AN INTEGRATED, EVOLUTIONARY APPROACH TO FACILITY LAYOUT AND DETAILED DESIGN

by

Charles Raoul Shebanie II

BS, Case Western Reserve University, 2002

Submitted to the Graduate Faculty of

School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2004

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This thesis was presented

by

Charles Raoul Shebanie II

It was defended on

July 12, 2004

and approved by

Harvey Wolfe, Professor, Industrial Engineering Department

Jayant Rajgopal, Associate Professor, Industrial Engineering Department

Thesis Advisor: Bryan A. Norman, Associate Professor, Industrial Engineering Department

AN INTEGRATED, EVOLUTIONARY APPROACH TO FACILITY LAYOUT AND DETAILED DESIGN

Charles R. Shebanie II, M.S.

University of Pittsburgh, 2004

The unequal-area, shape constrained facility layout problem is a *NP*-hard combinatorial optimization problem concerned with minimizing material handling costs. An integrated methodology that incorporates a genetic algorithm and a constructive heuristic is developed to simultaneously solve the traditional block layout problem of locating and shaping departments and the detailed design problem of locating the input/output stations of departments. These problems have received much attention over the past half-century with the majority of research focused on solving them individually or sequentially. This thesis aims to show that an integrated methodology which combines the problems and solves them in parallel is preferable to sequential approaches.

The complexity of the integrated layout problem is reduced through a Flexbay formulation and through pre-assigned intra-departmental flow types. A genetic algorithm with a two-tiered solution structure generates and maintains a population of block layout solutions throughout an evolutionary process. Genetic operators reproduce and alter solutions in order to generate better solutions, find new search directions, and prevent premature convergence of the algorithm. An adaptive penalty mechanism guides the search process and reduces the computational overhead of the algorithm. Through the placement of input/output stations, the optimization of a block layout's material flow network is implemented as a subroutine to the genetic algorithm. A contour distance metric is used to evaluate the costs associated with material movement between the input/output stations of departments and aids in constructing practical aisle structures. A constructive placement heuristic places the input/output stations and perturbs them until no further improvement to a layout can be realized.

The integrated approach is applied to several well known problems over a comprehensive test plan. The results from the integrated approach indicate moderate variability in the solutions and considerable computational expense. To compare the integrated methodology to prior methodologies, some of the best results from the unequal-area facility layout problem are selected from prior research and the I/O optimization heuristic is applied to them. The results of the integrated approach uniformly and significantly outperform the results obtained through sequential optimization. The integrated methodology demonstrates the value of a simultaneous approach to the unequal-area facility layout problem.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOTATION

n = the number of departments in a layout

f_{ij} = the total flow between departments i and j

d_{ij} = the contour distance between departments i and j

α = the maximum allowable aspect ratio for all departments in a layout

l_i = the length of department i

w_i = the width of department i

m = the number of infeasible departments in a layout

Z_{feas} = the best feasible objective function value

Z_{all} = the best overall objective function value

$Z(\pi)$ = objective function value for layout π

k = the severity parameter for the penalty function

$p(m)$ = the penalty value for a layout with m infeasible departments

Loc_i = the set of candidate I/O points for department i

$d_{i,l,j,m}$ = the shortest contour distance from I/O point l of department i to I/O point m of department j

A = the total area of a layout

H = the height of a layout

W = the width of a layout

w_b = the width of bay *b*

A_b = the total area of the departments in bay *b*

a_i = the area of department *i*

$U(\ )$ = uniform random number

P_c = the probability of selecting alleles from parent *1*

P_m = the probability of mutation

P_r = the probability of a mutant replacing its original parent

P_b = the probability of operating on the bay structure of a mutant

P_{me} = the probability of merging two bays during mutation

P_s = the probability of splitting two bays during mutation

ACKNOWLEDGEMENTS

I would like to thank Dr. Bryan A. Norman for affording me the opportunity to work on this research effort. His continual support, guidance, and inspiration have led me through this research experience. As a result, I have gained a much larger breadth of knowledge in the field of operations research and I have a much greater sense of purpose with regard to applying this knowledge.

I would like to thank Dr. Norman, Dr. Alice E. Smith, Dr. Rifat A. Arapoglu, Dr. David M. Tate, and Dr. Wippawee Tharmmaphornphilas for their research efforts on the facility layout problem. Their hard work and achievements made this research possible. The insights gained from this research are largely attributed to them. I would also like to thank Dr. Jayant Rajgopal and Dr. Harvey Wolfe for their support as members of my thesis committee.

I would also like to express my deepest gratitude to my mother, Laura, for her continual support and encouragement throughout my education and to my father, Charles, for instilling in me the work ethic to succeed. Finally, I would like to thank my sisters, Lauren and Alexis, for brightening my spirits throughout the rigors of my academic experiences.

1.0 INTRODUCTION

The problem of designing a facility layout that achieves feasible functionality, while at the same time realizing minimal operational costs, is well studied. This work combines aspects of traditional block layout optimization and detailed design optimization to provide insight into the effectiveness of an integrated, evolutionary optimization strategy for the unequal-area facility layout problem (FLP). The unequal-area FLP of minimizing material handling costs is known to be *NP*-hard. A genetic algorithm (GA) framework is utilized to alleviate the complexity and combinatorial nature of the problem. A constructive heuristic is integrated within a block layout GA to address the detailed design problem of locating input/output (I/O) points for all departments. In addition, a perturbation scheme is implemented to improve layout solutions throughout the evolutionary process. In contrast to the more prevalent methodology of optimizing a layout sequentially, the integrated, evolutionary approach optimizes the block layout and the detailed design in a unified manner without the need of a promising initial solution. The mathematical and computational difficulties that arise as a result of an integrated and realistic approach to the unequal-area FLP, inclusive of detailed design aspects, are discussed.

The motivation for this thesis originates from the unequal-area FLP formulated by Armour and Buffa [4]. An unequal-area layout problem consists of optimally locating and shaping a specified number of departments with predefined areas and other constraints. In order to approach the problem from a computationally feasible perspective, the problem must be restricted to reduce the infinite number of possible layouts. Existing techniques, such as Flexbay [36] formulations and adaptive penalty methods [11], are used to this effect because of their demonstrated effectiveness, both in alleviating CPU requirements and in providing solution quality. The I/O optimization, indicative of how material travels from one department to another, is implemented as a subroutine to the block layout GA. To limit the search space for optimally locating I/O points, restrictions are made in regards to the number of candidate I/O points for each department and to the type of flow existing within each department.

The goal of optimizing both the unequal-area block layout and the I/O location simultaneously is a relatively novel concept. The problem is *NP*-complete and requires heuristic procedures to approach test problems of even moderate size, although exact methods of computation are utilized as well. This thesis aims to bridge the gap of performing both optimization tasks in a cohesive fashion. With explicit consideration given to computational feasibility and solution quality, some of the most well known and promising FLP methodologies are selected for implementation. Prior attempts at solving this combinatorial optimization problem have been summarily focused on sequential analysis. Once an optimal, or near optimal, layout has been found, the detailed optimization of locating I/O points is then performed. Sequential approaches however, limit the number of candidate layouts that are considered. Thus, optimal layouts are extremely difficult to obtain when performing block layout and detailed layout separately. Due to the expense of performing both simultaneously, sequential procedures are currently the status quo for the unequal-area FLP.

In this thesis, the optimization of departmental shape, departmental location, I/O location, and implied flow paths are all taken into account. The constraints of this particular problem include predetermined flow types within departments [29], areas and minimum side requirements for departments, and maximum allowable aspect ratios for departments. A contour distance metric [27] is used in evaluating the total flow costs of a layout. Material flow both between departments (inter-departmental) and within departments (intra-departmental) is an important distinction of this thesis. All of the aforementioned aspects of facility layout design are applicable to a variety of industries and enhance the ability of an analyst to design and interpret a layout.

Most facility layout optimization strategies can be classified as either construction algorithms or improvement algorithms. Construction algorithms iteratively build layouts from scratch by building and placing departments, one at a time, until complete. Improvement algorithms assume an initial layout is given and enhance the layout by interchanging positions of departments based on the magnitude of improvement. Both constructive and improvement techniques are utilized in the proposed integrated, evolutionary methodology.

Embedding the detailed design optimization with the block layout optimization will be shown to significantly improve upon prior results. The use of evolutionary strategies versus traditional optimization techniques, such as integer programming, will be shown to produce results with little or no variance from their exact counterparts. The computational effort required to achieve near optimal layouts will be highlighted. Relevance and comparisons to prior research in the arena of the unequal-area FLP is accomplished by using problem-specific data from three of the most well known FLPs in the literature [4], [6], [37]. These problems range from small numbers of departments to moderately-sized layouts of up to twenty departments. All solution strategies are verified through comparisons to the results of recent research [2], [3], [27], [29], [35].

The problem of designing a facility layout is pertinent to a wide array of industries, where operations can be thought of in terms of a flow path, and is not necessarily restricted to manufacturing operations. The techniques that produce operational benefits are well known and have been rigorously studied for over half a century. Related applications include VLSI design, component placement for printed circuitry, and hospital design, to name a few. It is the aim of this work to bring some of the most significant advances of the past and the present together to meet the demands of designing a facility layout with minimal costs and minimal design time. It is important to note that this thesis would not have been readily conceivable, or executable, without prior advancements [2], [3], [11], [27], [29], [35], [36] that were largely done in the hopes of achieving a superior concurrent optimization strategy.

The primary goal of this thesis is to validate the idea that block layout and I/O placement should be optimized simultaneously. Performing block layout optimization without consideration for the material flow network defined by I/O stations limits a layout designer's ability to truly achieve the best possible layout configuration for a facility. If the placement of I/O stations is performed after a block layout has been determined, the layout designer is restricted to only one block layout. Considering the explicit aisle structure formed by I/O stations during the optimization of a block layout will be shown to produce superior layouts.

In addition, this thesis aims to demonstrate the value of a genetic algorithm approach to facility layout planning. Genetic algorithms have proven to be very effective for solving the unequal-area FLP. The integration of the block layout optimization problem and the I/O location optimization problem will further establish a genetic algorithm approach as a powerful tool for facility layout. The implementation of a contour distance metric will be shown to produce layouts that are physically realizable. That is, the explicit aisle structure created by using the contour distance metric will alleviate problems that could arise during the physical setup of a facility. The consideration of intra-departmental flow orientation within departments will also be shown to contribute to more practical layouts and to superior layouts.

There are several other contributions made to solving the unequal-area FLP in this thesis. Successful techniques, as indicated by prior research, were selected and integrated in a way that would likely produce superior results with reasonable CPU requirements. A rigorous test plan was developed to determine the usefulness of the integrated, evolutionary methodology. Significant analysis was performed on the quality of solutions attained, the computational effort required, the comparison of results with those of prior methodologies, and the tendencies and characteristics of the integrated, evolutionary methodology. Finally, recommendations are made as to future advancements for optimizing the unequal-area FLP considered in this thesis.

A discussion of the literature pertaining to this thesis follows in Section 2. The development of the FLP from past to present as well as the prevailing methodologies used in solving the FLP will be highlighted. Alternative FLP techniques and indications as to the next generation of FLP approaches will also be discussed. Section 3 presents the specific form of the unequal-area FLP that is considered in this thesis. Section 4 details the methodology of the integrated, evolutionary approach for the unequal-area FLP. The selection of test problems and the design of the computational test plans are explained in Section 5. The results of this thesis and comparisons to the best known results of prior research are presented in Section 6. The effectiveness of the integrated, evolutionary approach and considerations for future developments are discussed in Section 7.

2.0 LITERATURE SURVEY

This chapter discusses literature pertaining to this thesis and to the unequal-area FLP in general.

2.1 HISTORY OF FACILITY LAYOUT AND PROBLEM FORMULATIONS

The FLP and location theory, in general, have been well studied since the 1960s. Many significant research publications involving normative approaches to solving location problems from an operations research perspective are outlined in a bibliography by Francis and Goldstein (1974) [13]. This list includes much of the innovative and pioneering work that was done in the 1960s and 1970s for the FLP. Two more recent literature surveys on the FLP are done by Kusiak and Heragu (1987) [18] and Meller and Gau (1996) [21]. Meller and Gau discuss recent trends in facility layout research. They highlight a tendency towards concurrent design strategies, which incorporate design of a facility layout with design of a production system. In addition, a comparison is made between the state-of-the-art in facility layout software and the state-of-the-art in facility layout research [21].

The unequal-area FLP was originally formulated by Armour and Buffa (1963) [4]. The authors assume there to be a given fixed rectangular region, or facility, of dimensions H and W, where H is the height and W is the width. The number of departments, the area of each department, and the flow values associated with each pair of departments are assumed to be known. The basic formulation is given by,

n = the number of departments
f_{ij} = the total flow between departments i and j, where $i,j = 1,2,\ldots,n$
d_{ij} = the distance between departments i and j, where $i,j = 1,2,\ldots,n$

The objective function for minimizing the total product flow throughout a facility is given by,

$$Total\ Cost = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} f_{ij} d_{ij} \qquad (2\text{-}1)$$

The goal was to partition the region into departmental sub-regions so as to minimize the total material movement and associated costs of the entire facility. The combinatorial nature of this formulation made the consideration of all possible layouts virtually impossible. The authors provide a computer-aided heuristic algorithm that alleviated the need to consider all possible permutations of a layout and made the cumbersome methodology of using qualitative judgment to select several promising location patterns much less desirable. Their algorithm considered piecewise departmental exchanges in which adjacent departments of equal area could be relocated, or interchanged, if an improvement to the objective function was realized. Their methodology, formally known as CRAFT, can be classified as an improvement algorithm.

A variation of the quadratic assignment problem (QAP), that was modified to accommodate the FLP, was formulated and solved by Bazaraa (1975) [6]. The same rectangular layout from Armour and Buffa [4] is broken down into blocks of equal-area and uniform shape. An analyst must then provide a combination of blocks that represent the basic underlying structure of each department. Each department is then constructed from its basic combination of blocks, one department at a time. This technique can be classified as a construction algorithm. This greatly reduced the number of candidate layouts in the search space while still allowing departments to assume different areas and different shapes. It also greatly increased the effort required by an analyst because there was a need to provide a substantial amount of information, especially when considering a large number of departments. By using a branch and bound optimization scheme with the QAP formulation, optimal layouts for small numbers of departments could be obtained.

The simplistic representation of departments as groups of uniformly sized blocks was significantly upgraded by van Camp et al. (1991) [37]. By considering departments of fixed area and variable dimensioned rectangular shape, the authors formulated a nonlinear optimization layout technique (NLT) that is characterized by three sets of constraints. Two sets are classified as “hard-constraints,” or those that enforce the desired structure of the layout. One set ensures that departments may not overlap, while the other set prohibits departments from locating outside the facility boundaries. The third set is much less rigid and depends more on the specific problem as defined by an analyst. That is, the minimum and maximum allowable lengths for the shortest side of each department, as well as for the facility itself, are defined and incorporated into the constraint set. Through the use of a penalty method in the objective function, the NLT methodology will produce at worst a local minimum solution.

With the obstacle of representing departments as combinations of uniform blocks overcome, the concept of optimizing departmental shapes as well as departmental location was enhanced with the flexible bay (Flexbay) structure provided by Tong [36]. An example of a Flexbay layout is shown in Figure 2.1. The darkened vertical lines represent the partitions between bays. The Flexbay structure forces departments to share identical widths within each bay, dramatically reducing the search space of the problem. A constructive heuristic is used to simultaneously determine the number of bays and the sequence of departments within each bay. Although many layout configurations are eliminated from consideration, the Flexbay formulation proves to be an effective means of layout planning.

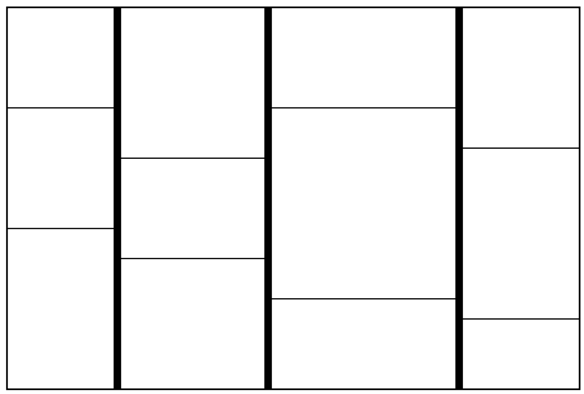

Figure 2.1 A Flexbay Layout

2.2 INTEGER PROGRAMMING

The quadratic assignment problem (QAP) of assigning discrete entities to discrete locations is one of the most well known and difficult combinatorial optimization problems in operations research. The unequal-area FLP represents a more complex optimization problem than the QAP. It is possible to adapt the QAP to the unequal-area FLP by breaking departments into small grids with equal areas and by not allowing the separation of grids of the same department by assigning large artificial flows between them. An unequal-area FLP formulated as a QAP substantially increases the number of decision variables needed, so that even solving such a problem presents a significant challenge. To date, attempts to model the unequal-area FLP using exact optimization methods, such as mixed integer programming (MIP), have been limited due to the complexity of the problem.

Montreuil (1990) [24] provides a MIP that has become somewhat of a baseline for extension, improvement, and comparison. In order to bound the nonlinear area constraint of the unequal-area FLP, a surrogate perimeter constraint is used. This constraint, however, does not guarantee that department areas, known a priori, will be realized upon final solution. Meller et al. (1999) [22] point out that Montreuil's perimeter constraint is modeled such that increases to the aspect ratio of a FLP directly correspond to increases in errors between the actual and solved areas. The difference between original and surrogate areas can be as much as 11%, 25%, 36%, and 44% for aspect ratios of 2, 3, 4, and 5, respectively [22]. The authors propose an improved surrogate perimeter constraint that provides a more realistic and effective implementation by forcing a department's area to rigorously adhere to changes in its perimeter. The authors report optimal solutions for FLPs with up to eight departments. Sherali et al. [34] provide a similar approach that significantly reduces errors in department areas by using a polyhedral outer approximation of the area constraints and branching priorities. Using the polyhedral approximation and other innovative techniques, the authors report solutions for FLPs with up to nine departments.

A new MIP formulation for the unequal-area FLP using the Flexbay structure is presented by Konak et al. [17]. Contrary to the work of Meller et al. [22] and Sherali et al. [34], the nonlinear department area constraints are modeled on a continuous plane without using any surrogate constraints and without utilizing linear relaxation techniques. This improvement permits the use of the Flexbay formulation in the form of a MIP. This is the first MIP in which area constraints are enforced 100% for the unequal-area FLP. The authors argue that although

Flexbay formulations restrict possible layout combinations, it forms the basis of an aisle structure that facilitates the user transferring the block design into an actual facility design. Several extensions to this formulation are suggested, such as tightening the lower bound of the linear program, removing the mirror effect, modeling the number of bays as a decision variable, and modeling monuments or fixed regions within departments. A mirror effect occurs when decision variables assume different values, but produce layouts that are identical in terms of inter-departmental distances and total cost. Restricting a department's centroid to a quarter of a layout eliminates the mirror effect and three fourths of all layout combinations. Modeling the number of bays as a decision variable was attempted, but the formulation was inefficient. Modeling fixed monuments within a department limits the shape and location changes that are possible for that department. A monument can be thought of as a predefined, fixed rectangle that acts to constrain a department's potential location. For example, a large broach could be a monument in a machining department. The machining department can be relocated, but its new location must still contain the broach. The authors present results for a fourteen department problem. By reverting to a centroid-to-centroid distance metric, no consideration is made for I/O location. Also, the determination of the number of bays a priori could present a limitation when considering construction of a new facility. Although the aforementioned MIP papers provide substantial improvements to the number of departments that can be solved to optimality, each work acknowledges computational limitations of using MIP for the FLP.

2.3 CONSTRUCTIVE AND IMPROVEMENT PROCEDURES

As mentioned earlier, the majority of facility layout solution strategies can generically be identified as either constructive heuristics or improvement heuristics. The same generalization can be made in regards to the QAP. Due to the intractability of the QAP for any but small problems (n = 5 to 20 departments maximum), heuristic techniques have generally been preferable to the exact methods discussed in the previous section. The tradeoff between gains to CPU efficiency and degradations to solution quality through the use of heuristic procedures for the QAP has been shown to be minimal (1981) [19].

The advantages and disadvantages of constructive and improvement procedures being utilized separately, sequentially, or cohesively, have been thoroughly discussed [19] and continue to be an active area of research [3]. Liggett [19] provides a computational analysis of utilizing constructive and improvement techniques on several well known test problems. The differences between improvements to a random solution through pair-wise exchange versus constructed solutions are highlighted. Liggett concludes that constructive procedures are preferable since better solutions can be obtained at less cost. It should be noted that computers today are much faster and make it easier to use improvement algorithms. The author suggests that attempts to reduce the number of exchanges may alter the preference towards constructive procedures. In particular, selecting the exchange that leads to the maximum cost improvement rather than selecting the first exchange that leads to improvement was not shown to be worth the addition in computational time. The results of Liggett [19] and Arapoglu et al. [3] indicate that improvements to solution quality and computational feasibility through the use of heuristic procedures do not have to come at the expense of one or the other. In addition, the question of preference between constructive and improvement schemes is answered to some degree. More importantly, the potential for improvement through the use of constructive and improvement strategies in some manner of cohesion becomes a distinct possibility.

The use of a cut tree methodology in the context of the construction and improvement of a facility layout was introduced by Montreuil and Ratliff (1989) [25]. The authors propose the use of a design skeleton because it has several desirable properties. If a designer wishes to place emphasis on the separation of certain departments, versus focusing strictly on departmental adjacency, a cut tree permits the partitioning of departments into two subsets and indicates the optimum partition. The links on a cut tree indicate average material movement. If the skeleton is used as the aisle structure, the designer is provided with valuable insight about how to increase or decrease aisle lengths. Furthermore, if an aisle structure is made up of segments of uniform length, then a cut tree will provide the aisle structure that minimizes total flow. Figure 2.2 displays a slicing tree representation and its corresponding unequal-area layout. Departments are placed at the leaves of the tree and internal nodes describe the operators indicating the direction of cuts. Four cut operators are shown: "u" represents an up cut, "b" a bottom cut, "l" a left cut, and "r" a right cut. The binary representation of the departments and the cut operators defines department sequence and adjacency, as well as the partitioning of the layout. Departments *2* and *3* in Figure 2.2 are separated by a bottom cut and departments *4* and *5* are divided by a right cut. Departments *2* and *3* are separated from departments *4* and *5* by a right cut, and departments *2*, *3*, *4*, and *5* are all separated from department *1* by an up cut. The authors point to an advantage of the cut tree methodology when used for multiple floor layout planning. They also mention that a layout designer's goal is ultimately to generate a satisfactory design rather than a theoretical optimum. In such a context, a cut tree provides an efficient and effective means of enhancing layout design. However, giving a layout designer the decision making power to manually select cuts in a design skeleton could potentially lead to a satisfactory layout design that is considerably deviant from its optimal solution.

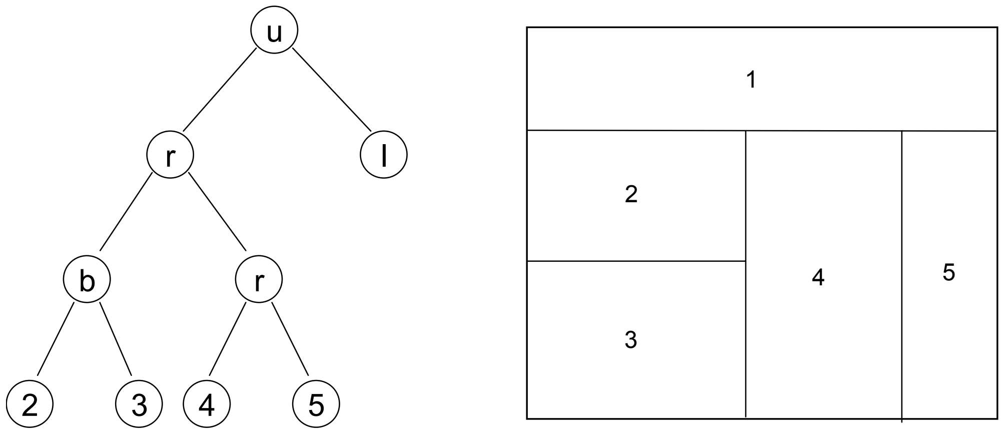

Figure 2.2 A Slicing Tree Layout Representation

The issue of unreliable and frequently changing flow values between departments in a manufacturing environment, caused by product demand variability, is addressed by Kim and Klein [16]. The authors discuss two heuristic algorithms used for locating pickup and delivery points, or I/O points, once a department configuration for a layout has been fixed. The motivation of the research is the potential for significant reduction in material handling costs by adjusting the I/O points of fixed departments. Changes to department configuration are intuitively much more expensive than adjustments to I/O configuration. The first heuristic, algorithm *A*, assumes the location of one pickup point and one delivery point for each department are known. The algorithm iteratively compares the relative location of pickup and delivery points between departments and changes locations based on the largest decrease in total flow cost for a single department. The second heuristic, algorithm *B*, takes advantage of a specific layout configuration in determining I/O locations. Subsets of departments are identified based on common flow paths. That is, departments whose outward material flows will pass through common nodes are grouped into subsets for evaluation. The complexity of the problem is thus reduced and computational expense is minimized. The authors suggest using algorithm *B* to produce an initial layout for algorithm *A*. The computational expense of algorithm *A* is then reduced since the initial layout has been previously optimized.

A methodology for designing a layout based on the metric of the shortest path along aisles or corridors (SPAAC) was introduced by Benson and Foote [8]. A constructive heuristic, known as DoorFAST, is used to optimize the aisle structure and door placement once the department locations and a general aisle structure have been determined. By using the SPAAC metric, the solutions produced by DoorFAST are much more interpretable than results that use centroid-to-centroid distance measures or other metrics that do not provide a clear indication of a material flow network.

Chittratanawat and Noble [10] put forth an integrated facility layout methodology that incorporates department location, qualitative relationships, I/O location, and material handling equipment selection. The authors assume equal area departments of identical shape and that the flow of material within a department is adjustable to the selection of an I/O station. A construction algorithm generates an initial layout and a Tabu Search algorithm improves the initial solutions. The inclusion of material handling equipment selection is justified as a means of ensuring the best overall material handling costs for a manufacturing layout. If the selection of material handling equipment were not included, a best layout configuration might not correlate to the lowest material handling costs. The initial construction of a layout is determined to be a critical factor in the solution quality of the Tabu Search. Their approach performs markedly better with constructed initial solutions than with those that are generated randomly.

Also, the methodology does not ensure diversity in the search space because Tabu moves are not recorded during the search process.

An integrated layout methodology specifically designed for semiconductor fabrication facilities is given by Peters and Yang [30]. The authors mention the proclivity of semiconductor manufacturers to adopt bay structured layouts that are not always the most efficient. A bay structure is accommodating to this industry since material movement is generally done through automated material handling systems that are designed as a either a spine configuration through some central location of a layout or as a perimeter configuration around a layout. Spine configurations restrict the number of departments in each bay to one to ensure interaction with a material handling system, whereas perimeter configurations mandate two departments within a bay to achieve the same functionality. These restrictions aid in optimizing both the block layout and the material handling system. Using space filling curves to construct an initial material handling configuration, multiple layouts with different flow sequences can be generated for further consideration. Once an initial layout and flow sequence have been constructed, the location of I/O points and the determination of crossover points are made. Crossover points indicate positions on a material handling system at which flow can reverse direction by crossing over to the parallel path of the system. A flow network is constructed from the candidate I/O points and candidate crossover points. A steepest-descent-pairwise-interchange heuristic improves initial solutions. Space filling curves could be used as a means of flow path configuration for the unequal-area FLP considered in this thesis. Assuming that a simple spine or perimeter configuration is not sufficient and the restrictions on bay structure are alleviated, space filling curves might aid in limiting the number of potential layouts for consideration. This would be especially useful if the material handling system under consideration was very expensive.

2.4 BLOCK LAYOUT AND DETAILED DESIGN

The literature discussed in this section forms the basis for the integrated, evolutionary approach. The work of the key contributors to this thesis is discussed in detail. In regards to block layout, the key contributors have researched and refined the concepts of the Flexbay formulation, unequal-area FLP genetic search, adaptive penalty functions, and aspect ratios. In terms of detailed design topics, the key contributors have produced the ideas for a contour distance metric, for consideration of inter-departmental and intra-departmental flow, and for identifying the most likely successful I/O placement algorithm to be used with a block layout GA. Detailed design topics can be defined as those elements of the unequal-area FLP that help to further specify the problem, advance solution quality, and enhance physical interpretation.

Using the Flexbay [36] formulation, Tate and Smith [35] applied genetic optimization with an adaptive penalty function to the unequal-area FLP. Genetic search was used because QAP formulations of unequal-area problems are much less tractable than the corresponding equal-area formulations. Equal-area formulations that interchange blocks of departments when given an initial layout restrict the possible shapes that departments can assume. Thus, elongated department shapes that might improve a layout are not considered. The lack of consideration for such configurations is potentially damaging to the search direction of a problem and could exclude advantageous layout configurations. The constructive genetic formulation [35] provides much greater potential for considering a substantial breadth of layout configurations.

Implementation of genetic search is often avoided due to the lack of guidance to near-feasible regions of the solution. To enhance guidance to feasible regions, a maximum allowable aspect ratio, similar to the constraints of van Camp et al. [37], is given by,

$$\alpha = \left(\frac{\max\{l_i, w_i\}}{\min\{l_i, w_i\}} \right) \qquad (2\text{-}2)$$

where α = aspect ratio
l_i = length of department i, $i=1,\ldots,n$
w_i = width of department i, $i=1,\ldots,n$

The aspect ratio ensures layout solutions that transform well into physical reality. A department's aspect ratio is defined a priori. Since department areas are also predetermined, the aspect ratio limits the possible perimeter dimensions for each department. Consider the following example. A department with an area of *10* is defined to have a minimum side length of *1* and a predetermined aspect ratio. Assuming that the length of this department is measured in increments of *0.2*, the number of possible shape configurations for this department would be *15*, *22*, and *25* for aspect ratios of *3*, *5*, and *7* respectively. As the aspect ratio becomes smaller, a problem becomes highly constrained since the allowable department perimeter dimensions become more restricted. Feasible solutions are difficult to find for smaller ratios ($\alpha \leq 2$) even with extensive computing power. Large aspect ratios ($\alpha \geq 7$) also require significant CPU power because there are more department shape configurations to consider, and consequently more layout solutions to evaluate To alleviate these problems and help to guide the search to feasible solutions, an adaptive penalty function is given by,

$$p(m) = (m)^k (V_{feas} - V_{all}) \qquad (2\text{-}3)$$

where m = number of infeasible departments
V_{feas} = best feasible objective function value yet found
V_{all} = best overall objective function value yet found
k = parameter that adjusts the severity of the penalty function

The penalty function changes as the problem evolves through the genetic process. In a sense, it learns the degree of constraint for a specific problem through retaining and utilizing prior knowledge in its evaluation. This addition to the GA structure was especially important for guiding the search process to feasible solutions for highly constrained problems, or those problems with very small aspect ratios and/or a large number of departments.

Similar to the SPAAC [8] metric, a contour distance metric was formally introduced in Norman et al. [27]. The contour measure is defined as the shortest rectilinear distance along department contours between the candidate I/O station l of department i and the candidate I/O station m of department j as given by,

$$d_{i,j} = Min\{d_{i,l,j,m} : l \in Loc_i , m \in Loc_j\} \qquad (2\text{-}4)$$

where Loc_i and Loc_j are the set of candidate I/O locations for departments i and j respectively, and $d_{i,l,j,m}$ is the shortest distance, following departmental contours, from the candidate I/O location l of department i to the candidate I/O location m of department j. The objective function for this problem incorporates Equation (2-1) with a modified version of the penalty function from Equation (2-3) as given by,

$$Z(\prod) = \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} f_{i,j} d_{i,j} + m^3 (Z_{feas} - Z_{all}) \qquad (2\text{-}5)$$

where m is the number of departments in layout $\prod$ that violate the aspect ratio constraint, Z_{feas} is the objective function value of the best feasible solution found so far, and Z_{all} is the unpenalized objective function value of the best solution found so far. In using a contour measure, the total flow cost of a layout is much more understandable and realistic because it follows an aisle structure permitted by the departments, versus crossing through the midpoints of department boundaries. The authors assumed that departments could possess multiple I/O points. Thus, flow paths between departments could be drawn to the most convenient point, resulting in shorter paths than if limited to only one input and one output station per department. Multiple I/O locations would actually result in additional costs during physical interpretation of the layout. Furthermore, multiple I/O locations may not be allowable when designing a layout.

To make the layout structure and the total cost consideration more realistic, the placement of I/O locations was restricted to one input station and one output station per department in Arapoglu et al [3]. This restriction produces more practical layouts and encourages the integration of block layout with I/O placement. Four optimization methodologies were explored with the intent of identifying the most suitable I/O location strategy for use as a nested routine within block layout optimization. A constructive heuristic was found to be the best choice in regards to computational feasibility. The pseudo-code for this heuristic is shown in Figure 2.2. A genetic search framework for I/O location was found to achieve superior results compared to the constructive heuristic, but required greater computational expense. The authors recommended using the constructive I/O heuristic for the majority of a block layout optimization. Then, near or at the end of the search, use the I/O GA methodology to improve the final layout.

Notation:
A: Set of assigned departments
$S = \{1,2,3,\ldots,n\}$ (set of all departments)
$tot\, f_{ik}$: Total material flow using I/O point k of department i

Algorithm:
Step 1:
$A = \phi$
Set tot $f_{i,k} = 0$ for all $i \in S$ and $k \in Loc_i$
Step 2:
Let $tot_{r,s} = Max\{totf_{i,k} : i \in S \setminus A, k \in Loc_i\}$ $A \leftarrow A \cup \{r\}$.
Assign I/O point s to department r.
If A=S then stop, all departments have been assigned an I/O point.
Else
tot $f_{i,k} = 0$ for all $i \in S \setminus A$ and $k \in Loc_i$.
Using the Floyd-Warshall algorithm, find the shortest path between all node pairs in the network consisting of all candidate I/O locations for all departments in $S \setminus A$ and the selected I/O locations in all departments in A. For each flow, $f_{i,j}$ that uses I/O point k of department i and I/O point l of department j, update tot $f_{i,k} = totf_{i,k} + f_{i,j}$ and $totf_{j,l} + f_{i,j}$

Figure 2.3 Constructive I/O Heuristic

Further enhancement to the scope of detailed facility design was accomplished in Norman et al. [29] by considering intra-departmental flows that occur within departments. A department of any rectangular area and dimension was defined to have a discrete set of candidate I/O points as shown in Figure 2.3. Each department could assume only one input point and one output point. Sharing of the input and output point was allowed, but not enforced. Three distinct intra-departmental flow types were defined: U-shaped (U), linear (L), and C-shaped (C). Figure 2.4 illustrates the possible orientations for each flow type with regards to the candidate I/O locations.

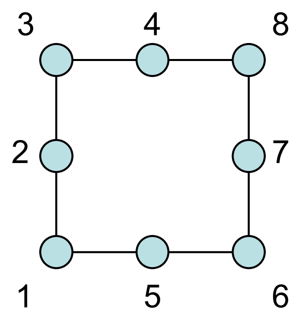

Figure 2.4 Candidate I/O Locations

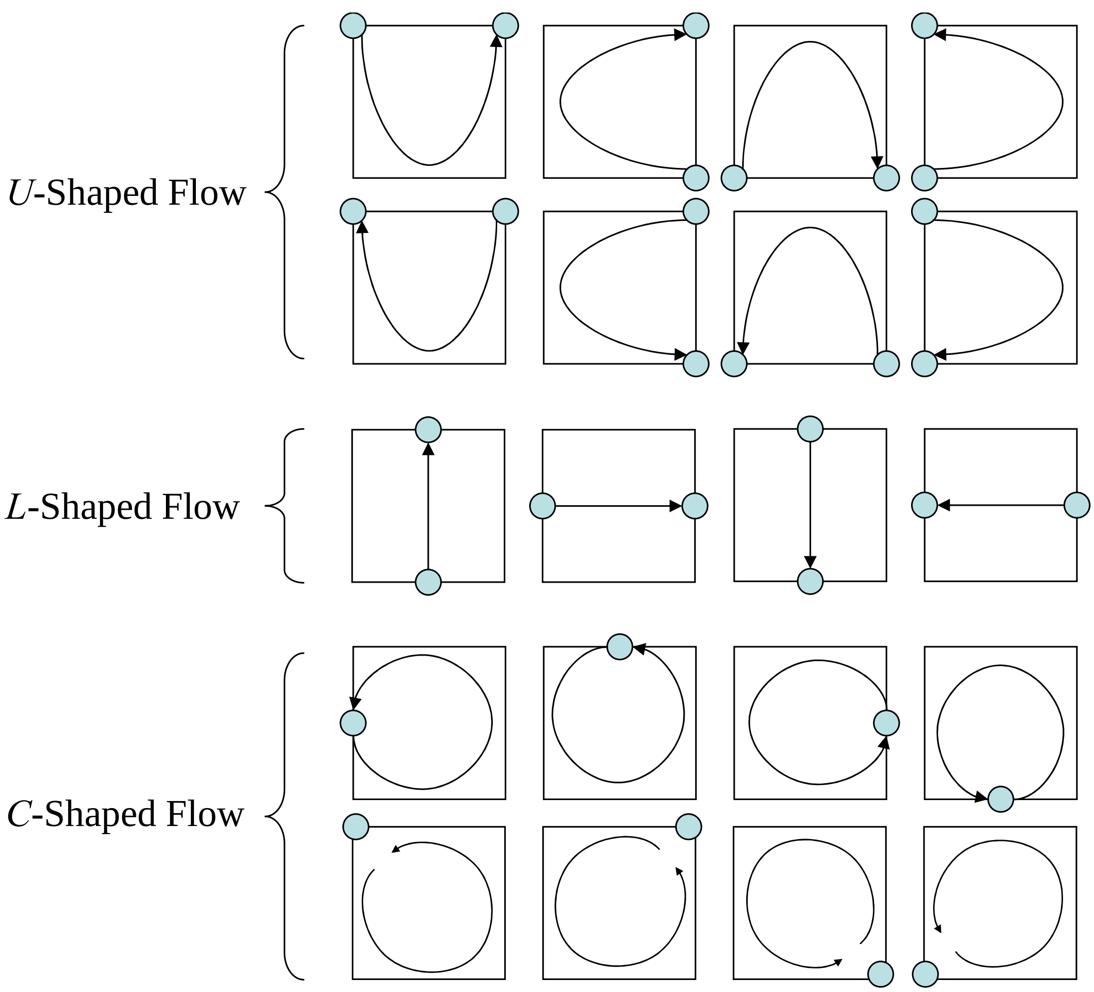

Figure 2.5 Intra-departmental Flow Types

Combining the candidate I/O points in Figure 2.3 with the flow type orientations in Figure 2.4 created a set of feasible I/O pairs for each intra-departmental flow. The sets of feasible pairs for each flow type are shown below.

U-Shaped:	(1,3),	(1,6),	(3,8),	(3,1),	(6,1),	(6,8),	(8,3),	(8,6)
L-Shaped:	(2,7),	(4,5),	(5,4),	(7,2)				
C-Shaped:	(1,1),	(2,2),	…,	(p_i, p_i)		$i=1,\ldots,8$		

Although only eight configurations are shown for *C*-flow in Figure 2.5, there can be more than eight candidate I/O locations for a department with *C*-flow. Points *9* and greater for *C*-flow refer to any additional points where the department under consideration intersects either the midpoints or corner points of the side of another department. Thus, the set of candidate I/O points for any department with *C*-flow constitutes a finite, dominant set [2]. The numbering for points *9* and greater starts on the left side of the department in Figure 2.3, continues to the bottom side, followed by the top side, and then to the right side. Knowing the intra-departmental flow type and the location of each department a priori, a shortest path network consisting of the candidate I/O points, or nodes, for each department could be constructed. An I/O GA was used to optimize the flow path. The resulting ailse structure was improved through the use of an elitist strategy, a greedy local search, and a perturbation scheme. The MIP formulation for the single I/O placement problem with consideration of intra-departmental flow is provided in Figure 2.5.

Parameters:

I:	Set of departments	$i = 1,2,\ldots,n$
J:	Set of departments	$j = 1,2,\ldots,n$
K_i:	Set of I/O points for department i	$k = 1,2,\ldots,p_i$
L_i:	Set of I/O points for department i	$l = 1,2,\ldots,q_i$
A:	Set of I/O points of flow type U	$a = 1,3,6,8$
B:	Set of I/O points of flow type L	$b = 2,4,5,7$
f_{ij}:	Amount of flow from department i to j	

d_{ikjl}: Distance from output point k of department i to input point l of department j

Decision Variables:

x_{ik}	=	1	If department i has point k as an output point
		0	Otherwise
y_{ik}	=	1	If department i has point k as an input point
		0	Otherwise
z_{ikjl}	=	1	If flow from output point k of department i to input point l of department j
		0	Otherwise
d_{ikjl}	=	Distance of the shortest path from output point k of department i to input point l of department j	
d_{ij}	=	Distance of flow path between department i and department j	
c_{ik}	=	1	If department i has flow type C and uses point k as an I/O point
		0	Otherwise
u_{ia}	=	1	If department i has flow type U and uses point a as an output point
		0	Otherwise
l_{ib}	=	1	If department i has flow type L and uses point b as an output point
		0	Otherwise
uu_{ia}	=	1	If department i has flow type U and uses point a as an input point
		0	Otherwise
ll_{ib}	=	1	If department i has flow type L and uses point b as an input point
		0	Otherwise
c_i	=	1	If department i has flow type C
		0	Otherwise
u_i	=	1	If department i has flow type U
		0	Otherwise
l_i	=	1	If department i has flow type L
		0	Otherwise
m_i	=	a{0,1}	A binary variable for each department i

Model:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{ij} \qquad (1)$$

s.t.

$$\sum_{k=1}^{p_i} x_{ik} = 1 \qquad \forall i \qquad (2)$$

$$\sum_{k=1}^{p_i} y_{ik} = 1 \qquad \forall i \qquad (3)$$

$$\sum_{l=1}^{q_j} \sum_{k=1}^{p_i} z_{ikjl} = 1 \qquad \forall i, j \qquad (4)$$

$$z_{ikjl} \le x_{ik} \qquad \forall i, k \qquad (5)$$

$$z_{ikjl} \leq x_{lj} \qquad \forall j, l \qquad (6)$$

$$\sum_{l=1}^{q_j} \sum_{k=1}^{p_i} z_{ikjl} dist_{ikjl} = d_{ij} \qquad \forall i, j \qquad (7)$$

$$x_{i1} = c_{i1} + u_{i1} \qquad \forall i \qquad (8)$$

$$x_{i2} = c_{i2} + l_{i2} \qquad \forall i \qquad (9)$$

$$x_{i3} = c_{i3} + u_{i3} \qquad \forall i \qquad (10)$$

$$x_{i4} = c_{i4} + l_{i4} \qquad \forall i \qquad (11)$$

$$x_{i5} = c_{i5} + l_{i5} \qquad \forall i \qquad (12)$$

$$x_{i6} = c_{i6} + u_{i6} \qquad \forall i \qquad (13)$$

$$x_{i7} = c_{i7} + l_{i7} \qquad \forall i \qquad (14)$$

$$x_{i8} = c_{i8} + u_{i8} \qquad \forall i \qquad (15)$$

$$x_{ik} = c_{ik} \qquad \forall i, k \geq 9 \qquad (16)$$

$$y_{i1} = c_{i1} + uu_{i1} \qquad \forall i \qquad (17)$$

$$y_{j2} = c_{j2} + ll_{j2} \qquad \forall j \qquad (18)$$

$$y_{i3} = c_{i3} + uu_{i3} \qquad \forall i \qquad (19)$$

$$y_{i4} = c_{i4} + ll_{i4} \qquad \forall i \qquad (20)$$

$$y_{i5} = c_{i5} + ll_{i5} \qquad \forall i \qquad (21)$$

$$y_{i6} = c_{i6} + uu_{i6} \qquad \forall i \qquad (22)$$

$$y_{i7} = c_{i7} + ll_{i7} \qquad \forall i \qquad (23)$$

$$y_{i8} = c_{i8} + uu_{i8} \qquad \forall i \qquad (24)$$

$$y_{ik} = c_{ik} \qquad \forall i, k \geq 9 \qquad (25)$$

$$\sum_{a \in A} u_{ia} = u_i \qquad \forall i \qquad (26)$$

$$\sum_{a \in A} uu_{ia} = u_i \qquad \forall i \qquad (27)$$

$$\sum_{b \in B} l_{ib} = l_i \qquad \forall i \qquad (28)$$

$$\sum_{b \in B} ll_{ib} = l_i \qquad \forall i \qquad (29)$$

$$\sum_{k=1}^{p_i} c_{ik} = c_i \qquad \forall i \qquad (30)$$

$$u_i + l_i + c_i = 1 \qquad \forall i \qquad (31)$$

$$l_{i2} - ll_{i7} = 0 \qquad \forall i \qquad (32)$$

$$l_{i7} - ll_{i2} = 0 \qquad \forall i \qquad (33)$$

$$l_{i4} - ll_{i5} = 0 \qquad \forall i \qquad (34)$$

$$l_{i5} - ll_{i4} = 0 \qquad \forall i \qquad (35)$$

$$u_{i1} - uu_{i3} \leq m_i \qquad \forall i \qquad (36)$$

$$u_{i1} - uu_{i6} \leq 1 - m_i \qquad \forall i \qquad (37)$$
$$u_{i3} - uu_{i1} \leq m_i \qquad \forall i \qquad (38)$$
$$u_{i3} - uu_{i8} \leq 1 - m_i \qquad \forall i \qquad (39)$$
$$u_{16} - uu_{i1} \leq m_i \qquad \forall i \qquad (40)$$
$$u_{i6} - uu_{i8} \leq 1 - m_i \qquad \forall i \qquad (41)$$
$$u_{i8} - uu_{i3} \leq m_i \qquad \forall i \qquad (42)$$
$$u_{i8} - uu_{i6} \leq 1 - m_i \qquad \forall i \qquad (43)$$

Figure 2.6 MIP Formulation for Single I/O Placement with Intra-departmental Flow

The objective function (1) represents the sum-product of the flow values and distance between departments. Explicitly, the flow value represents the unidirectional volume of material passing from one department to another. The double sum accounts for the unidirectional flow that occurs in the opposite direction. The distance represents the contour measurement of aisle length between two departments. The objective is to minimize the total material travel throughout a facility. Thus, sets of departments having significant flow values between them will likely have much shorter paths and be located near each other. Binary variables are used in all constraint equations to ensure the restrictions for the path layout problem. Constraints (2) and (3) ensure that no department is permitted to have more or less than one input and one output station. Constraint (4) is necessary in determining exactly one flow path for all sets of departments that have flow values between them. Constraints (5) and (6) ensure that a singular flow path can only exist when the input point and output point in question exist. Constraint (7) makes certain that the distance between an output point and an input point is only substantiated in the total flow path when that singular path actually exists. The details of each flow type and their I/O candidates were discussed earlier. Two sets of nine equations for each flow type are necessary for establishing the restrictions between flow type and choice of I/O. Equations (8) – (16) guarantee that only one flow type can be chosen for the output point of each department. Similarly, constraints (17) – (25) provide the same restriction for the input point of each department. Constraints (26) – (30) guarantee model integrity by checking that the input and output point of each department adhere to the flow type of each department. Constraint (31) ensures certain that no department may have multiple flow types. The possibility of an input point being assigned to a different flow type than the same department's output point is prevented by constraints (36) – (43).

The MIP formulation described is not a practical methodology for implementation due to extensive CPU requirements for large scale problems. This method is not ideal for implementation as a subroutine to block layout optimization. The described MIP model indicates the level of difficulty and constraint required to optimize the aisle structure of a layout when considering multiple flow types and distinct I/O points.

2.5 GENETIC ALGORITHMS

Genetic algorithms (GAs) are a family of global optimization heuristics that were described by Holland (1975) [15] to model adaptation processes. GAs are based on the principle of evolution, or survival of the fittest. GAs represent a subset in the more general field of study of evolutionary algorithms (EAs). Most differences between the classification of EA and GA methodologies are due to problem specific goals (i.e. predicting changes in an environment, randomly arriving at optimal solutions, etc.) and choice of solution encoding (binary or floating point). Specific problems merit different choices, combinations, and designs of the fundamental

EA building blocks: solution encodings (data structures) genetic operators (selection, crossover, and mutation), and fitness functions (evaluation and penalty). The basic components of the integrated, evolutionary GA will be fully discussed in Section 4. Two books by Michalewicz (1996) [23] and Goldberg (2002) [14] provide an in-depth overview of the nature and purpose of GAs, as well as a substantial analysis of promising techniques and problem specific methodologies. From an operations research perspective, GAs are global search methods that provide the capability to simplify difficult mathematical programming problems without sacrificing solution quality.

Critical to the usefulness of GAs for the unequal-area FLP was the development of an adaptive penalty function by Coit, Tate, and Smith [11]. Similar to the technique of Lagrangian relaxation for combinatorial optimization problems, the adaptive penalty function uses feedback obtained throughout the genetic process to encourage a GA to explore within the feasible region. The notion of a penalty function can generally be described by considering the optimization problem of,

$$\begin{aligned} \min \quad & z(x) && \text{(P1)} \\ \text{s.t.} \quad & x \in A \\ & x \in B \end{aligned}$$

where x is a vector of decision variables, the constraints x in A are easy to satisfy, and the constraints x in B are hard to satisfy. The problem (P1) is reformulated as,

$$\begin{aligned} \min \quad & z(x) + p(d(x, B)) && \text{(P2)} \\ \text{s.t.} \quad & x \in A \end{aligned}$$

where $d(x,B)$ is a metric function describing the distance of the vector x from the region B, and $p(.)$ is a monotonically non-decreasing penalty function such that $p(0) = 0$. If the penalty function grows quickly enough outside of B, the optimal solution of P1 will also be the optimal solution of P2. Furthermore, any optimal solution of P2 will provide a lower bound on the optimum for P1. This bound will be tighter than that obtained by optimizing $z(x)$ over A. By using a distance based dynamic penalty function, highly infeasible solutions are allowed early in a genetic search, while continually increasing the penalty imposed to eventually move the final solution to the feasible region. The penalty function of Tate and Smith introduces the concept of a near feasibility threshold (*NFT*) corresponding to a constraint or set of constraints. The *NFT* is defined as the threshold distance from the feasible region at which a user would consider the search as "getting warm." The penalty function encourages the GA to explore the feasible region and the *NFT*-neighborhood and discourages search beyond the threshold. The *NFT* is both problem specific and constraint specific. Thus, a penalty function should scale itself to a particular problem based on the severity of the constraints imposed. The gap between the best feasible value and the best infeasible value during a genetic search indicates the severity of a problem-specific constraint set. The generalized form of an adaptive penalty function for a minimization problem with n constraints takes the form of,

$$F_p(x) = F(x) + (F_{feas} - F_{all}) \sum_{i=1}^{n} \left(\frac{d_i(x,B)}{NFT_i} \right)^k \qquad (2\text{-}6)$$

where $F(x)$ is the unpenalized objective function value for solution x, F_{all} denotes the un-penalized value of the best solution yet found, F_{feas} denotes the value of the best solution yet found, the exponent k is a user-specified severity parameter, and the NFT_i is the near-feasible threshold for constraint i. The *NFT* takes the form of,

$$NFT = \frac{NFT_0}{1+\Lambda} \qquad (2\text{-}7)$$

where NFT_0 is some upper bound for the *NFT*, and Λ is a dynamic search parameter that adjusts the *NFT* based on the search history. If Λ is set to *0*, a static *NFT* results. Λ can be defined as a function of the search by,

$$\Lambda = f(g) = \lambda g \qquad (2\text{-}8)$$

where it is based on the generation number, g. .A positive value of Λ results in a monotonically moving *NFT* and a larger Λ more quickly decreases the *NFT* as the search progresses.

To adapt the generalized penalty function to the unequal-area FLP, the authors discuss the nature of the problem. Consider that a layout with many infeasible departments might require costly repair, whereas a layout with one infeasible department may be made feasible by shifting the department to an adjoining bay. Asserting that the degree of infeasibility of any one department is less important than the number of infeasible departments, Equation (2-6) is adjusted to the unequal-area FLP by,

$$C_{ip}(\prod) = C_i(\prod) + (C_{feas} - C_{all})\left(\frac{n_i}{NFT}\right)^k \qquad (2.9)$$

where n_i is the number of infeasible departments for a layout $\prod$, k is the severity parameter, $C_i(\pi)$ is the un-penalized objective function value, and $C_{ip}(\pi)$ is the penalized objective function value. The *NFT* for this problem is an integer value, thus a static *NFT* is employed. The results of Coit, Tate, and Smith [11] indicate a very robust and effective means of guiding a genetic search to the feasible region of a search space. The authors also indicate that specification of penalty parameters was not necessarily crucial for attaining quality solutions.

Schnecke and Vonberger [33] introduce a hybrid GA for the unequal-area FLP. A novel genotype, or solution, representation based on binary trees is proposed to limit the tendency of GAs to create infeasible layouts. Shape functions for departmental shape constraints and weighted matching for iteratively pairing departments based on connectivity, or flow, are incorporated into the hybrid GA. The multiple objectives of shaping and locating departments in an unequal-area FLP can be solved in one optimization step. The model is made more efficient by converting continuous shape functions into discrete shape functions as shown in Figure 2.6. The continuous shape function is equivalent to the aspect ratio that is used in the integrated, evolutionary approach. Transforming the continuous function into a smaller set of discrete shape possibilities dramatically reduces the computational effort required to solve the unequal-area FLP. The probability of producing a superior layout using discrete shape functions is less likely than in the continuous case. Discrete shape functions could be considered in the case of severely restricted departments (i.e. large machinery, furnaces, etc.) to reduce the size of the search space.

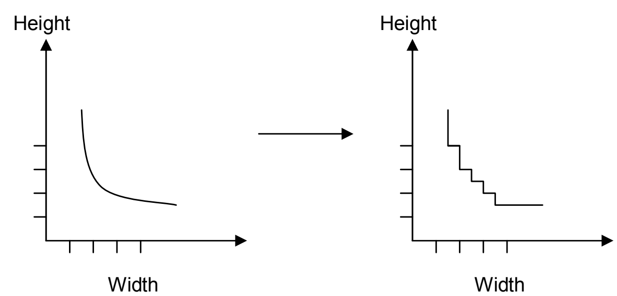

Figure 2.7 Transformation of Continuous Shape Function into Discrete Shape Function

With the intent of maintaining diversity of a GA search process, Aggarwal et al. [1] propose an optimized crossover mechanism for the independent set problem. Crossover is a genetic operation that mimics the natural process of breeding. The authors' crossover operator works to produce the best objective function value from the feasible set of children, as well as a solution that helps to maintain diversity in the search space. A common problem when using GAs is the propensity of the algorithm to prematurely converge and not explore the entire search space. By using the information from two independent sets, or parents, another independent set is created by the optimized crossover mechanism through the bipartite matching algorithm. Although the independent set problem is not the same as the unequal-area FLP, the concept of constructing knowledge-based crossover mechanisms that exploit the structure of a solution, rather than the encoding, presents a challenging design issue. The optimized crossover is better suited to problems in which the number of conflicting genes between parents is sufficiently small. For the case of the unequal-area FLP, a knowledge-based crossover mechanism, similar to the knowledge-based adaptive penalty function of Equation (2-6), may lead to reductions in design time and to better solution quality.

Banerjee et al. [5] employ genetic search for the FLP through a unique string representation that maps to a graphical form. Each node in their graph corresponds to a department and the edges connecting them indicate the flow pattern. Representation of the continuous plane, or unequal-area, layout is achieved by linking each node in an encoding to a subset of node locations. When a layout is changed by the relocation of a single node, the location of all other nodes in the encoding relative to the altered node do not have to be recomputed. A polar coordinate system is used as the basis for mapping a string to a graph. Instances of linear programs are created by graphical operators that remove edge overlaps, perturb ill-located nodes, and constrain the dimensions of nodes in a layout. New solutions are produced through a GA with the direction of the search enhanced by the graphical improvements described. Thus, an efficient search mechanism is provided.

The choice of solution encoding has a tremendous impact on the efficiency of a GA [23]. A good encoding can enhance the performance of genetic operations, such as crossover and mutation [23]. To that affect, Eklund et al. [12] compare two chromosome encodings for the Flexbay FLP. The results suggest that consideration of chromosome encodings should be made with relevance to genetic operators, and vice versa. Since GA encodings interact with genetic operators, the specification of encoding and of genetic operations is largely problem-specific [23]. Eklund at al. [12], also indicate that GAs exhibit some of the same interactive relationships, but to a much lesser degree.

2.6 DYNAMIC AND STOCHASTIC MODELS

Due to the complexity of the FLP, modeling the problem with dynamic programming and stochastic models, in general, has been limited. The issue of variable product demand over periods of time however, continues to be an ever increasing consideration in facility layout. In 1965, the inventor of dynamic programming (DP), Richard Bellman, produced an application for location-allocation problems [7]. He described how quasi-linearization could be used to transform a minimizing function into a DP problem which could possibly be solved computationally. The model is limited by the assumption of convexity that is required to perform quasi-linearization. The method can be used however, to obtain a sequence of lower bounds for a general location-allocation problem. Konak et al. [17] discuss the possibility of tightening lower bounds for the unequal-area FLP to creating a more robust MIP formulation.

A DP algorithm that utilizes an approximation in policy space to achieve a least cost matrix configuration was presented by Flores and Roberts (1964) [31]. A policy refers to a layout configuration with respect to equal-area departments of identical dimensions. The authors assume there exist a set of initial policies indicated by physically reasonable layout configurations. Iteration in policy space occurs by interchanging the locations of two departments with respect to an initial policy. Evaluating each interchange individually with respect to the initial configuration indicates an improved cost configuration to be used in a successive policy iteration. Their method always yields a local minimum for an initial policy. A global minimum is only guaranteed when all possible configurations are used as a starting point for policy evaluation. Enumerating all initial policies is computational inhibitive since there exist n! layout configurations for n departments. Extending the concept of policy iteration to the unequal-area FLP is considerably more computationally infeasible. To improve the efficiency of the DP policy evaluation technique, the authors run trials of initial policies for smaller problems to determine the average number of interchanges needed to reach a local minimum. The results indicate their method is favorable to evaluating all possible configurations. The method can also be enhanced by generating a lower bound from a flow matrix.

Norman and Smith [28] approach uncertainty in material handling costs through a random keys GA that incorporates expected value and standard deviations of product forecasts. The solution encoding assigns a random $U(0,1)$ variate, or random key, to each department as shown in the example on the following page. Sorting the random keys in ascending order provides the sequence of departments in a layout. The authors invoke the central limit theorem as a means of approximating a distribution for a set of independent products with an expected demand and a standard deviation per unit of time.

Department	A	B	C	D	E
Random *U*(0,1) Variate	0.34	0.76	0.63	0.15	0.97

Asserting that the total material handling costs depict a Gaussian probability distribution, expected values can be assigned to a layout for a range of scenarios. The authors claim that an analyst would likely prefer those layouts that perform well for both an overestimated and a less than expected production forecast. Evaluating a set of layouts pertaining to various expected values ($z_{1-\alpha}$), the dominant layouts that minimize material handling costs for a range of uncertainties (α) can be identified. Plotting objective function values over a range of uncertainties will indicate the dominant layouts. The approach of Norman and Smith [28] circumvents the need for specifying probabilities or random variable distributions.

An attempt to incorporate the uncertainty of material flow in the FLP through the use of fuzzy numbers was made by Cheng et al. [9]. Using a GA framework, the total cost of a layout and the flows between departments are modeled as trapezoidal fuzzy numbers that represent many possible real numbers. Fuzzy number solutions are ranked through the use of possibility theory and fuzzy integrals, since the evaluation of a layout solution using fuzzy numbers is not possible. By modeling fuzzy interflow between departments as a quadruple of estimations from best-case to worst-case, a layout solution that is satisfactory for all cases is produced. An advantage of this approach is the elimination of the requirement for a planner to provide a probability distribution for the interflows.

Another approach for incorporating uncertainty into the solution of FLPs comes from Rosenblatt [32]. He provides a DP formulation to deal with the uncertainty that occurs over time in maintaining an efficient layout. The objective is to select the sequence of layouts which would minimize the overall sum of flow costs over time. His approach offers an optimal or heuristic formulation depending on the CPU efficiency with which the static formulation of the FLP at each sequence in time can be solved. It is clear from his work, that a large number of departments would require a reduction of states at each sequence in time to achieve CPU efficiency. Such a formulation in the context of the integrated, evolutionary approach in this thesis would have to be heuristic to extend the problem to a stochastic environment.

2.7 QUALITATIVE APPROACHES

The multiple-criteria approach of Malakooti and Tsurushima [20] uses both quantitative and qualitative factors to influence the design of a facility layout. The authors argue that the classical formulation of the FLP as a well-structured mathematical problem does not permit the applicability of an ill-structured problem that considers qualitative decision making. The qualitative criterion upon which a layout is designed is enhanced by their model. It is conceivable that qualitative decision rules could be embedded within the integrated, evolutionary approach to produce layouts that are user-preferable, mathematically feasible, and at or near a minimum solution.

Montreuil and Venkatadri [26] propose a goal-oriented qualitative approach to the FLP that accommodates product flow uncertainty in a layout over time. An analyst is required to provide the perfect layout that an organization desires or conceptualizes at some point in the future. This conceptualization can be thought of as an expansion or reduction to the present layout structure. Using the mature, or future, facility layout, a set of scenarios representing possible layouts can be created and used to work in a backwards fashion generating intermediate layouts. A linear programming formulation is provided for their goal-oriented facility layout approach.

3.0 PROBLEM STATEMENT

The unequal-area FLP is defined within a fixed rectangular region, or layout, of area HxW, where H is the height and W is the width. The number of departments and the area of each department are known a priori, with the sum of departmental areas equivalent to the total layout area as indicated in Equation (3-2).

$$A = HxW \qquad (3\text{-}1)$$

$$A = \sum_{i=1}^{n} l_i w_i \qquad (3\text{-}2)$$

where
A = the total area of layout
H = the height of a layout
W = the width of a layout
n = the number of departments
l_i = the length of department i
w_i = the width of department i

Although departmental areas are defined, shapes and orientations for each are not. To restrict the number of possible departmental shapes and orientations, minimum side requirements are provided for each department. This provision ensures realistic, or physically realizable, layout solutions. A matrix of flow values indicating material movement between each pair of departments is also provided. In addition, the intra-departmental flow type (U, L, or C) within each department is given. These intra-departmental flow patterns provide a discrete set of feasible I/O locations for each department, which aids in reducing the complexity of the problem.

The objective of this problem is to minimize the total material travel within a facility layout as defined by the objective function in Equation (2-1). This function is amended to include the adaptive penalty function of Equation (2-3) to give the form of Equation (2-5). To begin to define a layout solution for the unequal-area FLP, the sequence of departments and the number of bays are determined simultaneously using the Flexbay formulation. The departmental sequence and the bay structure are the primary elements of a block layout. The process moves to establish the number of departments to be placed in each bay and the corresponding width of each bay.

Once a block layout structure has been set, the detailed design optimization of the unequal-area FLP begins. The candidate I/O points for each department, as indicated by flow type, form a material flow network of nodes and arcs consisting of the departmental edges connecting these nodes. Distances between nodes, or I/O points, are defined as rectilinear contour paths that follow departmental perimeters. Travel through departments, as indicated by a centroid metric, is not permitted. Although the path or movement of material flow within a department is explicitly accounted for by flow type, the distance of intra-departmental travel is not included. Each department is restricted to have one input point and one output point. Although these points may occupy the same location for a given department, sharing of I/O points between departments is not permitted. A shortest path network of the I/O points is optimized through a constructive placement heuristic. A perturbation scheme is used in conjunction with the constructive heuristic to further improve flow paths. The detailed design of locating I/O points implies an aisle structure for the flow of material throughout the entire facility. When the block layout and I/O placement have completed, a fully defined layout is obtained and the analyst is provided with a substantial amount of information.

4.0 METHODOLOGY

This chapter discusses the methodology of the integrated, evolutionary approach.

4.1 UNEQUAL-AREA, SHAPE CONSTRAINED BLOCK LAYOUT

Block layout refers to the concept of a facility layout that is comprised of departments with rectangular areas. The unequal-area, shape-constrained layout problem consists of a rectangular region with dimensions $H \times W$, a sequence of departments from 1 to n with pre-specified areas, the flow values indicating material travel between all sets of departments, and restrictions as to the shapes that departments can assume. Realistic departmental shapes are enforced through pre-defined minimum side lengths for each department and allowable aspect ratios. The objective is to identify a layout that minimizes the total material travel between all sets of departments.

Solutions for this problem are formulated using the Flexbay structure [36]. The Flexbay structure forces departments into bays where every department within a bay must adhere to the bay's width. The number of bays, the width of each bay, the sequence of departments, the number of departments in each bay, and the height of each department within a bay are all determined in the Flexbay formulation. The Flexbay formulation eliminates many unequal-area layouts from consideration. There are $2^{n-3}n!$ distinct flexible bay layouts for n departments [36], which is a substantial benefit to CPU implications when considering the unequal-area FLP. The

Flexbay reduction is further justified because it has been shown that straight line aisles in one direction of a facility layout help to minimize material flow. The Flexbay formulation for the unequal-area FLP provides a means of generating layouts with reasonable CPU requirements.

An initial population of candidate layouts is generated at the beginning of the integrated GA of this thesis. The first step of the Flexbay formulation is to initialize the departmental sequence of the n departments in numerical order. Departments are then randomly selected and placed one at a time. The first selection is designated as the last department in the sequence, the second as the next to last, and so on, until all departments have been placed. Bay structure is then determined for a layout. Using Equation (4-1), the number of bays is determined as a function of the number of departments and the dimensions of a layout.

$$\text{Bay Probability} = \frac{1}{\sqrt{n\left(\frac{W}{H}\right)}} \qquad (4\text{-}1)$$

where n = the number of departments
W = the width of the facility
H = the height of the facility

The bay probability is compared with a pseudo-random variate that is generated from a random number seed supplied to a problem. For every department, a new random variate is calculated and compared with the static bay probability. When the pseudo-random variate is less than the bay probability, a bay is added to a layout. At the same time, the departments to be placed within each bay are determined. Knowing the number of bays and the departments to be placed in each, the dimensions of each bay and each department are found. The width of a bay is calculated as given by Equation (4-2). The width of each bay mandates the width of every department in that bay. The height of each department is determined by Equation (4-3).

$$w_b = \frac{A_b}{H} \qquad (4\text{-}2) \qquad\qquad\qquad h_i = \frac{a_i}{w_b} \qquad (4\text{-}3)$$

where w_b is the width of bay *b*, A_b is the total area of the departments in bay *b*, H is the height of a layout, h_i is the height of department *i*, a_i is the area of department *i*. Finally, if the width or height of any department is less than its pre-specified minimum side length, that department is labeled infeasible. Once the initial population of candidate layouts has been created, the integrated GA begins the evolutionary process.

4.2 THE BLOCK LAYOUT GENETIC ALGORITHM

The premise of GAs is based on the biological phenomena of evolution. Analogous to the natural genetic processes of breeding and mutation that can improve upon the deficiencies of prior species, a mathematically genetic process may improve upon the deficiencies of prior results. From an operations research perspective, the concept of evolutionary computation contrasts significantly with traditional optimization techniques that require information about constraints, derivatives, solution surfaces, etc. Also, the objective function is not required to be smooth, continuous or unimodal. GAs instead, rely on the representation of a chromosome as a data structure to affect better search methods. Chromosome representations of solutions permit a means of evaluation and the use of genetic operators.

GAs are stochastic heuristic programs that lack the use of a constraint set, thus alternative methods are required to explore feasible regions of a search space. This is primarily achieved through an adaptive penalty that is included in the objective function, or evaluation mechanism.

It should be noted that a penalty function does not replace the unequal-area FLP constraint set. Rather, it emulates the effect of the constraint set.

In the GA framework, a population of candidate solutions is maintained rather than generating a sequence of candidate solutions one at a time. The solution encoding imitates a chromosome on which specific genes describe various aspects of an individual. For the unequal-area FLP, an individual is a layout solution that can be drawn from the information provided in chromosome encodings. The permissible values for each gene in a chromosome are known as alleles. A breeding mechanism produces new individuals using parent encodings from the population. A mutation mechanism perturbs an encoding to produce a nearby solution. While breeding, or crossover, produces new solutions, mutation helps to ensure a diverse population of solutions. A GA is defined by a set of parameters that dictate the size of the population to be maintained and the frequency and probabilities with which genetic operators function.

The GA used in this thesis demonstrates properties that are common to GA methodologies in general. The search process of the GA is highly parallel due to the various search directions that result from members of the population. GAs have been shown to operate effectively under varying parameter settings. Thus, an emphasis on tuning parameters for this particular problem is not as critical as is the case with other optimization methodologies. A potential drawback of GAs results when solution encodings with nearly identical allele values produce substantially variable objective function values. If such problems can largely be avoided, GAs will generally perform well and find near optimal solutions.

GAs have exhibited significant potential for efficiently producing results at or near optimality for the both the unequal-area FLP [35] and the I/O location problem [3]. This thesis employs a block layout GA for the unequal-area FLP. An I/O placement GA could be used as well, but CPU limitations prevent integration of dual block layout and I/O GAs. An I/O GA could however be used on the most promising solutions from the integrated, evolutionary approach for further improvement. A much quicker method of I/O location is used, and the results of the integrated approach will be shown to be preferable to using separate or sequential GAs. Solution quality and CPU efficiency will demonstrate the reasoning for the integrated approach. The details of the integrated GA used in this thesis, the evaluation mechanism, the genetic operators, and the parameter settings are described in the following sections. Common GA methodologies not used in this thesis are also discussed in the interest of justifying the techniques that are used.

4.2.1 Solution Structure

The solution structure of any GA plays a substantial role in the ease of implementation and in the effectiveness of the methodology. In biological terms, chromosomes are the elements of an organism that describe its anatomy. A genotype refers to the complete set of chromosomes needed to define an organism. In a GA setting, chromosomes are the building blocks of a solution encoding. For the unequal-area FLP, a fully defined layout is comprised of two chromosomes, one for the block layout and one for the I/O locations. The chromosomes are comprised of genes that contain all the relevant information necessary to construct a detailed block layout. Each gene possesses a set of alleles for the purpose of indicating departmental configuration, bay structure, or I/O location.

A conflict arises when considering the encoding of a solution. Providing too much information might require exhaustive CPU requirements and significantly increase the production of infeasible solutions throughout the genetic process, while too little information might prohibit full definition of a solution. The integration of block layout and I/O location required a combination of encoding schemes. The encodings were designed to limit the occurrence of infeasible solutions while maintaining full definition of a layout solution.

The encoding of the block layout, the upper chromosome, is broken into two segments. The first segment is a string of *n* departments, which indicate departmental order from bottom to top and left to right within a layout's boundaries. The second segment of the upper chromosome indicates the number of bays in a layout and the placement of those bays. An example upper chromosome is shown below.

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

The translation of this encoding is as follows: twenty departments in the order of bottom to top and left to right with five bay breaks occurring at departments 12, 11, 18, 19, and 1. Figure 4.1 illustrates the layout for this encoding.

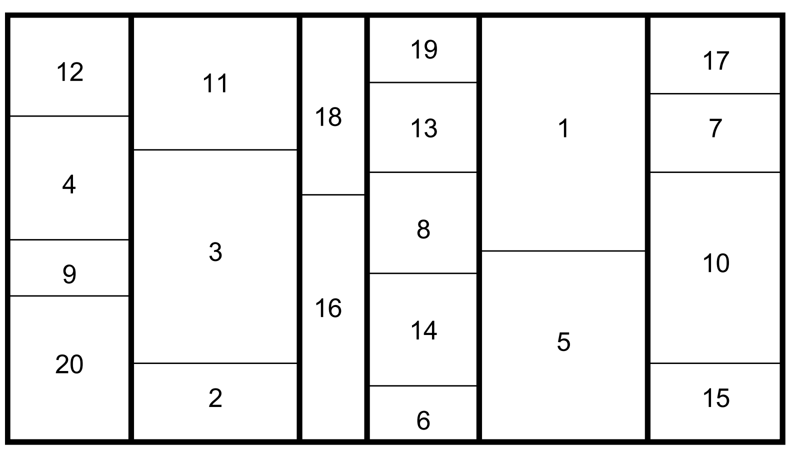

Figure 4.1 Example Layout of an Upper Chromosome Encoding

The second encoding, or lower chromosome, contains the information needed to define I/O locations and identify an aisle structure for material flow. Each departmental gene from the upper chromosome carries with it a gene holding two allele values that indicate the location of its input point and the location of its output point. The example below defines a lower chromosome for a layout with five departments, each with a predetermined flow type.

Department	**1**	**2**	**3**	**4**	**5**
Flow Type	***C***	***U***	***L***	***C***	***U***
Input Point	6	6	2	8	3
Output Point	6	8	7	8	1

The interpretation of this chromosome is straightforward. Department *1* is characterized by *C*-flow and has its input and its output point located in the *4* position, with reference to the set of candidate I/O locations defined in Figure 2.3. Department *2* is assigned *U*-flow and has its input point located in the *6* position and its output point located in the *1* position. The other departments are defined in the same way. Figure 4.2 provides a depiction of the example lower chromosome and the flow path between I/O points that it implies.

Each department in the lower I/O chromosome has a one to one identity with the same numbered department from the upper block chromosome. Thus, the sequence of departments in the lower chromosome is not relevant for proper interpretation. The information carried in the upper and lower chromosome is the only data pertaining to a layout that is eligible for genetic operations. The upper and lower chromosomes carry sufficient information to fully describe a detailed block layout.

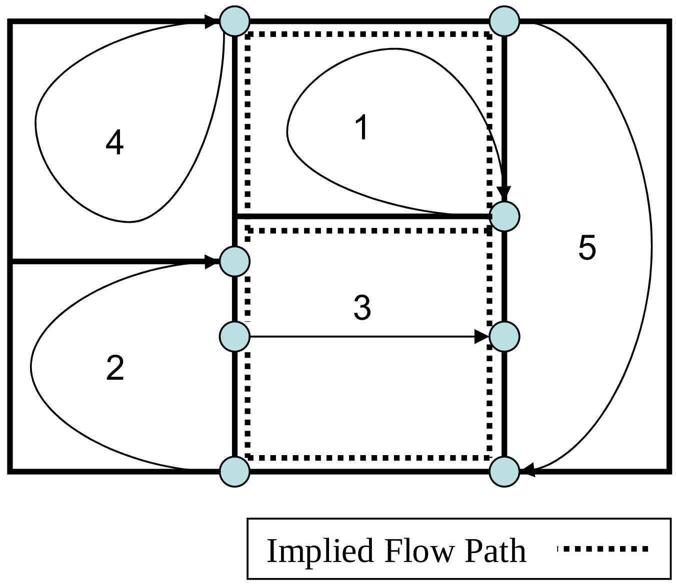

Figure 4.2 Example Layout of a Lower Chromosome

4.2.2 Evaluation Mechanism and Adaptive Penalty Function

The integrated, evolutionary GA evaluates layout encodings as defined by Equation (2-5). As mentioned earlier, this function is a combination of the total flow cost between all departments of Equation (2-1) and the adaptive penalty function of Equation (2-3). The distance measure, d_{ij}, is defined as the shortest rectilinear path along departmental contours between the input of a department and the output of another department. The penalty is adaptive because the severity of the penalty imposed is a function of the results from the entire search process, not just of the solution being evaluated. The evaluation mechanism enables the GA to rank layout solutions based on their score, or fitness. Solutions that score poorly are more likely to be eliminated from the population as generations succeed. This is important to the success of the integrated GA because it prevents repetitive consideration of solutions that would likely not add to solution quality. However, solutions with undesirable fitness are not entirely discriminated against as will be discussed later. The number of infeasible departments is raised to an exponential. Layouts with large numbers of infeasible departments are more severely penalized since they would require substantial repair at great CPU expense. However, small numbers of infeasible departments are not penalized as stringently as their value in terms of different search direction may be worth the expense to repair them. The other portion of the penalty helps to restrict the search from exploring solutions that are significantly deviant in value from the best known feasible solution found thus far. However, if the best feasible solution is much worse than the best overall solution, the candidate layout is likely to eventually be eliminated. This mechanism helps the GA search to continue in directions that have demonstrated the most promise.

4.2.3 Genetic Operators

The primary elements of a GA that allow for successful imitation of a genetic, evolutionary process are known as selection, crossover, and mutation. As stated earlier, crossover can be thought of as breeding. A population is sorted according to the evaluation mechanism in order of declining fitness, or from best to worst. The selection mechanism identifies solutions for genetic operations. A crossover mechanism combines the alleles of two solutions with one another in the hopes of creating a new solution that will demonstrate improvement to prior solutions. Mutation is the process by which a solution is altered in the interest of maintaining a divergent set of candidate solutions. The primary genetic operators mentioned require user defined information for effective usage. The details of these operators and the settings provided them are discussed in the following sections.

4.2.3.1 Selection

Selection is the process of choosing individuals from the current population to undergo genetic operations. Selected individuals are referred to as parents, which in turn will create new individuals, or children. There are many selection techniques available in the GA literature [23]. Selection mechanisms are usually designed to realize stochastic or deterministic effects, or a combination of both.

Roulette wheel selection is a stochastic method that affords each individual in a population an opportunity to become a parent based solely on fitness. Equation (4-4) determines the probability, p_i, with which an individual with fitness f_i will be selected from a population of size *popsize*. This method of selection is not used in this thesis because it presents a distinct possibility that an individual with superior fitness will dominate all other individuals with a high probability, thus limiting the search direction.

$$p_i = \frac{f_i}{\sum_{k=1}^{popsize} f_k} \qquad (4\text{-}4)$$

Ranking selection sorts a population from best to worst and assigns probabilities to each individual as a function of rank. Thus, the actual fitness of an individual is not directly used in its selection. The ranking function is commonly linear or exponential. Ranking selection is ideal for reducing the effects that non-uniform fitness values or wide ranging fitness values of a population can have on a selection process. It is not used for the unequal-area FLP because layouts with very poor fitness are likely to have many infeasible departments and require costly repair. Such layouts are ideally eliminated from the population. Tournament selection randomly picks a set of individuals from a population and then selects the fittest individual from the chosen set. Choosing the fittest individual of a set is known as elitist selection. This method provides stochastic and deterministic selection.

The selection process for the integrated, evolutionary approach is a rank-based quadratic method [35]. Prior to initiating the selection mechanism, the current population of layouts is sorted from best to worst based on fitness and an ordinal number ranking is assigned. Selection begins with a random seed, supplied as input to a problem, being sent to a uniform pseudo-random number generator. The pseudo-random variate, $U(seed)$, is used with the size of the population, *popsize*, in Equation (4-5) to create a random number between 1 and $\sqrt{popsize}$. The random number is squared, truncated, and taken to be the rank of the parent to be selected.

$$Random\ Number = \left\lfloor \left(\sqrt{popsize+1}-1\right) * U(seed) \right\rfloor + 1 \qquad (4\text{-}5)$$

The selection process is repeated until two distinct parents have been identified. The parents are then sent to the crossover mechanism for reproduction. The rank-based quadratic selection method gives preference to higher ranked individuals while allowing all individuals a chance to reproduce. It provides a desirable balance of improving solution quality and enhancing search direction without exhaustive CPU requirements.

4.2.3.2 Crossover

Crossover is the genetic process of combining two parent chromosomes to produce a single child chromosome. A child chromosome receives all of its allele values from one parent or the other so that it will maintain similar features of the parent chromosomes. In this thesis, crossover is performed each iteration of the genetic search with the hope of identifying a better solution, or a solution that will lead to an advantageous search direction. There are several common crossover techniques from the literature. The crossover method for the integrated, evolutionary approach and alternative crossover methods will be discussed.

Aside from the specific operations, several factors influence the effectiveness of a crossover mechanism. The selection process defined in the previous section is used to find parent chromosomes for reproduction. The choice of parents weighs heavily on the effect of crossover. For instance, selecting parents with infeasible solutions is likely to generate a child that is even more infeasible. However, completely avoiding infeasible parent solutions may limit the search process. The rank-based quadratic selection mechanism and the adaptive penalty function used in this thesis help to reduce this potential conflict. Another consideration for crossover design is what to do with a child solution once it has been produced. A decision must be made as to whether the child will unconditionally enter the candidate population or be evaluated for entry. If the child is to enter the population, the decision of which individual it will replace must be considered since this thesis maintains a constant population size throughout genetic search. To illustrate the design of the crossover mechanism for the unequal-area FLP and the concept of crossover in general, single point, uniform, and parameterized uniform crossover will be discussed.

Single point crossover selects a point, or gene location, that is used to cut, or split, parent chromosomes into two pieces. The cut point indicates the segments of data from each parent chromosome that are interchanged to produce two children. An example of single point crossover performed on binary chromosomes is shown below.

Cut Point					↓					
Parent 1	1	0	0	1	|	0	1	0	1	1
Parent 2	0	1	0	0	|	1	0	1	0	1
Child 1	1	0	0	1	|	0	1	0	1	1
Child 2	0	1	0	0	|	0	1	0	1	1

The cut point is shown between the fourth and fifth genes of parents *1* and *2*. The segments to the right of the cut point from parents *1* and *2* are swapped with one another to produce two children. Single point crossover can be applied to other forms of solution encoding. The example below demonstrates single point crossover for alphabetic encodings, which could represent a facility layout with eight departments. The same operation as in the binary example produces two candidate layouts.

Cut Point					↓				
Parent 1	H	B	E	C	|	A	I	D	F
Parent 2	D	C	I	F	|	B	H	E	A
Child 1	H	B	E	C	|	B	H	E	A
Child 2	D	C	I	F	|	A	I	D	F

Inspection of child *1* reveals two instances of departments *H*, *B*, and *E*. This layout is not feasible and would require repair. Single point crossover is not used for the unequal-area FLP because of the likelihood of such occurrences.

Uniform crossover is the process by which two parents contribute allele values to one offspring. In contrast to single point crossover, single allele values are combined versus segments of consecutive allele values. Alleles are selected from either parent. The fitter parent is labeled as parent *1* and is given a higher probability of passing alleles to the offspring. The selection probability, P_c, is an approximate indication of what percentage of alleles will come from parent *1*. Common probabilities of selection are $P_c \in [0.5, 0.8]$. The process generates a uniform random number, $U_i \sim U\ (0,1)$, that is used with P_c to determine which parent will provide each allele. The rule of uniform crossover states that the ith allele of parent *1* is inherited if $U_i < P_c$. Otherwise, the allele is taken from parent *2*. With P_c = 0.5, an example of uniform crossover is shown below.

Parent 1	H	B	E	C	A	I	D	F
Parent 2	D	C	I	F	B	H	E	A
$U_i \sim U\ (0,1)$	0.62	0.36	0.75	0.34	0.09	0.56	0.97	0.88
Child	D	B	I	C	A	H	E	A

Employing uniform crossover for the unequal-area FLP as described is not desirable because it does not inhibit production of infeasible solutions. Instead, a variant of uniform crossover is used for this thesis.

A parameterized uniform crossover mechanism [14] with repair is used for the integrated, evolutionary approach. The same operation as described for uniform crossover is employed with a few modifications. The crossover techniques described thus far all operate on single chromosomes. Although the integrated, evolutionary approach entails the use of two chromosomes per individual, the crossover mechanism is restricted to working one chromosome per individual. This will be the upper chromosome that defines a block layout. Common gene locations in the parents that hold the same department are directly carried over to the child. The example of uniform crossover shown before indicated that duplicate allele values, representing departments, might occur more than once in a child. To prevent duplicate departments in a child, the parameterized uniform crossover mechanism leaves empty any gene location in the child that is attempted to be occupied by a department that has already been placed. Departments that have not been selected are placed in empty gene locations at the end of the crossover process. This ensures construction of feasible solutions. Solutions are feasible in the sense that multiple instances of any department will not occur, but not necessarily feasible in the sense that a department will be located within a facility's boundaries. An example of parameterized uniform crossover is shown below.

Parent 1	D	C	A	H	F	E	B	G
Parent 2	F	A	E	H	B	D	C	G
Common Locations	-	-	-	H	-	-	-	G
$U_i \sim U(0,1)$	0.22	0.36	0.75	-	0.09	-	0.17	-
Random Choice	D	A	E	H	F	-	B	G
Leftover	C							
Child	D	A	E	H	F	C	B	G

The allele values of genes four and eight in parents *1* and *2* are identical, thus the child maintains departments *H* and *G* in the respective gene locations. Department *D* is taken from parent *1* in the first gene location and department *E* is taken from parent *2* in the third allele location. When the crossover mechanism arrives at the sixth gene location, departments *D* and *E* have already been selected and neither department is allowed to enter the child again. The sixth gene location remains empty until each gene location has been examined. It is filled with department *C*, which was leftover during the first phase of crossover. The example is devoid of the bay structure that is included in departmental chromosomes. The bay structure is taken from either parent, with no modification to the number or location of bay breaks, using the same selection technique as for the department sequence. In parameterized uniform crossover, the labeling of parent *1* and parent *2* is indicative of which parent was chosen first, not by their fitness.

Several parameters and rules are defined for the parameterized crossover mechanism. Crossover is performed once per generation. The resulting child always enters the candidate population regardless of solution quality. The child replaces the worst solution in the current population. A culling mechanism removes the worst solution to make room for the child and maintain a constant population size, but not before the child and the current population undergo mutation. The selection probability, P_c, was set to *0.5*. Thus, no reproductive preference was made towards the fitter parent and approximately *50%* of the selected alleles will come from parent*1* and the rest from parent *2*. If the pseudo-random variate for a gene location was less than the selection probability, the allele value from the gene location of parent *1* was given to the child. Once the I/O locations of a child are determined, the child enters the current population as a function of its fitness. The population must still undergo mutation before the next generation occurs.

4.2.3.3 Mutation

Mutation is the genetic process of altering one or more allele values in a chromosome to produce a new chromosome. Mutation provides additional variation to the population. If genetic search is restricted to crossover as the primary operator for generating new solutions, a genetic search is more likely to maintain a similar population in terms of the allele values at each gene location for a population. This results in a limited genetic search that is likely to converge prematurely. Using mutation enhances the randomness and variation of search direction throughout the genetic process and can dramatically increase the chances of finding local optima. However, mutation can also result in a completely random search if the specification and design of mutation operators are not carefully thought out. The frequency of mutation in genetic search is defined by a mutation probability, generally defined as in Equation (4-6). Low mutation probabilities (≤ 0.5) are common for GA implementation.

$$P_m = \frac{1}{length\_of\_string} \qquad (4\text{-}6)$$

where $length\_of\_string$ = the number of alleles in a chromosome

Three mutation operators are used in the integrated, evolutionary approach. As with the parameterized uniform crossover mechanism, the mutation operators are only allowed to operate on the alleles of the upper block layout chromosome. Two of the operators work on the bay structure of a chromosome, while the other alters the departmental sequence of a layout. One of the bay operators splits an existing bay into two adjacent bays and the other bay operator merges two adjacent bays. The sequence operator inverts a subsequence of departments while reserving the sequence of all other departments. The sequence mutation operator selects any two departments in the upper chromosome and reverses the sequence between and including the selected departments. An example of the inversion mutation operator is shown below.

Mutation Points		↓				↓		
Original Chromosome	H	B	E	C	A	I	D	F
Mutated Chromosome	H	I	A	C	E	B	D	F

The second and sixth gene locations holding departments *B* and *I* are selected. The sequence of departments from the second to the sixth gene location is reversed. The mutated chromosome indicates the new departmental sequence. Where the crossover mechanism alters the departmental sequence by selecting alleles from two parents, this operator alters the sequence within a selected chromosome. This operator is beneficial to the search direction, especially as the population of candidate solutions become more similar. If altering of sequence were solely accomplished through crossover, the search process would be severely restricted.

The bay split mutation operator selects a gene from the upper chromosome and uses the department of that gene as the split point for a new bay. The department chosen must reside in a bay that currently holds more than one department or the selection is not allowed. Also, the chosen department cannot be the last department in its current bay. After a legal selection has been made, the split operator designates the selected department as the last department in its current bay. It shifts the remaining departments of the current bay into an additional, adjacent bay. Thus, the department sequence is not altered and only the bay structure is modified. An example of the split operator is shown below.

									Bay Structure		
Split Point				↓							
Original Chromosome	H	B	E	C	A	I	D	F	B	A	
Mutated Chromosome	H	B	E	C	A	I	D	F	B	C	A

The original chromosome indicates three bays in its layout. Department *C* is chosen as the split point. The original chromosome holds departments *E*, *C*, and *A* in its second bay. The mutated chromosome indicates departments *E* and *C* are in the second bay, and department *A* is in the new bay. Thus, the mutated chromosome contains four bays.

The merge mutation operator selects a bay from the upper chromosome and merges it with an adjoining bay. If the bay chosen is the rightmost bay in the original chromosome, the bay is merged with the adjacent bay to the left. Otherwise, the selected bay is merged with the adjoining bay to its right. An example of the merge operator is shown below.

Merge Bay										↓
Original Chromosome	H	B	E	C	A	I	D	F	B	A
Mutated Chromosome	H	B	E	C	A	I	D	F	B	
									Bay Structure	

The second bay holding departments *E*, *C*, and *A* is chosen for merging. Since the second bay is not the last bay in the original chromosome, it is merged with the third bay. Thus, the number of bays is reduced to two and departments *E*, *C*, *A*, *I*, *D*, and *F* all reside in the same bay as indicated by the mutated chromosome.

The three mutation operators for the integrated, evolutionary approach are given probabilities of usage. Approximately *50%* of all mutations involve the departmental sequence operator, *25%* involve merging bays, and *25%* involve splitting bays. The decision to perform mutation on an individual in a population is made in the same way as for crossover. If a pseudo-random variate is less than the mutation probability, $P_m = 0.5$, the individual is mutated provided that a pre-specified number of mutants have not already been created. The maximum number of mutants is determined by the maximum mutant replacement percentage, $P_r = 0.8$, multiplied by the population size (*pop_size = 10*). Thus, no more than eight mutants are created during a single generation. Once each individual from the current population and the offspring from crossover have been evaluated, the mutation process ends. Each mutant replaces its original in the population, unless the mutant originates from the best feasible solution found thus far. Only if the mutant has better fitness than the best feasible solution found will the mutant replace the original. The new population is finalized by ranking each individual and removing the worst solution to maintain the population size.

4.2.4 Convergence

Since GAs are stochastic search methods, they require notification of when to terminate a search. There are several methods for specifying convergence of a GA. Search can be terminated after a specified number of generations, or a time limit could be imposed. As discussed in Section 2, a fitness threshold that indicates when the best fitness found is within a user-defined threshold can also be used to terminate genetic search. Population convergence indicates termination when the average fitness of a population is within a pre-specified threshold, and gene convergence indicates termination when the average value of a gene across the entire population is within a pre-specified threshold.

For this thesis, a generation number limit is imposed. Prior implementations of genetic search for the unequal-area FLP have indicated that quality solutions are found at various generations throughout the evolutionary process. Some are found at low generation numbers, while others become available nearly at the end of the generation number limit. In addition, tracking the fitness of individuals for the unequal-area FLP has shown that populations can remain stable for a significant portion of a genetic search before generating better solutions. Taking the recommendations of prior research and the nature of the methodology employed in this thesis into account, the number of generations is set to *100,000*.

4.2.5 Genetic Settings

The success of any GA can be enhanced through the selection of genetic parameters. Many optimization methodologies require parameter tuning prior to determining a set of parameters for implementation. The degree to which parameter optimization is a function of a methodology's usefulness is largely dependent on the methodology used and the specific problem to be solved. For the unequal-area FLP, GAs have demonstrated effectiveness under a variety of parameter settings [11]. Exhaustive tuning of parameters has not shown significant improvements to results. Each parameter used in this thesis was determined on the collective basis of improving solution quality, enhancing the search space, and reducing CPU requirements.

The number of generations was set to *100,000* for most test problems. The tests for the Armour and Buffa problem were set to *75,000* generations due to computational requirements. Due to the stochastic nature of GAs, each test problem was run for ten random seeds as shown below.

[(0123456789), (1234567890), …, (8901234567), (9012345678)]

The population size, *popsize*, for the number of solutions to be maintained in each generation was set to *10*. A larger value of population size might result in better solutions since more candidate layouts would be evaluated, but would require much greater CPU expense. The probability of selection, P_c, approximating the percentage of genes coming from parent *1* during crossover was set to *0.5*. The mutation rate, P_m, approximating the percentage of individuals from a population to undergo mutation was set to *0.5*. The maximum replacement percentage, P_r, determining the maximum number of mutations that could be performed each generation was set to *0.8*. The bay mutation probability, P_b, approximating the percentage of mutations affecting the bay structure was set to *0.5*. The bay merge, P_{me} , and bay split, P_s, mutation probabilities were both set to *0.25*. A summary of the genetic settings and parameters for the GA employed in this thesis are shown in Table 4.1.

Table 4.1 Summary of Genetic Parameters

# of Generations	100,000
Population Size	10
Random Seeds	[(0123456789), …, (9012345678)]
$\mathbf{P_c}$	0.5
$\mathbf{P_m}$	0.5
$\mathbf{P_r}$	0.8
$\mathbf{P_b}$	0.5
$\mathbf{P_{me}}$	0.25
$\mathbf{P_s}$	0.25

4.2.6 The Integrated Block Layout GA

A flow chart depicting the process of the integrated GA is shown in Figure 4.3. All elements of the integrated GA with the exception of those pertaining to I/O location have been discussed. The next section will describe the detailed design optimization of I/O placement that occurs in Figure 4.3.

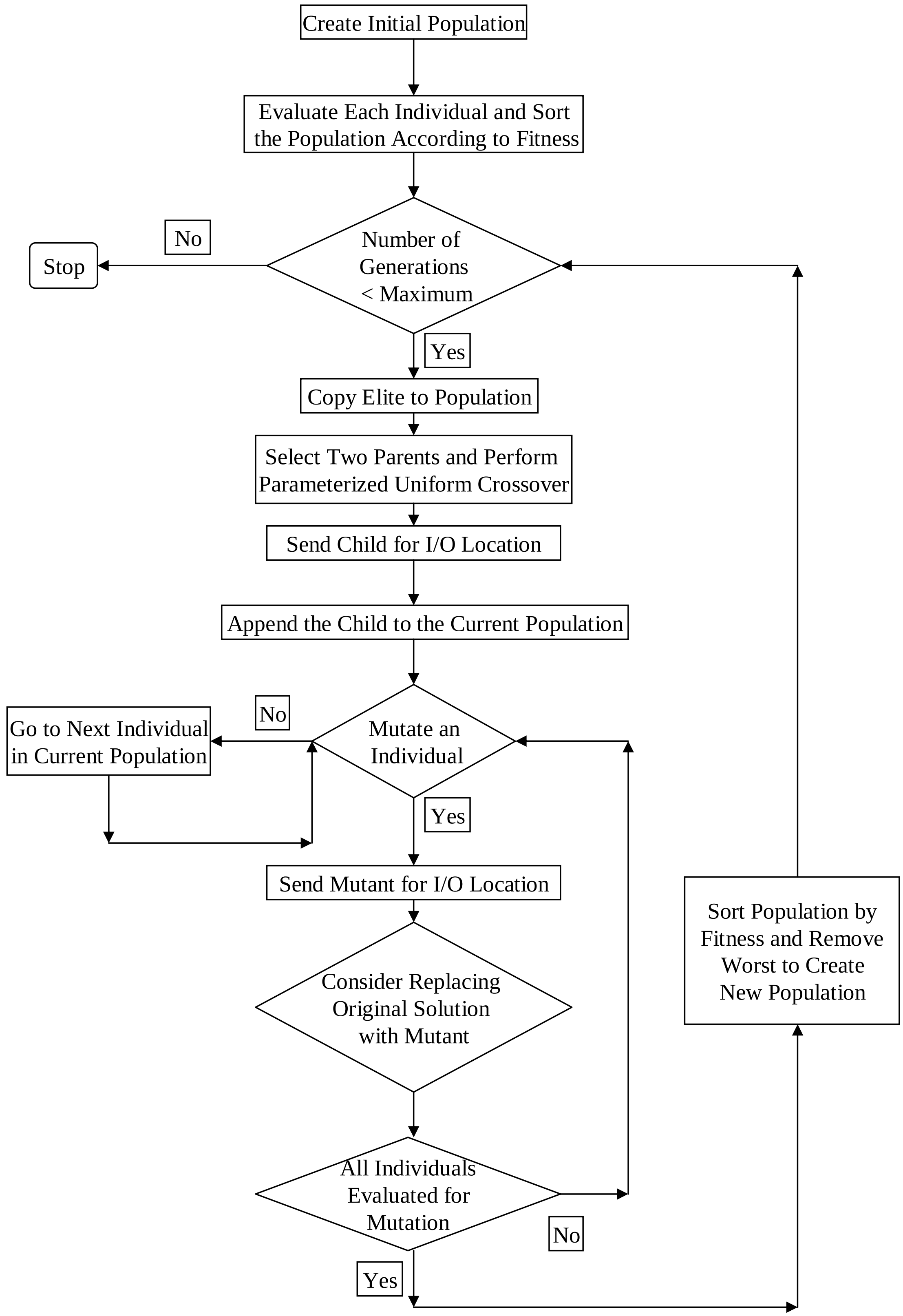

Figure 4.3 The Integrated GA Flow Diagram

4.3 DETAILED DESIGN

The detailed design problem of locating I/O points is described in this section. For every block layout that is considered during genetic search, a set of nodes, or I/O points, is connected by nonnegative arcs resulting in a material flow network. The arcs of the network connect each department's output point to every other department's input point. The arcs indicate the flow paths along departmental contours along which material travels through a facility. The I/O placement problem considered here is that of the single I/O MIP that was discussed in section 2. Each department is restricted to having only one input point and one output point per department. The single I/O placement problem is made possible by reducing the total number of candidate I/O points for a layout to the finite set of departmental intersection points. With this provision, the maximum number of candidate I/O points for a layout with n departments is $2N\text{-}2$ [2]. The total number of possible solutions for the single I/O placement problem is given by Equation (4-7) [2], where Loc_i is the set of candidate I/O points for department i.

$$\prod_{i=1}^{n} |\, Loc_i \,| \qquad \text{(4-7)}$$

The I/O placement problem is further defined by pre-assigned intra-departmental flow types, which will be discussed. The metric used to evaluate a network, as well as the methods of constructing the initial network and determining the final network, are described in the ensuing sections.

4.3.1 Contour Distance Metric

In order to quantify and evaluate a layout, a distance metric is needed. The choice of a distance metric for evaluating departmental separation in a layout is a critical factor in computational efficiency. The choice of distance measure is also extremely significant in terms of the physical interpretation of a layout. An optimized layout is not useful if the distance measure does not adhere to its physical environment. Several common distance metrics have been used throughout the course of facility layout research. Rectilinear distance and Euclidean distance are two frequently used metrics. Rectilinear, or Manhattan, distance is defined as the distance between two points measured along axes at right angles, as given by,

$$d(i,j) = \left|x_i - x_j\right| + \left|y_i - y_j\right| \qquad (4\text{-}8)$$

where $d(i,j)$ is the distance between points i and j with coordinates (x_i, y_i) and (x_j, y_j) respectively. The Euclidean distance metric is defined as the direct, or straight line, distance between two endpoints as given by,

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)}^2 \qquad (4\text{-}9)$$

with the same notation as the rectilinear metric. Figure 4.4 provides an illustration of the rectilinear and Euclidean metric. The key next to the figure identifies each metric. The metrics indicate the path of material flow between the centroids of departments *B* and *H*. The Euclidean metric traverses the boundary of department *E*, while the rectilinear metric traverses departments *E* and *G*. In an industrial environment, these implied paths may not be permitted due to physical barriers, such as walls. Operational complications, such as impeding production in the traversed departments, mixing of different production line quantities, or losing track of work-in-process inventories, can result from such movement.

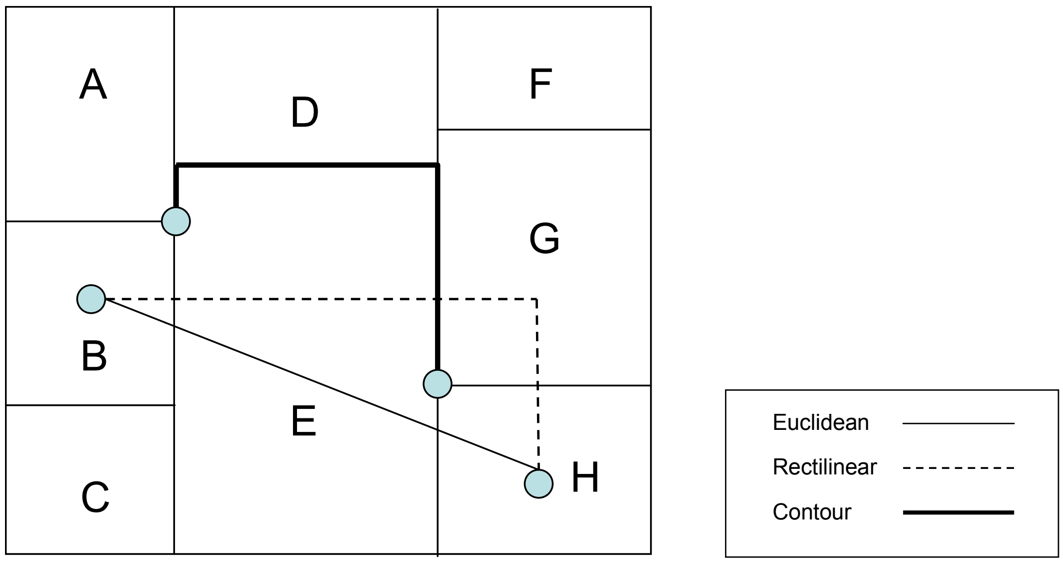

Figure 4.4 Three Distance Metrics

Figure 4.4 also illustrates the contour distance metric [27]. This measure represents the shortest rectilinear distance along departmental contours between the I/O points *j* and *k* as given by Equation (2-4). The contour path implies material flow from the output point of department *B* to the input point of department *H,* or vice versa. Using this measure, the choice of location for an I/O point becomes much more complicated as versus the assumption of a centroid I/O.

4.3.2 Inter-departmental and Intra-departmental Flow

As mentioned, the choice of the contour metric for this thesis invokes the consideration of material flow both within a department (intra-departmental) and between (inter-departmental) departments. As shown in Figure 4.4, the rectilinear and Euclidean metrics indicate material travel to and from the centroids of departments. Such material movement in a facility layout is generally not typical, especially in regards to the implied flows within the originating department and within the destination department. In using the contour measure, no indication of material travel within a department is given.

In order to define material flow within a facility, three distinct flow types were described in Section 2 [29]. Figure 2.4 illustrates the *U*-shaped, linear, and *C*-shaped flow types. The feasible set of I/O points along departmental perimeters as designated by these flow types was also discussed in Section 2. The justifications for restricting the possible I/O points of a department are both mathematical [2] and intuitive. Typically, material travel within a department can be classified as one of these three flow types. *U*-shaped flow indicates travel from one corner of a department to an adjacent corner of the same department, which is typical of material that enters and exits on the same aisle. Given this assumption, there are eight different *U*-shaped flow patterns which are shown in Figure 2.4. Linear flow depicts travel from one midpoint of a department to the opposite midpoint of the same department, which is typical of material that travels from one side of a department to its opposite side. There are two possible linear orientations. *C*-shaped flow specifies travel that enters at one of eight locations along a departmental perimeter, travels in a circular fashion within the department, and exits at the original point of entry. Theoretically, the possible I/O locations for a *C*-shaped flow pattern could reside anywhere on a departmental perimeter. Referring to the arguments of [2], it can be shown that a discrete set of locations constitutes a dominant set for *C*-shaped flow. This set includes the four corners of a given department and all locations where the given department intersects either the corner or midpoint of one side of any other department.

Although these flow types do not account for every possible path of material travel within a department, most forms of travel could be designated as one of the three. By assuming that intra-departmental flow types are known a prioi, the consideration of locating an I/O point anywhere on a departmental perimeter is not necessary. If the possible locations of I/O points were not limited, the CPU requirements for locating I/O points would be too overwhelming to consider in combination with locating and shaping departments.

4.3.3 Construction and Improvement of the I/O Network

In each generation of the block layout GA, every member of the population undergoes the detailed design optimization of I/O location. This entails the selection of a single input point and a single output point for each department. The candidate I/O points for each department are defined by the intra-departmental flow types that are assigned to each department. The process of forming a material flow network begins by calculating the coordinates of all candidate I/O points. Using the contour distance metric and the Floyd-Warshall all pairs shortest path algorithm, the shortest path network of all candidate I/O points is determined. The pseudo-code for the Floyd-Warshall algorithm is shown in Figure 4.5.

---

n = number of departments
$D^{(0)} = W$
for k = 1 to n
   for i = 1 to n
      for j = 1 to n
$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$
return $D^{(n)}$

Figure 4.5 Floyd-Warshall All Pairs Shortest Path Algorithm

The Floyd-Warshall algorithm used here produces the shortest contour paths from every department's set of output points to every other department's set of input points. Thus, when the input station and output station for all departments are chosen, the shortest contour paths between each pair of departments are already known. By explicitly constraining the number of candidate I/O stations per department and the location of each candidate I/O point, the detailed design problem of I/O location is considerably more difficult to solve than the shortest path problem since the evaluation routine must solve network problems of greater complexity.

Knowing the shortest paths between all I/O pairs, the definitive material flow network for a layout begins to take form. For each block layout in the population, a sub-population of material flow networks is created. Each member of the sub-population is represented by an I/O chromosome. To create an initial I/O sub-population for a block layout, I/O points are chosen at random from each department's feasible set. Once an initial I/O network has been identified for each member of the sub-population, the sub-population is evaluated for fitness and ranked. To avoid expending CPU time on non-promising layouts, a restriction to the size of the sub-population of I/O chromosomes is made for every layout that undergoes I/O placement. That is, if a layout contains two or more infeasible departments, the I/O sub-population size is restricted to one. Otherwise, a sub-population of ten I/O chromosomes is used.

Once the initial sub-population of I/O networks have been formed and evaluated for fitness, the detailed design heuristic moves to improve each member of the I/O sub-population. A perturbation movement mechanism deterministically improves each material flow network in the sub-population. For each department, every candidate I/O pair is evaluated one at a time to determine which I/O pair results in the minimum flow cost for a layout. As the I/O pairs of the department under consideration are being evaluated, the I/O stations of all other departments are fixed. When all candidate I/O pairs for a department have been evaluated, the pair resulting in the minimum total flow cost for a layout is chosen for that department. The mechanism then moves to the next department in the departmental sequence and performs that same perturbation and evaluation process. Figure 4.6 provides an illustration of the perturbation mechanism. Department 2 has a pre-assigned U-Type flow pattern, which corresponds to eight candidate I/O pairs or intra-departmental flow orientations. Figure 4.6 displays four of the candidate I/O pairs for department 2. The other four candidate I/O pairs are simply the reverse of the intra-departmental flow direction for department 2 in each of the four layouts shown. The perturbation mechanism evaluates each of these orientations and their resulting I/O locations. From left to right and top to bottom, Figure 4.6 depicts the perturbation process for department 2 as it moves from one I/O pair to another while keeping the rest of the I/O network fixed. The I/O pair resulting in the lowest total flow cost in the I/O sub-population is selected as the I/O pair for department 2. The process then moves to department 4 and the same process of evaluation and selection takes place. Once every department in the departmental sequence has been evaluated, the corresponding layout is kept for comparison to all other members of the I/O sub-population. Once each member of the I/O sub-population has been evaluated by the perturbation routine, the best individual from the I/O sub-population is retained along with the initial block layout under consideration.

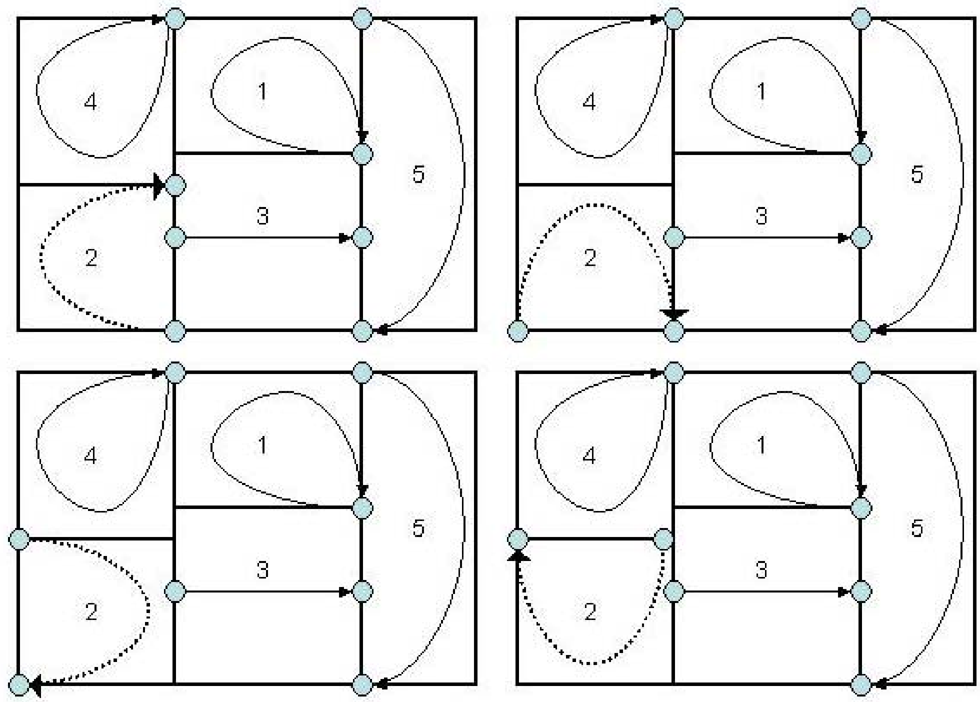

Figure 4.6 Illustration of Perturbation Mechanism

5.0 EXPERIMENTAL DESIGN

This chapter describes the experimental design.

5.1 PROBLEM SELECTION

The suite of test problems used as input for the integrated, evolutionary approach consists of four well known FLPs from the literature [4], [6], [37]. The problems herein are referred to as Armour and Buffa (AB) [4], Bazaraa12 (B12) [6], Bazaraa14 (B14) [6], and van Camp (VC) [37]. These problems were selected because they are traditionally well studied in facility layout research and afford the opportunity for comparison. Using these problems as a basis for comparison is particularly important for the integrated, evolutionary approach. The stochastic and heuristic nature of the methodology employed in this thesis necessitates the inclusion of commonly studied FLPs in order to effectively demonstrate the advantages and relevance of this thesis. If strictly exact methods of optimization had been used, demonstration of superiority and justification of the methodology could be shown without the need for comparison.

As mentioned, the four problems were used as input to the integrated, evolutionary code. Each problem consists of a predefined number of departments, the rectangular dimensions of the facility, the area of each department, and the matrix of inter-departmental flow values between all departments. All of this information is provided in Appendix A. Tables A1-A4 show the flow matrices for each of the four problems. Since each department is assumed to have no product flow interaction with itself, the diagonal elements of each flow matrix are equal to zero. Table A5 displays the number of departments, the heights, and the widths of layouts for each of the four problems. Table A6 details the areas and the minimum side requirements for all departments in each of the four problems. All problems adhere to Equation (3-2) in that the sum of departmental areas is equivalent to the overall area of a layout.

The details of each problem and the adjustments required to apply the integrated, evolutionary approach from each initial problem form are now discussed. For the Armour and Buffa problem, each department experiences product flow interaction with a subset of the other departments. The data for this problem is unchanged from its initial form with exception to the inclusion of minimum side requirements for each department. The input data for the two instances of Bazaraa problems were adjusted as described in [35]. Both instances originally had areas within a facility that were unassigned to departments. The intention was to depict an operating environment that did not utilize the entire space of a facility. This is a useful consideration since many facilities are designed to accommodate future growth and production. The first problem instance, B14, had a notch in the lower right hand corner that was unavailable for assignment. To accommodate usage of the problem with the integrated, evolutionary approach, the notch is modeled as one department with an area of two, no interactions, and unrestricted shape. The second problem instance, B12, required the adjustment that department 14 be unrestricted in shape. Both Bazaraa problems were given the additional restrictions of minimum side requirements, although the original formulation inherently assumed minimum sides. The van Camp problem was taken without change from its original form.

5.2 DETAILED DESIGN SPECIFICATION

With the four FLPs described in the previous section, the input data pertaining to block layout aspects of facility design was defined. The aspects of detailed layout design were determined by consideration and analysis of intra-departmental flow types. The three flow types, *U*-shaped (*U*), linear (*L*), and *C*-shaped ©, considered here are intended to provide an analyst with the ability to accurately depict the flow of operations within departments. An analyst can generally classify any department's product flow as *U*, *L*, or *C*. Intra-departmental flow patterns are largely dependent on the nature of the operations under consideration. The integrated, evolutionary approach is made applicable to a wide array of industries by developing six sets of intra-departmental flow patterns. Three of the sets classify all departments as having all *U*, all *L*, or all *C* flow types. The other three sets assign a mixture of flow patterns to departments. These are labeled as mixture 1 (*M1*), mixture 2 (*M2*), and mixture 3 (*M3*).

Since the four problem instances for this thesis have different numbers of departments, the three mixtures are not identical across problem instances. However, an effort was made to balance each mixture across the four problems. This was done by analyzing the amount of flow interaction between each set of flow types. There are nine possible interactions with regard to product flow between two departments with predefined intra-departmental flow. The nine occurrences are shown below.

U to U	*L to U*	*C to U*
U to L	*L to L*	*C to L*
U to C	*L to C*	*C to C*

It was determined that the three mixtures for each problem instance would approximately involve *33%*, *33%*, and *33%* of flow types *U*, *L* and *C* respectively for *M1*. For *M2* the mixture would resemble *50%*, *25%*, and *25%* of flow types *U*, *L* and *C* respectively. *M3* would have a balance of *25%*, *25%*, and *50%* respectively.

Balancing the percentages for each mixture does not ensure that the product flow between all sets of departments would assume a similar balance. This is due to the variability of product flow values in each problem's flow matrix. Analysis was done to achieve the percentage distributions of flow mixtures mentioned above. First, the number of departments that would assume each type of flow was defined. Then, a random assignment of the designated number of flow types to the set of departments for each problem was made. With regard to the nine intra-departmental flow interactions mentioned above, the total flow value for all nine types of interactions for each department was calculated. The total flow for each interaction over all departments was then calculated. These totals were then grouped and totaled according to the following logic. All flow values involved with *U*-flow, either from a department or to a department, were summed. The same was done for all flows involving *L*-flow and *C*-flow. Dividing the sum of each set by the sum of flow for the entire facility provided the actual percentage of flow either from or to departments of flow type *U*, *L*, or *C*. If the actual percentages were not close enough to the desired percentages, flow types were randomly exchanged between departments until an acceptable balance was reached. The actual balance of flow patterns for each of the three mixtures for each of the four problems is shown in Table A7.

5.3 TEST PLAN

The specification of the test plan for the integrated, evolutionary approach was done by considering the problems selected, the mixture of flows assigned, and the genetic parameters. The four problems were assigned to six flow mixtures at three different aspect ratios for ten random seeds. This created a test matrix of 720 simulations (4 x 6 x 3 x 10 = 720).

In order to demonstrate superiority of the integrated, evolutionary approach with prior methodologies, a comparison of the results of this thesis with the best known solutions of prior research was necessary. Three publications were chosen as the basis for comparison [3], [27], [35]. The selection of these prior research methodologies and the aim of the comparison were intended to show that solving the unequal-area FLP by integrating the optimization of block layout and detailed design was advantageous to solving each separately or sequentially. The results in the following section will validate this proposition. The results of prior research could not be directly evaluated versus the results of the integrated, evolutionary approach due to differences in the problem specifications and/or the methodologies employed. In order to fully achieve a relevant and fair comparison, additional test plans were needed to convert the results of prior research to the solution form of the integrated, evolutionary approach.

Seven layout solutions were chosen to be converted for comparison. The layouts were chosen to represent each of the four problems for some of the three aspect ratios. Since the layout solutions from each paper could not be attained in exact form, the encodings for each were used. That is, the exact dimensions of height and width for each department in a solution were not provided. However, the solution encodings provide enough information to assure that the block layouts produced in the comparison are identical to those shown in each publication. The use of the Flexbay structure ensures this.

The layout solutions for each problem inclusive of the departmental sequence, bay structure, and aspect ratio are provided in Table B1 of Appendix B. Encodings 1, 2, and 4 originate from [35]. These solutions were originally obtained through genetic search with no specification of I/O location. They were optimized based on centroid locations and the Flexbay structure. To permit comparison of these results to the results of this thesis, the block layout solutions for encodings 1, 2, and 4 were not changed. The six variants of flow assignments were assigned to each fixed block layout and only detailed design optimization was performed. Encodings 5 and 7 come from [3]. These results were originally obtained by taking previously optimized block layouts and performing I/O location optimization. However, intra-departmental flow was not included. Thus, the same procedure that was implemented for encodings 1, 2, and 4 was used again. Encoding 3 is from [27]. It was originally optimized for both block layout and I/O placement using the contour measure. The I/O placement for this problem was unconstrained in that departments could assume multiple I/Os.

None of the 720 problems from the test plan for the integrated, evolutionary approach were solved by exact methods of computation that could indicate an optimal solution. Thus, it is not possible to compare the results of this research to any known optima. In the interest of strengthening the results of this thesis, the best solution out of the ten seeds from all problem instances were supplied to the detailed design heuristic to try and further improve the results. Extensively performing the I/O placement heuristic on these fixed block layouts ensures that the optimal or near optimal solution has been found.

6.0 EXPERIMENTAL RESULTS

This chapter presents the results of this thesis.

6.1 INTEGRATED, EVOLUTIONARY RESULTS

A summary of the results for the 720 tests that were run using the integrated, evolutionary code are given in Appendix B. Amour and Buffa is shown in Table B1, Bazaraa12 in Table B2, Bazaraa14 in Table B3, and vanCamp in Table B4. Each table includes the flow type assignments and the aspect ratios that were used. Also indicated are the minimum basic feasible solution, the maximum basic feasible solution, the average basic feasible solution, the standard deviation, and the average generation of convergence over ten random seeds.

As mentioned earlier, GAs are stochastic search programs. Part of a GA's value is the measure of variability of the solutions it produces through varying random inputs. To gauge the variability of the GA implemented in this thesis, the standard deviation of each set of ten random solutions from each of the four problem instances was calculated. These values are shown in Tables B1-B4 along with the average standard deviation for each of the four problems. The algorithm performed somewhat consistently over all solutions for each problem instance. Increasing the number of random seeds for each problem instance would likely not help to reduce this variability. Even if the number of random seeds were increased and a better minimum was obtained as a result, the difference between the best minimum solutions over the smaller set and the larger set would most likely be marginal and not worth the CPU expense.

Another measure of a GA's robustness is its ability to converge to quality solutions without prematurely converging early in the search. It is also desirable that a GA has the ability to vary its direction of search in order to find better solutions. The average generation of convergence for each of the four problem instances is shown in Tables B1-B4. Each problem had an average below *50,000* with *34,000* (AB), *41,000* (B12), *34,000* (B14), and *29,000* (VC) average generations to convergence. Each test was run for *100,000* generations with the exception of AB, which ran for *75,000*. The averages indicate that the number of generations may have been set a little too high. However, *4* out of the *18* minimum solutions for the AB problem did not converge until after *55,000* generations and *3* out of the *18* minimum solutions for the B12 problem did not converge until after *80,000* generations. Conversely, *7* (AB), *5* (B12), *8* (B14), and *15* (VC) minimum solutions out of *18* each converged at or before *20,000* generations. It is reasonable to infer from these results that setting the generation limit for the integrated GA should be a function of the problem being solved. Factors such as cost, importance, and time should be considered when setting this parameter.

Another benchmark of a GA that relates to setting the generation limit is the CPU expense it requires. Table B5 provides the average running times over ten seeds for all problems. All problems were run on a Pentium (R) 4 CPU 3.00 GHz processor, with the exception of the B12 problems, which were run on a Pentium (R) 4 CPU 2.00 GHz processor. The average running times (in hours) for each of the four problem instances were *11.62* (AB), *7.99* (B12), *4.33* (B14), and *1.86* (VC). Due to the nature of the problem being solved, large running times were expected at the onset of this thesis. However, these times are quite high and indicate that the integrated approach can be improved from an efficiency viewpoint.

Appendix C contains eleven selected layouts from the integrated results. Figures C3 – C5 are layouts from the AB problem with aspect ratios of *3*, *5*, and *7* respectively. Figures C6 – C7 are layouts from the B12 problem with aspect ratios of *3* and *5* respectively. Figures C8 – C10 are layouts from the B14 problem with aspect ratios of *3*, *5*, and *7* respectively. Figures C11 – C13 are layouts from the VC problem with aspect ratios of *3*, *5*, and *7* respectively. Each layout represents the best solution for a particular problem instance. The intra-departmental flow types for each department are indicated next to each department's number. For layouts in which all departments have the same intra-departmental flow, the flow type is indicated only in the layout's caption, as in Figure C3. Each layout depicts the location of each department's I/O pair. The small arrows pointing to I/O points within each department indicate which I/O point belongs to which department.

6.2 COMPARISONS TO PRIOR RESULTS

The process of converting the best known results of prior research to allow for direct comparison with the results of this thesis was discussed earlier. Figure B1 provides the solution encodings that were converted by fixing the department structure, assigning flow types, and performing I/O placement. Table B6 contains the results of encodings #*5* and #*6* for the B14 problem. The best known solution is the value of the layout as reported in the publication from which each encoding originates. There is no practical interpretation for comparing these values to the results of this thesis. The solutions with flows applied are the results of the conversion process. These results are comparable to the integrated, evolutionary results, which are also shown in each table. The percentage change indicates the difference between the converted solution and the integrated, evolutionary results. Encoding #6 comes from [35] and was originally optimized by genetic search for the block layout with no I/O placement. Distances between departments were calculated by the rectilinear metric using departmental centroids as locators. After applying the six flow types and performing I/O placement, the integrated, evolutionary results are between 26% and 53% better in terms of total material handling costs for each problem. This does not indicate deficiency in the methodology of the prior research. It does indicate that performing the block layout and the I/O placement simultaneously results in a block layout that is more suited for I/O placement with the contour measure. Encoding #*5* for the B14 problem shows the same behavior as encoding #*6* with improvements ranging from *6.09%* to *22.61%*. Encoding #*4* from the B12 problem originates from [35] as well and showed a range of improvement from *5.81%* - *47.78%*.

Table B7 displays the comparison results from the AB encodings #*1*, #*2*, and #*3*, which originate from [35], [35], and [27] respectively. The original method of solution for #1 and #2 were discussed in the previous paragraph. Encoding #3 was originally solved through a GA using a contour distance metric. It concurrently optimized the block layout and I/O placement, but the I/O placement did not include intra-departmental flow types and there was no restriction as to the number of I/Os for each a department. In all but one case the integrated, evolutionary methodology outperformed the results of the AB encodings with flows applied. The improvements range from 2% to 70%, again confirming the usefulness of the proposed integrated methodology. An interesting observation among the comparisons for the three AB encodings is that the largest improvements occur for #3, which had originally been simultaneously optimized for block layout and I/O placement as versus encodings #1 and #2. The I/Os per department were unconstrained however, and the block layouts were formed on the assumption that any number of I/Os could be placed anywhere on a department perimeter. Thus, I/Os were originally located closest to their destination I/Os and vice versa. When the resulting layout is fixed and constrained to the single I/O problem, it experienced a poorer flow total than the integrated, evolutionary approach. Thus highlighting the necessity to constrain I/Os both for practical reasons (i.e. cost, physical obstruction) and performance of the layout. Encodings #7 for the vanCamp problem comes from [27] and demonstrated a range of improvement of *29.34% - 75.75%*.

7.0 CONCLUSION

This chapter discusses the successes and the difficulties of this thesis and suggests possible extensions.

7.1 IMPROVEMENTS TO FACILITY LAYOUT AND DETAILED DESIGN

The stated purpose of this thesis was to design and implement an integrated methodology for solving the unequal-area FLP and the detailed design I/O placement problem simultaneously, and this was accomplished. The Flexbay methodology for reducing the unequal-area block layout problem was utilized in a GA framework in conjunction with a detailed design constructive heuristic in a computationally feasible manner. A thorough test plan was designed and implemented to indicate the practicality and usefulness of the integrated methodology, both in terms of solution quality and CPU expense. Comparison of the integrated results to the best results of prior research was made possible through conversion of previously optimized results.

The intra-departmental flow types that depict material movement within departments further the definition of a layout solution. Accounting for the movement of material within each department helps to avoid I/O location problems. If the intra-departmental flows of each department were not considered before optimization, an analyst would be required to orient the movement of material within each department after a layout solution has been obtained. In such a case, an I/O station could be located in a detrimental position and require adjustment. Adjusting an I/O station once a layout solution is obtained could result in an inefficient layout structure. Thus, the consideration of intra-departmental flow types before performing optimization is an important aspect of this thesis.

A fundamental accomplishment of this thesis is the combination of block layout and detailed design methodologies into a single algorithmic design. The value of this integrated methodology was demonstrated through comparisons to prior research. As discussed in Section 6, performing the I/O location problem after a block layout has been optimized and fixed was shown not to perform as well as when performing both tasks cohesively. This is the most important achievement of this thesis. Although superiority of the integrated approach over sequential approaches was demonstrated, the integrated methodology did not perform particularly well in terms of variability and CPU expense. The variability of solutions was quite high for each of the four problems. Since the integrated approach is a heuristic methodology, variability in the solutions was expected. Moreover, GA implementations for combinatorial optimization problems are commonly found to exhibit such variability.

The CPU times for each set of problems are shown in Table B6. The average CPU time in hours for each of the four problems are approximately *12*, *8*, *4*, and *2* for problems of size $n = 20$, *16*, *14*, *10* respectively. These times are somewhat discouraging considering that heuristic procedures were used in place of exact methods in order to reduce large CPU overhead. This is a drawback when considering solving larger problems with the integrated methodology. However, an organization that wishes to optimize a facility layout would likely be willing to sacrifice the CPU cost of the integrated approach since only one layout, or perhaps a few layouts, needs to be optimized. Furthermore, redesigning a facility that was not designed correctly to begin with is much more expensive than allotting CPU time to solve the problem satisfactorily from the beginning.

An argument could be made that using a heuristic approach for this problem is no longer necessary. Recent research has shown that MIP methodologies are becoming more robust for the unequal-area FLP. However, the problems being solved are still small in size ($n \leq 15$). The integrated approach had an average CPU time of nearly a half a day for a problem of size $n = 20$. Thus, problems that are very large in size ($n \geq 50$) will continue to require the use of heuristic techniques for the unequal-area FLP. In addition, heuristic methodologies could be even more effective in solving the unequal-area FLP through the use of a supercomputer.

7.2 FUTURE CONSIDERATIONS

The objective function for the thesis here, as indicated by Equation (2-5), could readily be adjusted to include variable material handling costs. For example, the movement of material between a specific set of departments may require the use of expensive machinery, either in terms of the machinery itself or the labor required to operate it. In such an instance, the amount of product flow and distance between the departments would not be the only factors influencing their placement and specification. Such consideration was made by Armour and Buffa [4] without the addition of a variable to the objective function. The authors specified a material handling cost matrix and combined it with the product flow matrix, maintaining one cost variable in the objective function. Adding a second variable would be more desirable since modes of transportation frequently change and the resulting costs should be taken into consideration.

A more pressing consideration might be the cost of additional I/O points. This thesis assumes a single input and a single output point for all departments. This assumption was made for practical reasons. However, this restriction might not always be practical. Consider a flexible job shop environment in which numerous subcomponents are traveling to a single department. If the subcomponents are traveling to the destination department from different directions, it may be possible and advantageous to provide that department with more than one input point in order to further minimize material handling costs, especially when the material being transported is either large in quantity or expensive to move. Adding I/O points would require relaxing the single I/O restriction and consideration of the cost for additional I/Os. This would entail significant thought by the analyst in determining costs for specific I/O locations. This could be very time consuming and difficult since I/O costs will change as a function of the layout specification.

The unequal-area FLP is relevant to many industries. In the context of manufacturing, three-dimensional, or multi-floor, operational environments are becoming more prevalent. With product flow extending beyond two dimensions to three dimensions, the unequal-area FLP becomes even more difficult to solve. There has been much research done for three-dimensional VLSI applications in chip design. Such research is concerned with minimizing stacks of electrical networks that are vertically connected. Consideration of the block layout features discussed in this thesis in addition to path structure is an intriguing research direction. Such a problem would certainly require the use of heuristic procedures, such as the GA framework used in this thesis.

This thesis attempts to minimize a flow network, which is beneficial in reducing material handling costs. The aim could be changed to maximize a flow network, which could be useful to the retail industry. In the interest of keeping customers within stores for as much time as possible, a longest path network would be the objective. Simply modifying the objective function of this thesis to a maximization problem would not suffice. Constraints such as safety requirements (i.e. closeness to fire exits) and inventory replenishment (i.e. stocking shelves) would need to be considered.

Enhancing the solution quality of the integrated, evolutionary approach could be realized by replacing the I/O heuristic by the I/O GA of [29]. As computers become faster, the possibility of integrating both a block layout GA and an I/O GA becomes much more feasible. Since GAs have proven to produce results that are consistently optimal or near optimal for separately optimizing a block layout [35] and an aisle structure [29], integrating GAs for each optimization task presents an intriguing research direction for the unequal-area FLP. Furthermore, massively parallel hardware could be used to further accommodate the complexity of such an approach.

As discussed in the Section 2.6, the consideration of uncertainty with regard to product demand has not been given much attention in the course of facility layout research. Moreover, the ability to effectively solve FLPs with the inclusion of uncertainty has generally been limited. This is largely due to the complexity of such problems and to the inability to effectively predict product demand. However, the need for manufacturing organizations to consider variability in product demand and the introduction of new product lines is becoming vitally important. This is especially apparent in the United States, where many manufacturing environments rely on their ability to be operationally flexible in order to compete globally. The genetic random keys approach of Norman and Smith [27] could be incorporated with the integrated, evolutionary approach to produce detailed design layouts that account for demand volatility.

Another aspect of uncertainty that could be accounted for in the context of unequal-area FLP is intra-departmental flow patterns. Modeling a problem in which the flow types of some departments are not known might be beneficial for a couple of reasons. Given that upgrades in technology frequently occur in manufacturing environments, the flow type of a department could change many times in the course of a decade. It would be worthwhile to consider multiple configurations of intra-departmental flow if an analyst has reason to believe that a certain department will undergo such changes in the near future. Furthermore, catastrophic increases to the overall cost of material flow in a facility could result from modifying a department's inner flow pattern if that department had a substantial amount of interaction with other departments and provisions had not been made for such changes.

The Flexbay formulation used in this thesis aided in reducing the complexity of the block layout problem. Its use was justified by the argument that facilities are frequently constructed with aisles that extend from one side of a facility to the other. Since this assumption is not always satisfied, it would be useful to eliminate or modify the Flexbay formulation so that departmental shapes are not as restricted.

Figures C1 and C2 in Appendix C depict combinations of the contour distance metric with the Euclidean metric and the rectilinear metric. The contour measure used in this thesis does not account for flow distances within departments. By combining the contour measure with either the Euclidean or rectilinear metric, an even more practical layout design and interpretation could be gained. Although, the intra-departmental flow patterns that depict flow orientations within departments would need to be considered in regards to such a change to the distance metric.

Another possible upgrade to the integrated, evolutionary GA is improving its efficiency. Research has demonstrated that the encoding structure of any GA plays a crucial role in both solution quality and computational efficiency. It is possible that the encoding used in this thesis could be improved on both accounts.

APPENDIX A

PROBLEM DATA

Table A1 Armour and Buffa Flow Matrix

Armour and Buffa																				
Dept	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	-	1.8	1.2	0	0	0	0	0	0	1.04	1.12	0	0	1.2	0	0	0	0	0	0
2		-	0.96	24.45	0.78	0	13.95	0	1.2	1.35	0	0	0	0	0	0	0	0	6.9	0
3			-	0	0	0.21	0	0	3.15	3.9	0	0	0	13.05	0	0	0	0	13.65	0
4				-	1.08	5.7	7.5	0	2.34	0	0	1.4	0	0	0	0	0	1.5	15.75	0
5					-	0	2.25	1.35	0	1.56	0	0	0	0	1.35	0	0	0	0	0
6						-	6.15	0	0	0	0	0.45	0	0	0	0	0	1.05	0	0
7							-	24	0	1.87	0	0	0	0.96	0	0	0	1.65	0	3.75
8								-	0	0	0	0	0.6	0	0	0	0	0	7.5	33.45
9									-	0	0	0	0	7.5	0	0	7.5	0	0	0
10										-	0.36	12	0	18.6	1.92	0	0	0	5.25	0
11											-	2.25	0	3	0.96	22.5	0	0	0	0
12												-	0	0	1.65	0	15	0	8.4	0
13													-	8	1.04	6	0	0	0	0
14														-	9.75	0	0	0.9	0	0
15															-	0	5.25	0	0	0
16																-	12	0	0	0
17																	-	0	7.5	0
18																		-	4.65	0
19																			-	0
20																				-

Table A2 Bazaraa12 Flow Matrix

Bazaraa12																
Dept	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
1	-	288	180	54	72	180	27	72	36	0	0	9	0	0	0	0
2		-	240	54	72	24	48	160	16	64	8	16	0	0	0	0
3			-	120	80	0	60	120	60	0	0	30	0	0	0	0
4				-	72	18	18	48	24	48	12	0	0	0	0	0
5					-	12	12	64	16	16	4	8	0	0	0	0
6						-	18	24	6	12	3	3	0	0	0	0
7							-	0	6	6	3	6	0	0	0	0
8								-	16	16	16	4	0	0	0	0
9									-	4	4	2	0	0	0	0
10										-	2	2	0	0	0	0
11											-	2	0	0	0	0
12												-	0	0	0	0
13													-	0	0	0
14														-	0	0
15															-	0
16																-

Table A3 Bazaraa14 Flow Matrix

Bazaraa14														
Dept	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	-	72	162	90	108	27	0	0	18	27	18	0	0	0
2		-	72	80	0	48	0	48	32	0	16	8	0	0
3			-	45	54	27	27	27	0	27	0	9	18	0
4				-	30	0	30	30	20	0	20	10	10	0
5					-	18	0	18	12	18	24	0	0	0
6						-	9	9	0	0	6	6	6	0
7							-	9	12	9	6	3	0	0
8								-	6	9	0	3	0	0
9									-	6	4	6	2	0
10										-	6	3	6	0
11											-	2	0	0
12												-	4	0
13													-	0
14														-

Table A4 van Camp Flow Matrix

van Camp										
Dept	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	-	0	0	0	0	218	0	0	0	0
2		-	0	0	0	148	0	0	296	0
3			-	28	70	0	0	0	0	0
4				-	0	28	70	140	0	0
5					-	0	0	210	0	0
6						-	0	0	0	0
7							-	0	0	28
8								-	0	888
9									-	59.2
10										-

Table A5 Layout Parameters I

Problem	# of Departments	Height of Layout	Width of Layout
Armour and Buffa	20	2	3
Bazaraa12	16	6	10
Bazaraa14	14	7	9
van Camp	10	25	51

Table A6 Layout Parameters II

Dept	Armour and Buffa		Bazaraa12		Bazaraa14		van Camp	
	Area	Min Side	Area	Min Side	Area	Min Side	Area	Min Side
1	0.27	0.15	9	1	9	1	238	5
2	0.18	0.15	8	1	8	1	112	5
3	0.27	0.15	10	1	9	1	160	5
4	0.18	0.15	6	1	10	1	80	5
5	0.18	0.15	4	1	6	1	120	5
6	0.18	0.15	3	1	3	1	80	5
7	0.09	0.15	3	1	3	1	60	5
8	0.09	0.15	4	1	3	1	85	5
9	0.09	0.15	2	1	2	1	221	5
10	0.24	0.15	2	1	3	1	119	5
11	0.6	0.25	2	1	2	1		
12	0.42	0.2	2	1	1	1		
13	0.18	0.15	2	1	1	1		
14	0.24	0.15	1	1	3	0		
15	0.27	0.15	1	0				
16	0.75	0.3	1	0				
17	0.64	0.25						
18	0.41	0.2						
19	0.27	0.15						
20	0.45	0.2						

Table A7 Mixed Flow Analysis

Problem	Number of Departments Assigned to Each Flow Type			Percentage of Departments Assigned to Each Flow Type			Percentage of Material Transitions From or To Each Flow Type		
	U	L	C	U	L	C	U	L	C
AB M1	8	4	8	40%	20%	40%	41%	16%	44%
AB M2	10	5	5	50%	25%	25%	53%	19%	28%
AB M3	5	5	10	25%	25%	50%	32%	24%	44%
B12 M1	5	5	6	31%	31%	38%	41%	28%	30%
B12 M2	8	4	4	50%	25%	25%	51%	29%	19%
B12 M3	4	4	8	25%	25%	50%	28%	27%	45%
B14 M1	5	6	3	36%	43%	21%	24%	48%	28%
B14 M2	7	3	4	50%	21%	29%	48%	30%	22%
B14 M3	3	4	7	21%	29%	50%	19%	25%	57%
VC M1	4	4	2	40%	40%	20%	50%	38%	12%
VC M2	5	2	3	50%	20%	30%	63%	10%	27%
VC M3	2	3	5	20%	30%	50%	13%	21%	66%

APPENDIX B

RESULTS AND COMPARISONS

Table B1 Integrated, Evolutionary Results for Armour and Buffa

Armour and Buffa						
Flow Type	**Aspect Ratio**	**Min BFS**	**Max BFS**	**Avg BFS**	**Avg Std Dev**	**Avg Generation of Convergence**
U	3	407.10	529.18	464.22	38.71	42
L	3	670.56	802.24	743.80	37.31	29
C	3	278.86	355.30	316.07	25.01	34
M1	3	434.54	522.71	487.13	31.19	26
M2	3	416.08	552.22	501.70	41.12	33
M3	3	387.35	525.91	438.88	40.82	33
U	5	389.36	460.94	424.53	22.19	33
L	5	634.89	782.55	698.95	50.56	29
C	5	260.00	325.84	296.79	24.01	34
M1	5	404.95	484.86	443.10	22.73	32
M2	5	425.06	477.75	457.69	20.94	30
M3	5	391.14	488.09	436.64	38.01	32
U	7	318.86	436.71	374.56	38.75	40
L	7	590.88	765.39	668.44	52.70	33
C	7	247.94	318.99	278.20	23.63	48
M1	7	399.83	463.33	429.00	21.16	29
M2	7	423.24	524.44	457.61	29.35	41
M3	7	364.04	471.96	407.50	34.56	45
Averages		413.59	516.02	462.49	32.93	34

Table B2 Integrated, Evolutionary Results for Bazaraa12

Bazaraa12						
Flow Type	**Aspect Ratio**	**Min BFS**	**Max BFS**	**Avg BFS**	**Avg Std Dev**	**Avg Generation of Convergence**
U	3	4564.37	5304.04	4855.03	275.12	15
L	3	8585.57	9348.88	8782.99	225.13	55
C	3	4032.62	4775.76	4242.48	271.53	61
M1	3	5280.62	6133.40	5520.99	313.83	48
M2	3	5836.88	6758.24	6188.41	343.75	52
M3	3	5246.97	6878.22	5736.98	468.10	33
U	5	4230.48	5009.01	4487.24	256.06	38
L	5	7806.45	8499.71	7980.73	218.35	47
C	5	3763.23	4362.06	4013.18	204.30	40
M1	5	4977.86	6239.55	5656.87	400.29	37
M2	5	5311.64	6272.59	5829.90	260.72	53
M3	5	4427.55	6010.79	5154.45	549.33	42
U	7	2911.73	3723.68	3104.33	310.06	38
L	7	7709.86	8318.90	7981.72	209.63	30
C	7	2851.45	8318.90	5063.80	2561.04	35
M1	7	4783.87	5079.56	4954.90	91.17	37
M2	7	4785.61	5953.01	5278.56	375.02	38
M3	7	3555.67	3911.03	3711.90	1795.89	43
Averages		5036.80	6160.96	5474.69	507.18	41

Table B3 Integrated, Evolutionary Results for Bazaraa14

Bazaraa14						
Flow Type	**Aspect Ratio**	**Min BFS**	**Max BFS**	**Avg BFS**	**Avg Std Dev**	**Avg Generation of Convergence**
U	3	2604.17	3439.85	2948.46	272.03	24
L	3	4732.08	5259.91	4959.89	231.03	40
C	3	2392.67	2981.86	2642.61	196.65	34
M1	3	3964.43	4585.41	4156.89	229.13	34
M2	3	3276.70	3641.99	3467.30	113.81	57
M3	3	3058.32	3671.24	3253.69	241.52	46
U	5	2548.43	3081.63	2747.78	167.98	30
L	5	4735.68	5496.38	4992.94	296.25	40
C	5	2345.93	2671.93	2486.86	109.41	29
M1	5	3784.88	4386.93	4040.93	161.62	25
M2	5	2782.37	3317.89	3054.64	144.48	22
M3	5	2973.76	3248.32	3084.53	95.20	31
U	7	2222.81	3081.63	2584.49	277.39	40
L	7	4488.13	5496.38	4841.88	400.28	28
C	7	2002.10	2671.93	2324.95	244.91	29
M1	7	3769.95	4320.06	3910.13	168.01	53
M2	7	2590.10	3031.15	2801.96	138.13	28
M3	7	2590.10	3029.18	2909.26	123.36	34
Averages		3159.03	3745.20	3400.51	200.62	34

Table B4 Integrated, Evolutionary Results for van Camp

van Camp						
Flow Type	**Aspect Ratio**	**Min BFS**	**Max BFS**	**Avg BFS**	**Avg Std Dev**	**Avg Generation of Convergence**
U	3	2809.22	3382.70	3006.48	229.01	24
L	3	8586.12	8641.05	8591.61	17.37	7
C	3	3188.12	3778.82	3515.40	184.57	26
M1	3	5585.50	7349.46	6229.30	446.42	32
M2	3	4569.17	5256.30	4769.62	220.79	21
M3	3	5527.84	5855.69	5586.14	111.98	33
U	5	2134.02	3599.43	2542.37	527.42	41
L	5	7358.07	8612.13	8201.07	457.06	27
C	5	2963.66	3601.48	3055.15	210.79	39
M1	5	4417.29	6731.06	5152.23	839.57	42
M2	5	3005.52	3894.11	3450.69	364.89	17
M3	5	3839.20	3839.20	3839.20	0.00	20
U	7	1827.75	2956.37	2280.18	395.76	59
L	7	7358.07	8248.24	7635.36	242.66	37
C	7	2963.66	3471.30	3115.94	245.20	28
M1	7	4417.29	6362.60	5108.13	860.83	26
M2	7	3005.52	4569.65	3432.89	532.81	20
M3	7	3839.20	4015.44	3868.47	63.30	27
Averages		4299.73	5231.39	4632.24	330.58	29

Table B5 CPU Analysis

Flow Type	Aspect Ratio	Average Running Times (hours)			
		Armour and Buffa	Bazaraa12	Bazaraa14	vanCamp
U	3	8.94	7.20	3.74	1.63
L	3	5.50	4.17	2.27	1.21
C	3	11.81	8.77	4.43	2.31
M1	3	9.90	7.00	2.80	1.56
M2	3	8.58	6.76	3.65	1.85
M3	3	10.38	7.37	3.67	1.69
U	5	14.96	10.16	4.69	2.05
L	5	5.85	5.01	2.71	1.30
C	5	16.73	12.02	6.08	2.45
M1	5	10.46	9.21	3.73	1.61
M2	5	10.05	8.90	4.36	2.02
M3	5	11.62	9.28	4.30	1.78
U	7	18.61	10.47	5.52	2.11
L	7	6.35	5.48	3.01	1.42
C	7	20.49	12.50	7.15	2.69
M1	7	12.40	9.88	4.39	1.78
M2	7	12.64	10.04	5.54	2.05
M3	7	13.90	10.01	5.92	1.88
Averages		11.62	7.99	4.33	1.86

Problem	Encoding #	Aspect Ratio	Block Layout Encodings	
AB	1	3	1 11 16 15 13 17 14 10 9 12 3 19 4 2 5 6 8 7 18 20	16 17 12 2 7
AB	2	5	20 8 7 4 2 5 18 6 19 9 3 17 12 10 14 15 16 13 11 1	5 3 15 13
AB	3	7	5 9 3 4 7 8 20 15 14 10 2 19 6 18 1 13 17 12 11 16	20 1 11 16
B12	4	3	12 6 7 1 2 3 8 9 4 14 5 16 15 11 10 13	7 1 2 3 9 4 16 15
B14	5	3	7 9 10 8 12 13 11 4 2 3 5 14 6 1	10 11 5
B14	6	5	10 6 14 11 5 13 1 3 2 4 7 8 9 12	14 13 1 3 2 4
VC	7	5	1 2 6 9 10 7 8 4 5 3	1 6 9 7 4

Figure B1 Block Layout Encodings for Comparison

Table B6 Comparison Results for Bazaraa14

Bazaraa14						
Encoding #	**Flow Type**	**Aspect Ratio**	**Best Known Solution**	**Solution with Flows Applied**	**Best Integrated, Evolutionary Solution**	**% Change**
5	-	3	1343.2			
	U	3		3063.33	2604.17	14.99%
	L	3		5184.82	4732.08	8.73%
	C	3		3091.78	2392.67	22.61%
	M1	3		4605.31	3964.43	13.92%
	M2	3		3276.70	3276.70	6.09%
	M3	3		3562.36	3058.32	14.15%
6	-	5	4991.78			
	U	5		5080.46	2548.43	49.84%
	L	5		7460.15	4735.68	36.52%
	C	5		5048.43	2345.93	53.53%
	M1	5		5147.27	3784.88	26.47%
	M2	5		5118.79	2782.37	45.64%
	M3	5		5156.15	2973.76	42.33%

Table B7 Comparison Results for Armour and Buffa

Armour and Buffa						
Encoding #	**Flow Type**	**Aspect Ratio**	**Best Known Solution**	**Solution with Flows Applied**	**Best Integrated, Evolutionary Solution**	**% Change**
1	-	3	345.89			
	U	3		454.50	407.10	10.43%
	L	3		684.34	670.56	2.01%
	C	3		355.51	278.86	21.56%
	M1	3		470.23	434.54	7.59%
	M2	3		470.13	416.08	11.50%
	M3	3		474.95	387.35	18.44%
2	-	5	552.47			
	U	5		437.82	389.36	11.07%
	L	5		624.27	634.89	-1.70%
	C	5		364.46	260.00	28.66%
	M1	5		507.19	404.95	20.16%
	M2	5		488.51	425.06	12.99%
	M3	5		495.83	391.14	21.11%
3	-	7	785.8			
	U	7		871.81	318.86	63.43%
	L	7		1108.60	590.88	46.70%
	C	7		828.07	247.94	70.06%
	M1	7		857.82	399.83	53.39%
	M2	7		908.66	423.24	53.42%
	M3	7		856.93	364.04	57.52%

Table B8 Comparison Results for Bazaraa12

Bazaraa12						
Encoding #	**Flow Type**	**Aspect Ratio**	**Best Known Solution**	**Solution with Flows Applied**	**Best Integrated, Evolutionary Solution**	**% Change**
4	-	3	8861			
	U	3		8470.57	4564.37	46.11%
	L	3		9115.24	8585.57	5.81%
	C	3		7721.80	4032.62	47.78%
	M1	3		8819.75	5280.62	40.13%
	M2	3		8454.10	5836.88	30.96%
	M3	3		8597.67	5246.97	38.97%

Table B9 Comparison Results for van Camp

vanCamp						
Encoding #	**Flow Type**	**Aspect Ratio**	**Best Known Solution**	**Solution with Flows Applied**	**Best Integrated, Evolutionary Solution**	**% Change**
7	-	5	7239.03			
	U	5		8800.39	2134.02	75.75%
	L	5		10413.32	7358.07	29.34%
	C	5		7455.40	2963.66	60.25%
	M1	5		11521.55	4417.29	61.66%
	M2	5		8622.15	3005.52	65.14%
	M3	5		9686.24	3839.20	60.36%

APPENDIX C

SELECTED LAYOUTS

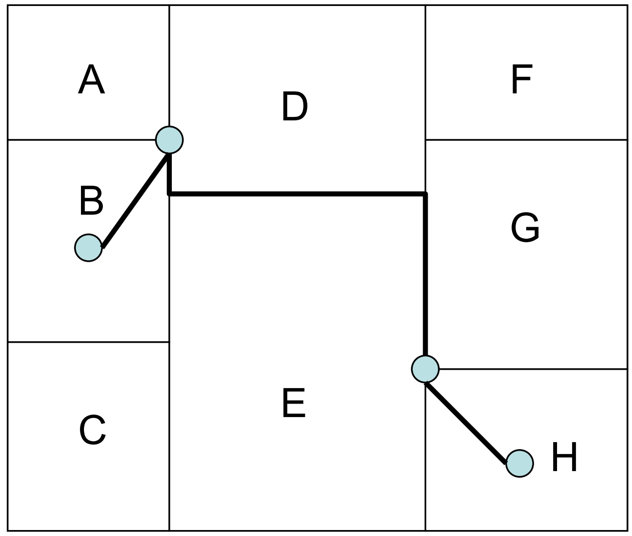

Figure C1 Combined Contour and Euclidean Metric

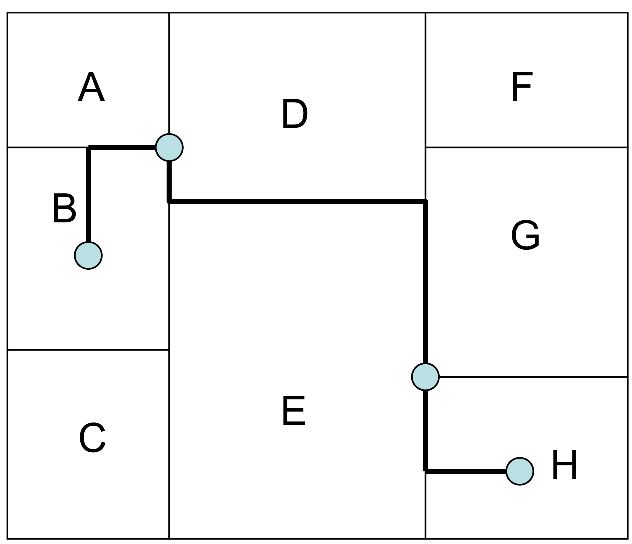

Figure C2 Combined Contour and Rectilinear Metric

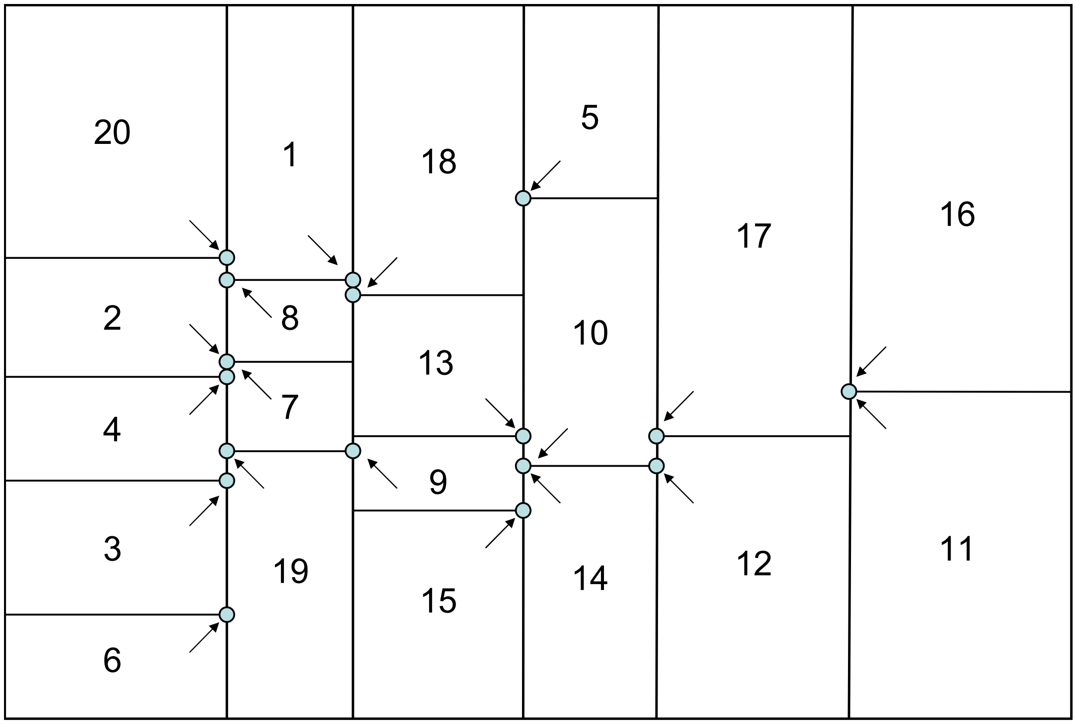

Figure C3 AB Layout with α = 3 and All Circular Flows

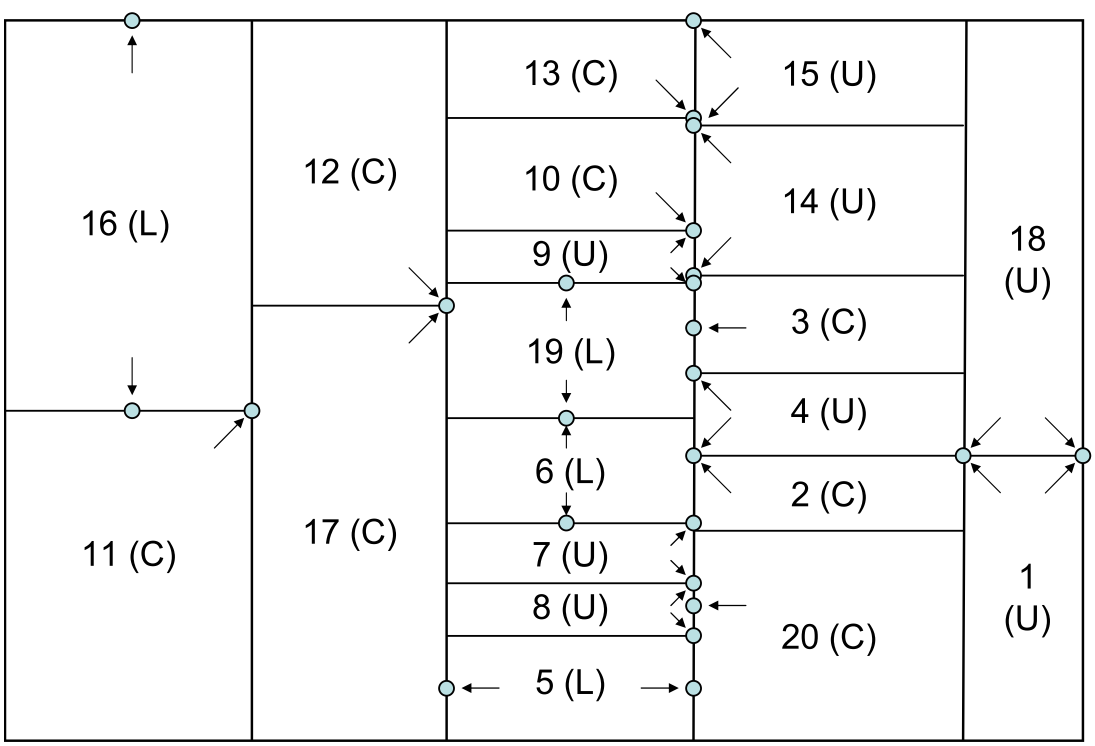

Figure C4 AB Layout with α = 5 and M1 Flow Pattern

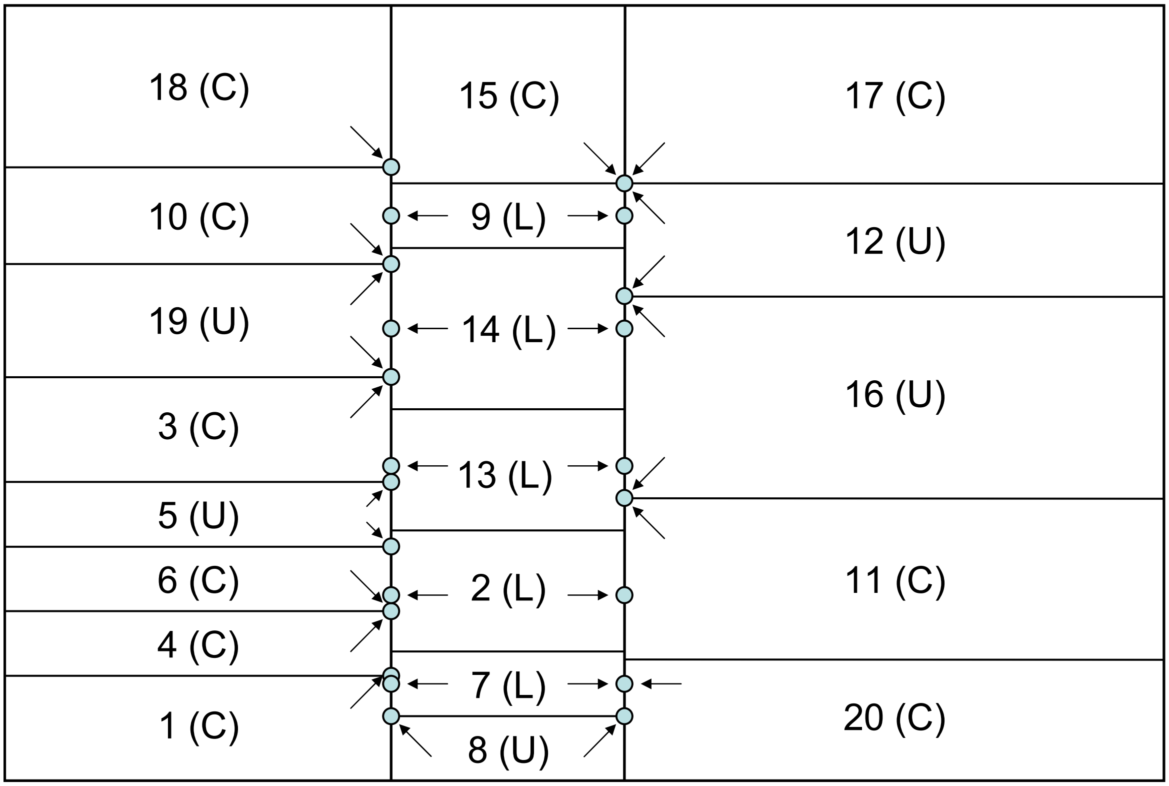

Figure C5 AB Layout with α = 7 and M3 Flow Pattern

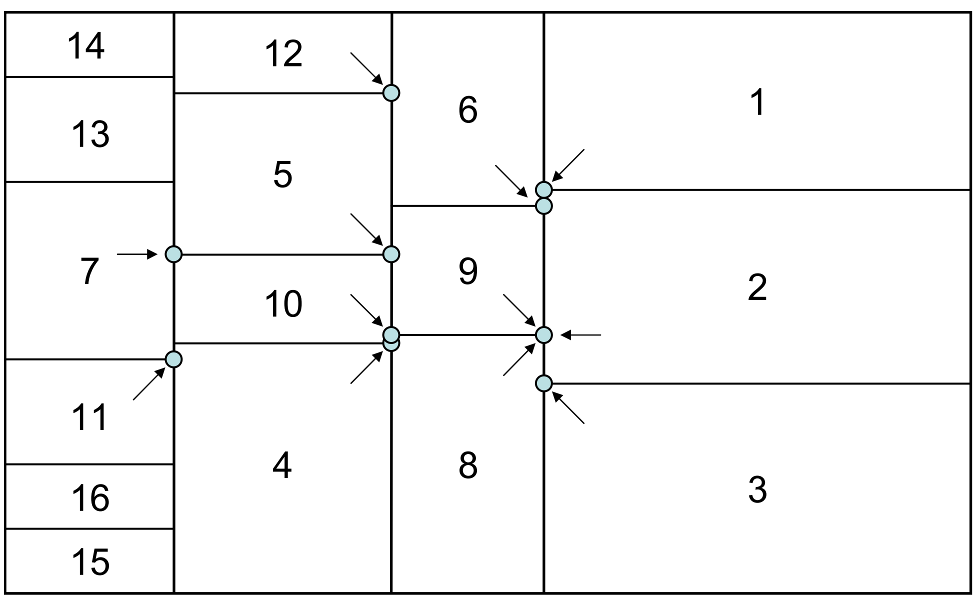

Figure C6 B12 Layout with α = 3 and All Circular Flows

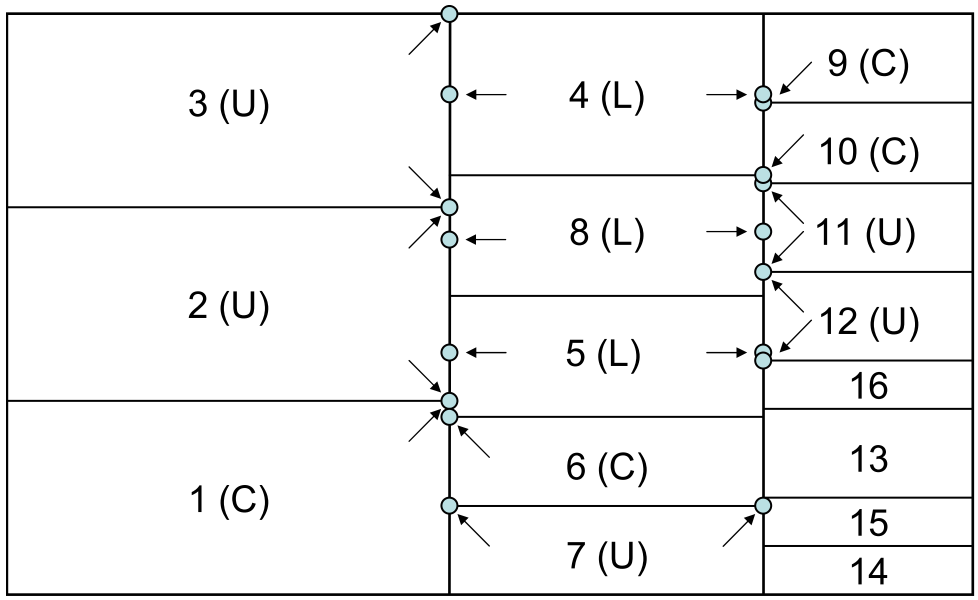

Figure C7 B12 Layout with α = 5 and M1 Flow Pattern

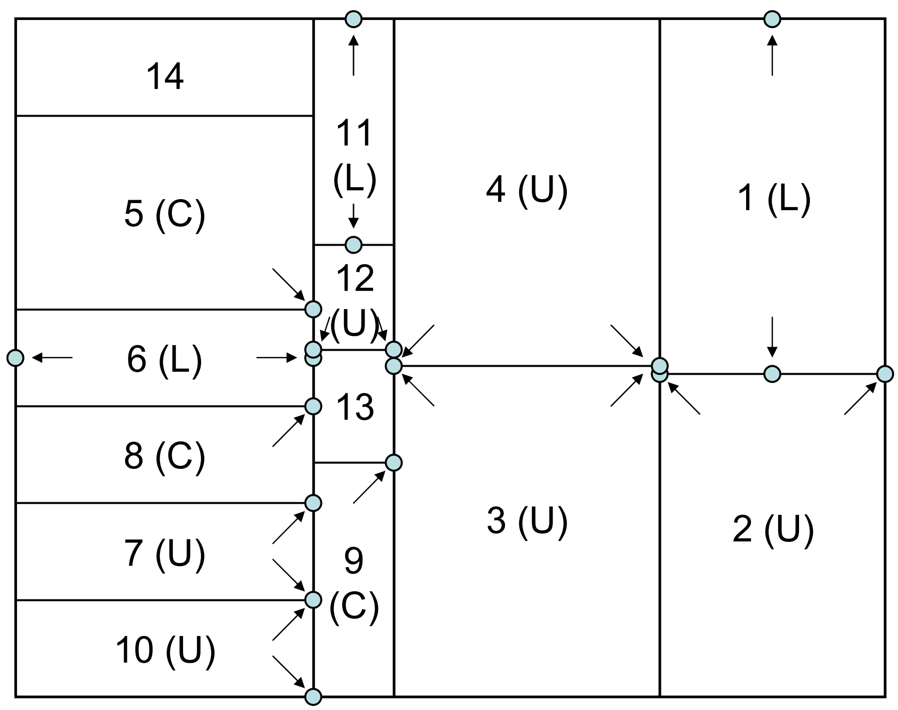

Figure C8 B14 Layout with α = 3 and M2 Flow Pattern

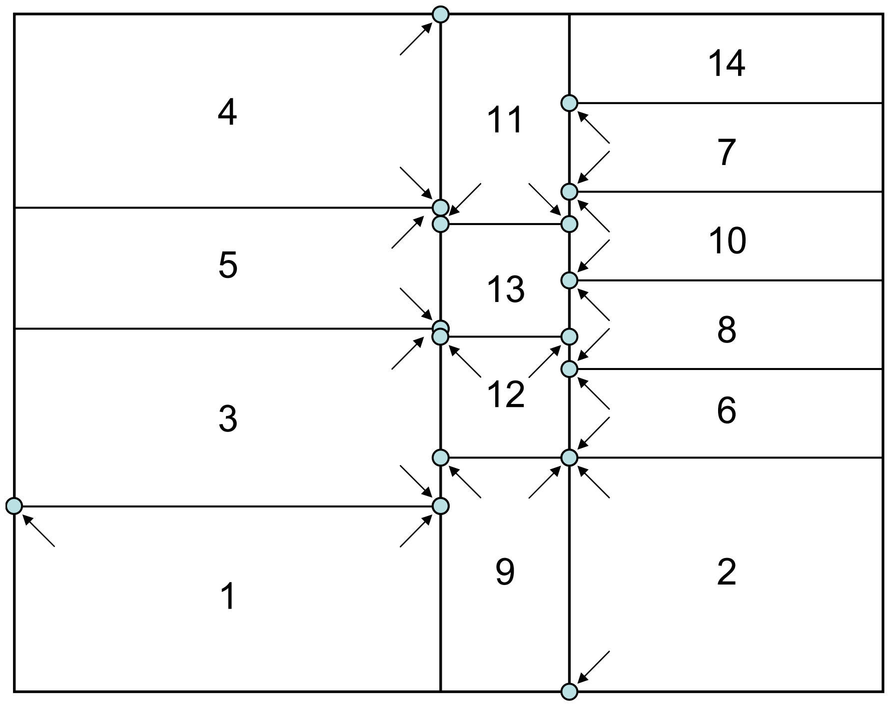

Figure C9 B14 Layout with α = 5 and All U-Shaped Flows

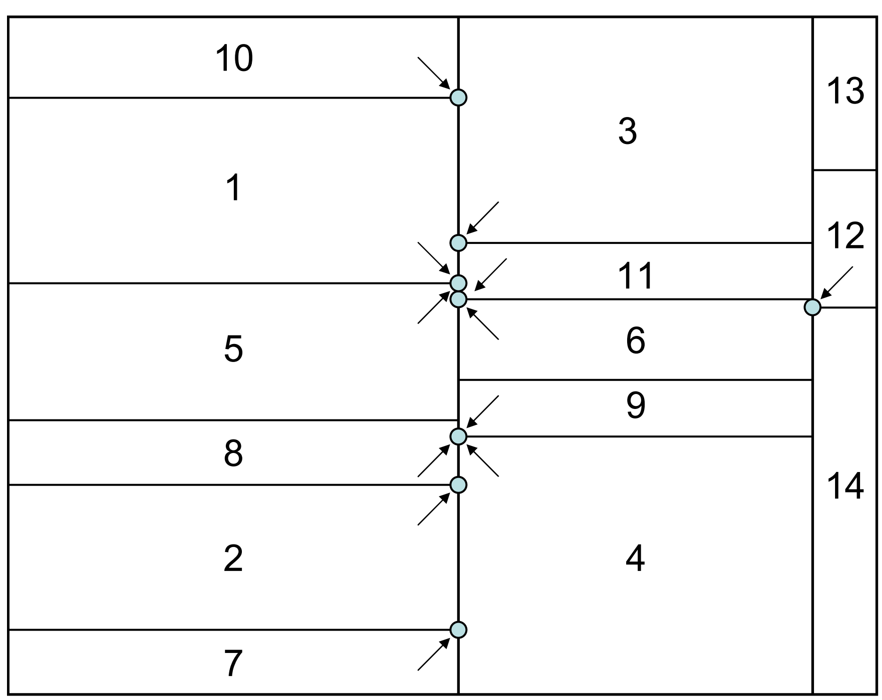

Figure C10 B14 Layout with α = 7 and All Circular Flows

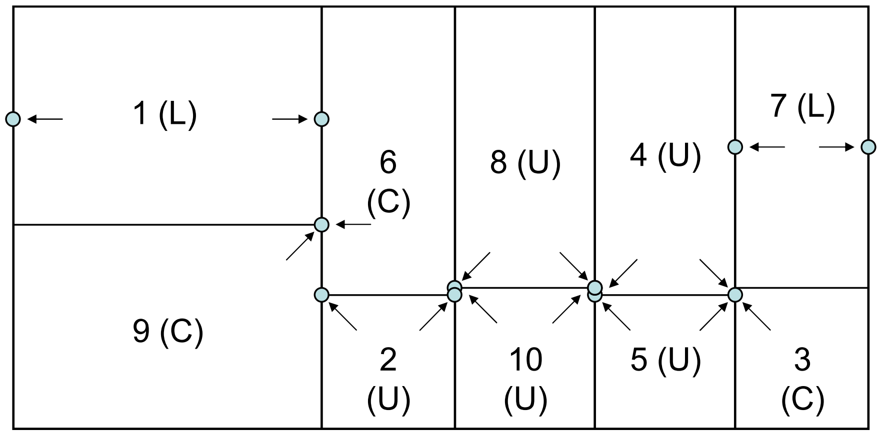

Figure C11 VC Layout with α = 3 and M2 Flow Pattern

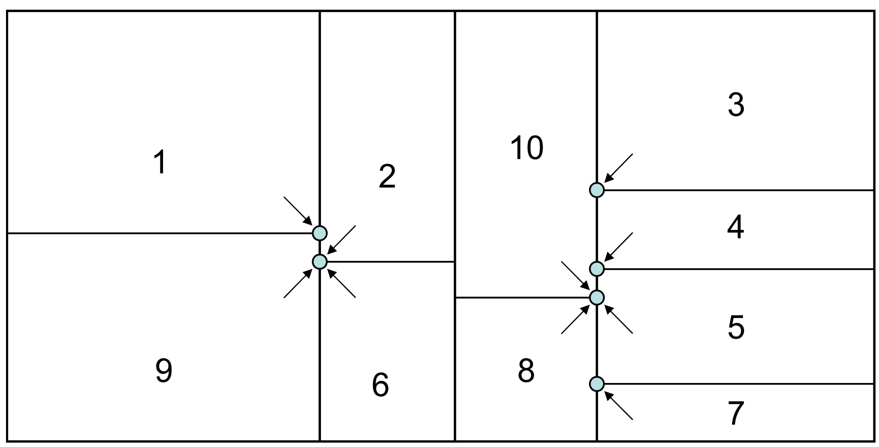

Figure C12 VC Layout with α = 5 and All Circular Flows

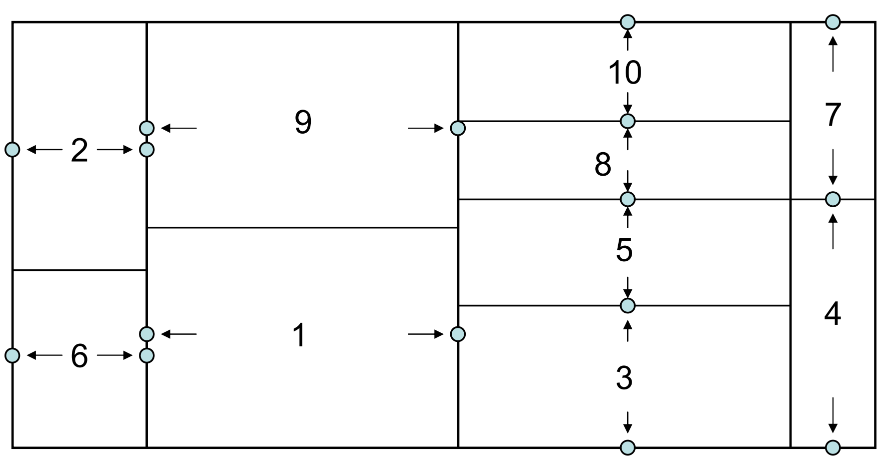

Figure C13 VC Layout with α = 7 and All Linear Flows

BIBLIOGRAPHY

[1] Aggarwal, C.C., Orlin, J.B. and Tai, R.P. (1997) Optimized Crossover for the Independent Set Problem. *Operations Research*, **45**(2), 226-234.

[2] Arapoglu, R.A. (2000) Simultaneous Layout Design in Facility Layout. Unpublished Doctoral Dissertation. Department of Industrial Engineering, University of Pittsburgh.

[3] Arapoglu, R.A., Norman, B.A. and Smith, A.E. (2001) Locating Input and Output Points in Facilities Design – A Comparison of Constructive, Evolutionary, and Exact Methods. *IIE Transactions on Evolutionary Computation*, **5**(3), 192-203.

[4] Armour, G.C. and Buffa, E.S. (1963) A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities, *Management Science*, **9**(2), 294-309.

[5] Banerjee, P., Zhou, Y. and Montreuil, B. (1997) Genetically Assisted Optimization of Cell Layout and Material Flow Path Skeleton, *IIE Transactions*, **29,** 277-291.

[6] Bazaraa, M.S. (1975) Computerized Layout Design: A Branch and Bound Approach, *AIIE Transactions*, **7**(4), 432-437.

[7] Bellman, R. (1965) An Application of Dynamic Programming to Location-Allocation Problems. *SIAM Review*, **7**, 126-128.

[8] Benson, B. and Foote, B.L. (1997) DoorFAST: A Constructive Procedure to Optimally Layout a Facility Including Aisles and Door Locations Based on an Aisle Flow Distance Metric. *International Journal of Production Research*, **35**(7), 1825-1842.

[9] Cheng, R., Gen M. and Tozawa, T. (1996) Genetic Search for Facility Layout Design Under Interflows Uncertainty. *IEEE*, 400-405.

[10] Chittratanawat, S. and Noble J.S. (1999) An Integrated Approach for Facility Layout, P/D Location and Material Handling System Design. *International Journal of Production Research*, **37(**3), 683-706.

[11] Coit, D.W., Smith, A.E. and Tate, D.M. (1996) Adaptive Penalty Methods for Genetic Optimization of Constrained Combinatorial Problems. *INFORMS Journal on Computing*, **8**(2), 173-182.

[12] Eklund, N.H., Embrechts, M.J. and Goetschalckx, M. (2003) An Efficient Chromosome Encoding and Problem-Specific Mutation Methods for the Flexible Bay Facility Layout Problem. *IEEE International Workshop on Soft Computing in Industrial Applications*, 109-113.

[13] Francis, R.L. and Goldstein, J.M. (1974) Location Theory: A Selective Bibliography. *Operations Research*, **22**, 400-410.

[14] Goldberg, D.E. (2002) *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers, Boston, MA.

[15] Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.

[16] Kim, J. and Klein, C.M. (1996) Location of Departmental Pickup and Delivery Points for an AGV System. *International Journal of Production Research*, **34**(2), 407-420.

[17] Konak, A., Kulturel-Konak, S., Norman, B.A. and Smith, A.E. (2004) A New Mixed Integer Programming Formulation for Optimal Facility Layout Design. *Submitted to Operations Research Letters*, 1-38.

[18] Kusiak, A. and Heragu, S.S. (1987) The Facility Layout Problem. *European Journal of Operational Research*, **29**(3), 229-251.

[19] Liggett, R.S. (1981) The Quadratic Assignment Problem: An Experimental Evaluation of Solution Strategies. *Management Science*, **27**, 442-458.

[20] Malakooti, B. and Tsurushima, A. (1989) An Expert System Using Priorities for Solving Multiple-Criteria Facility Layout Problems. *International Journal of Production Research*, **27**(5), 793-808.

[21] Meller, R.D. and Gau, K.Y. (1996) The Facility Layout Problem: Recent and Emerging Trends and Perspectives. *Journal of Manufacturing Systems,* **15**, 351-366.

[22] Meller, R.D., Narayanan, V. and Vance, P.H. (1998) Optimal Facility Layout Design. *Operations Research Letters*, **23**(3-5), 117–127.

[23] Michalewicz, Z. (1996) *Genetics Algorithms + Data Structures = Evolution Programs*, WNT, Warsaw.

[24] Montreuil, B. (1990) A Modeling Framework for Integrating Layout Design and Flow Network Design. *Proceedings of the Materials Handling Research Colloquium*, Hebron, KY, 43-58.

[25] Montreuil, B. and Ratliff, H. D. (1989) Utilizing Cut Trees as Design Skeletons for Facility Layout. *IIE Transactions*, **21**, 136-143.

[26] Montreuil, B. and Venkatadri, U. (1991) Strategic Interpolative Design of Dynamic Manufacturing Systems Layout. *Management Science*, **37**, 682-694.

[27] Norman, B.A., Arapoglu, R.A. and Smith, A.E. (2001) Integrated Facilities Layout Using a Contour Distance Measure. *IIE Transactions*, **33**(4), 337-344.

[28] Norman, B.A. and Smith, A.E. (1997) Random Keys Genetic Algorithm with Adaptive Penalty Function for Optimization of Constrained Facility Layout Problems. *Proceedings of the IEEE Conference on Evolutionary Computing*, Indianapolis, IN, 407-411.

[29] Norman, B.A., Smith, A.E., Yildirim, E. and Tharmmaphornphilas, W. (2001) An Evolutionary Approach to Incorporating Intradepartmental Flow into Facilities Design. *Advances in Engineering Software*, **32**(6), 443-453.

[30] Peters, B.A. and Yang, T. (1997) Integrated Facility Layout and Material Handling System Design in Semiconductor Fabrication Facilities. *IEEE Transactions on Semiconductor Manufacturing*, **10**(3), 360-369.

[31] Roberts, S.M. and Flores, B. (1965) Solution of a Combinatorial Problem by Dynamic Programming. *Operations Research*, **13**, 146-156.

[32] Rosenblatt, M.J. (1986) The Dynamics of Plant Layout. *Management Science*, **32**, 76-86.

[33] Schnecke, V. and Vornberger, O. (1997) Hybrid Genetic Algorithms for Constrained Placement Problems. *IEEE Transactions on Evolutionary Computation* , **____** .

[34] Sherali, H.D., Fraticelli, B.M.P. and Meller, R.D (2003) Enhanced Model Formulations for Optimal Facility Layout. *Operations Research*, **51**(4), 629–644.

[35] Tate, D.M. and Smith, A.E. (1995) Unequal-Area Facility Layout by Genetic Search. *IIE Transactions*, **27**, 465-472.

[36] Tong, X. (1991) SECOT: A Sequential Construction Technique for Facility Design. Unpublished Doctoral Dissertation, Department of Industrial Engineering, University of Pittsburgh.

[37] van Camp, D.J., Carter, M.W. and Vannelli, A. (1991) A Nonlinear Optimization Approach for Solving Facility Layout Problems. *European Journal of Operational Research*, **57**, 174-189.