacturing systems. *International Journal of Production Research, 31*(2), 279-297. https://doi.org/10.1080/00207549308956725

Dilworth, J. B. (1996). *Operation management*. McGraw Hill.

Drira, A., Pierreval, H., & Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control, 31*(2), 255-267. https://doi.org/10.1016/j.arcontrol.2007.04.001

Dweiri‡, F., & Meier, F. A. (1996). Application of fuzzy decision-making in facilities layout planning. *International Journal of Production Research, 34*(11), 3207-3225. https://doi.org/10.1080/00207549608905085

Ejeh, J. O., Liu, S., & Papageorgiou, L. G. (2018). Optimal multi-floor process plant layout with production sections. *Chemical Engineering Research and Design, 137*, 488-501. https://doi.org/https://doi.org/10.1016/j.cherd.2018.07.018

Evanst†, G. W., Wilhelm†, M. R., & Karwowskij, W. (1987). A layout design heuristic employing the theory of fuzzy sets. *International Journal of Production Research, 25*(10), 1431-1450. https://doi.org/10.1080/00207548708919924

Fischer, A., Fischer, F., & Hungerländer, P. (2019). New exact approaches to row layout problems. *Mathematical Programming Computation : A Publication of the Mathematical Optimization Society, 11*(4), 703-754. https://doi.org/10.1007/s12532-019-00162-6

Fruggiero, F., Lambiase, A., & Negri, F. (2006). *Design and optimization of a facility layout problem in virtual environment.* Proceeding of ICAD (pp. 13-16).

Gen, M., Ida, K., & Cheng, C. (1995). Multirow machine layout problem in fuzzy environment using genetic algorithms. *Computers & Industrial Engineering, 29*(1), 519-523. https://doi.org/https://doi.org/10.1016/0360-8352(95)00127-M

Goetschalckx, M., & Irohara, T. (2007). *Efficient Formulations for the Multi-Floor Facility Layout Problem with Elevators.*

Gogna, A., & Tayal, A. (2013). Metaheuristics: review and application. *Journal of Experimental & Theoretical Artificial Intelligence, 25*(4), 503-526. https://doi.org/10.1080/0952813X.2013.782347

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of genetic algorithms* (Vol. 1, pp. 69-93). Elsevier.

[Record #44 is using a reference type undefined in this output style.]

Grobelny, J. (1987). The fuzzy approach to facilities layout problems. *Fuzzy Sets and Systems, 23*(2), 175-190. https://doi.org/https://doi.org/10.1016/0165-0114(87)90057-1

Gupta, T., & Seifoddini, H. I. J. T. i. j. o. p. r. (1990). Production data based similarity coefficient for machine-component grouping decisions in the design of a cellular manufacturing system. *28*(7), 1247-1269.

Hahn, P., MacGregor Smith, J., & Zhu, Y.-R. (2010). The Multi-Story Space Assignment Problem. *Annals of Operations Research, 179*(1), 77-103. https://doi.org/10.1007/s10479-008-0474-3

Hamann, T., & Vernadat, F. (1993). The intra-cell layout problem in automated manufacturing systems. In *Manufacturing Research and Technology* (Vol. 16, pp. 231-243). Elsevier.

Harvey, I. (2011). *The Microbial Genetic Algorithm.* Berlin, Heidelberg (pp. 126-133).

Hassanat, A., Almohammadi, K., Alkafaween, E. a., Abunawas, E., Hammouri, A., & Prasath, V. B. S. (2019). Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach. *10*(12), 390. https://www.mdpi.com/2078-2489/10/12/390

Heragu, S. S., & Kusiak, A. (1988). Machine Layout Problem in Flexible Manufacturing Systems. *Operations Research, 36*(2), 258-268.

Jaradat, G., Ayob, M., & Almarashdeh, I. (2016). The effect of elite pool in hybrid population-based meta-heuristics for solving combinatorial optimization problems. *Applied Soft Computing, 44*, 45-56. https://doi.org/https://doi.org/10.1016/j.asoc.2016.01.002

Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. J. O. r. (1989). Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning. *37*(6), 865-892.

Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. J. O. r. (1991). Optimization by simulated annealing: an experimental evaluation; part II, graph coloring and number partitioning. *39*(3), 378-406.

Johnson, R. V. (1982). Spacecraft for Multi-Floor Layout Planning. *Management Science, 28*(4), 407-417.

Kaku, B. K., Thompson, G. L., & Baybars, I. (1988). A heuristic method for the multi-story layout problem. *European Journal of Operational Research, 37*(3), 384-397. https://doi.org/https://doi.org/10.1016/0377-2217(88)90202-0

Kanwal, M. S., Ramesh, A. S., & Huang, L. A. (2013). A novel pseudoderivative-based mutation operator for real-coded adaptive genetic algorithms. *F1000Research, 2*, 139. https://doi.org/10.12688/f1000research.2-139.v2

Kesavan, V., Kamalakannan, R., Sudhakarapandian, R., & Sivakumar, P. (2020). Heuristic and meta-heuristic algorithms for solving medium and large scale sized cellular manufacturing system NP-hard problems: A comprehensive review. *Materials Today: Proceedings, 21*, 66-72. https://doi.org/https://doi.org/10.1016/j.matpr.2019.05.363

Kim, J.-G., & Kim, Y.-D. J. I. J. o. P. R. (2000). Layout planning for facilities with fixed shapes and input and output points. *38*(18), 4635-4653.

Kim, J. G., & Kim, Y. D. (1999). A Branch and Bound Algorithm for Locating Input and Output Points of Departments on the Block Layout. *The Journal of the Operational Research Society, 50*(5), 517-525.

Kim, J. Y., & Kim, Y. D. (1995). Graph theoretic heuristics for unequal-sized facility layout problems. *Omega, 23*(4), 391-401. https://doi.org/https://doi.org/10.1016/0305-0483(95)00016-H

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics, 34*(5), 975-986. https://doi.org/10.1007/BF01009452

Kochhar, J. S. (1998). MULTI-HOPE: A tool for multiple floor layout problems. *International Journal of Production Research, 36*(12), 3421-3435.

Korejo, I., Yang, S., Brohi, K., & Khuhro, Z. (2013). Multi-population methods with adaptive mutation for multi-modal optimization problems.

Kouvelis, P., & Chiang, W.-C. (1992). A simulated annealing procedure for single row layout problems in flexible manufacturing systems. *International Journal of Production Research, 30*(4), 717-732. https://doi.org/10.1080/00207543.1992.9728452

Kouvelis, P., & Kim, M. W. (1992). Unidirectional Loop Network Layout Problem in Automated Manufacturing Systems. *Operations Research, 40*(3), 533-550. https://doi.org/10.1287/opre.40.3.533

Kouvelis, P., Kurawarwala, A. A., & Gutierrez, G. J. J. E. j. o. o. r. (1992). Algorithms for robust single and multiple period layout planning for manufacturing systems. *63*(2), 287-303.

Kusiak, A., & Heragu, S. S. (1987). The facility layout problem. *European Journal of Operational Research, 29*(3), 229-251. https://doi.org/10.1016/0377-2217(87)90238-4

Lacksonen, T. A., & Enscore, E. E. (1993). Quadratic assignment algorithms for the dynamic layout problem. *International Journal of Production Research, 31*(3), 503-517. https://doi.org/10.1080/00207549308956741

Lacksonen, T. A. J. I. J. o. P. R. (1997). Preprocessing for static and dynamic facility layout problems. *35*(4), 1095-1106.

Lee, G.-C., & Kim, Y.-D. J. O. (2000). Algorithms for adjusting shapes of departments in block layouts on the grid-based plane. *28*(1), 111-122.

Lee, K.-Y., Roh, M.-I., Jeong, H.-S. J. C., & Research, O. (2005). An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. *32*(4), 879-899.

Leung, J. (1992). A graph-theoretic heuristic for designing loop-layout manufacturing systems. *European Journal of Operational Research, 57*(2), 243-252. https://doi.org/https://doi.org/10.1016/0377-2217(92)90046-C

Liggett, R. S., & Mitchell, W. J. (1981). Optimal space planning in practice. *Computer-Aided Design, 13*(5), 277-288. https://doi.org/https://doi.org/10.1016/0010-4485(81)90317-1

Matai, R., Singh, S., & Mittal, M. J. I. J. o. P. R. (2013). Modified simulated annealing based approach for multi objective facility layout problem. *51*(14), 4273-4288.

Matsuzaki, K., Irohara, T., Yoshimoto, K. J. C., & Engineering, I. (1999). Heuristic algorithm to solve the multi-floor layout problem with the consideration of elevator utilization. *36*(2), 487-502.

McKendall, A. R., Shang, J., & Kuppusamy, S. (2006). Simulated annealing heuristics for the dynamic facility layout problem. *Computers & Operations Research, 33*(8), 2431-2444. https://doi.org/https://doi.org/10.1016/j.cor.2005.02.021

Meller, R. D., & Bozer, Y. A. (1997). Alternative approaches to solve the multi-floor facility layout problem. *Journal of Manufacturing Systems, 16*(3), 192-203. https://doi.org/10.1016/S0278-6125(97)88887-5

Meller, R. D., & Bozer, Y. A. J. I. J. o. P. R. (1996). A new simulated annealing algorithm for the facility layout problem. *34*(6), 1675-1692.

Meller, R. D., Narayanan, V., & Vance, P. H. (1998a). Optimal facility layout design. *Operations Research Letters, 23*(3-5), 117-127. https://doi.org/10.1016/S0167-6377(98)00024-8

Meller, R. D., Narayanan, V., & Vance, P. H. (1998b). Optimal facility layout design11This study was supported in part by an Auburn College of Engineering infrastructure award made to Drs. Meller and Vance, and NSF CAREER Grants DMII 9623605 and DMII 9502502. *Operations Research Letters, 23*(3), 117-127. https://doi.org/https://doi.org/10.1016/S0167-6377(98)00024-8

Meng, G., Heragu *, S. S., & Zijm, H. (2004). Reconfigurable layout problem. *International Journal of Production Research, 42*(22), 4709-4729. https://doi.org/10.1080/00207540420002645 90

Misevičius, A., & Verenė, D. J. E. (2021). A Hybrid Genetic-Hierarchical Algorithm for the Quadratic Assignment Problem. *23*(1), 108.

Neri, F., & Tirronen, V. (2010). Recent advances in differential evolution: a survey and experimental analysis. *Artificial Intelligence Review, 33*(1), 61-106. https://doi.org/10.1007/s10462-009-9137-2

Nordin, N. N., & Lee, L. S. J. P. J. o. S. R. R. (2016). Heuristics and metaheuristics approaches for facility layout problems: a survey. *2*(3).

Nourani, Y., Andresen, B. J. J. o. P. A. M., & General. (1998). A comparison of simulated annealing cooling strategies. *31*(41), 8373.

Patsiatzis, D. I., & Papageorgiou, L. G. (2002). Optimal multi-floor process plant layout. *Computers & Chemical Engineering, 26*(4), 575-583. https://doi.org/https://doi.org/10.1016/S0098-1354(01)00781-5

Proth, J.-M. (1992). *Conception et gestion des systèmes de production*. Presses universitaires de France reedition numeri.

Raoot, A. D., & Rakshit, A. (1991). A 'fuzzy' approach to facilities lay-out planning. *International Journal of Production Research, 29*(4), 835-857. https://doi.org/10.1080/00207549108930105

Reeves, C. R. J. O. R.-K. P. (1996). Heuristic search methods: A review. 122-149.

Rostami, B., & Malucelli, F. (2014). A revised reformulation-linearization technique for the quadratic assignment problem. *Discrete Optimization, 14*, 97-103. https://doi.org/10.1016/j.disopt.2014.08.003

Shayan, E., & Chittilappilly, A. (2004). Genetic algorithm for facilities layout problems based on slicing tree structure. *International Journal of Production Research, 42*(19), 4055-4067.

Sherali, H. D., & Alameddine, A. (1992). A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization : An International Journal Dealing with Theoretical and Computational Aspects of Seeking Global Optima and Their Applications in Science, Management, and Engineer, 2*(4), 379-410. https://doi.org/10.1007/BF00122429

Sherali, H. D., Fraticelli, B. M. P., & Meller, R. D. (2003). Enhanced Model Formulations for Optimal Facility Layout. *Operations Research, 51*(4), 629-644. https://doi.org/10.1287/opre.51.4.629.16096

Shukla, A., Pandey, H. M., & Mehrotra, D. (2015). *Comparative review of selection techniques in genetic algorithm.* 2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE) (pp. 515-519).

Singh, S. P., & Sharma, R. R. K. (2006). A review of different approaches to the facility layout problems. *The International Journal of Advanced Manufacturing Technology, 30*(5-6), 425-433. https://doi.org/10.1007/s00170-005-0087-9

Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M., & Trevino, J. (1996). *Facilities Planning*. John Weily.

Tsuchiya, K., Bharitkar, S., & Takefuji, Y. J. E. J. o. O. R. (1996). A neural network approach to facility layout problems. *89*(3), 556-563.

Umbarkar, A. J., & Sheth, P. D. J. I. j. o. s. c. (2015). Crossover operators in genetic algorithms: a review. *6*(1).

Wang, M.-J., Hu, M. H., & Ku, M.-Y. J. C. i. I. (2005). A solution to the unequal area facilities layout problem by genetic algorithm. *56*(2), 207-220.

Welgama, P. S., & Gibson, P. R. (1993). A construction algorithm for the machine layout problem with fixed pick-up and drop-off points. *International Journal of Production Research, 31*(11), 2575-2589. https://doi.org/10.1080/00207549308956884

Wong, W., & Ming, C. I. (2019). *A review on metaheuristic algorithms: recent trends, benchmarking and applications.* 2019 7th International Conference on Smart Computing & Communications (ICSCC) (pp. 1-5).

APPENDIX A CLIENTS DATA

Table A- 1, list of departments and service points and their space requirements

Departments			Service Points				
Description	**Code**	**ID**	**Description**	**Code**	**ID**	**Area (Min)**	**Area (Max)**
Surgery Clinic	D00117	SG011	Operating room for major surgeries	D00021	SG001	34	37
			Operating room for minor surgeries	D00022	SG002	34	36
			Preparation & anesthesia room for major surgery	D00023	SG003	21	23
			Preparation & anesthesia room for minor surgery	D00024	SG004	21	23
			Sterilization room for operating room	D00025	SG005	17	20
			Recovery room	D00026	SG006	17	20
			Post-anesthesia care unit 1	D00027	SG007	12	14.5
			Post-anesthesia care unit 2	D00028	SG008	12	14.5
			Post-anesthesia care unit 3	D00029	SG009	12	14.5
			Post-anesthesia care unit 4	D00030	SG010	12	14.5
Outpatient Clinic	D00116	OC007	Doctors' office 1	D00036	AD004	20	22
			Waiting room	D00037	GS015	10	12
			Staff lounge in the outpatient clinic	D00038	AD005	12	14
			Inpatient emergency caring room	D00039	OC001	45	49
			Outpatient clinic room 1	D00040	OC002	14	16
			Outpatient clinic room 2	D00041	OC003	14	16
			Admission office	D00043	OC005`	7	9.5
			Waiting room (outpatient clinic)	D00044	GS016	10	11
			Stretchers room	D00045	OC006	10	11
Medical Imaging Test clinic	D00118	DC009	Echocardiography	D00046	MI001	20	23
			X-Ray	D00047	MI002	25	31
			The waiting room for the MI department	D00048	MI003	35	37
			Doctor's office in MI department	D00049	MI004	15	18
			Staff lounge	D00050	MI005	10	11
			Reception office in MI department	D00051	MI006	20	23
			MRI	D00145	MI007	30	33.5
Specialized Clinic	D00119	SC012	Registration & admission office	D00058	AD010	12	15
			Outpatient surgery room 1	D00059	SC001	16	19
			Outpatient surgery room 2	D00060	SC002	16	19
			Waiting room 1 at SC	D00061	SC003	15	17
			Waiting room 2 at SC	D00062	SC004	15	17
			Gynecology & obstetrics doctor office 1	D00063	SC005	15	17
			Gynecology & obstetrics doctor office 2	D00064	SC006	15	17
			Orthopedic doctor office	D00065	SC007	15	17
			Neurology doctor office	D00066	SC008	15	17
			Cardiology doctor office	D00067	SC009	15	17
			Hematology doctor office	D00068	SC010	15	17

Continued next page

Table A- 1, list of departments and service points and their space requirements

Departments			Service Points				
Description	**Code**	**ID**	**Description**	**Code**	**ID**	**Area (Min)**	**Area (Max)**
Surgery Clinic	D00117	SG011	Operating room for major surgeries	D00021	SG001	34	37
			Operating room for minor surgeries	D00022	SG002	34	36
			Preparation & anesthesia room for major surgery	D00023	SG003	21	23
			Preparation & anesthesia room for minor surgery	D00024	SG004	21	23
			Sterilization room for operating room	D00025	SG005	17	20
			Recovery room	D00026	SG006	17	20
			Post-anesthesia care unit 1	D00027	SG007	12	14.5
			Post-anesthesia care unit 2	D00028	SG008	12	14.5
			Post-anesthesia care unit 3	D00029	SG009	12	14.5
			Post-anesthesia care unit 4	D00030	SG010	12	14.5
Outpatient Clinic	D00116	OC007	Doctors' office 1	D00036	AD004	20	22
			Waiting room	D00037	GS015	10	12
			Staff lounge in the outpatient clinic	D00038	AD005	12	14
			Inpatient emergency caring room	D00039	OC001	45	49
			Outpatient clinic room 1	D00040	OC002	14	16
			Outpatient clinic room 2	D00041	OC003	14	16
			Admission office	D00043	OC005`	7	9.5
			Waiting room (outpatient clinic)	D00044	GS016	10	11
			Stretchers room	D00045	OC006	10	11
Medical Imaging Test clinic	D00118	DC009	Echocardiography	D00046	MI001	20	23
			X-Ray	D00047	MI002	25	31
			The waiting room for the MI department	D00048	MI003	35	37
			Doctor's office in MI department	D00049	MI004	15	18
			Staff lounge	D00050	MI005	10	11
			Reception office in MI department	D00051	MI006	20	23
			MRI	D00145	MI007	30	33.5
Specialized Clinic	D00119	SC012	Registration & admission office	D00058	AD010	12	15
			Outpatient surgery room 1	D00059	SC001	16	19
			Outpatient surgery room 2	D00060	SC002	16	19
			Waiting room 1 at SC	D00061	SC003	15	17
			Waiting room 2 at SC	D00062	SC004	15	17
			Gynecology & obstetrics doctor office 1	D00063	SC005	15	17
			Gynecology & obstetrics doctor office 2	D00064	SC006	15	17
			Orthopedic doctor office	D00065	SC007	15	17
			Neurology doctor office	D00066	SC008	15	17
			Cardiology doctor office	D00067	SC009	15	17
			Hematology doctor office	D00068	SC010	15	17

Continued next page

Table A- 1, list of departments and service points and their space requirements

Departments			Service Points				
Description	**Code**	**ID**	**Description**	**Code**	**ID**	**Area (Min)**	**Area (Max)**
Surgery Clinic	D00117	SG011	Operating room for major surgeries	D00021	SG001	34	37
			Operating room for minor surgeries	D00022	SG002	34	36
			Preparation & anesthesia room for major surgery	D00023	SG003	21	23
			Preparation & anesthesia room for minor surgery	D00024	SG004	21	23
			Sterilization room for operating room	D00025	SG005	17	20
			Recovery room	D00026	SG006	17	20
			Post-anesthesia care unit 1	D00027	SG007	12	14.5
			Post-anesthesia care unit 2	D00028	SG008	12	14.5
			Post-anesthesia care unit 3	D00029	SG009	12	14.5
			Post-anesthesia care unit 4	D00030	SG010	12	14.5
Outpatient Clinic	D00116	OC007	Doctors' office 1	D00036	AD004	20	22
			Waiting room	D00037	GS015	10	12
			Staff lounge in the outpatient clinic	D00038	AD005	12	14
			Inpatient emergency caring room	D00039	OC001	45	49
			Outpatient clinic room 1	D00040	OC002	14	16
			Outpatient clinic room 2	D00041	OC003	14	16
			Admission office	D00043	OC005`	7	9.5
			Waiting room (outpatient clinic)	D00044	GS016	10	11
			Stretchers room	D00045	OC006	10	11
Medical Imaging Test clinic	D00118	DC009	Echocardiography	D00046	MI001	20	23
			X-Ray	D00047	MI002	25	31
			The waiting room for the MI department	D00048	MI003	35	37
			Doctor's office in MI department	D00049	MI004	15	18
			Staff lounge	D00050	MI005	10	11
			Reception office in MI department	D00051	MI006	20	23
			MRI	D00145	MI007	30	33.5
Specialized Clinic	D00119	SC012	Registration & admission office	D00058	AD010	12	15
			Outpatient surgery room 1	D00059	SC001	16	19
			Outpatient surgery room 2	D00060	SC002	16	19
			Waiting room 1 at SC	D00061	SC003	15	17
			Waiting room 2 at SC	D00062	SC004	15	17
			Gynecology & obstetrics doctor office 1	D00063	SC005	15	17
			Gynecology & obstetrics doctor office 2	D00064	SC006	15	17
			Orthopedic doctor office	D00065	SC007	15	17
			Neurology doctor office	D00066	SC008	15	17
			Cardiology doctor office	D00067	SC009	15	17
			Hematology doctor office	D00068	SC010	15	17

Continued next page

Table A- 1, list of departments and service points and their space requirements

Departments			Service Points				
Description	**Code**	**ID**	**Description**	**Code**	**ID**	**Area (Min)**	**Area (Max)**
Surgery Clinic	D00117	SG011	Operating room for major surgeries	D00021	SG001	34	37
			Operating room for minor surgeries	D00022	SG002	34	36
			Preparation & anesthesia room for major surgery	D00023	SG003	21	23
			Preparation & anesthesia room for minor surgery	D00024	SG004	21	23
			Sterilization room for operating room	D00025	SG005	17	20
			Recovery room	D00026	SG006	17	20
			Post-anesthesia care unit 1	D00027	SG007	12	14.5
			Post-anesthesia care unit 2	D00028	SG008	12	14.5
			Post-anesthesia care unit 3	D00029	SG009	12	14.5
			Post-anesthesia care unit 4	D00030	SG010	12	14.5
Outpatient Clinic	D00116	OC007	Doctors' office 1	D00036	AD004	20	22
			Waiting room	D00037	GS015	10	12
			Staff lounge in the outpatient clinic	D00038	AD005	12	14
			Inpatient emergency caring room	D00039	OC001	45	49
			Outpatient clinic room 1	D00040	OC002	14	16
			Outpatient clinic room 2	D00041	OC003	14	16
			Admission office	D00043	OC005`	7	9.5
			Waiting room (outpatient clinic)	D00044	GS016	10	11
			Stretchers room	D00045	OC006	10	11
Medical Imaging Test clinic	D00118	DC009	Echocardiography	D00046	MI001	20	23
			X-Ray	D00047	MI002	25	31
			The waiting room for the MI department	D00048	MI003	35	37
			Doctor's office in MI department	D00049	MI004	15	18
			Staff lounge	D00050	MI005	10	11
			Reception office in MI department	D00051	MI006	20	23
			MRI	D00145	MI007	30	33.5
Specialized Clinic	D00119	SC012	Registration & admission office	D00058	AD010	12	15
			Outpatient surgery room 1	D00059	SC001	16	19
			Outpatient surgery room 2	D00060	SC002	16	19
			Waiting room 1 at SC	D00061	SC003	15	17
			Waiting room 2 at SC	D00062	SC004	15	17
			Gynecology & obstetrics doctor office 1	D00063	SC005	15	17
			Gynecology & obstetrics doctor office 2	D00064	SC006	15	17
			Orthopedic doctor office	D00065	SC007	15	17
			Neurology doctor office	D00066	SC008	15	17
			Cardiology doctor office	D00067	SC009	15	17
			Hematology doctor office	D00068	SC010	15	17

Continued next page

Table A- 2, list of departments and service points and their space requirements (cont’d and end)

Table A- 4, the matrix of flow between service points

	D00001	D00002	D00003	D00004	D00005	D00006	D00007	D00008	D00009	D00010	D00011	D00012	D00013	D00014	D00015	D00016	D00017	D00018	D00019	D00020	D00031	D00032	D00033	D00034	D00035	D00042	D00052	D00053	D00054	D00055	D00056	D00057	D00069	D00070	D00071	D00084	D00085	D00090	D00113	D00114	D00115	D00116	D00117	D00118	D00119	D00120	D00121	D00122	D00123	D00127	D00128	D00129	D00130	D00131	D00132	D00134	D00136	D00139	D00141	D00142
D00001	0	0	0	0	0	0	0	0	0	0	0	0	0	600	600	600	600	600	600	600	0	0	0	0	60	0	0	60	0	0	0	0	0	0	0	0	0	0	6000	0	0	540	420	420	420	300	420	540	420	0	0	0	0	0	0	0	0	0	0	0	0
D00002	0	0	0	0	180	180	180	60	0	0	0	0	0	120	120	120	120	120	120	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	180	180	180	180	180	180	180	0	0	0	0	0	0	0	0	0	0	0	0
D00003	0	0	0	0	180	180	0	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00004	0	0	0	0	180	180	0	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00005	0	0	0	0	0	0	180	0	0	12	12	12	12	480	480	480	480	480	480	480	12	12	12	12	12	12	12	12	12	12	12	12	12	12	30	12	12	12	30	30	30	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
D00006	0	0	0	0	0	0	180	0	0	12	12	12	12	480	480	480	480	480	480	480	12	12	12	12	12	12	12	12	12	12	12	12	12	12	30	12	12	12	30	30	30	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12
D00007	0	0	0	0	0	0	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	720	180	180	180	360	360	180	0	0	0	0	0	0	0	0	0	0	0	0
D00008	0	0	0	0	0	0	0	0	12	12	12	30	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	30	12	12	12	30	30	30	30	60	60	30	60	60	60	30	12	12	12	12	12	30	12	12	12	12	30
D00009	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	1500	1500	1500	1500	1500	1500	0	0	0	300	12	12	0	180	0	0	0	0	0	600	360	0	0	120	0	0	0	180	180	120	60	180	180	180	60	0	0	0	0	0	0	0	0	0	0	0	0
D00010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00012	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00013	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00014	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	1500	1500	1500	1500	1500	0	0	0	180	0	0	0	0	0	0	0	0	180	180	180	0	0	60	0	0	0	180	360	360	180	360	360	480	180	0	0	0	0	0	0	0	0	0	0	0	0
D00015	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	1500	1500	1500	1500	0	0	0	180	0	0	0	0	0	0	0	0	180	180	180	0	0	60	0	0	0	180	360	360	180	360	360	480	180	0	0	0	0	0	0	0	0	0	0	0	0
D00016	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	1500	1500	1500	0	0	0	180	0	0	0	0	0	0	0	0	180	180	180	0	0	60	0	0	0	180	360	360	180	360	360	480	180	0	0	0	0	0	0	0	0	0	0	0	0
D00017	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	1500	1500	0	0	0	180	0	0	0	0	0	0	0	0	180	180	180	0	0	60	0	0	0	180	360	360	180	360	360	480	180	0	0	0	0	0	0	0	0	0	0	0	0
D00018	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	1500	0	0	0	180	0	0	0	0	0	0	0	0	180	180	180	0	0	60	0	0	0	180	360	360	180	360	360	480	180	0	0	0	0	0	0	0	0	0	0	0	0
D00019	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	0	0	0	180	0	0	0	0	0	0	0	0	180	180	180	0	0	60	0	0	0	180	360	360	180	360	360	480	180	0	0	0	0	0	0	0	0	0	0	0	0
D00020	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0	0	0	0	0	0	0	0	180	180	180	0	0	60	0	0	0	180	360	360	180	360	360	480	180	0	0	0	0	0	0	0	0	0	0	0	0
D00031	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6000	0	900	0	0	1500	1500	600	600	0	0	300	300	0	0	300	180	600	900	360	720	180	180	480	180	180	0	0	0	0	0	0	0	0	0	0	0	0
D00032	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00033	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0	0	720	1800	360	360	0	0	180	1800	0	0	3000	0	900	1800	900	900	300	300	300	600	300	300	0	0	0	0	0	0	0	0	0	0	0
D00034	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00035	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	0	300	180	720	360	0	0	120	120	0	0	120	180	360	720	480	360	180	180	180	360	180	180	0	0	0	0	0	0	0	0	0	0	0	0
D00042	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	360	0	720	0	0	0	0	360	0	0	360	0	3000	600	3000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00052	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00053	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	1080	720	0	0	0	720	0	0	1080	120	600	1800	480	240	240	360	240	240	300	480	0	0	0	0	0	0	0	0	0	0	0	0
D00054	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	720	0	0	720	0	480	960	360	720	180	180	120	180	120	180	0	0	0	0	0	0	0	300	0	0	0	0
D00055	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	360	0	0	0	720	0	0	720	0	7200	14400	4800	1200	3600	2400	2400	1200	3600	4800	0	0	0	0	0	0	0	2400	0	0	0	0
D00056	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	0	0	300	0	60	300	600	150	450	300	300	150	450	600	0	0	0	0	0	0	60	0	0	0	0	0
D00057	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00069	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	360	0	0	360	0	1200	24000	3000	720	1080	1500	720	720	720	3000	0	0	0	0	0	0	0	900	0	0	0	0
D00071	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	0	600	6000	3000	1200	900	900	900	900	900	900	0	0	0	0	0	0	0	300	0	0	0	0
D00084	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00085	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00090	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	900	4500	1500	1500	2400	1200	900	1800	3000	0	0	0	0	0	0	0	300	0	0	0	0
D00113	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	300	120	300	120	180	120	480	0	0	0	0	0	0	0	0	0	0	0	0
D00114	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2160	540	1620	1080	1080	540	1620	2160	0	0	0	0	0	0	0	0	0	0	0	0
D00115	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4320	1080	3240	2160	2160	1080	3240	4320	0	0	0	0	0	0	0	0	0	0	0	0
D00116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	720	1440	1080	360	300	1380	1500	0	0	0	0	0	0	0	0	0	0	0	0
D00117	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	360	360	360	180	720	120	0	0	0	0	0	0	0	0	0	0	0	0
D00118	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	600	360	360	0	600	0	0	0	0	0	0	0	0	0	0	0	0
D00119	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	300	720	120	0	0	0	0	0	0	0	0
D00120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
D00121	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300	0	0	0	0	0	0	0
D00122	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	720	0	0	0	0	0	0
D00123	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00127	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00128	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00131	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00136	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00141	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D00142	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APPENDIX B GA-BASED HEURISTIC OPTIMIZER

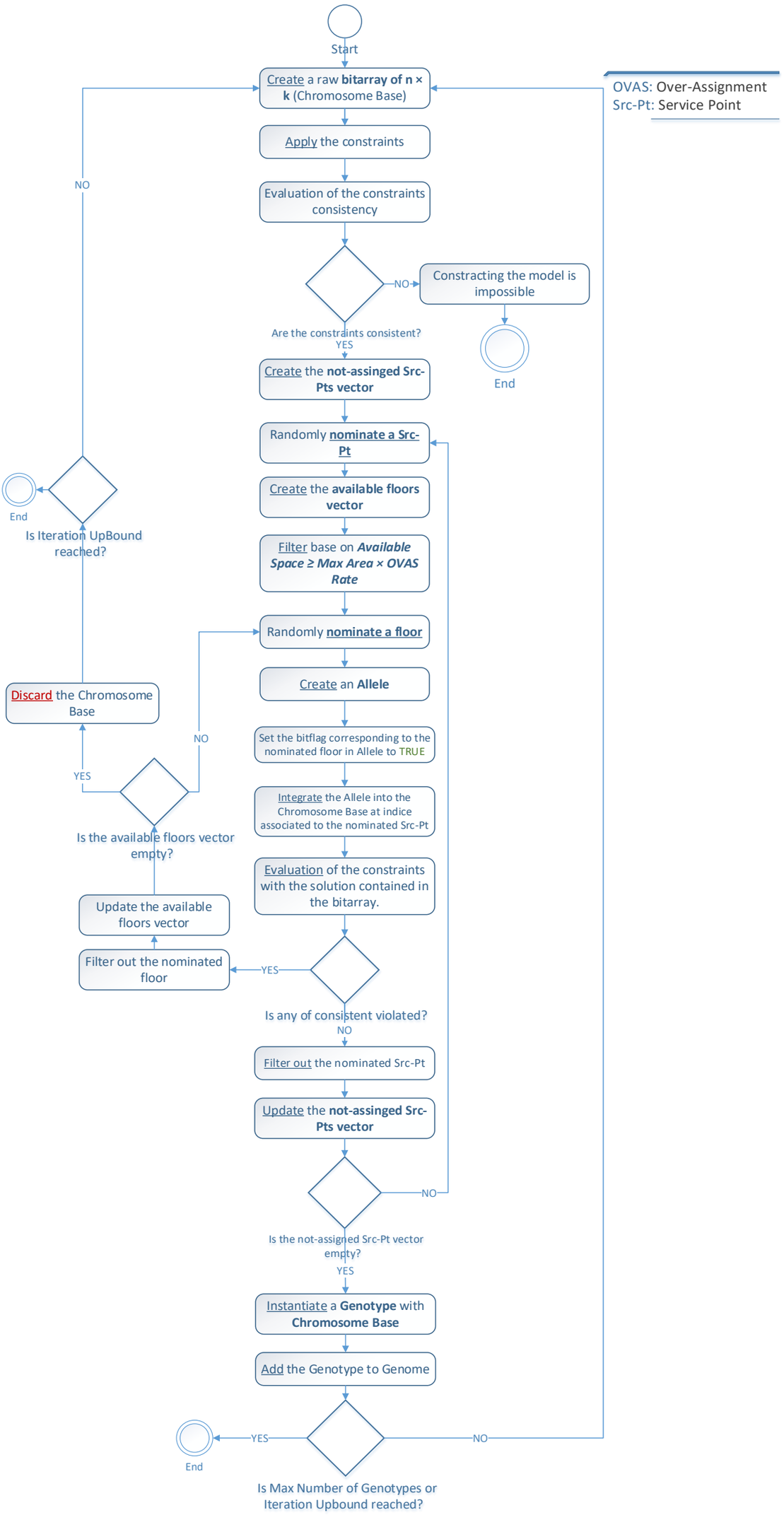


Figure B- 1, Genetic Algorithm, initial population flow process chart

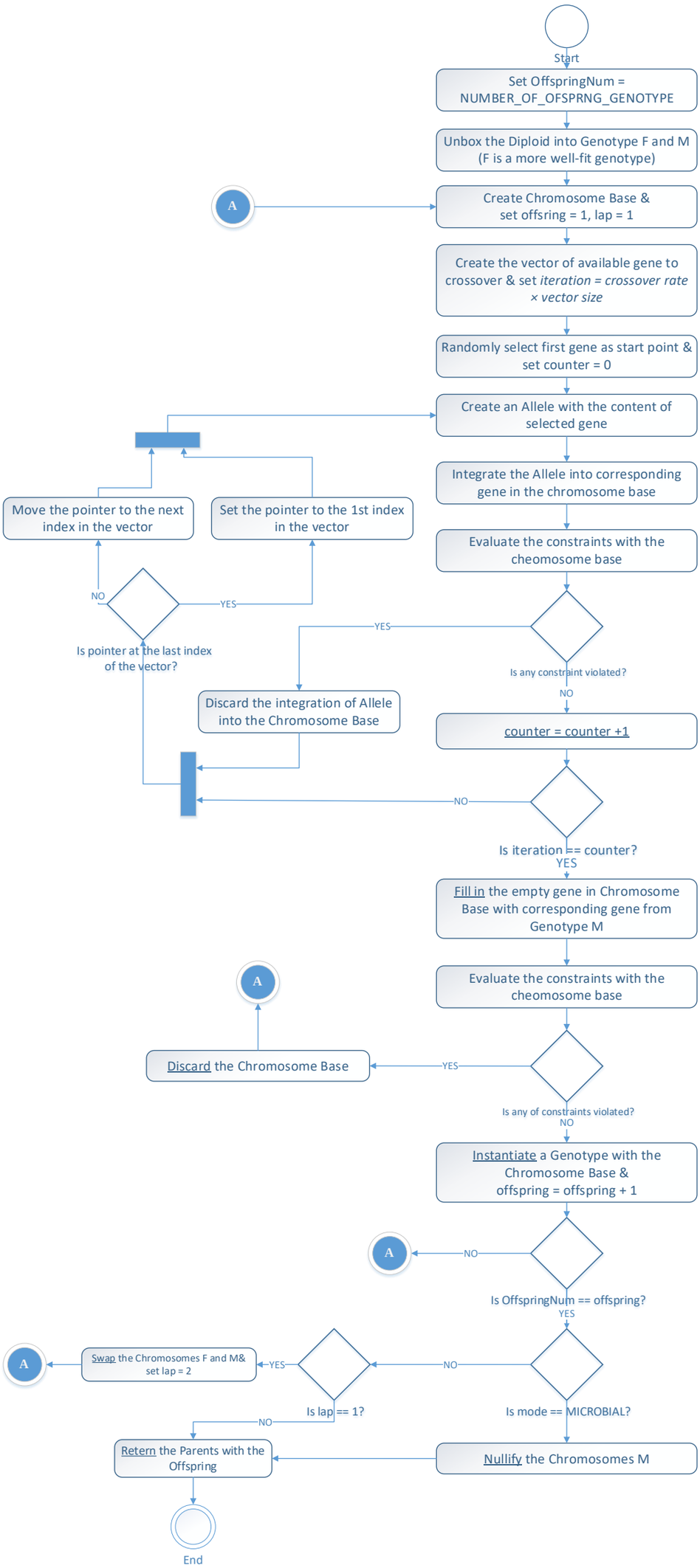


Figure B- 2, Genetic Algorithm, crossover flow process chart

APPENDIX C RESULT AND DATA OF THE EMPIRICAL STUDY ON THE GA OPTIMIZER

```
---------------------------------------------------------------------------- [Round 1] ----------------------------------------------------------------------------
Info:
                            size |FTV Uniqueness Rate |            Mean FTV |          Variance FTV |             Min FTV |             Max FTV | Dive. Rate|
       [Base population]     3000 |             100.0% | 2.385182699424546e7 |   7.81614598213146e12 |1.1790052125493102e7 |  3.0520801417035 79e7 |    100.0% |
      [Elite Candidates]       30 |             100.0% |1.5496006068916863e7 | 1.0631150207860765e12 |1.1790052125493102e7 |  1.6501069150289679e7 |    100.0% |
   [Selected Population]     1970 |              66.35% |2.2317482894224867e7 |  5.660783967795295e12 | 1.652130532357305e7 |  2.8195911661091793e7 |    66.35% |
       [Candidates Pool]     2000 |              66.85% | 2.221516074184525e7 |    6.2791149098564e12 |1.1790052125493102e7 |  2.8195911661091793e7 |    66.85% |
              [Diploids]     1905 |               N.A. |               N.A. |                  N.A. |                N.A. |                  N.A. |    99.95% |

Scattering of fitness values in range of minimum and maximum around the average:
            Min            0.25             0.5            Mean            0.75             Max
            [----------------|----------------|----------------|----------------|----------------]
                  16            367            569             568            480
[ Info: Crossover launched!
[ Info: Crossover done!
[ Info: Mutation launched!
[ Info: Mutation done!
Info: [Post-Execution Stats]
[Top 3 solutions]
       GNR No.|          TC          |      OVAS Cost       |      Fitness Value     |  Round Improv.          |      Global Improv.       |Atmpt. No.|
    ***     0 |  1.1790052125493102e7 |                  0.0 |  1.1790052125493102e7 |                  =  0 |                  =  0 |     3    |
     **     0 |  1.3504731092565805e7 |                  0.0 |  1.3504731092565805e7 |                  =  0 |                  =  0 |     1    |
      *     0 |   1.378437901731658e7 |                  0.0 |   1.378437901731658e7 |                  =  0 |                  =  0 |     1    |
[Last 2 solutions]
            0 |  2.8013201749410596e7 |                  0.0 |  2.8013201749410596e7 |                  =  0 |                  =  0 |     3    |
            0 |  2.8195911661091793e7 |                  0.0 |  2.8195911661091793e7 |                  =  0 |                  =  0 |     4    |
      Stat info:
               size |FTV Uniqueness Rate |           Mean FTV |          Round Improv. |      Global Improv.  |          Variance FTV | Dive. Rate|
               2962 |              76.1% |2.2185656584265612e7 |                 =  0.0 |             =  0.0 |  6.305748592851259e12 |    77.62% |

Scattering of fitness values in range of minimum and maximum around the average:
            Min            0.25             0.5            Mean            0.75             Max
            [----------------|----------------|----------------|----------------|----------------]
                  28            597            758             977            602
```

Figure C- 1, BSE-3000-2000 first optimization round summary reports.

```
---------------------------------------------------------------------------- [Round 100] ----------------------------------------------------------------------------
[ Info:
                            size |FTV Uniqness Rate |            Mean FTV |          Variance FTV |             Min FTV |             Max FTV | Dive. Rate|
       [Base population]     5945 |           7.25% | 5.775454951916003e6 |  8.209212212341733e10 | 5.723667888845888e6 |  1.094972164835322e7 |    93.93% |
     [Elite Candidates]       59 |           1.69% | 5.723667888845884e6 |1.4117060011397464e-17 | 5.723667888845888e6 |  5.723667888845888e6 |    98.31% |
  [Selected Population]     1941 |            1.6% |  5.72475096315757e6 |   4.9201952126715l1e8 | 5.723667888845888e6 |  6.647382273244202e6 |    78.46% |
      [Candidates Pool]     2000 |           1.55% | 5.724719012465362e6 |   4.775312853990603e8 | 5.723667888845888e6 |  6.647382273244202e6 |    78.85% |
             [Diploids]     1878 |            N.A. |                N.A. |                N.A. |                N.A. |                N.A. |    100.0% |

Scattering of fitness values in range of minimum and maximum around the average:
          Min             Min             0.25              0.5              0.75              Max
          [----------------|----------------|----------------|----------------|----------------]
              1121            877              1                0                1

[ Info: Crossover launched!
[ Info: Crossover done!
[ Info: Mutation launched!
[ Info: Mutation done!
[ Info: [Post-Execution Stats]
[Top 3 solutions]
       GNR No.|          TC          |      OVAS Cost        |      Fitness Value     |   Round Improv.           |      Global Improv.       |Atmpt. No.|
    ***    99 |   5.723667888845888e6 |                   0.0 |    5.723667888845888e6 |                   =  0.0 |   ↑ (6.066384236647214e6) |     1    |
     **    99 |   5.723667888845888e6 |                   0.0 |    5.723667888845888e6 |                   =  0.0 |   ↑ (7.781063203719917e6) |     1    |
      *    99 |   5.723667888845888e6 |                   0.0 |    5.723667888845888e6 |                   =  0.0 |   ↑ (8.060711128470692e6) |     1    |
[Last 2 solutions]
         100 |  1.0202698440929705e7 |                   0.0 |   1.0202698440929705e7 |     ↑ (92255.14928521775) |   ↑ (1.781050330848089e7) |     0    |
         100 |  1.0294663250018151e7 |                   0.0 |   1.0294663250018151e7 |     ↑ (655058.3983350694) |   ↑ (1.790124841107364e7) |     0    |
     Stat info:
              size |FTV Uniqness Rate |           Mean FTV |          Round Improv. |         Global Improv.  |          Variance FTV | Dive. Rate|
              6457 |           6.16% |5.77301212277057956e6 |   ↑ (2442.829145424068) |↑ (1.6412644461495033e7) |  7.737981691356567e10 |    93.45% |

Scattering of fitness values in range of minimum and maximum around the average:
          Min             Mean            0.25              0.5              0.75              Max
          [----------------|----------------|----------------|----------------|----------------]
              6033            358              41               12               13
```

Figure C- 2, BSE-3000-2000 100th optimization round summary reports.

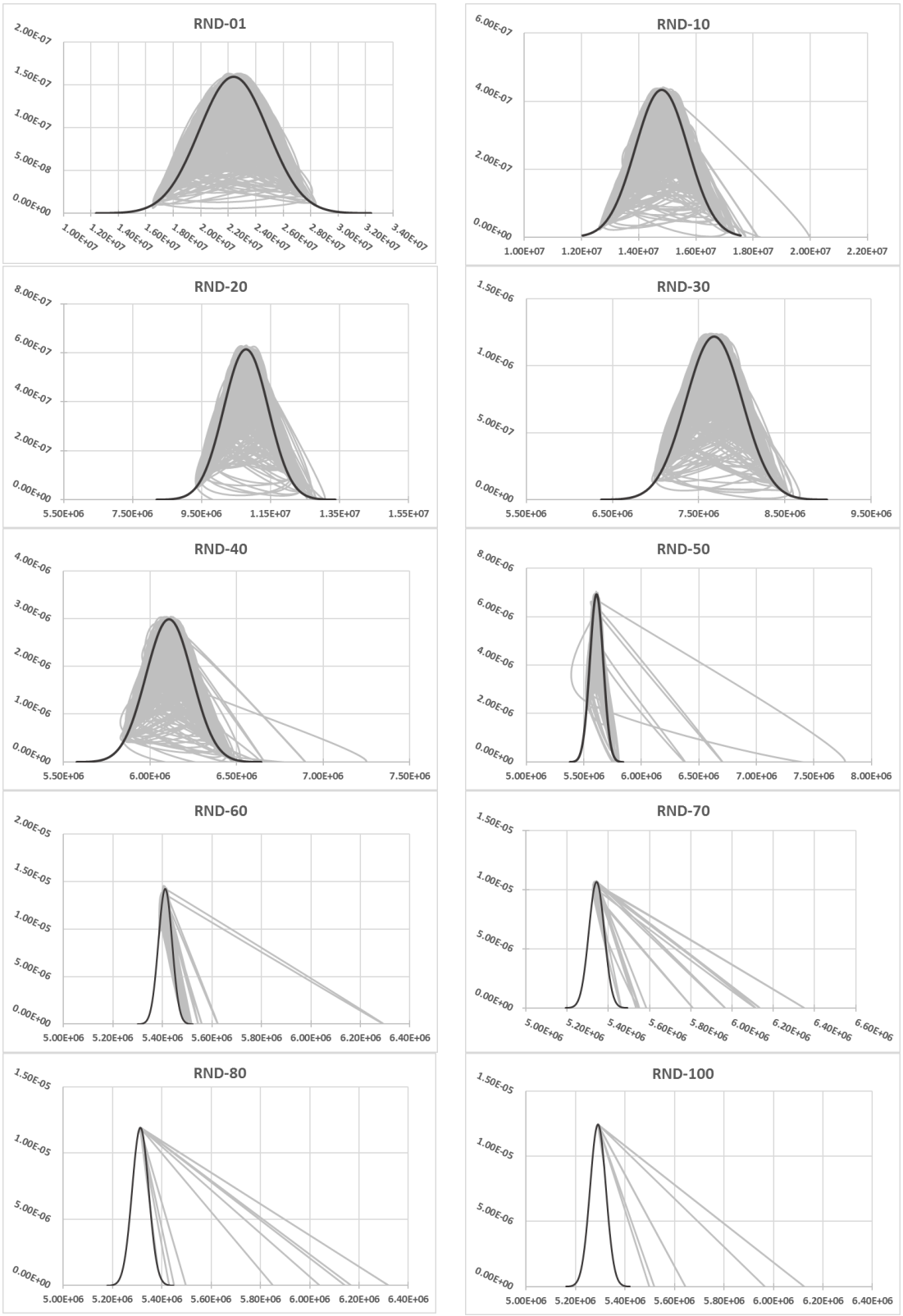


Figure C- 3, the candidate pool's population's good fit into the normal bell curve for PPL

Table C- 1, candidate pool's demographic heat map for the PPL-10000-6000.

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	19	166	801	1812	2274	871	57	0	0.05289	-0.41026
10	31	165	725	1821	2383	803	67	5	0.62789	-0.38949
20	15	185	738	1848	2316	807	82	9	0.32507	-0.24723
30	31	210	692	1821	2397	767	80	2	0.45859	-0.47328
40	1	191	724	1973	2250	762	77	22	1.60664	0.10639
50	0	51	561	2395	2546	397	33	17	366.09403	11.21645
60	0	1	190	3557	1928	236	64	24	380.24981	15.44658
70	0	0	0	3493	2461	21	5	20	439.37948	19.02070
80	0	0	0	5916	58	1	3	22	529.57900	22.13149
90	0	0	0	2989	2994	1	2	14	1219.24259	32.41046
100	0	0	0	5954	23	2	3	18	542.39214	22.59778

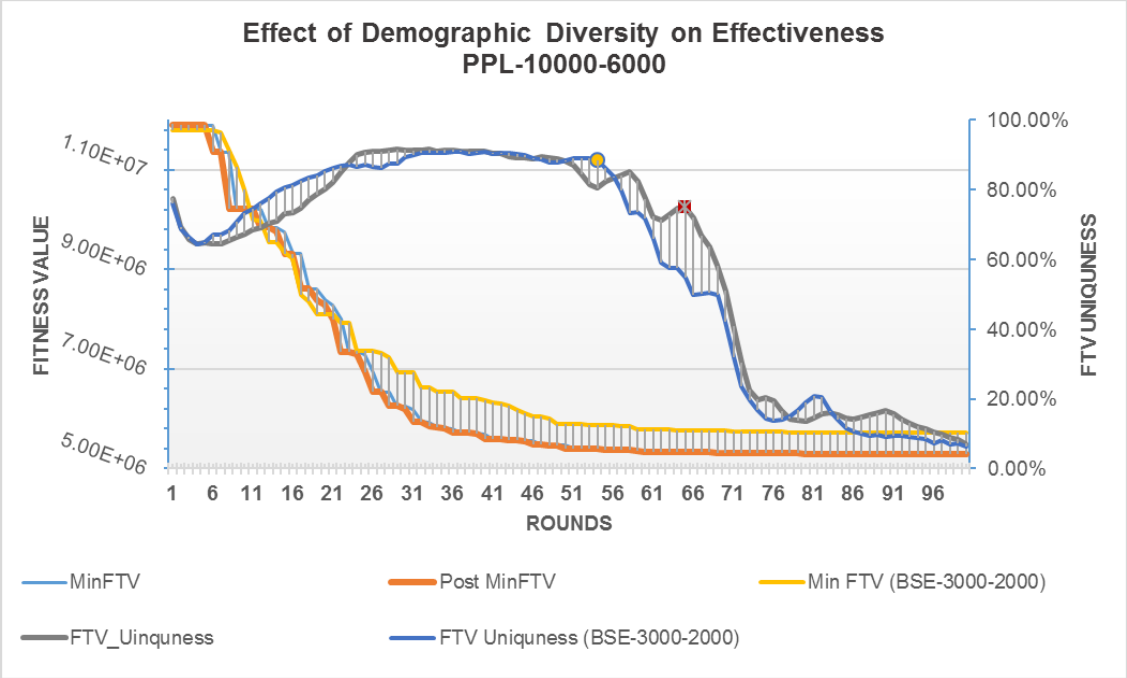


Figure C- 4, effect of demographic diversity on process effectiveness.

Table C- 2, performance benchmark with the base experiment (BSE-3000-2000).

	Best-fit's FTV	RND with Improv.	Qty of Imprv.	Crossover's Contribution Number	Crossover's Contribution Quantity	Mutation's Contribution Number	Mutation's Contribution Quantity
PPL-10000-6000	5.2881064319E+06	70	6.627895045E+06	60	4.936077614E+06	10	1.691817431E+06
Compare to BSE-3000-2000	7.61%	6.00%	8.47%	2.00%	2.88%	4.00%	5.59%

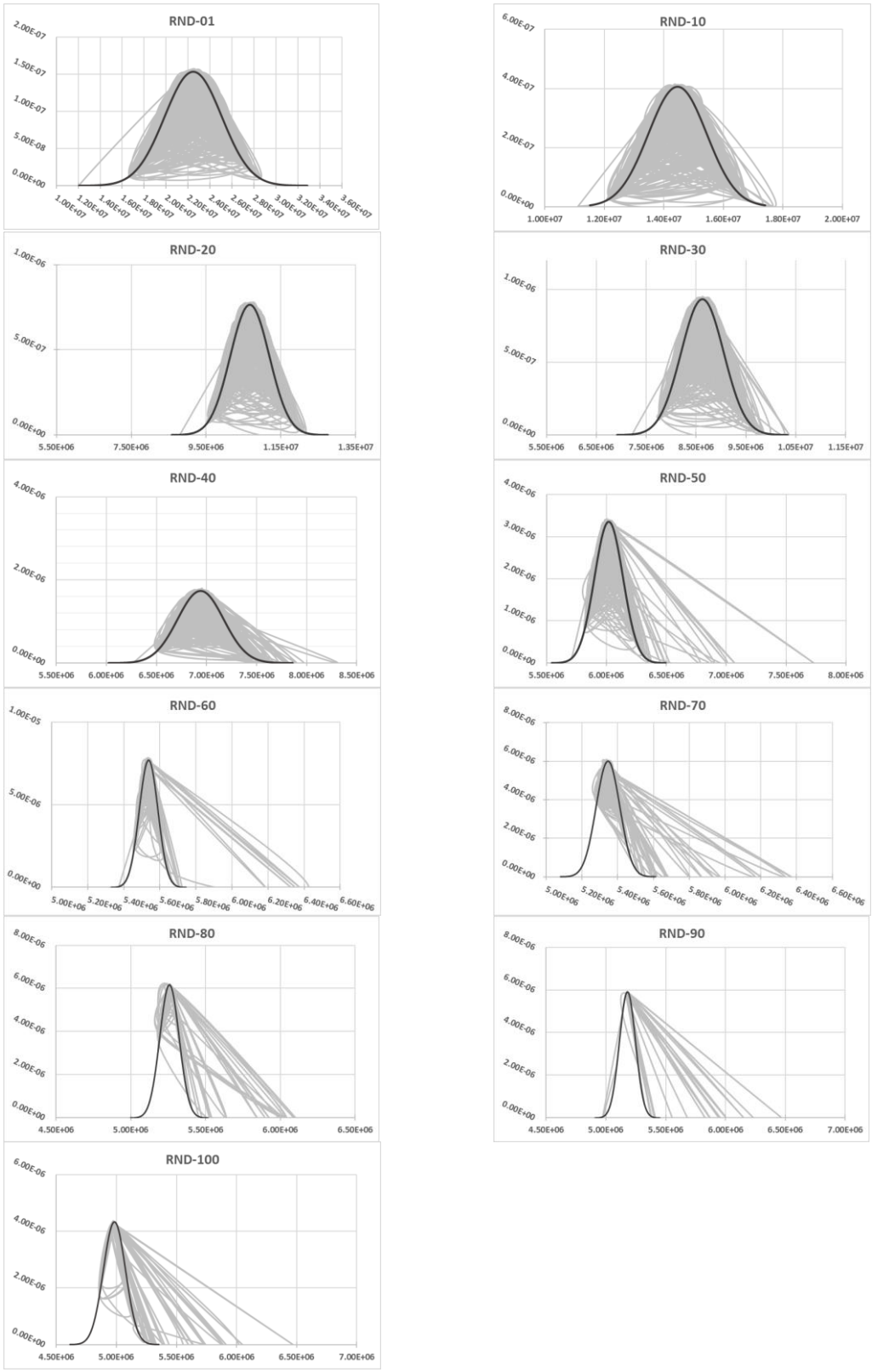


Figure C- 5, the candidate pool's population's good fit into the normal bell curve for MTN60-3000-2000

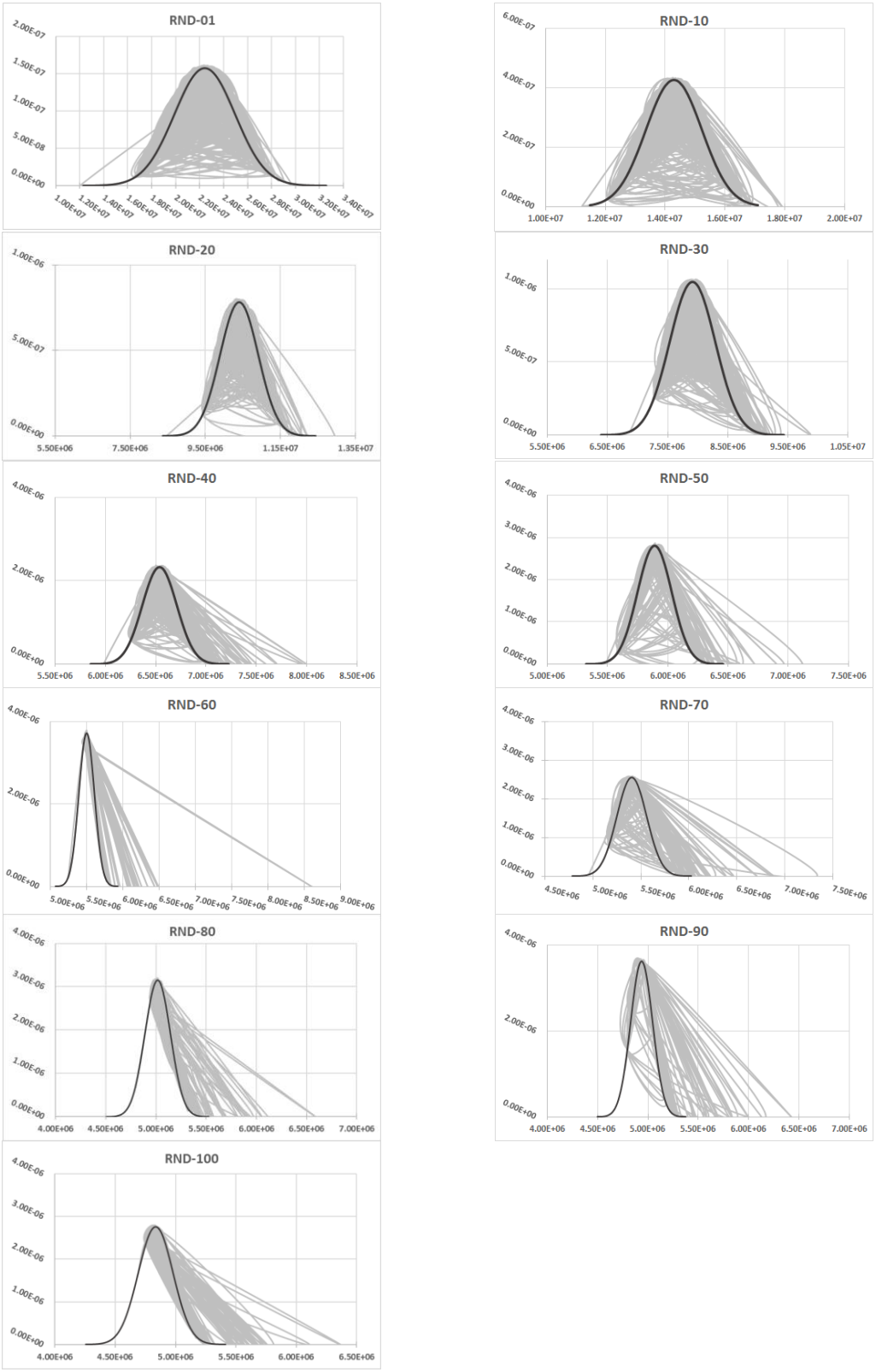


Figure C- 6, the candidate pool's population's good fit into the normal bell curve for MTN100-3000-2000.

Table C- 4, candidate pool demographic heat map table for MTN60-3000-2000

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	6	61	273	594	754	291	21	0	-0.05860	-0.34661
10	6	76	193	641	778	285	18	3	0.35046	-0.41298
20	8	62	249	638	756	250	37	0	0.35658	-0.30660
30	2	65	253	603	793	238	43	3	0.26977	-0.17466
40	0	48	245	701	734	226	33	13	1.25903	0.26488
50	0	33	215	712	855	160	15	10	31.97207	2.80811
60	1	23	131	838	949	39	9	10	146.39339	9.18337
70	0	0	13	1320	584	63	3	17	88.18503	7.16789
80	0	0	134	269	1569	5	4	19	93.32779	8.32519
90	3	1	4	1929	33	10	4	16	174.16786	12.36526
100	0	0	104	1694	33	142	3	24	80.90663	7.43158

Table C- 5, candidate pool demographic heat map table for MTN100-3000-2000

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	7	61	279	575	770	291	17	0	-0.03983	-0.40271
10	6	84	201	650	801	223	32	3	0.61440	-0.31655
20	14	58	240	609	818	218	37	6	0.97086	-0.26061
30	0	26	289	705	677	246	45	12	0.48945	0.35772
40	1	22	180	904	670	168	29	26	9.20880	1.59067
50	0	135	107	207	1419	79	34	19	8.75085	0.63517
60	0	4	1	1595	353	12	10	25	382.90159	15.82381
70	0	22	285	158	1449	36	10	40	28.71702	3.27004
80	0	0	1	1442	259	246	15	37	29.89308	4.34629
90	0	0	195	185	1546	24	21	29	55.11888	6.02855
100	0	0	5	1466	322	107	62	38	20.23140	3.75998

Table C- 3, performance benchmark with the base experiment (BSE-3000-2000).

	Best-fit's FTV	RND with Improv.	Qty of Imprv.	Crossover's Contribution Number	Crossover's Contribution Quantity	Mutation's Contribution Number	Mutation's Contribution Quantity
MTN60-3000-2000	4.8517798049E+06	83	7.187139779E+06	61	4.396639867E+06	22	2.790499912E+06
Compare to BSE-3000-2000	15.23%	19.00%	15.59%	3.00%	-4.85%	16.00%	20.44%
MTN100-3000-2000	4.6822207792E+06	67	7.356698805E+06	45	5.327110690E+06	22	2.029588115E+06
Compare to BSE-3000-2000	18.20%	3.00%	17.54%	-13.00%	7.91%	16.00%	9.63%

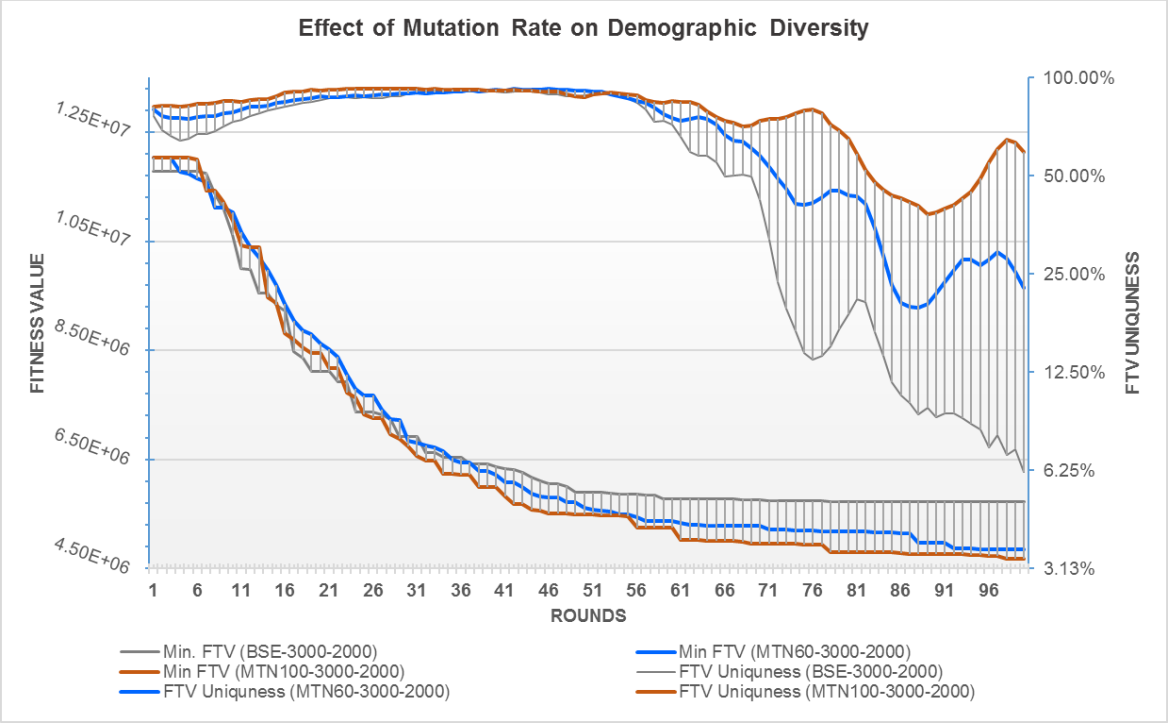


Figure C- 7, the impact of population diversity on optimization's effectiveness.

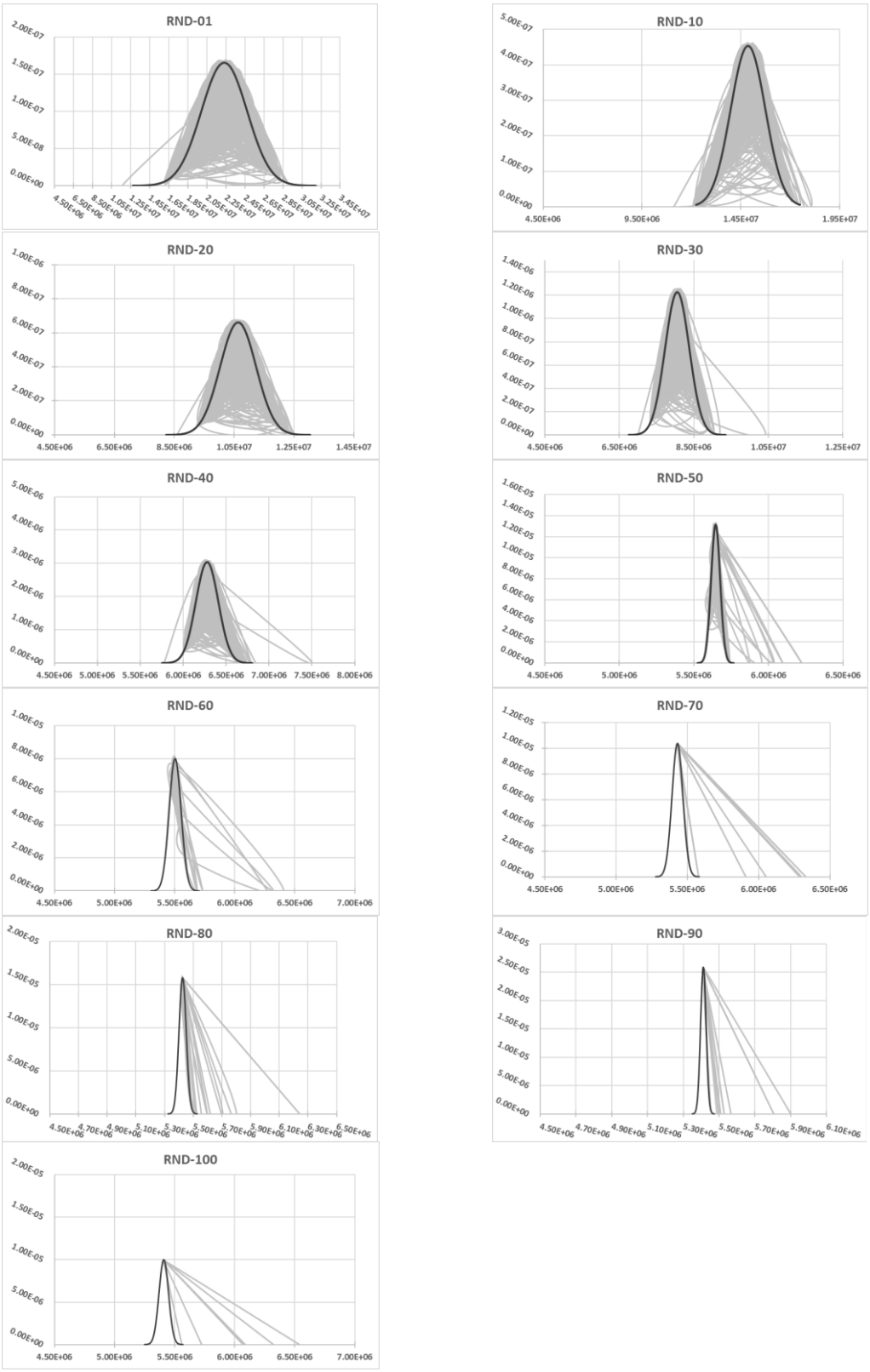


Figure C- 8, the candidate pool's population's good fit into the normal bell curve for ELT05-3000-2000.

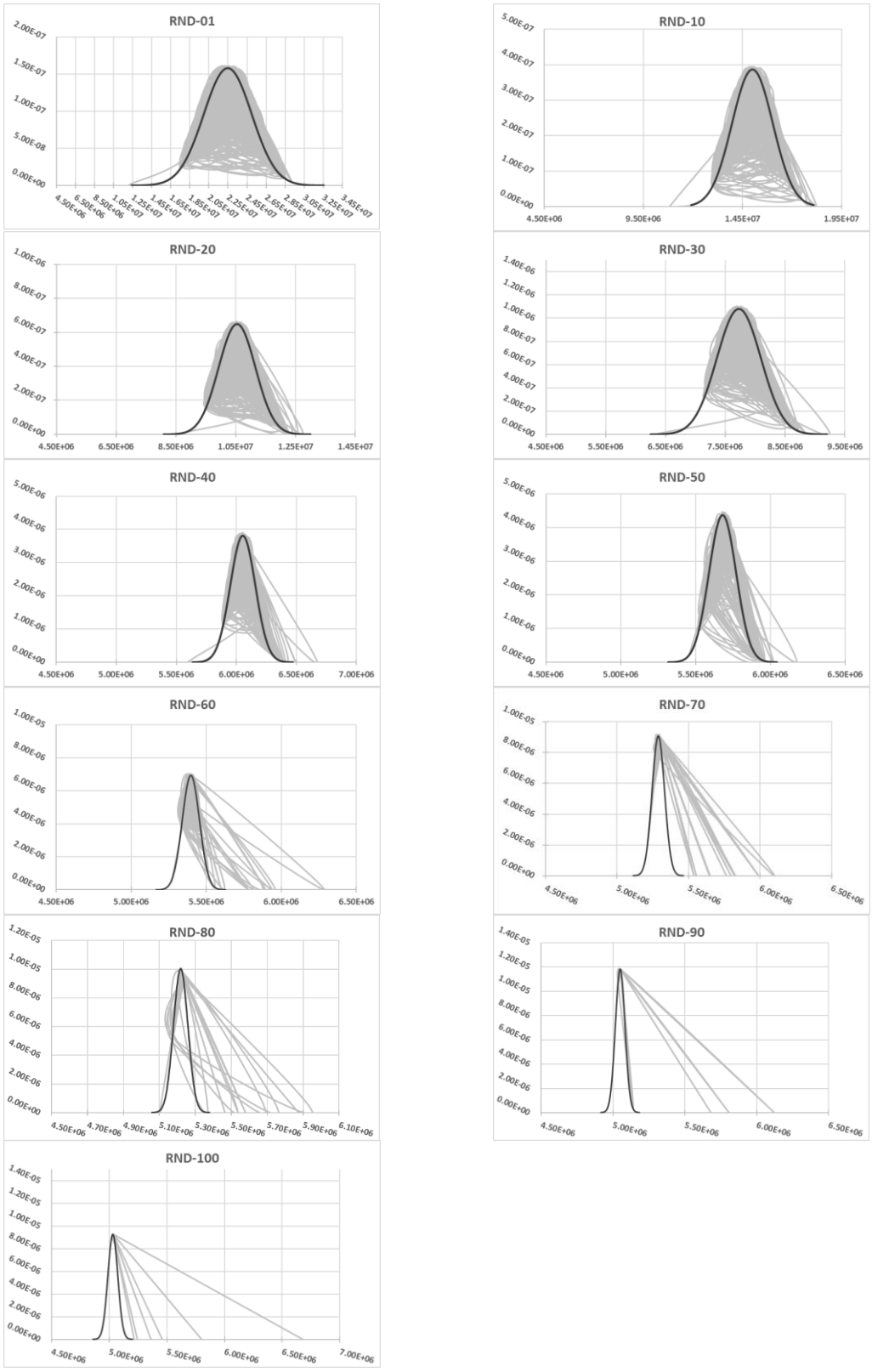


Figure C- 9, the candidate pool’s population's good fit into the normal bell curve for ELT20-3000-2000.

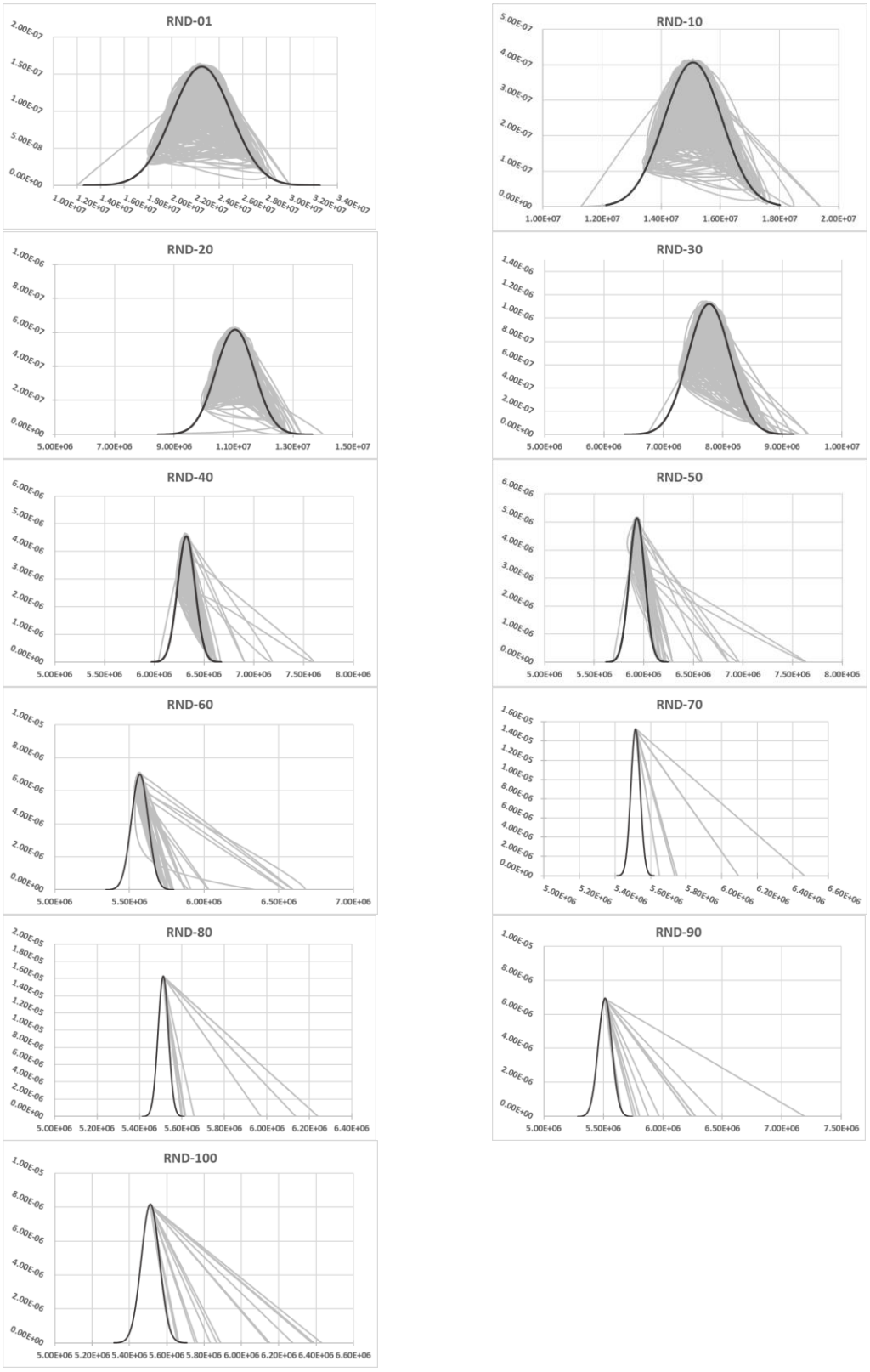


Figure C- 10, the candidate pool's population's good fit into the normal bell curve for ELT30-3000-2000.

Table C- 6, candidate pool demographic heat map table for ELT05-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	5	52	279	612	730	298	24	0	-0.01262	-0.31964
10	22	39	199	664	851	204	18	3	1.38387	-0.69561
20	7	40	279	684	672	283	34	1	-0.07810	-0.15341
30	4	53	269	624	772	251	22	5	1.37354	-0.07116
40	6	37	235	771	660	233	47	11	4.49435	0.55494
50	0	5	178	760	951	91	7	8	121.87315	7.84093
60	0	0	17	1378	182	414	1	8	117.62811	7.55065
70	0	0	0	1921	71	0	2	6	446.39020	20.76511
80	0	0	0	1950	32	1	2	15	611.22941	21.97544
90	0	0	0	1639	350	2	1	8	734.88920	25.64408
100	0	0	0	1988	4	0	0	8	537.69902	22.38272

Table C- 7, candidate pool demographic heat map table for ELT20-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	10	48	271	619	723	308	21	0	0.05184	-0.39262
10	7	52	272	604	762	283	19	1	0.07181	-0.33349
20	0	44	267	688	675	283	40	3	-0.17095	0.02439
30	8	53	228	669	724	294	22	2	0.32442	-0.31292
40	10	47	278	555	847	232	26	5	1.15473	-0.27553
50	0	44	402	231	1222	35	61	5	0.60307	-0.15548
60	0	1	425	603	867	93	1	10	43.77606	3.80864
70	0	0	221	250	1517	0	0	12	166.51565	11.21578
80	0	14	409	64	1502	1	1	9	129.97625	8.21239
90	0	0	0	495	1500	0	2	3	704.61452	25.17148
100	0	0	0	1948	46	0	0	6	1133.00318	31.74041

Table C- 8, candidate pool demographic heat map table for ELT30-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	8	60	268	560	803	290	11	0	0.12182	-0.48556
10	16	45	234	707	662	314	20	2	0.49320	-0.32585
20	5	51	256	643	738	278	26	3	0.21178	-0.21374
30	0	31	319	645	688	278	34	5	0.08637	0.15436
40	1	19	265	786	640	241	41	7	27.62678	2.38944
50	1	16	175	721	977	93	7	10	143.73930	7.54659
60	0	0	0	1311	548	116	13	12	168.52568	9.98625
70	0	0	0	1547	448	0	0	5	1002.82045	30.24749
80	0	0	0	1979	12	1	0	8	671.19466	25.16025
90	0	0	0	1977	7	3	2	11	464.76780	19.74212
100	0	0	0	1977	2	5	2	14	237.63889	14.88553

Table C- 9, performance benchmark with the base experiment (BSE-3000-2000).

				Crossover's Contribution		Mutation's Contribution	
	Best-fit's FTV	RND with Improv.	Qty of Imprv.	Number	Quantity	Number	Quantity
ELT05-3000-2000	5.4062408911E+06	59	6.216254175E+06	48	4.595074543E+06	11	1.621179632E+06
Compare to BSE-3000-2000	5.55%	-5.00%	2.41%	-10.00%	-2.41%	5.00%	4.82%
ELT20-3000-2000	5.0189779096E+06	65	7.091885632E+06	48	6.129160570E+06	17	9.627250628E+05
Compare to BSE-3000-2000	12.31%	1.00%	14.46%	-10.00%	19.52%	11.00%	-5.06%
ELT30-3000-2000	5.5090893287E+06	58	6.509296771E+06	54	5.782873318E+06	4	7.264234538E+05
Compare to BSE-3000-2000	3.75%	-6.00%	6.80%	-4.00%	15.95%	-2.00%	-9.14%

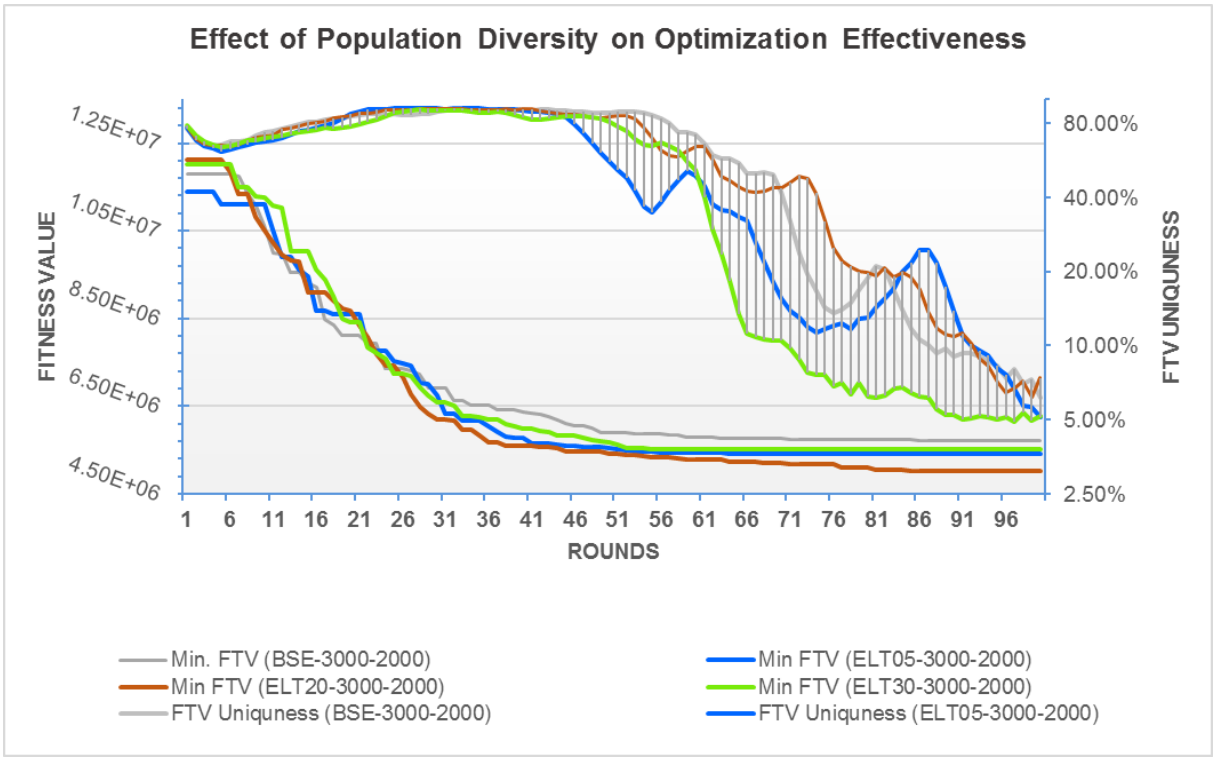


Figure C- 11, effect of population diversity on optimization effectiveness.

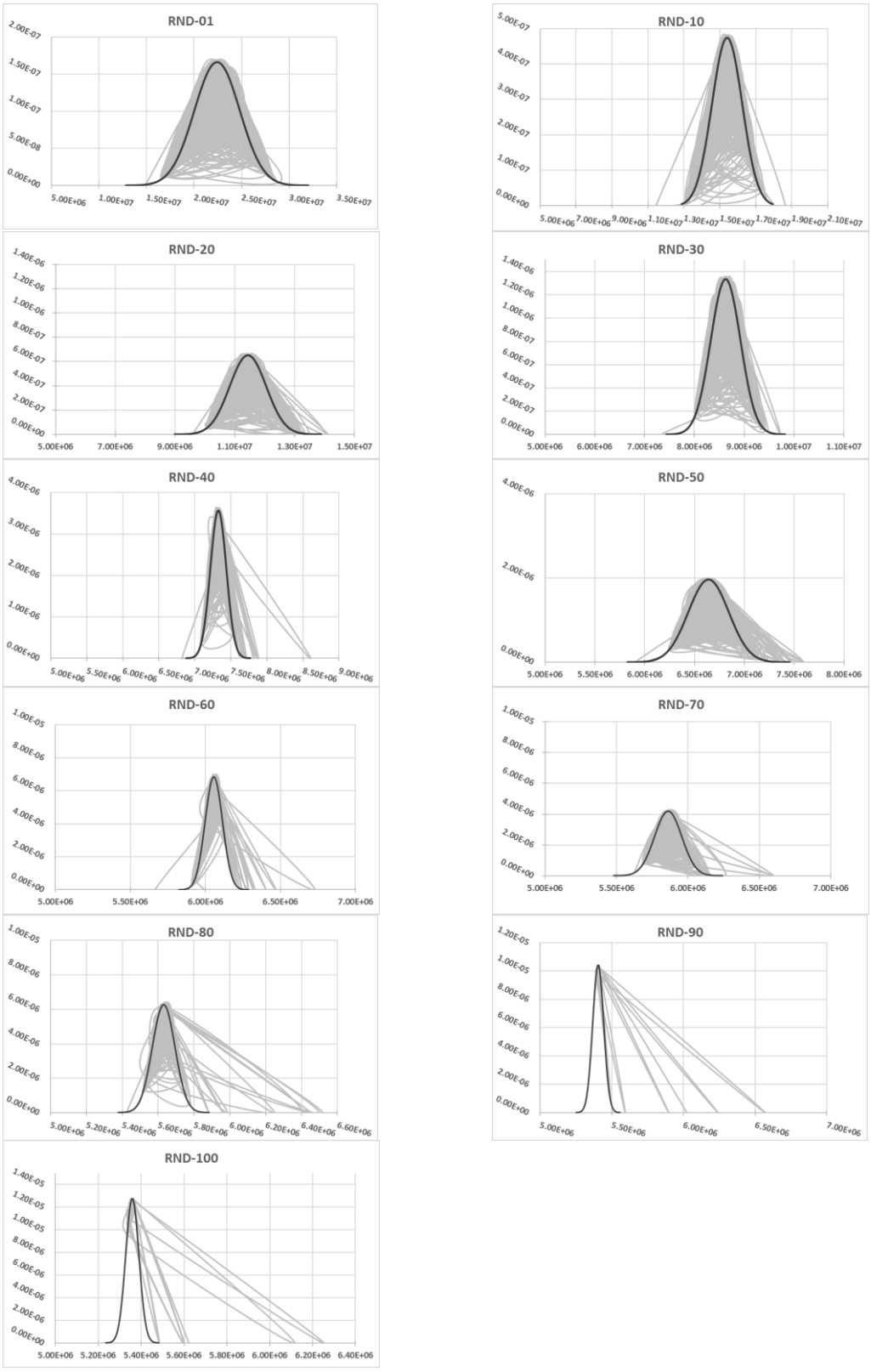


Figure C- 12, the candidate pool's population's good fit into the normal curve for CSO50-3000-2000.

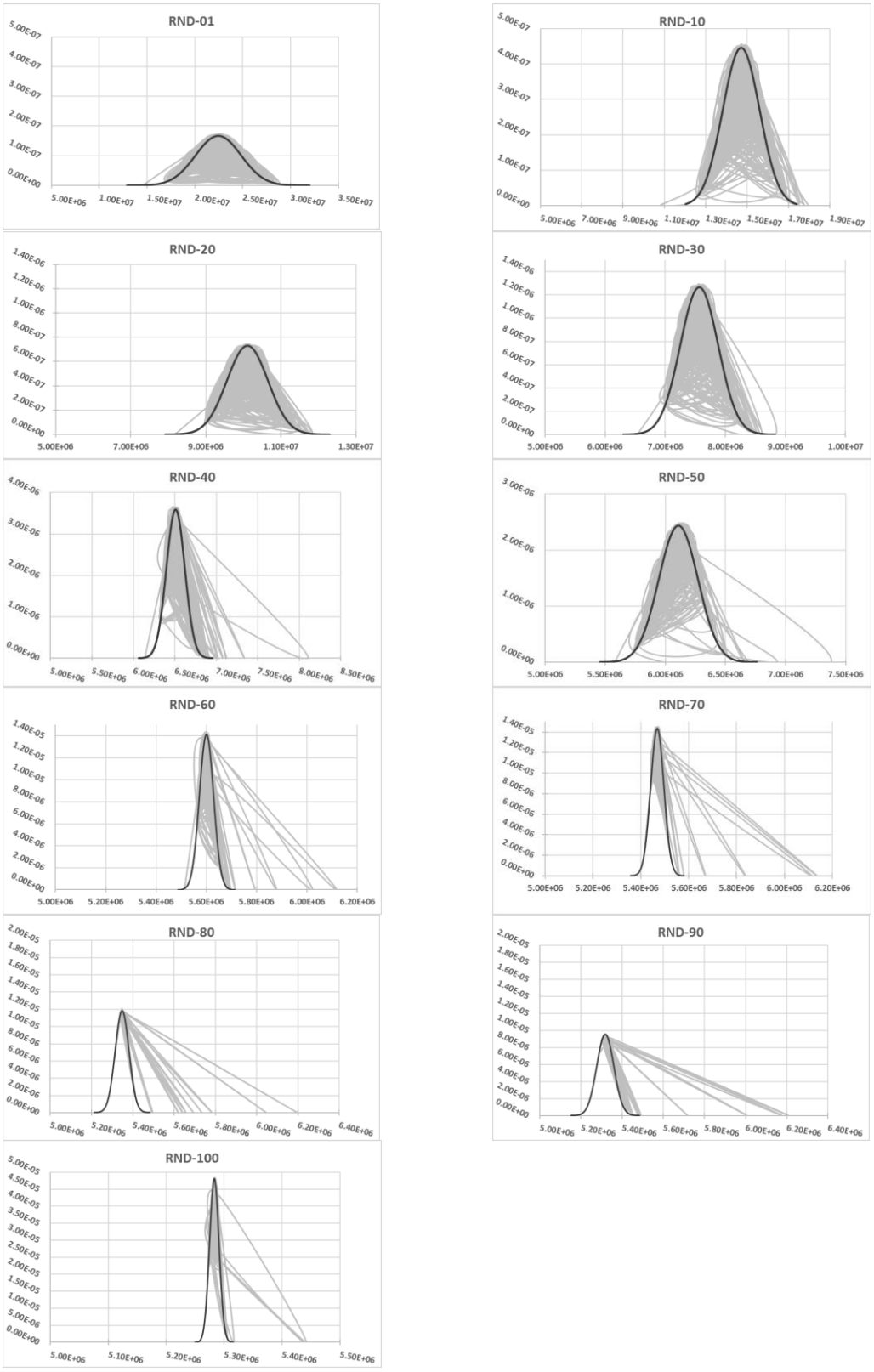


Figure C- 13, the candidate pool’s population's good fit into the normal curve for CSO70-3000-2000.

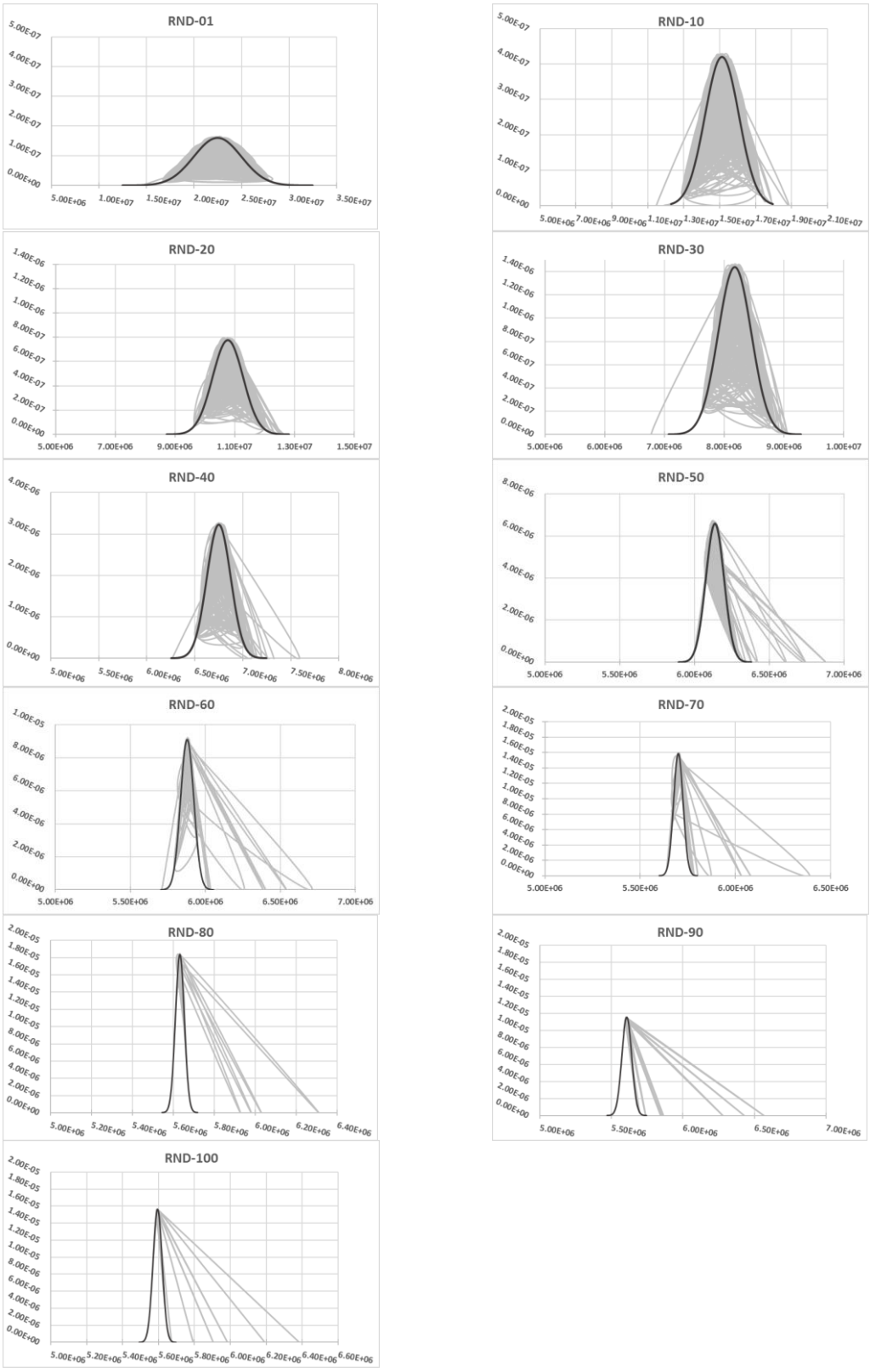


Figure C- 14, the candidate pool's population's good fit into the normal curve for CSORDM-3000-2000.

Table C- 10, candidate pool demographic heat map table for CSO50-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	5	55	273	587	765	297	18	0	-0.19019	-0.37181
10	24	59	176	678	802	246	14	1	1.33605	-0.79647
20	0	62	247	678	730	240	39	4	0.37926	-0.07294
30	9	63	241	571	845	244	26	1	0.55007	-0.50565
40	21	63	180	621	945	145	15	10	11.01786	0.30533
50	1	39	306	601	792	224	26	11	0.71200	0.10903
60	51	76	33	594	1209	22	6	9	16.91059	-0.79733
70	0	26	396	248	1158	156	10	6	2.97045	0.03903
80	2	37	207	515	1164	65	2	8	65.82639	4.76867
90	0	0	0	902	1091	1	1	5	593.33386	22.64120
100	0	0	1	586	1405	0	0	8	512.01716	19.74452

Table C- 11, candidate pool demographic heat map table for CSO70-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	1	81	257	586	770	289	16	0	-0.17614	-0.39297
10	8	52	249	686	722	248	29	6	0.48781	-0.19356
20	10	59	231	635	785	250	27	3	0.90033	-0.40227
30	8	74	197	653	778	264	25	1	0.47216	-0.44794
40	5	63	255	540	968	139	25	5	7.58071	0.18305
50	0	8	386	496	935	125	34	16	6.57879	0.75784
60	1	10	212	793	684	284	9	7	55.12954	3.42009
70	0	0	0	1073	919	1	0	7	816.19171	25.72235
80	0	0	0	1924	69	2	0	5	617.43463	23.75945
90	0	0	0	1981	5	1	0	13	233.68392	14.74068
100	0	0	0	1980	8	4	0	8	389.88574	19.25251

Table C- 12, candidate pool demographic heat map table for CSORDM-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ2	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ2	Excess Kurtosis	Skewness
1	1	73	264	583	752	317	10	0	-0.33927	-0.41296
10	5	67	270	602	734	305	16	1	-0.01238	-0.38941
20	4	63	236	641	771	249	32	4	0.30442	-0.30712
30	5	58	262	655	702	286	30	2	0.18780	-0.26387
40	10	56	279	521	907	205	16	6	1.64557	-0.21625
50	0	1	77	1115	445	328	27	7	21.91252	2.73389
60	6	64	152	400	1322	47	0	9	114.48853	6.57196
70	0	42	240	331	1372	3	3	9	320.80990	12.90377
80	0	0	0	1180	799	17	0	4	577.20643	21.52699
90	0	0	0	1247	743	0	3	7	539.51152	21.96866
100	0	0	0	1782	209	3	0	6	649.41819	24.22196

Table C- 13, performance benchmark with the base experiment (BSE-3000-2000).

				Crossover's Contribution		Mutation's Contribution	
	Best-fit's FTV	RND with Improv.	Qty of Imprv.	Number	Quantity	Number	Quantity
CSO50-3000-2000	5.3229855939E+06	63	9.612152675E+06	55	9.223240063E+06	8	3.889126122E+05
Compare to BSE-3000-2000	7.00%	-1.00%	36.89%	-3.00%	46.59%	2.00%	-9.70%
CSO70-3000-2000	5.1349769900E+06	78	9.594994108E+06	67	7.880048710E+06	11	1.714945397E+06
Compare to BSE-3000-2000	10.29%	14.00%	36.78%	9.00%	32.68%	5.00%	4.10%
CSORDM-3000-2000	5.5704300887E+06	69	9.159541009E+06	60	8.753276674E+06	9	4.062643350E+05
Compare to BSE-3000-2000	2.68%	5.00%	33.77%	2.00%	43.76%	3.00%	-9.99%

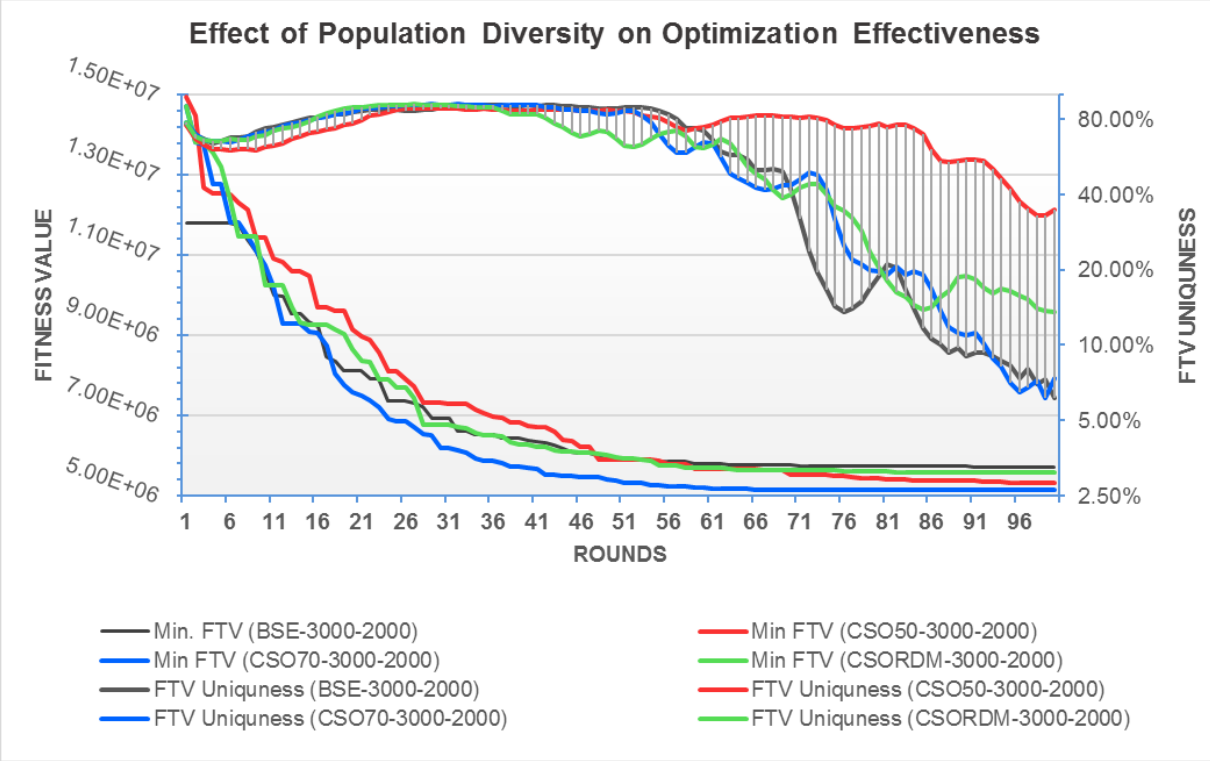


Figure C- 15, effect of population diversity on optimization effectiveness.

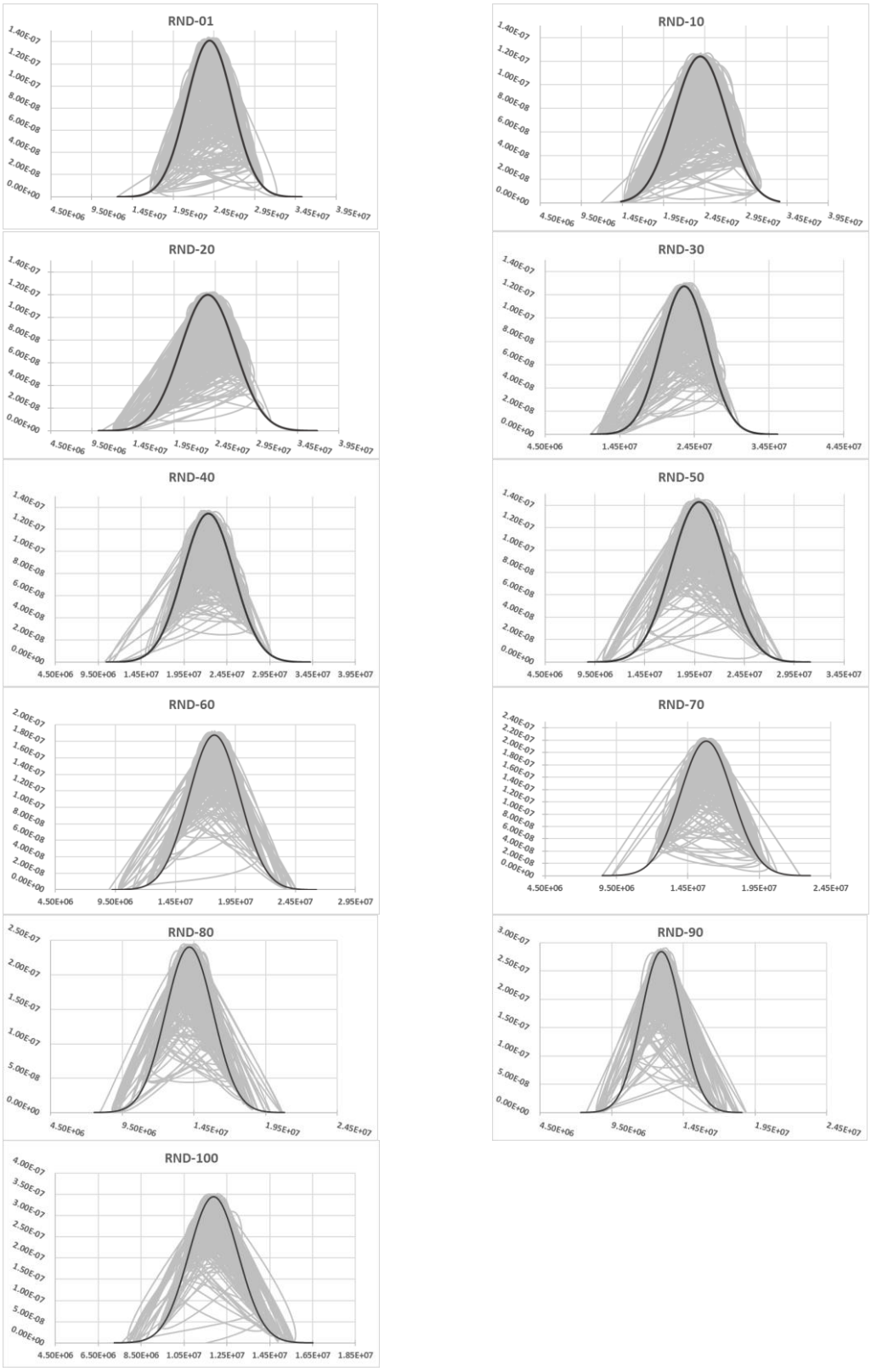


Figure C- 16, the candidate pool's population's good fit into the normal curve for TNS01-3000-2000.

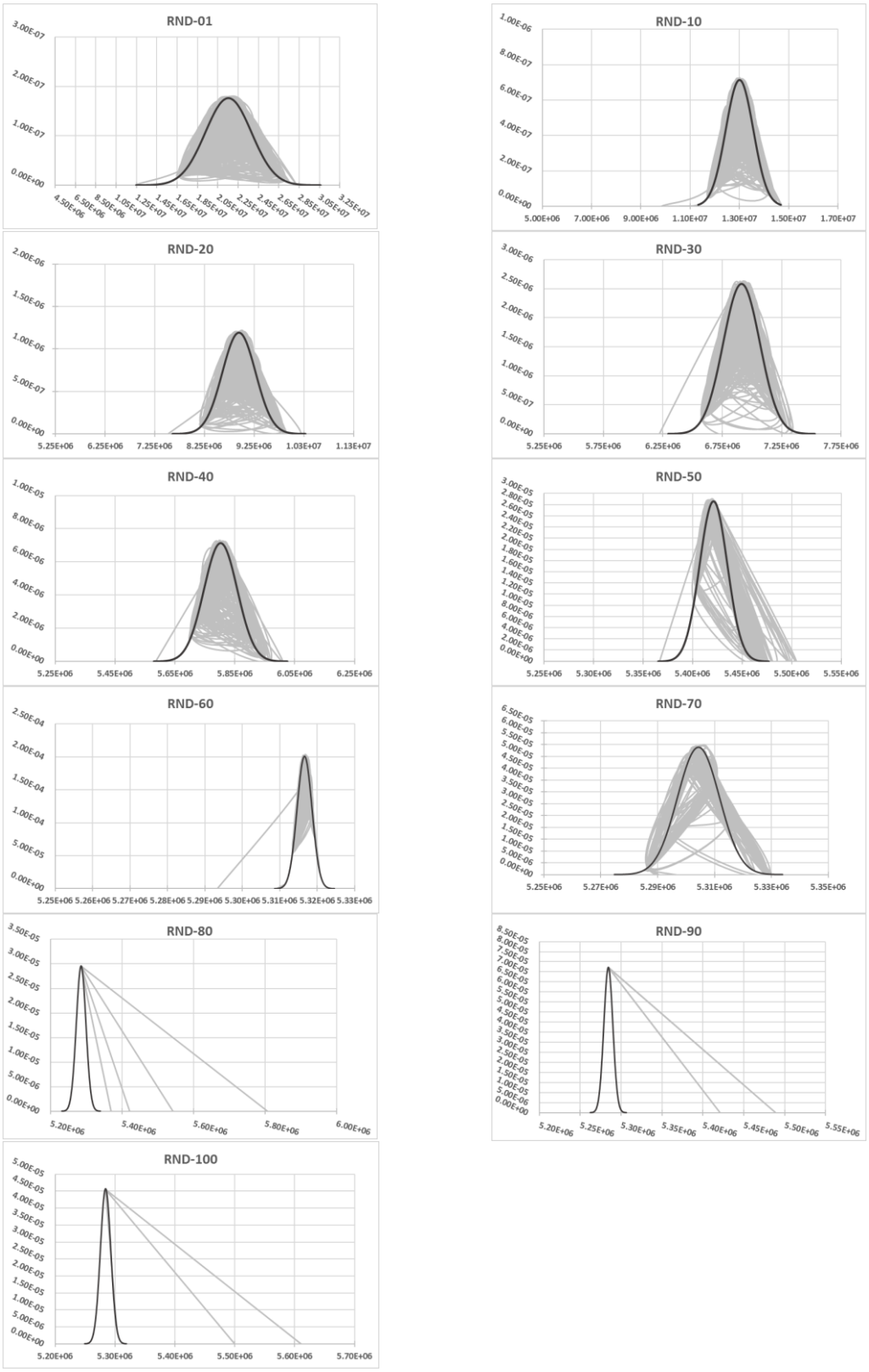


Figure C- 17, the candidate pool’s population's good fit into the normal curve for TNS03-3000-2000.

Table C- 14, candidate pool demographic heat map table for TNS01-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ	Excess Kurtosis	Skewness
1	7	67	245	606	759	301	15	0	0.19126	-0.53305
10	16	85	196	542	915	237	9	0	1.02546	-0.96593
20	53	36	186	544	952	228	1	0	2.21037	-1.30814
30	61	15	146	628	905	242	3	0	3.91698	-1.52445
40	50	4	152	719	820	251	4	0	4.09442	-1.39945
50	74	2	70	850	733	248	22	1	3.55020	-1.15115
60	74	8	70	671	1025	135	15	2	6.21526	-1.90716
70	69	2	110	716	888	195	19	1	4.62969	-1.51916
80	80	6	85	742	872	182	29	4	3.41314	-1.04509
90	78	6	77	752	898	152	25	12	4.27855	-1.02605
100	81	9	46	793	877	159	29	6	4.42038	-1.27638

Table C- 15, candidate pool demographic heat map table for TNS03-3000-2000.

Opt. RND	Less Than μ - 3σ	μ - 3σ	μ - 2σ	μ - σ	μ + σ	μ + 2σ	μ + 3σ	Greater Than μ + 3σ	Excess Kurtosis	Skewness
1	3	45	273	631	725	298	25	0	-0.19550	-0.24450
10	13	43	243	619	794	261	26	1	1.04323	-0.51280
20	10	64	208	702	689	302	24	1	0.71397	-0.41995
30	23	65	211	562	870	250	19	0	1.21399	-0.77529
40	11	36	389	335	1057	149	21	2	0.33080	-0.49868
50	18	14	96	1242	467	42	29	92	7.49782	1.75966
60	24	21	179	493	1245	37	1	0	28.46729	-3.93759
70	0	126	42	1120	590	68	3	51	2.56253	0.08811
80	0	0	0	1996	0	0	0	4	1188.28990	32.97072
90	0	0	0	1998	0	0	0	2	1144.59712	33.33004
100	0	0	0	1995	2	0	1	2	1145.79176	33.29974

Table C- 16, performance benchmark with the base experiment (BSE-3000-2000).

				Crossover's Contribution		Mutation's Contribution	
	Best-fit's FTV	RND with Improv.	Qty of Imprv.	Number	Quantity	Number	Quantity
TSN01-3000-2000	7.5795074655E+06	30	5.087907823E+06	12	1.907736376E+06	18	3.180171447E+06
Compare to BSE-3000-2000	32.42%	-34.00%	-19.23%	-46.00%	-55.76%	12.00%	36.53%
TSN03-3000-2000	5.2840504471E+06	63	7.113153725E+06	53	5.206899626E+06	10	1.906254099E+06
Compare to BSE-3000-2000	7.68%	-1.00%	14.72%	-5.00%	6.50%	4.00%	8.22%

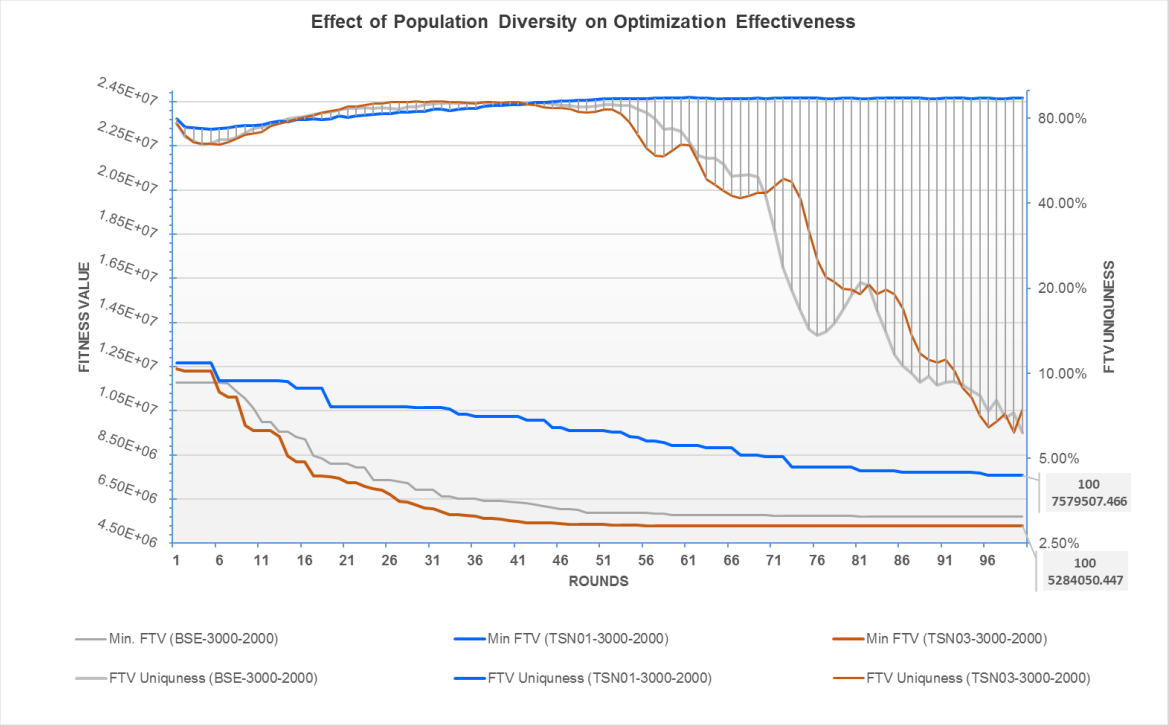


Figure C- 18, the impact of population diversity on optimization's effectiveness.

Table C- 17, assignment table presenting the solution with no constraints.

Place Code	Description	Min. Area	Max. Area
MBFF	**First floor, the middle building**	**110**	**150**
AD008	Information office	18	20
GS012	Elevator	0	0
GS023	Cafe	16	18
GS027	Northern Entrance	30	50
GS028	Main Entrance	30	50
GS041	East Entrance	10	12
MBGF	**Ground floor, middle building**	**550**	**600**
GS005	Maintenance Workshop 1	20	25
GS006	Maintenance Workshop 2	20	25
GS009	Warehouse Office	12	15
GS010	Washroom	10	12
GS019	Pharmacy	90	110
SC011	Warehouse at specialty clinic	10	12
SG011	Surgery Dvision	230	265
WH001	Warehouse 1	20	25
WH002	Warehouse 2	20	25
WH003	Warehouse 3	20	25
WH004	Warehouse 4	20	25
WH005	Warehouse 5	20	25
WH006	Warehouse 6	20	25
WH007	Warehouse 7	20	25
WH008	Warehouse 8	10	12
NBBS	**Basement, the north building**	**550**	**680**
GS011	Service Lobby 1	22	24
GS017	Washroom at Mid. building, ground floor	9	11
GS033	Service Lobby in ground floor middle building	65	70
GS035	Patio	60	70
NBFF	**First floor, north building**	**780**	**900**
GS001	Car Parking	600	620
GS003	Heating	30	35
GS004	Watertank	45	50
GS008	Electrical Department	33	38
GS014	Washroom 2	19	23
GS026	Car Entrance	50	70

Continued on next page

Table C- 17, assignment table presenting the solution with no constraints (continued and end)

Place Code	Description	Min. Area	Max. Area
NBGF	**Ground floor, noth building**	**535**	**680**
GC001	General Clinic	360	395
GS046	Stairways on first floor	24	24
SBBS	**Basment, south building**	**600**	**670**
GS002	Washing Center	40	45
GS007	Central Sterilization Department	40	45
GS021	Washroom in physiotherapy clinic	12	14
NH001	Nephrology & Hemodialysis Clinic	215	245
PB012	Medical Test Lab (Blood & Pathology)	270	300
SBFF	**First floor, south building**	**760**	**900**
GS013	Service Lobby 2	22	24
GS018	Washroom at southern building, ground floor	17	19
GS022	Washroom in Nephrology clinic	12	14
GS031	Service Lobby in Outpatient Clinic	10	12
GS036	Stairways	23	26
GS037	Elevators	15	15
GS039	Service Lobby on the ground floor, south building	230	270
GS044	Patient waiting room 2 on first floor, north building	22	25
GS047	Elevators on first floor	15	15
PH009	Physiotherapy Clinic	385	440
SBGF	**Ground floor, south building**	**980**	**1024**
AD001	Residences supervisory room	19	23
AD002	Shift rotation room	22	25
AD003	General Manager's office	20	25
AD006	Administration department	140	160
AD007	On-call room	20	24
AD009	Archive	16	18
DC009	Medical Imaging Test Center	150	175
GS020	Restaurant	140	160
GS034	Washroom on ground floor, middle building	10	10
OC004	Police station	10	12
OC007	Outpatient clinic	221	260
SC012	Specialized Clinic	195	220

Table C- 18, list of the restricted assignment constraints (RAC List)

Constraint ID	Dept(s) ID	MUST (NOT)	Const. Type	Floor(s) ID	Active/Inactive
CN000002	D00010	TRUE	RAC	F01;F04	TRUE
CN000003	D00011	TRUE	RAC	F01	TRUE
CN000004	D00013	TRUE	RAC	F01	TRUE
CN000005	D00113	TRUE	RAC	F04	TRUE
CN000006	D00114	TRUE	RAC	F02	TRUE
CN000007	D00115	TRUE	RAC	F05	TRUE
CN000008	D00136	TRUE	RAC	F05	TRUE
CN000010	D00127	TRUE	RAC	F02	TRUE
CN000014	D00134	TRUE	RAC	F05	TRUE
CN000018	D00129	TRUE	RAC	F07	TRUE
CN000019	D00132	TRUE	RAC	F07	TRUE
CN000020	D00131	TRUE	RAC	F07	TRUE
CN000021	D00128	TRUE	RAC	F07	TRUE
CN000022	D00130	TRUE	RAC	F07	TRUE
CN000024	D00141	TRUE	RAC	F08	TRUE
CN000025	D00142	TRUE	RAC	F08	TRUE
CN000028	D00002	TRUE	RAC	F01;F04	TRUE
CN000029	D00003	TRUE	RAC	F01;F04	TRUE
CN000030	D00004	TRUE	RAC	F01;F04	TRUE
CN000031	D00116	TRUE	RAC	F02;F05;F07	TRUE
CN000032	D00008	TRUE	RAC	F01;F04	TRUE
CN000037	D00001	TRUE	RAC	F01;F04	TRUE
CN000038	D00005	TRUE	RAC	F01;F04	TRUE
CN000039	D00006	TRUE	RAC	F01;F04	TRUE
CN000040	D00007	TRUE	RAC	F01;F04	TRUE
CN000041	D00009	TRUE	RAC	F01;F04	TRUE
CN000042	D00052	TRUE	RAC	F07	TRUE
CN000042	D00139	TRUE	RAC	F05	TRUE
CN000043	D00057	TRUE	RAC	F05	TRUE
CN000043	D00012	TRUE	RAC	F04	TRUE

Table C- 19, optimizer parameters configuration table

Parameter Name	Value	Description
Number of service points	60	
Number of floors	8	
Over-assignment allowed	FALSE	
Strategy for departments area	Min Area	
Strategy for floor area availability	Min Area	
Horizontal transport cost	1 unit	
Vertical transport cost	10 unit	
Initial population	3000	
Genotype max attempts	8	Maximum attempts to generate a feasible genotype.
Selected population in each round	2000	
Max. round to create initial population	25000	Maximum round to generate the initial population
Tournament size	2	
Elite rate	0.01	1% of base population in each round
Crossover rate	0.5	60% of first parent DNA with 40% of second parent DNA
Mutation rate	1.0	30% of candidate population are mutated each round
Crossover mode	Double	
Population screening rates	N.A.	The population screening operator does not apply to the optimization process.
Floor over-assignment rates	N.A.	

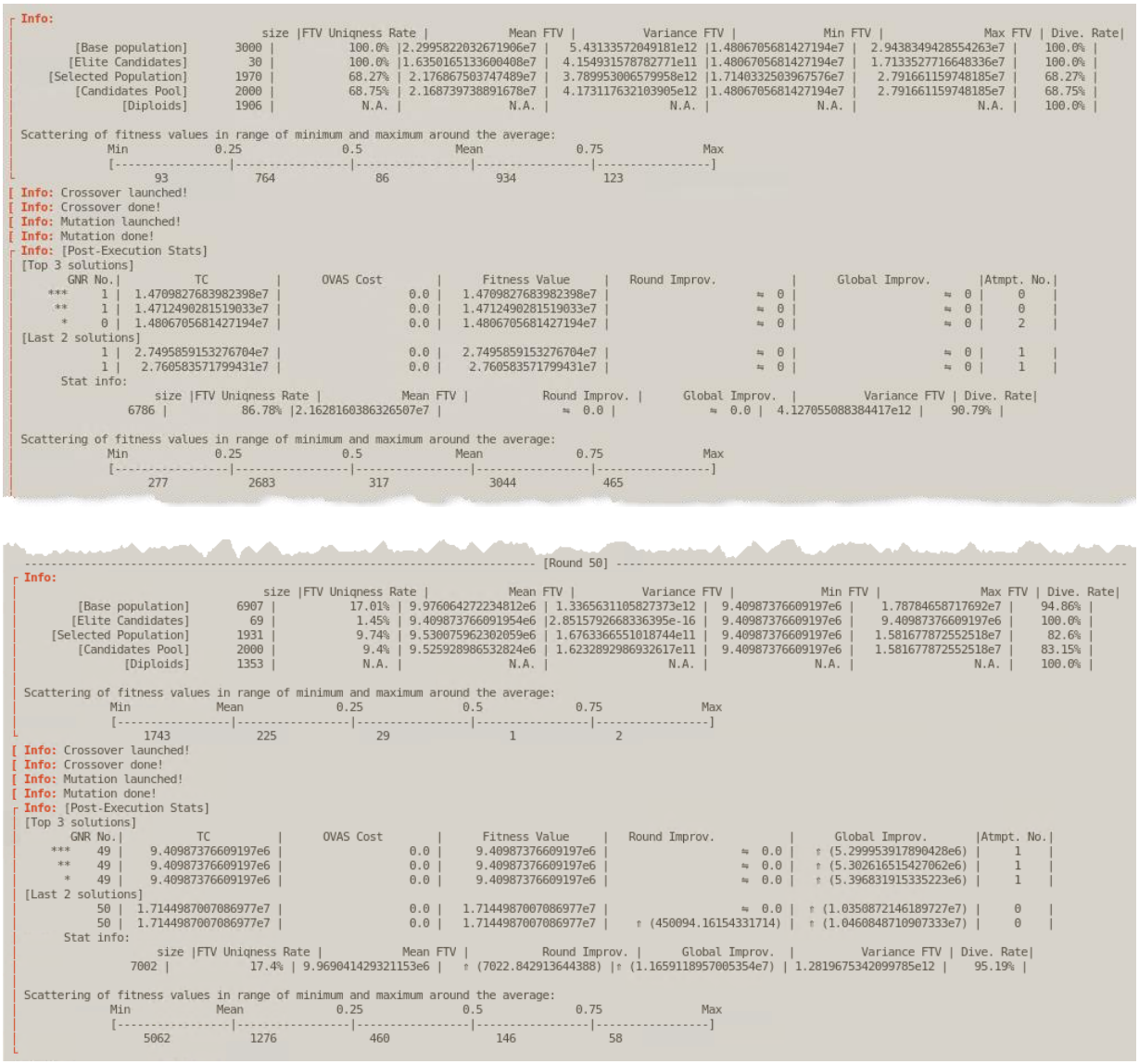


Figure C- 19, summary reports of the first and last labs after applying RACs

Table C- 20, assignment table presenting the best solution after applying RACs

SP ID	SP Code	Description	Min. Area	Max. Area
MBFF	**F08**	**First floor, the middle building**	**110**	**150**
AD002	D00033	Shift rotation room	22	25
AD008	D00055	Information office	18	20
GS023	D00090	Cafe	16	18
GS046	D00141	Stairways on first floor	24	24
GS047	D00142	Elevators on first floor	15	15
OC004	D00042	Police station	10	12
MBGF	**F07**	**Ground floor, middle building**	**550**	**600**
AD001	D00031	Residences supervisory room	19	23
AD003	D00035	General Manager's office	20	25
AD007	D00054	On-call room	20	24
AD009	D00056	Archive	16	18
GS017	D00052	Washroom at Mid. building, ground floor	9	11
GS033	D00128	Service Lobby in ground floor middle building	65	70
GS034	D00129	Washroom on ground floor, middle building	10	10
GS035	D00130	Patio	60	70
GS036	D00131	Stairways	23	26
GS037	D00132	Elevators	15	15
PB012	D00122	Medical Test Lab (Blood & Pathology)	270	300
NBBS	**F01**	**Basement, the north building**	**550**	**680**
GS002	D00002	Washing Center	40	45
GS003	D00003	Heating	30	35
GS004	D00004	Watertank	45	50
GS005	D00005	Maintenance Workshop 1	20	25
GS006	D00006	Maintenance Workshop 2	20	25
GS007	D00007	Central Sterilization Department	40	45
GS008	D00008	Electrical Department	33	38
GS009	D00009	Warehouse Office	12	15
GS010	D00010	Washroom	10	12
GS011	D00011	Service Lobby 1	22	24
GS013	D00013	Service Lobby 2	22	24
GS014	D00032	Washroom 2	19	23
GS021	D00084	Washroom in physiotherapy clinic	12	14
GS022	D00085	Washroom in Nephrology clinic	12	14
SC011	D00069	Warehouse at specialty clinic	10	12
WH001	D00014	Warehouse 1	20	25
WH002	D00015	Warehouse 2	20	25
WH003	D00016	Warehouse 3	20	25
WH004	D00017	Warehouse 4	20	25
WH005	D00018	Warehouse 5	20	25
WH006	D00019	Warehouse 6	20	25
WH007	D00020	Warehouse 7	20	25
WH008	D00034	Warehouse 8	10	12

Continued on next page

Table C- 20Table C- 20, assignment table presenting the best solution after applying RACs

(continued and end)

SP ID	SP Code	Description	Min. Area	Max. Area
NBFF	**F03**	**First floor, north building**	**780**	**900**
AD006	D00053	Administration department	140	160
GC001	D00123	General Clinic	360	395
SG011	D00117	Surgery Dvision	230	265
NBGF	**F02**	**Ground floor, noth building**	**535**	**680**
GS027	D00114	Northern Entrance	30	50
GS031	D00127	Service Lobby in Outpatient Clinic	10	12
NH001	D00121	Nephrology & Hemodialysis Clinic	215	245
SC012	D00119	Specialized Clinic	195	220
SBBS	**F04**	**Basment, south building**	**650**	**670**
GS001	D00001	Car Parking	580	600
GS012	D00012	Elevator	0	0
GS026	D00113	Car Entrance	50	70
SBFF	**F06**	**First floor, south building**	**760**	**900**
PH009	D00120	Physiotherapy Clinic	385	440
SBGF	**F05**	**Ground floor, south building**	**980**	**1024**
DC009	D00118	Medical Imaging Test Center	150	175
GS018	D00057	Washroom at southern building, ground floor	17	19
GS019	D00070	Pharmacy	90	110
GS020	D00071	Restaurant	140	160
GS028	D00115	Main Entrance	30	50
GS039	D00134	Service Lobby on the ground floor, south building	230	270
GS041	D00136	East Entrance	10	12
GS044	D00139	Patient waiting room 2 on first floor, north building	22	25
OC007	D00116	Outpatient clinic	221	260

Table C- 21, list of the adjacency constraints (AJC List)

Constraint ID	Dept(s) ID	MUST (NOT)	Const. Type	Floor(s) ID	Active/Inactive
CN000046	D00055;D00115	TRUE	AJC	NaN	TRUE
CN000047	D00042;D00114; D00116;D00127	TRUE	AJC	NaN	TRUE
CN000048	D00084;D00120	TRUE	AJC	NaN	TRUE
CN000049	D00085;D00121	TRUE	AJC	NaN	TRUE
CN000050	D00069;D00119	TRUE	AJC	NaN	TRUE
CN000051	D00070;D00115	TRUE	AJC	NaN	TRUE
CN000052	D00071;D00115	TRUE	AJC	NaN	TRUE

Table C- 22, assignment table presenting the best solution after applying AJCs (AJC01-3000-2000)

Place ID	Place Code	Description	Min. Area	Max. Area
MBFF	**F08**	**First floor, the middle building**	**110**	**150**
AD002	D00033	Shift rotation room	22	25
GS023	D00090	Cafe	16	18
GS046	D00141	Stairways on first floor	24	24
GS047	D00142	Elevators on first floor	15	15
WH007	D00020	Warehouse 7	20	25
WH008	D00034	Warehouse 8	10	12
MBGF	**F07**	**Ground floor, middle building**	**550**	**600**
AD001	D00031	Residences supervisory room	19	23
AD003	D00035	General Manager's office	20	25
AD006	D00053	Administration department	140	160
AD007	D00054	On-call room	20	24
AD009	D00056	Archive	16	18
DC009	D00118	Medical Imaging Test Center	150	175
GS017	D00052	Washroom at Mid. building, ground floor	9	11
GS033	D00128	Service Lobby in ground floor middle building	65	70
GS034	D00129	Washroom on ground floor, middle building	10	10
GS035	D00130	Patio	60	70
GS036	D00131	Stairways	23	26
GS037	D00132	Elevators	15	15
NBBS	**F01**	**Basement, the north building**	**550**	**680**
GS002	D00002	Washing Center	40	45
GS003	D00003	Heating	30	35
GS004	D00004	Watertank	45	50
GS005	D00005	Maintenance Workshop 1	20	25
GS006	D00006	Maintenance Workshop 2	20	25
GS007	D00007	Central Sterilization Department	40	45
GS008	D00008	Electrical Department	33	38
GS009	D00009	Warehouse Office	12	15
GS010	D00010	Washroom	10	12
GS011	D00011	Service Lobby 1	22	24
GS013	D00013	Service Lobby 2	22	24
GS014	D00032	Washroom 2	19	23
WH001	D00014	Warehouse 1	20	25
WH002	D00015	Warehouse 2	20	25
WH003	D00016	Warehouse 3	20	25
WH004	D00017	Warehouse 4	20	25
WH005	D00018	Warehouse 5	20	25
WH006	D00019	Warehouse 6	20	25

Continued on next page

Table C- 22, , assignment table presenting the best solution after applying AJCs (continued and end)

Place ID	Place Code	Description	Min. Area	Max. Area
NBFF	**F03**	**First floor, north building**	**780**	**900**
GS022	D00085	Washroom in Nephrology clinic	12	14
NH001	D00121	Nephrology & Hemodialysis Clinic	215	245
PB012	D00122	Medical Test Lab (Blood & Pathology)	270	300
SG011	D00117	Surgery Dvision	230	265
NBGF	**F02**	**Ground floor, noth building**	**535**	**680**
GS027	D00114	Northern Entrance	30	50
GS031	D00127	Service Lobby in Outpatient Clinic	10	12
OC004	D00042	Police station	10	12
OC007	D00116	Outpatient clinic	221	260
SC011	D00069	Warehouse at specialty clinic	10	12
SC012	D00119	Specialized Clinic	195	220
SBBS	**F04**	**Basment, south building**	**650**	**670**
GS001	D00001	Car Parking	590	610
GS012	D00012	Elevator	0	0
GS026	D00113	Car Entrance	50	70
SBFF	**F06**	**First floor, south building**	**760**	**900**
GS021	D00084	Washroom in physiotherapy clinic	12	14
PH009	D00120	Physiotherapy Clinic	385	440
SBGF	**F05**	**Ground floor, south building**	**980**	**1024**
AD008	D00055	Information office	18	20
GC001	D00123	General Clinic	360	395
GS018	D00057	Washroom at southern building, ground floor	17	19
GS019	D00070	Pharmacy	90	110
GS020	D00071	Restaurant	140	160
GS028	D00115	Main Entrance	30	50
GS039	D00134	Service Lobby on the ground floor, south building	230	270
GS041	D00136	East Entrance	10	12
GS044	D00139	Patient waiting room 2 on first floor, north building	22	25

Table C- 23, list of the adjacency constraints (AJC List)

Constraint ID	Dept(s) ID	MUST (NOT)	Const. Type	Floor(s) ID	Active/Inactive
CN000046	D00055;D00115	TRUE	AJC	NaN	TRUE
CN000047	D00042;D00114; D00116;D00127	TRUE	AJC	NaN	TRUE
CN000048	D00084;D00120	TRUE	AJC	NaN	TRUE
CN000049	D00085;D00121	TRUE	AJC	NaN	TRUE
CN000050	D00069;D00119	TRUE	AJC	NaN	TRUE
CN000051	D00070;D00115	TRUE	AJC	NaN	TRUE
CN000052	D00071;D00115	TRUE	AJC	NaN	TRUE
CN000053	D00032	TRUE	RAC	F02	TRUE
CN000054	D00009;D00014; D00015;D00016; D00017;D00018; D00019;D00020	TRUE	AJC	NaN	TRUE
CN000055	D00034;D00070	TRUE	AJC	NaN	TRUE
CN000056	D00117;D00122	FALSE	AJC	NaN	TRUE

Table C- 24, assignment table presenting the best solution with after applying AJCs outlined in Table C- 23. (AJC02-3000-2000)

Place Code		Description	Min. Area	Max. Area
MBFF	**F08**	**First floor, the middle building**	**110**	**150**
AD001	D00031	Residences supervisory room	19	23
AD002	D00033	Shift rotation room	22	25
GS023	D00090	Cafe	16	18
GS046	D00141	Stairways on first floor	24	24
GS047	D00142	Elevators on first floor	15	15
MBGF	**F07**	**Ground floor, middle building**	**550**	**600**
DC009	D00118	Medical Imaging Test Center	150	175
GS017	D00052	Washroom at Mid. building, ground floor	9	11
GS033	D00128	Service Lobby in ground floor middle building	65	70
GS034	D00129	Washroom on ground floor, middle building	10	10
GS035	D00130	Patio	60	70
GS036	D00131	Stairways	23	26
GS037	D00132	Elevators	15	15
SC011	D00069	Warehouse at specialty clinic	10	12
SC012	D00119	Specialized Clinic	195	220
NBBS	**F01**	**Basement, the north building**	**550**	**680**
GS002	D00002	Washing Center	40	45
GS003	D00003	Heating	30	35
GS004	D00004	Watertank	45	50
GS005	D00005	Maintenance Workshop 1	20	25
GS006	D00006	Maintenance Workshop 2	20	25
GS007	D00007	Central Sterilization Department	40	45
GS008	D00008	Electrical Department	33	38
GS009	D00009	Warehouse Office	12	15
GS010	D00010	Washroom	10	12
GS011	D00011	Service Lobby 1	22	24
GS013	D00013	Service Lobby 2	22	24
WH001	D00014	Warehouse 1	20	25
WH002	D00015	Warehouse 2	20	25
WH003	D00016	Warehouse 3	20	25
WH004	D00017	Warehouse 4	20	25
WH005	D00018	Warehouse 5	20	25
WH006	D00019	Warehouse 6	20	25
WH007	D00020	Warehouse 7	20	25

Continued on next page

Table C- 24, assignment table presenting the best solution with after applying AJCs

(Continued and end)

Place Code		Description	Min. Area	Max. Area
NBFF	**F03**	**First floor, north building**	**780**	**900**
GS022	D00085	Washroom in Nephrology clinic	12	14
NH001	D00121	Nephrology & Hemodialysis Clinic	215	245
PB012	D00122	Medical Test Lab (Blood & Pathology)	270	300
NBGF	**F02**	**Ground floor, noth building**	**535**	**680**
AD006	D00053	Administration department	140	160
AD007	D00054	On-call room	20	24
GS014	D00032	Washroom 2	19	23
GS027	D00114	Northern Entrance	30	50
GS031	D00127	Service Lobby in Outpatient Clinic	10	12
OC004	D00042	Police station	10	12
OC007	D00116	Outpatient clinic	221	260
SBBS	**F04**	**Basment, south building**	**650**	**670**
GS001	D00001	Car Parking	590	610
GS012	D00012	Elevator	0	0
GS026	D00113	Car Entrance	50	70
SBFF	**F06**	**First floor, south building**	**760**	**900**
GS021	D00084	Washroom in physiotherapy clinic	12	14
PH009	D00120	Physiotherapy Clinic	385	440
SG011	D00117	Surgery Dvision	230	265
SBGF	**F05**	**Ground floor, south building**	**980**	**1024**
AD003	D00035	General Manager's office	20	25
AD008	D00055	Information office	18	20
AD009	D00056	Archive	16	18
GC001	D00123	General Clinic	360	395
GS018	D00057	Washroom at southern building, ground floor	17	19
GS019	D00070	Pharmacy	90	110
GS020	D00071	Restaurant	140	160
GS028	D00115	Main Entrance	30	50
GS039	D00134	Service Lobby on the ground floor, south building	230	270
GS041	D00136	East Entrance	10	12
GS044	D00139	Patient waiting room 2 on first floor, north building	22	25
WH008	D00034	Warehouse 8	10	12

Table C- 25, assignment table presenting the best solution after applying AJCs explained in Applying FUZZY Adjacency Constraints (6.3.13) AJC-3000-2000.

Place Code		Description	Min. Area	Max. Area
MBFF	**F08**	**First floor, the middle building**	**110**	**150**
AD001	D00031	Residences supervisory room	19	23
AD002	D00033	Shift rotation room	22	25
GS023	D00090	Cafe	16	18
GS046	D00141	Stairways on first floor	24	24
GS047	D00142	Elevators on first floor	15	15
MBGF	**F07**	**Ground floor, middle building**	**550**	**600**
DC009	D00118	Medical Imaging Test Center	150	175
GS017	D00052	Washroom at Mid. building, ground floor	9	11
GS033	D00128	Service Lobby in ground floor middle building	65	70
GS034	D00129	Washroom on ground floor, middle building	10	10
GS035	D00130	Patio	60	70
GS036	D00131	Stairways	23	26
GS037	D00132	Elevators	15	15
SC011	D00069	Warehouse at specialty clinic	10	12
SC012	D00119	Specialized Clinic	195	220
NBBS	**F01**	**Basement, the north building**	**550**	**680**
GS002	D00002	Washing Center	40	45
GS003	D00003	Heating	30	35
GS004	D00004	Watertank	45	50
GS005	D00005	Maintenance Workshop 1	20	25
GS006	D00006	Maintenance Workshop 2	20	25
GS007	D00007	Central Sterilization Department	40	45
GS008	D00008	Electrical Department	33	38
GS009	D00009	Warehouse Office	12	15
GS010	D00010	Washroom	10	12
GS011	D00011	Service Lobby 1	22	24
GS013	D00013	Service Lobby 2	22	24
WH001	D00014	Warehouse 1	20	25
WH002	D00015	Warehouse 2	20	25
WH003	D00016	Warehouse 3	20	25
WH004	D00017	Warehouse 4	20	25
WH005	D00018	Warehouse 5	20	25
WH006	D00019	Warehouse 6	20	25
WH007	D00020	Warehouse 7	20	25

Continued on next page

Table C- 25, , assignment table presenting the best solution after applying AJCs

(Continued and end)

Place Code		Description	Min. Area	Max. Area
NBFF	**F03**	**First floor, north building**	**780**	**900**
GS022	D00085	Washroom in Nephrology clinic	12	14
NH001	D00121	Nephrology & Hemodialysis Clinic	215	245
PB012	D00122	Medical Test Lab (Blood & Pathology)	270	300
NBGF	**F02**	**Ground floor, noth building**	**535**	**680**
AD006	D00053	Administration department	140	160
AD007	D00054	On-call room	20	24
GS014	D00032	Washroom 2	19	23
GS027	D00114	Northern Entrance	30	50
GS031	D00127	Service Lobby in Outpatient Clinic	10	12
OC004	D00042	Police station	10	12
OC007	D00116	Outpatient clinic	221	260
SBBS	**F04**	**Basment, south building**	**650**	**670**
GS001	D00001	Car Parking	590	610
GS012	D00012	Elevator	0	0
GS026	D00113	Car Entrance	50	70
SBFF	**F06**	**First floor, south building**	**760**	**900**
GS021	D00084	Washroom in physiotherapy clinic	12	14
PH009	D00120	Physiotherapy Clinic	385	440
SG011	D00117	Surgery Dvision	230	265
SBGF	**F05**	**Ground floor, south building**	**980**	**1024**
AD003	D00035	General Manager's office	20	25
AD008	D00055	Information office	18	20
AD009	D00056	Archive	16	18
GC001	D00123	General Clinic	360	395
GS018	D00057	Washroom at southern building, ground floor	17	19
GS019	D00070	Pharmacy	90	110
GS020	D00071	Restaurant	140	160
GS028	D00115	Main Entrance	30	50
GS039	D00134	Service Lobby on the ground floor, south building	230	270
GS041	D00136	East Entrance	10	12
GS044	D00139	Patient waiting room 2 on first floor, north building	22	25
WH008	D00034	Warehouse 8	10	12

Table C- 26, assignment table presenting the best solution found by OVAS01-3000-2000

Place Code		Description	Min. Area	Max. Area
MBFF	**F08**	**First floor, the middle building**	**110**	**150**
AD001	D00031	Residences supervisory room	19	23
AD007	D00054	On-call room	20	24
AD009	D00056	Archive	16	18
GS023	D00090	Cafe	16	18
GS046	D00141	Stairways on first floor	24	24
GS047	D00142	Elevators on first floor	15	15
MBGF	**F07**	**Ground floor, middle building**	**550**	**600**
DC009	D00118	Medical Imaging Test Center	150	175
GS017	D00052	Washroom at Mid. building, ground floor	9	11
GS033	D00128	Service Lobby in ground floor middle building	65	70
GS034	D00129	Washroom on ground floor, middle building	10	10
GS035	D00130	Patio	60	70
GS036	D00131	Stairways	23	26
GS037	D00132	Elevators	15	15
SC011	D00069	Warehouse at specialty clinic	10	12
SC012	D00119	Specialized Clinic	195	220
NBBS	**F01**	**Basement, the north building**	**550**	**680**
GS002	D00002	Washing Center	40	45
GS003	D00003	Heating	30	35
GS004	D00004	Watertank	45	50
GS005	D00005	Maintenance Workshop 1	20	25
GS006	D00006	Maintenance Workshop 2	20	25
GS007	D00007	Central Sterilization Department	40	45
GS008	D00008	Electrical Department	33	38
GS009	D00009	Warehouse Office	12	15
GS010	D00010	Washroom	10	12
GS011	D00011	Service Lobby 1	22	24
GS013	D00013	Service Lobby 2	22	24
WH001	D00014	Warehouse 1	20	25
WH002	D00015	Warehouse 2	20	25
WH003	D00016	Warehouse 3	20	25
WH004	D00017	Warehouse 4	20	25
WH005	D00018	Warehouse 5	20	25
WH006	D00019	Warehouse 6	20	25
WH007	D00020	Warehouse 7	20	25

Table C- 26, assignment table presenting the best solution found by OVAS01-3000-2000

(Continued and end)

Place Code		Description	Min. Area	Max. Area
NBFF	**F03**	**First floor, north building**	**780**	**900**
GS021	D00084	Washroom in physiotherapy clinic	12	14
PB012	D00122	Medical Test Lab (Blood & Pathology)	270	300
PH009	D00120	Physiotherapy Clinic	385	440
NBGF	**F02**	**Ground floor, noth building**	**535**	**680**
GS014	D00032	Washroom 2	19	23
GS022	D00085	Washroom in Nephrology clinic	12	14
GS027	D00114	Northern Entrance	30	50
GS031	D00127	Service Lobby in Outpatient Clinic	10	12
NH001	D00121	Nephrology & Hemodialysis Clinic	215	245
OC004	D00042	Police station	10	12
OC007	D00116	Outpatient clinic	221	260
SBBS	**F04**	**Basment, south building**	**650**	**670**
GS001	D00001	Car Parking	590	610
GS012	D00012	Elevator	0	0
GS026	D00113	Car Entrance	50	70
SBFF	**F06**	**First floor, south building**	**760**	**900**
AD006	D00053	Administration department	140	160
SG011	D00117	Surgery Dvision	230	265
SBGF	**F05**	**Ground floor, south building**	**980**	**1024**
AD002	D00033	Shift rotation room	22	25
AD003	D00035	General Manager's office	20	25
AD008	D00055	Information office	18	20
GC001	D00123	General Clinic	360	395
GS018	D00057	Washroom at southern building, ground floor	17	19
GS019	D00070	Pharmacy	90	110
GS020	D00071	Restaurant	140	160
GS028	D00115	Main Entrance	30	50
GS039	D00134	Service Lobby on the ground floor, south building	230	270
GS041	D00136	East Entrance	10	12
GS044	D00139	Patient waiting room 2 on first floor, north building	22	25
WH008	D00034	Warehouse 8	10	12

Table C- 27, assignment table presenting the best solution found by OVAS02-3000-2000

Place Code		Description	Min. Area	Max. Area	Floor Assigned Area
MBFF	**F08**	**First floor, the middle building**	**110**	**150**	
AD001	D00031	Residences supervisory room	19	23	116
AD002	D00033	Shift rotation room	22	25	
AD007	D00054	On-call room	20	24	
GS023	D00090	Cafe	16	18	
GS046	D00141	Stairways on first floor	24	24	
GS047	D00142	Elevators on first floor	15	15	
MBGF	**F07**	**Ground floor, middle building**	**550**	**600**	
AD003	D00035	General Manager's office	20	25	423
AD009	D00056	Archive	16	18	
GS017	D00052	Washroom at Mid. building, ground floor	9	11	
GS033	D00128	Service Lobby in ground floor middle building	65	70	
GS034	D00129	Washroom on ground floor, middle building	10	10	
GS035	D00130	Patio	60	70	
GS036	D00131	Stairways	23	26	
GS037	D00132	Elevators	15	15	
SC011	D00069	Warehouse at specialty clinic	10	12	
SC012	D00119	Specialized Clinic	195	220	
NBBS	**F01**	**Basement, the north building**	**550**	**680**	
GS002	D00002	Washing Center	40	45	434
GS003	D00003	Heating	30	35	
GS004	D00004	Watertank	45	50	
GS005	D00005	Maintenance Workshop 1	20	25	
GS006	D00006	Maintenance Workshop 2	20	25	
GS007	D00007	Central Sterilization Department	40	45	
GS008	D00008	Electrical Department	33	38	
GS009	D00009	Warehouse Office	12	15	
GS010	D00010	Washroom	10	12	
GS011	D00011	Service Lobby 1	22	24	
GS013	D00013	Service Lobby 2	22	24	
WH001	D00014	Warehouse 1	20	25	
WH002	D00015	Warehouse 2	20	25	
WH003	D00016	Warehouse 3	20	25	
WH004	D00017	Warehouse 4	20	25	
WH005	D00018	Warehouse 5	20	25	
WH006	D00019	Warehouse 6	20	25	
WH007	D00020	Warehouse 7	20	25	

Continued on next page

Table C- 27, assignment table presenting the best solution found by OVAS02-3000-2000

(Continued and end)

Place Code		Description	Min. Area	Max. Area	Floor Assigned Area
NBFF	**F03**	**First floor, north building**	**780**	**900**	
GC001	D00123	General Clinic	360	395	590
SG011	D00117	Surgery Dvision	230	265	
NBGF	**F02**	**Ground floor, noth building**	**535**	**680**	
GS014	D00032	Washroom 2	19	23	
GS022	D00085	Washroom in Nephrology clinic	12	14	
GS027	D00114	Northern Entrance	30	50	
GS031	D00127	Service Lobby in Outpatient Clinic	10	12	517
NH001	D00121	Nephrology & Hemodialysis Clinic	215	245	
OC004	D00042	Police station	10	12	
OC007	D00116	Outpatient clinic	221	260	
SBBS	**F04**	**Basment, south building**	**650**	**670**	
GS001	D00001	Car Parking	590	610	
GS012	D00012	Elevator	0	0	640
GS026	D00113	Car Entrance	50	70	
SBFF	**F06**	**First floor, south building**	**760**	**900**	
AD006	D00053	Administration department	140	160	
GS021	D00084	Washroom in physiotherapy clinic	12	14	537
PH009	D00120	Physiotherapy Clinic	385	440	
SBGF	**F05**	**Ground floor, south building**	**980**	**1024**	
AD008	D00055	Information office	18	20	
DC009	D00118	Medical Imaging Test Center	150	175	
GS018	D00057	Washroom at southern building, ground floor	17	19	
GS019	D00070	Pharmacy	90	110	
GS020	D00071	Restaurant	140	160	
GS028	D00115	Main Entrance	30	50	987
GS039	D00134	Service Lobby on the ground floor, south building	230	270	
GS041	D00136	East Entrance	10	12	
GS044	D00139	Patient waiting room 2 on first floor, north building	22	25	
PB012	D00122	Medical Test Lab (Blood & Pathology)	270	300	
WH008	D00034	Warehouse 8	10	12	

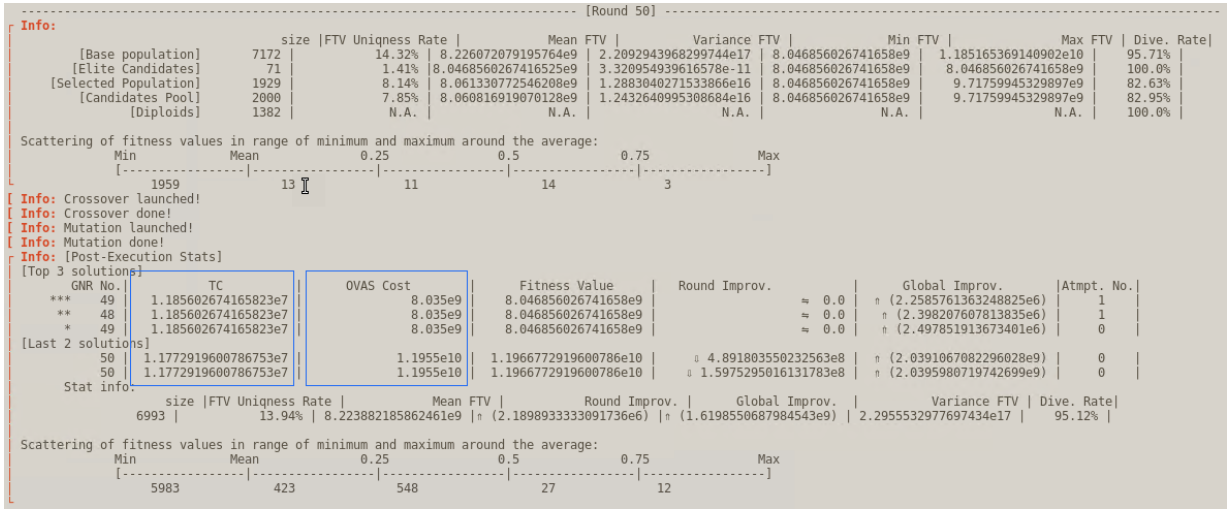


Figure C- 20, summary reports of one of the optimization round in case OVAS02-3000-2000