2017

Improvement of Turret Punch Tool Layout for the Production of Nested Parts

Anderson, David C

UNIVERSITY OF CALGARY

Improvement of Turret Punch Tool Layout for the Production of Nested Parts

by

David C. Anderson

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF ENGINEERING

GRADUATE PROGRAM IN MECHANICAL AND MANUFACTURING

ENGINEERING

CALGARY, ALBERTA

JANUARY, 2017

ABSTRACT

Sheet metal parts processed by CNC turret punches are often grouped together onto single sheets of material, known as nests, in random combinations based on current demand. The content and configuration of each nest is highly variable, resulting in unique hole locations and quantities.

A hybrid genetic algorithm (HGA) is presented for the development of a robust turret layout given a set of parts with known tool requirements and flexible operation sequences. HGA population members are improved through an iterative local search heuristic that alternately considers part operation sequences and turret layout. Improved members replace their unimproved predecessors in the population.

HGA solutions are tested for robustness using a modified form of the Layout Configuration Robustness Index (LCRI). The HGA solutions are shown to offer a statistically significant decrease in total turret rotation distance when compared to a population of randomly generated layouts.

ACKNOWLEDGEMENTS

The topic of this thesis is rooted in a challenge that I encountered early in my professional career which sparked an interest in combinatorial optimization. In addition to providing guidance and support throughout this journey, I would like to thank my supervisor, Dr.Y.Tu, for allowing me to pursue my own research interests and finally develop the better solution that I knew must exist.

I would also like to thank the members of my committee, Dr. T. Freiheit, Dr. D. Xue and Dr. J. Balakrishnan for the time that they have spent reviewing my thesis and the valuable suggestions that they have provided.

Finally, this thesis would not have been possible without the loving support from my family. Special thanks are owed to my wife Maria for supporting me through the successes and setbacks that this project entailed. I must also thank my sons David and Adam for their patience and understanding on all of the weekends when Daddy couldn't play.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS, ABBREVIATIONS,NOMENCLATURES

Symbols

a – sequence step to which an operation is assigned

b – operation to be completed on a part

C-Cost of assigning a facility to a location

D- Distance between locations

F – Flow occurring between facilities

i -index indicating source facility

j -index indicating destination facility

k -index indicating candidate location for source facility

l -index indicating candidate location for destination facility

$m(l)$ – The estimated mean cost of producing a demand schedule using layout *l*

$\overline{m(l)}$ –The estimated mean cost of producing a demand schedule across all layouts

$M(l)$ – The actual mean cost of producing a demand schedule using layout l

$\overline{M(l)}$ - The actual mean cost of producing a demand schedule across all layouts

n – The number of facilities and locations in the assignment problem

p -indicator of part number

P – total number of parts

N – number of possible layouts in robustness measurement

s_p – number of sequence positions associated with part p

$s(l)$ – The estimated variance in cost for producing a demand schedule using layout l

$S(l)$ - The actual variance in cost for producing a demand schedule using layout l

$\overline{s(l)}$ – The estimated mean variance in cost for producing a demand schedule across all layouts

$\overline{S(l)}$ – The actual mean variance in cost for producing a demand schedule across all layouts

T_{pbi} - An initial assignment of facility i to sequence number b of part p.

V_p – Expected production volume of part p

X-Assignment of facilities to locations

Y_p - A permutation matrix representing the sequence of operations within part p

Abbreviations

ACA – Ant colony algorithm

CRAFT – Computerized relative allocation of facilities technique

CSA – Coupled simulated annealing

CNC - Computer numerically controlled

DFLP –Dynamic facility layout problem

DP – Dynamic programming

ETS – Enhanced tabu search

FLP –Facility layout problem

GA –Genetic algorithm

HGA – Hybrid genetic algorithm

IHGA – Improved hybrid genetic algorithm

IHO – Iterative hold and optimize heuristic used by Kumar & Veeramani (1995)

ILS – Iterated local search

ISP – iterated swap procedure

ITS – Iterated tabu search

LCRI – Layout cost robustness index

LPRI – Layout problem robustness index

LSH – Local search heuristic – a method proposed in this thesis

MAP – Minimax assignment problem

MBTD – Moving board with time delay

MNS – Modified neighbourhood search – a method proposed in this thesis

MST – Minimum spanning tree

NP-Nondeterministic, polynomial time

NNR – Nearest neighbour

PCB – Printed circuit board

QAP – Quadratic assignment problem

SDFLP – Stochastic dynamic facility layout problem

SFLP – Stochastic facility layout problem

SMD –Surface mounted device

SUB – A substitution heuristic used by Kumar & Veeramani (1995)

TS – Tabu search

TSP –Traveling salesman problem

VNZ - Vollmann, Nugent and Zartler heuristic for the quadratic assignment problem

CHAPTER 1: INTRODUCTION

1.1 CNC Turret Punches

CNC turret punch presses are used to create two dimensional patterns in sheet metal by moving a flat piece of metal in a horizontal plane and striking the piece with a punching tool in the vertical direction at locations specified in a proprietary CNC program. Top and front views of a typical gantry-style punching machine are shown in Figure 1-1. During the punching operation, the punching tool, typically positioned above the sheet metal, is struck by a ram. The downward force from the ram drives the punching tool through the sheet metal and into a die located on the underside of the sheet, forming the hole. The punch is then returned upward to its home position by a spring so that the sheet is free to move to the next punching location. (Tlusty, 2000)

To increase the number and type of holes that can be included in the pattern, multiple punching tools are mounted into a circular turret that rotates about a vertical axis in a plane between the ram and the workpiece. A second turret containing matching dies rotates synchronously beneath the workpiece. To create the desired hole, the turret is rotated, or indexed, until the desired tool is positioned directly beneath the ram at a fixed position in the plane of the work piece. It is common for a turret to hold 40-50 tools and rotate at speeds of ~30rpm (e.g. (Amada GmbH, 2011), (Murata Machinery, Ltd., 2014).

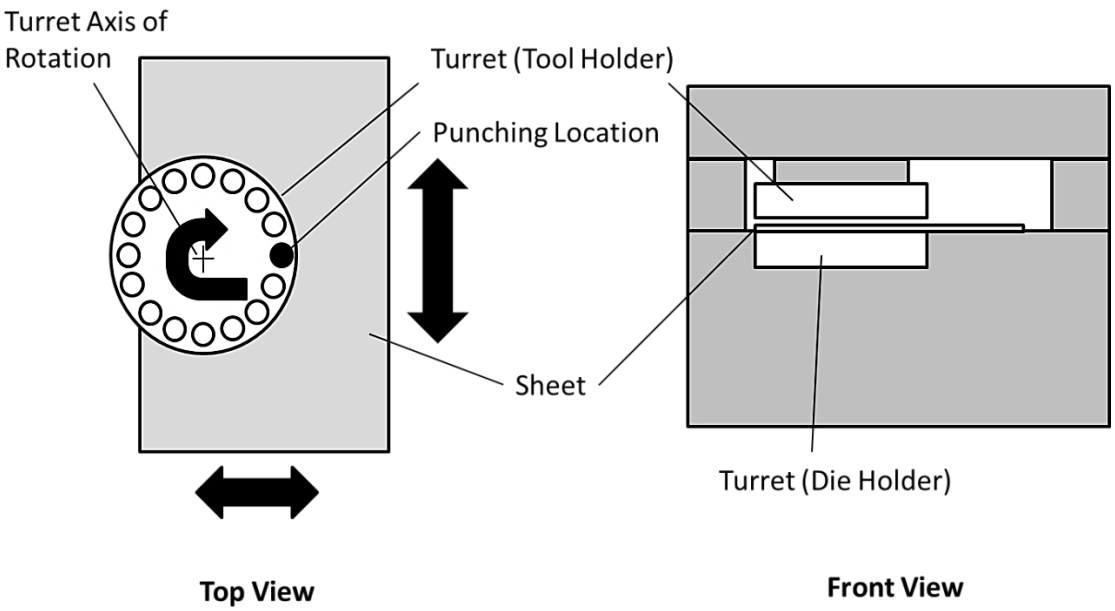


Figure 1-1 Typical configuration of a turret punch

Kumar and Veeramani (1995) divided the time required to process a piece of sheet metal on a turret punch into two categories: actual punching time and inter-hit delay. While the actual punching time is limited by the machine capabilities and the tool being used, Kumar and Veeramani (1995) considered the inter-hit delay to be improvable and subdivided it into the additional two categories of sheet travel time and turret rotation time. Functionally, sheet travel time is considered in both the x and y axes separately since each axis is commonly controlled by an independent drive motor. In the operation of a turret punch, these three movements (turret rotation, x-axis translation and y-axis translation) occur concurrently, and the inter-hit delay is determined by the greater of the two metrics. The movements required between holes are shown in Figure 1-2. This

notion is similar to the moving board with time-delay problem (MBTD) (Leu, et al., 1993) that has been studied for the installation of surface mounted devices (SMD) in printed circuit board (PCB) manufacturing.

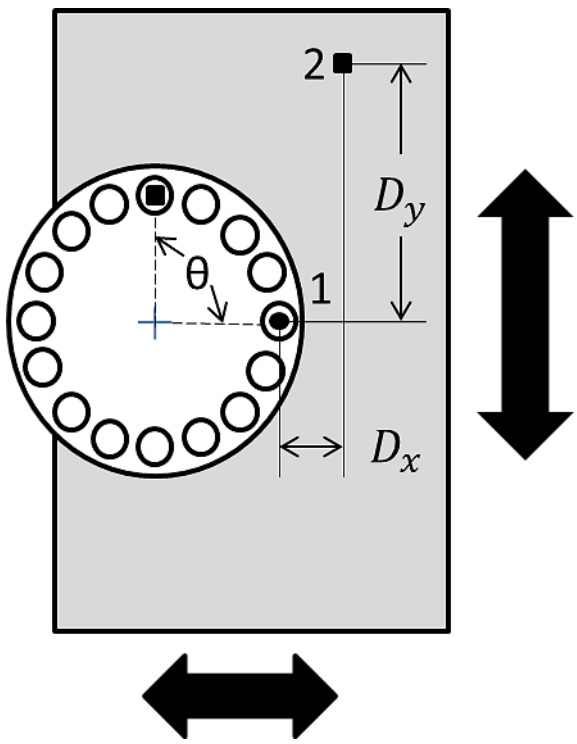


Figure 1-2 Required movements between punch locations

In the production of a sheet metal part, the movement between hole locations is dictated by the location of holes within the part. As real design requirements of the part, these locations cannot be changed and will consequently influence the production time required to make the part. In contrast the turret rotation time is a characteristic of the machine and can be minimized by an effective layout of tools within the turret.

1.1.1 Nesting

Historically, punched parts were produced by first shearing a piece of sheet metal to the required outside dimensions of the part resulting in a ‘blank’. Blanks were then loaded individually into the turret punch for the production of a single part. A single CNC program was run repeatedly, once for each individual part. This process is shown schematically in Figure 1-3. Many manufacturers have eliminated the shearing operation by moving to a process known as nesting.

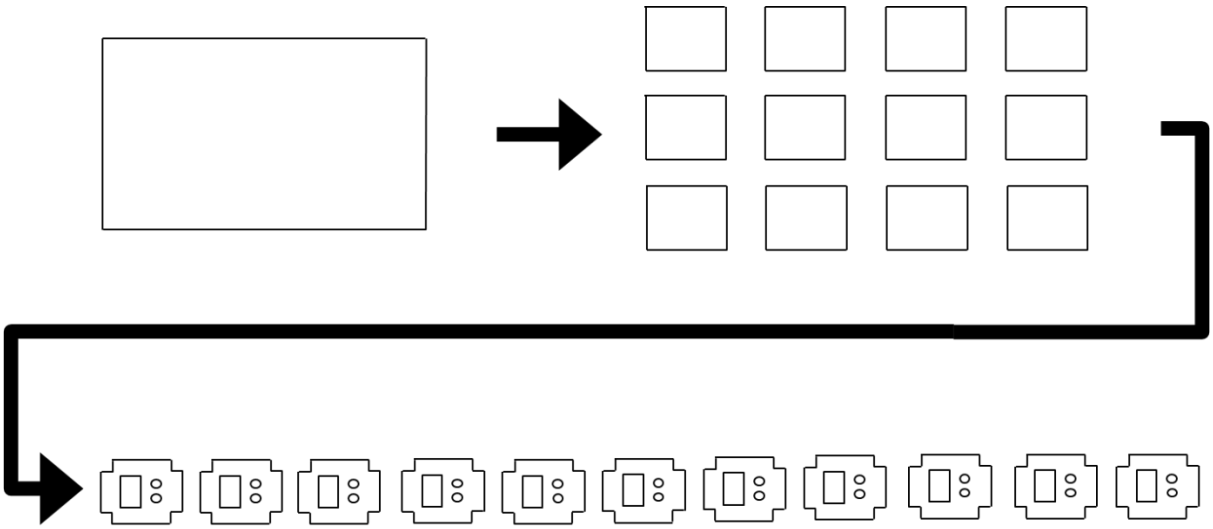


Figure 1-3 Schematic representation of historical punching process

Nesting is described by Herrmann & Delalio (2001) as the process of forming batches that can be manufactured efficiently by forming combinations from within a pre-specified set of orders. In the context of the current research, this means that several parts, each much smaller than a piece of sheet metal, can be combined on a single piece of sheet

metal and processed simultaneously through the punching operation. Early nesting efforts were fixed programs, sometimes called static nests, designed to produce multiple pieces of the same part out of a single piece of sheet metal as shown schematically in Figure 1-4.The current research investigates the case where multiple different parts are combined into a single sheet based on current demand, sometimes called dynamic nests, as shown in Figure 1-5.

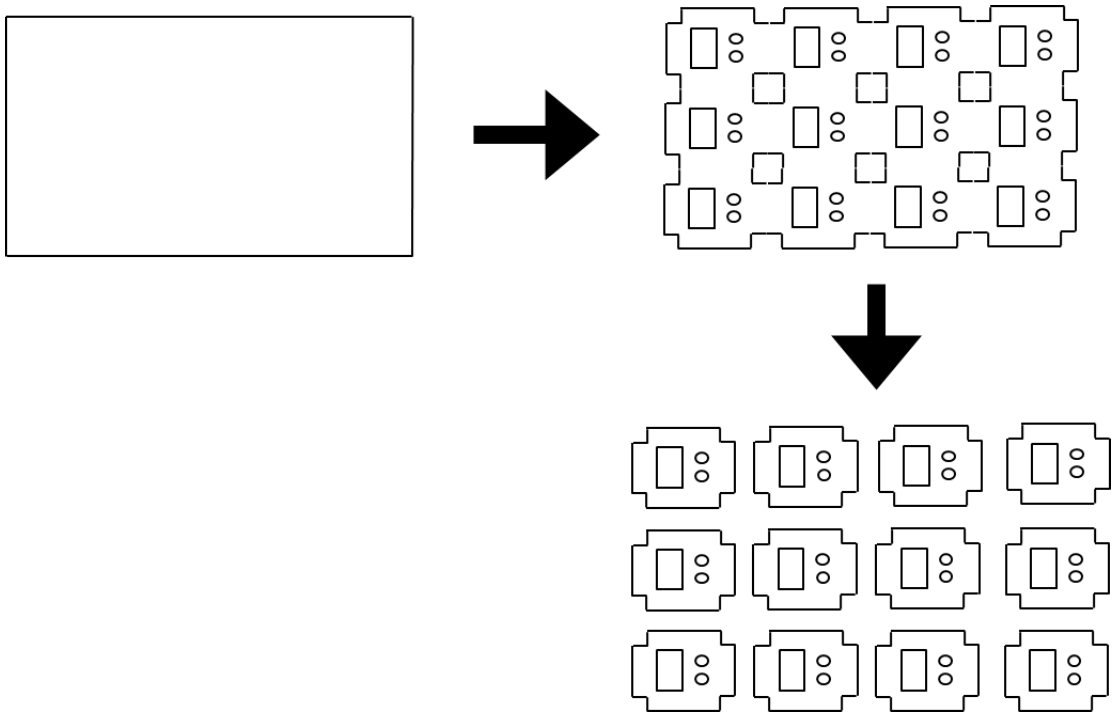


Figure 1-4 Schematic representation of static nesting

While Herrmann & Delalio are not attempting to optimize the programming for the batches, their work provides important context for the rising popularity of dynamic

nesting. Reviews of trade publications further demonstrate the prevalence of nesting as well as the tool path issues that are related such as the need to punch in sections for increased accuracy (Binder, 2008) and minimized distortion (Ripka, 2014).

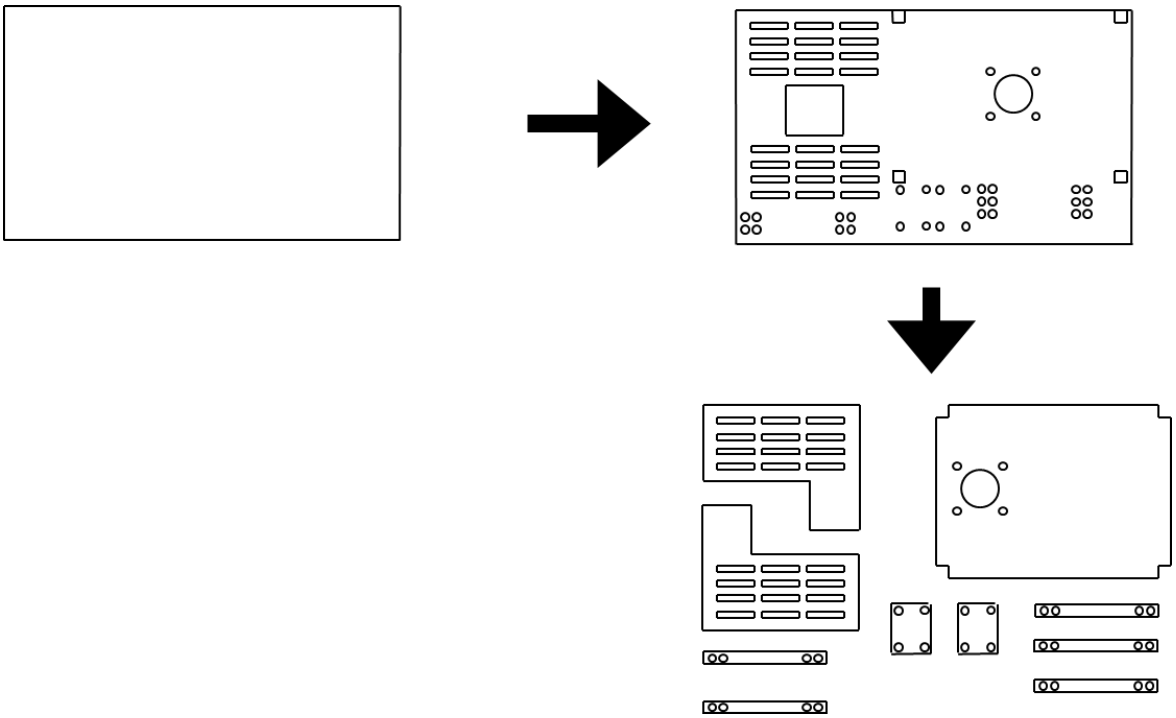


Figure 1-5 Schematic representation of dynamic nesting

1.1.2 Process Planning for Nested Sheets

Chauny et al. (1987) presents a heuristic for optimizing the tool path of a dynamically nested sheet. In their paper, they summarize the steps of production related to nesting as:

1. Creation of production lots
2. Sequencing of the lots
3. Positioning parts on sheets (cutting stock problem)
4. Positioning tools in the sockets of the turret
5. Clamps positioning on the table and on the sheet.
6. Sequencing of punch operations
7. Sequencing of the sheets in a production lot.

As is often noted in the general tool switching problem (e.g. (Chaves, et al., 2016) (Crama, et al., 2007) (Konak, et al., 2008)) and specifically with respect to CNC turret punches (Marvizadeh & Choobineh, 2013), changing the setup of the turret between production lots significantly increases the unproductive time of the machine. This setup time has been eliminated by many manufacturers through the introduction of standard turret configurations that can be used for all production routed through a single machine with no changes in tool content or tool position. This improvement has effectively eliminated steps 1 and 2 from the sequence presented by Chauny, et al. (1987) and has transformed step 4 into a design problem rather than a routine operations problem. Steps 3, 5 and 6 are performed by a CAM system in which the standard turret configuration is provided as an input.

The sequence of punching operations for nested parts (step 6 above) has been studied in in two specific contexts. Firstly, Wang & Xie (2005) consider tool path generation for nests processed on punch-and-laser combination machines. As with a single part

program, the tool path is found by modelling as a travelling salesman problem and solving using an ant colony optimization method. Xie & Tu (2011) continued the work of Wang & Xie (2005) by applying a genetic algorithm approach to the same sample problem and found that the genetic algorithm approach was more effective than the ant colony technique that was used in the original research. Both of these studies assumed that all holes requiring a particular tool should be completed before changing tools used.

Secondly, Veeramani & Kumar (1998) study the tool path of a part nibbling tool that is used to separate the individual parts from the sheet at the end of the punching process. In their solution method, the main problem of path optimization is subdivided into the two sub-problems of pierce point determination and part sequence determination. While this work is an important contribution to the field, it is not readily applicable to most tools since the decision of pierce point is not relevant for most punching operations.

Historically, complete optimization of a punching operation has been thought to require consideration of both the turret configuration and the punching sequence. The close interrelationship of these problems has led to them being solved simultaneously by both Walas & Askin (1984) and Kumar & Veeramani (1995) for the optimization of single programs. These same approaches cannot be applied to the case of dynamic nesting since the composition of the sheet is subject to wide variation. Consider the schematic shown in Figure 1-6. Small variations in the demand for any single part within a group of parts can change the composition of the nested sheet. For each variation, the number of each

hole type and the locations of the holes on the sheet can change significantly. This research focusses on the need for a method to use part tool requirements to develop a turret layout that can efficiently produce all nest variations.

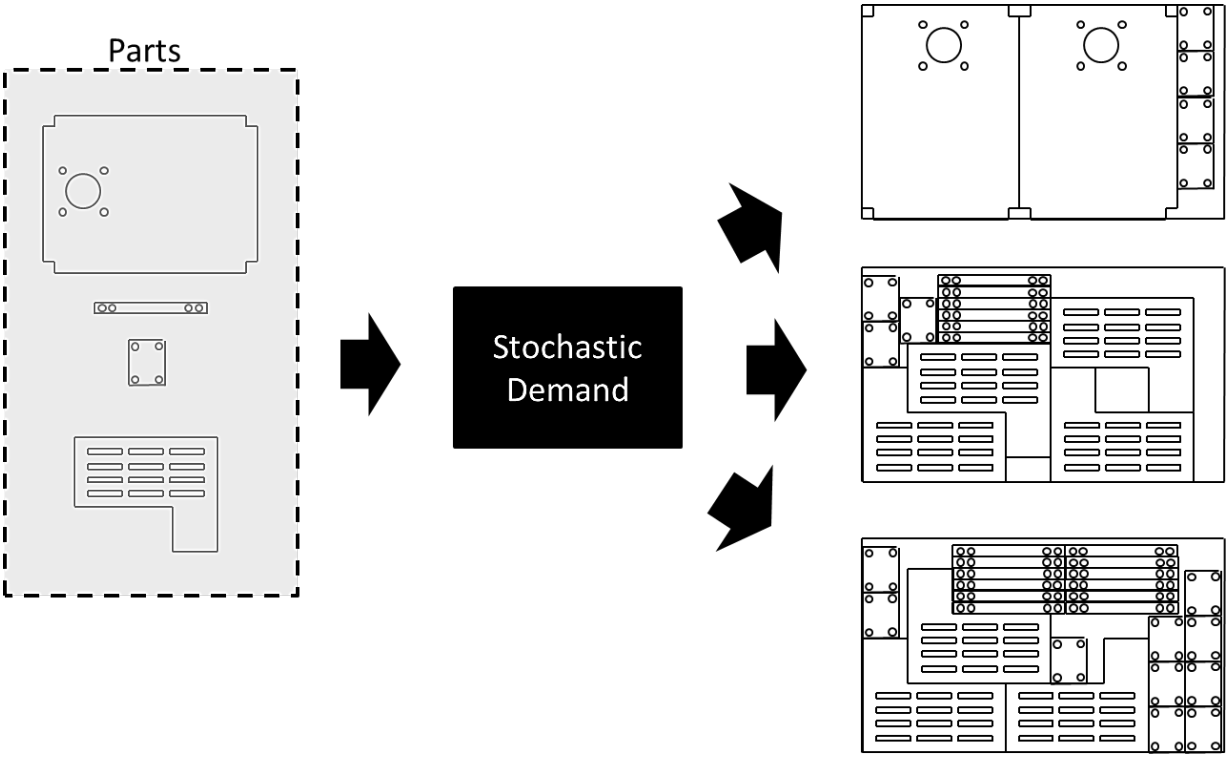


Figure 1-6 Variations in dynamic nesting caused by stochastic demand

1.1.3 Required Robustness

In the production of PCBs, the relationship between decreased setup time and increased production time is well known (e.g. (Chen & Chyu, 2002), (Yilmaz, et al., 2007)). The extension of this observation is further magnified in a nested part environment since,

unlike PCBs, each nest is a unique entity that is formed from the demand for parts at the moment that the nest is created.

When a standard turret is introduced, setup time is eliminated. However, it would be unacceptable to allow the production time to increase without bound in order to remove all setup time. Consequently, it is desirable to develop a layout within the turret that strives to deliver efficient performance across the continuously variable range of nests that it is used to produce. In the stochastic facility layout problem (SFLP) this characteristic is described as robustness.

Robustness was introduced by Gupta & Rosenhead (1968) as a measure of flexibility. Although the study of robustness of layouts has included several measurement approaches, such as penalty approaches (e.g. (Rosenblatt & Lee, 1987)), continuous cost distributions (e.g. (Norman & Smith, 2006)) and performance indices (e.g. (Braglia, et al., 2003)), robustness is consistently considered to reflect a layout that will provide above average performance across a range of demand conditions.

1.2 Research Objectives

The considerable body of literature that seeks to optimize the performance of turret punches reflects an outdated paradigm in which parts are produced individually. It is notable that while some research has been conducted to improve the tool path planning for nested parts, no research has sought to improve the layout of the turret for the production of randomly generated nests that is common in modern sheet metal

manufacturing. Industry observations suggest that an increasing number of companies are adopting standard turrets that require no tool changes, but there is no evidence of optimization within the standard turret layout.

The characteristics of processing nests using a standard turret that make it different from historical single part production are:

a. The sequence of tool use for a nest is determined by the composition of the nest, not the individual part as occurs in single part production. Tool sequence will be selected based on the set of tools required to produce the nest.
b. Nests are rarely duplicated so there is no motivation to optimize the turret layout for a single nest as there is in the production of single parts.
c. Tool path planning is executed by CAM software at the time of nest creation and its efficiency is limited by the potential of the turret configuration that is available.

The objectives for this research are to provide a method for developing a robust, standard turret configuration for use in a nesting environment by:

1. Generalizing the turret layout problem to a facility layout problem characterized by flexible sequencing and developing a suitable heuristic for accurately solving the problem for an industrial scaled problem.

2. Measuring the robustness of the result by applying the developed heuristic to the layout of a turret punch based on a simulated set of tools and testing the result using tools developed for the SFLP.

1.3 Thesis Layout

The remainder of this thesis is presented in the following manner:

Chapter 2: Literature Review

Chapter 3: Methods

Chapter 4: Results

Chapter 5: Conclusions

CHAPTER 2: LITERATURE REVIEW

2.1 Classic Combinatorial Optimization Problems

Many past efforts to model turret punch operation have built upon two well-known problems in combinatorial optimization; the quadratic assignment problem (QAP) and the travelling salesman problem (TSP). The QAP is generally used as a method for assigning the required tools to locations in a turret as shown in Figure 2-1.

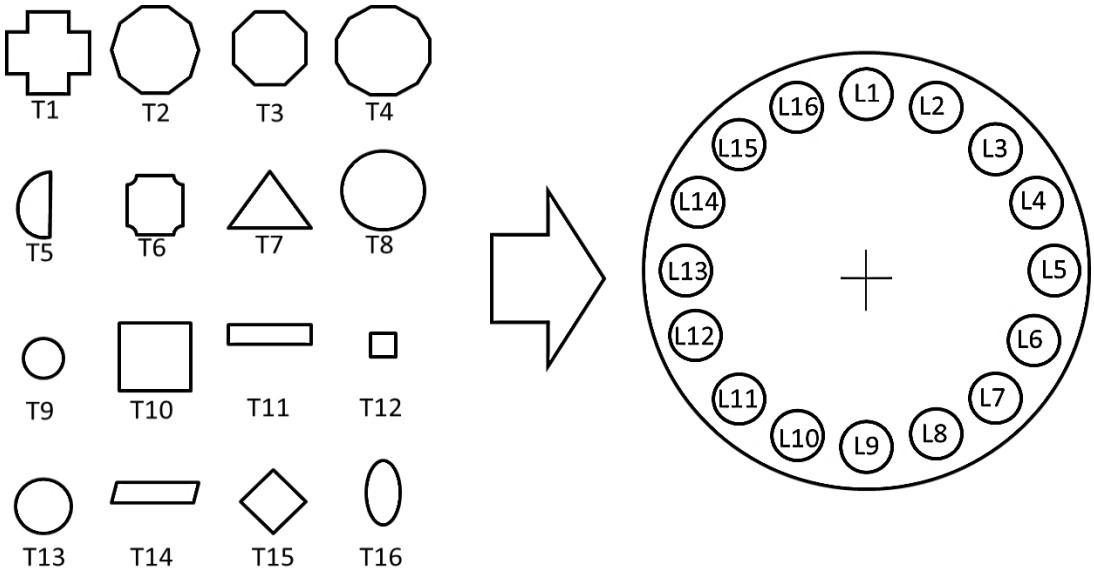


Figure 2-1 QAP applied to turret layout

In Figure 2-1, tools are denoted by 'T' and locations are denoted with 'L'. Since each tool makes a unique shape, different combinations of tools are required to make each part. The QAP framework is used to reduce the amount of turret rotation that is needed in

order to position all required tools at the punching position of the turret in the sequence that they are needed.

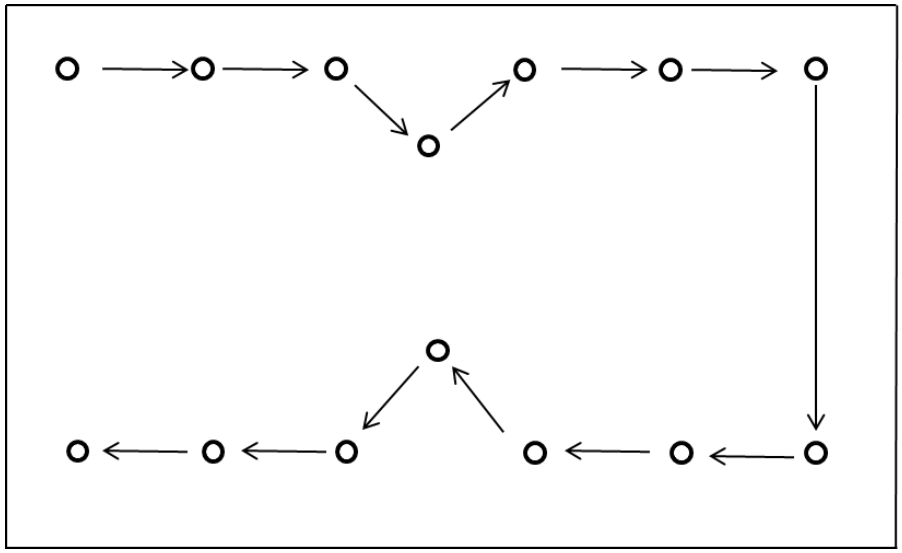


Figure 2-2 Traditional TSP for sheet metal processing

Historically, the TSP has been used to find the shortest travel distance through a series of punching locations as shown in Figure 2-2. However, in the current research, we apply the TSP model to the problem of travelling between tool holder locations as shown in Figure 2-3.

A brief introduction into the structure and solution methods for these problems is presented before further exploring their applications to the current research problem.

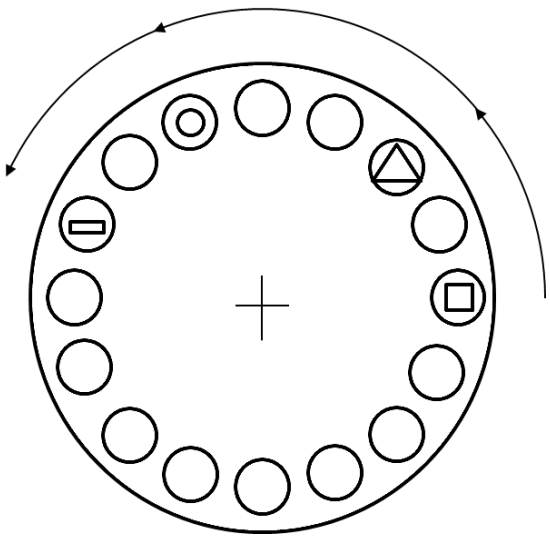


Figure 2-3 TSP applied to turret rotation

2.1.1 The Quadratic Assignment Problem

Drira et al. (2007) broadly defined facilities as the components of a plant that are required to produce a good or deliver a service. The facility layout problem (FLP) is the arrangement of these facilities within a plant such that an operational target, most often the minimization of material handling costs, is achieved. This problem has been studied many times and in many different variations. Good summaries can be found in Drira et al. (2007), Meller & Gau (1996)and Kusiak & Heragu (1987) .

The QAP is a specific subset of the FLP in which the available locations are identically sized and located in discrete locations. The QAP was introduced by Koopmans & Beckmann (1957) as a means of assigning *n* facilities to *n* locations such that the

economic gain of the system was maximized. The general form of this problem is presented by Burkard, et al. (2009) in matrix form as:

$$TC = F \cdot XDX^T + CX$$

where:

F represents the flow between facilities

X is a permutation matrix defining the assignment of facility *i* to location *j*

D is the a matrix summarizing the distances between locations

C is the cost associated with assigning facility *i* to location *j*.

The QAP is thought to be NP-Hard (Sahni & Gonzales, 1976). Consequently there has been a significant amount of research into heuristic methods, such as genetic algorithms, simulated annealing, ant colony search, iterated local search and tabu search which can be applied to the QAP. Details of heuristic developments can be found in the online QAPLIB (Hahn & Anjos, 2011).

2.1.1.1 Exact Solution Methods for the QAP

Considered to be NP-Hard, exact solutions for the QAP are difficult to find for large scale problems. As noted by Brixius (2000) all three instances of the Steinberg wiring problem provided by Steinberg (1967) remained unsolved at that time, 33 years after introduction.

An early branch and bound technique was introduced independently by Gilmore (1962) and Lawler (1963) and is known as the Gilmore-Lawler bound. The method begins by linearizing the model and then recursively exploring the costs resulting from partial assignments. Branches that cannot possibly improve on a best known solution are abandoned. A good implementation of this method can be found in Francis & White (1974).

Improvements in exact solution methods generally rely on finding improved lower bounds to allow faster elimination of branches or applying computational improvements such as parallel computing. A good summary of these techniques can be found in Brixius (2000) or Loiola, et al. (2006). Since a modern turret punch typically holds more than 50 tools, exact solutions are not feasible for the current application. Instead, we focus on heuristic techniques that can find near optimum results quickly.

2.1.1.2 Heuristic Methods for the QAP

Taillard (1995) classified different QAP examples into four categories: random and uniform distances and flows, random flows on grids, real life problems, and non-uniform random problems. Taillard tested different algorithms on different classes and found that

the hybrid genetic algorithm (HGA) method was more effective than the other metaheuristics considered for both random flows on grids and real life processes. These are the classes of problems closest to the one in the current research. In a study of iterated tabu search (ITS) performance, Misevicius (2012) reinforced this observation by noting that although ITS outperformed other variants of tabu search on real-life like instances, it could not match the performance of an improved hybrid genetic algorithm (IHGA) that used tabu search as a local improvement tool. This builds on an earlier finding of Misevicius (2005) that compared the performance of an enhanced tabu search (ETS) to the HGA introduced by Fleurent & Ferland (1994) and found that the ETS and HGA performance was similar in the class of real-life like instances. The later developed IHGA (Misevicius, 2004) incorporates elements of the ETS into the HGA. Similarly, Stützle (2006) considered eight variations of iterated local search (ILS) on different examples from QAPLIB. The variation of ILS that included evolutionary search characteristics demonstrated the best performance on real life instances and randomly generated real-life like instances. From the literature, it is clear that the solution algorithm used should match the type of problem and that the family of hybrid genetic algorithms is well suited to the class of real-life problems.

The success of a heuristic method relies on both its explorative and exploitative capabilities. By incorporating an additional local search tool, HGA approaches improve the exploitative nature of the GA by searching the neighbourhood surrounding a good solution. As with a classic GA, the explorative nature is created by mating chromosomes

to find new areas of the solution space. Fleurent & Ferland (1994) are often credited with introducing this method to the QAP. Drezner (2008) explored the hybrid genetic algorithm (HGA), or memetic algorithm, by combining a typical genetic algorithm with a localized search applied to each member of the population. In this work, simple tabu (Drezner, 2003) robust tabu (Taillard, 1991)and modified robust tabu were explored as the localized search tool. Modified robust tabu search was found to be the best improvement method. Similar results were found by Misevicius (2004) in the introduction of the improved hybrid genetic algorithm (IHGA) that used a variation of robust tabu search as a localized improvement method. Misevicus & Guogis (2012) considered variants of the HGA in which children were formed by either a single parent or two parents. The crossover method involving two parents was found to offer slightly better performance. Drezner & Misevicius (2013) suggest a differential improvement variation of the HGA in which a predetermined number of randomly sampled population members be improved prior to the crossover portion of the HGA. While the method showed improved results for large problems, improvement with respect to both time and accuracy for small ($n<50$) instances was small.

For further information on the QAP the reader is directed to the text by Burkard et al (2009), or the review paper by Loiola, et al. (2006). Additional information about the QAP and historic problems are available online through QAPLIB (Hahn & Anjos, 2011).

2.1.2 The Traveling Salesman Problem

The Traveling Salesman Problem is a well-known problem in operational research in which it is necessary to find the shortest possible route for a Salesman to travel through *n* locations, starting and ending at the same node. Applegate et al. (2006) and Punnen (2007)describe the messenger problem of Menger (1932)as a variation of the TSP in which it is necessary to visit all of the cities in a group, but it is not necessary to return to the initial location. Applegate et al (2006) explains that the messenger problem can be modelled as a TSP by adding an additional node at the start and end of the route with a distance of zero to all other nodes.

Many methods have been proposed to solve the TSP and can be reviewed in general textbooks (eg (Applegate, et al., 2006) (Gutin & Punnen, 2007)). Two popular methods, the nearest neighbour heuristic and dynamic programming are described below.

2.1.2.1 Nearest Neighbour Heuristic

The nearest neighbour heuristic (NNR) is a well-known greedy construction heuristic that is used to form initial solutions to the TSP. The NNR method begins by considering all points to be visited, such as punch hole locations, as nodes in a complete graph. A starting node is selected at random. From the starting node, the arc leaving with the lowest arc weight is selected and the node at the opposite end becomes the new active node. At the second node, all arcs connecting to unvisited nodes are considered and the lowest arc weight is selected again. The process is continued until all nodes have been

visited. The arc connecting the final node to the starting node is then added to the sum of all selected arc weights, and the result is an approximation of the shortest route through all of the points.

2.1.2.2 Dynamic Programming for TSP

For small problems, the dynamic programming (DP) method proposed separately by both Held & Karp (1962) and Bellman (1962)was proposed and has been implemented in several variations, including the approach outlined by Smith (1991). Applegate et al. (2006) maintains that the guarantee of $O(n^2 2^n)$ remains the fastest for a general solution TSP algorithm. Though effective at finding an exact method, the DP approach is often overlooked because of its capacity limitations. Bellman (1962) suggested a feasibility limit of 17 cities for the DP algorithm based on available computer memory at the time. Advances in computer hardware have increased this limit, but it is still too low for many instances of the TSP. For most industrial applications the number of processes performed on a single part or assembly is well within the capacity limitations of the DP method making it suitable for the present application.

2.2 History of Turret Punch Optimization

2.2.1 Reductions in turret rotation time

As shown in Figure 2-3, it is necessary for a turret to rotate between tool locations in order to process a part. Walas & Askin (1984) are often credited with being the first attempt at minimizing turret rotation time. In their approach, both the turret rotation time

and sheet travel time are considered in the final solution by treating the optimization as two subproblems solved iteratively. Sheet travel time was minimized by modelling the sequence of required punch locations as a TSP and solving to find the least amount of sheet travel necessary. The TSP was solved using the NNR heuristic, and alternate start points were considered during each iteration of the solution. The turret layout subproblem was modelled as a QAP and solved using the Vollmann, Nugent and Zartler (1968) (VNZ) procedure. In the QAP formulation, the angular distance between punches was used as the distance component of the problem and the number of changes between tools, as found during solution of the TSP for the current iteration, was used as the weighting between tools. Since the turret considered was capable of bidirectional rotation, the angular distance considered was the shortest angular movement between tools. The proposed algorithm was found to decrease the cycle time for the actual parts, and produce results with an average time within approximately 12% of the minimum estimated by a lower bound.

Lin (1998) commented on the work of Walas & Askin (1984) and presented an exact algorithm that could be used to find the global minimum. Lin acknowledges that the heuristics are computationally more efficient, but presents the exact approach as a means of assessing the performance of the heuristic since it allows the heuristic to be compared to the true minimum.

Nayanzin & Romanov (1988) considered the turret rotation time for a small turret producing a single part with only one direction of rotation. Using an Euler orientated graph method, the authors present a technique that can be used to optimize tool positions within a turret. In the paper, the method is not tested on actual or simulated parts, and no summary data of the improvement achievable is provided.

Like Walas & Askin (1984), Kumar & Veeramani (1995) minimize the total processing time of a sheet by treating it as coupled sub-problems where sub-problem one is the creation of an optimal turret configuration and sub-problem two is the creation of an optimal hit sequence. Both sub-problems are solved using a paired-interchange technique, but the solution of the sub-problems is combined in three different approaches; an iterative hold-one optimize-other heuristic (IHO), a coupled simulated annealing heuristic (CSA), and a substitution heuristic (SUB). The three heuristics were tested on sample data sets that used total punch hits ranging from 40-200 and number of tool types ranging from 10-40. The performance of the heuristics were measured based on the percentage improvement in the solutions between a random starting configuration and the final result. It was found that the SUB heuristic produced the best results with a 63-76% improvement. In comparison CSA yielded an improvement range of 52-60% and IHO delivered improvements of only 25-40%. When CPU time was considered, the worst case sample (40 tools, 200 hits) required a computation time of close to 0s for IHO, 100s for CSA and 1800s for SUB. Because of the substantial time requirement for SUB, the

authors recommended CSA as the best overall solution, providing reasonable improvement within an acceptable amount of CPU time.

Both Walas & Askin (1984) and Kumar & Veeramani (1995) were able to demonstrate improvements in the productivity of a turret punch using pairwise interchange methods for minimizing the turret rotation time. In a study of all CNC tool changer processes (i.e. lathes, mills and other machining centres in addition to punches), Dereli & Filiz (2000) modelled the tool changer as a TSP, and used a genetic algorithm (GA) to minimize the total indexing time for a single part program. Results of their approach for two specific samples were compared to solutions generated by a group of 10 average technicians. No attempts to apply their results to a larger group of random samples were made. There were also no comparisons made to existing algorithms, such as the VNZ algorithm that had been used by Walas & Askin (1984).

In a similar fashion, Krishna & Rao (2006) modelled the problem as a TSP and used an ant colony algorithm (ACA) to minimize the turret rotation time of a tool magazine. Like Dereli & Filiz (2000), this work was targeted at tool magazines used on all CNC machine tools, and was not specifically limited to the case of the turret punch. The algorithm was tested using a machined part similar to the one used by Dereli & Filiz (2000), but no comparisons of the result quality between GA and ACA were made. In a comparison between genetic algorithm GA and ACA processing time, it was found that the ACA was able to find the minimized solution in 14s, 12s faster than was required using GA. Since

only one example is provided, the expected improvement over GA cannot be accurately assessed.

All of the above studies have focussed on the optimisation of machine movements for a single part or, in the case of Kumar and Veeramani (1995), a single combination of parts. In practice, an increasing number of companies are capitalizing on the flexible nature of turret punches, and using them to process dynamically nested sheets of parts. A dynamically nested sheet consists of a combination of parts based on the current demand load that has been arranged on the sheet in a way that allows scrap metal to be minimized. Since demand is always changing, it would be rare for two successive sheets to be identical. Using any of the above approaches would require the turret to be reconfigured before each sheet. The time required for this would be greater than the time savings achieved by optimising the turret.

In an investigation of optimization for diverse product requirements, Niemi (2003) used simulated production schedules and simulated parts programs to investigate the benefits of optimizing the turret configuration for an entire production batch. Although Niemi applied his approach to both buffered (Niemi, 2004) and unbuffered (Niemi, 2003) rotating tool holders, the application to the unbuffered design is most relevant to the case of the turret punch. In his approach, Niemi modelled the problem as a QAP where the distance between stations was measured in angular rotation and the weighting between tools was based on the number of changes between tools i and j (and vice versa) in a

simulated production schedule. In Niemi's simulation, NC programs were created as a randomized sequence of 50 tool changes, where the tool changes were weighted such that 24% of the tools represented 80% of the tool changes. Each NC program was then repeated a random number of times based on an exponential distribution with a mean of 10 repetitions. NC programs were then combined into simulated production schedules, with each schedule containing 1000-4000 tool changes. The turret configuration was optimised for each simulated production schedule using the CRAFT heuristic ((Armour & Buffa, 1963), (Buffa, et al., 1964)), and improvement was measured based on the decrease in wait time achieved between the initial random configuration and the final optimised one. It was found that optimisation is most beneficial when the magazine is large and when the tool usage frequency distribution is non-uniform.

Niemi (2003) is the only work to investigate the effects of optimizing a turret for the production of a combined workload, but it does not investigate the effect of changes in the production schedule on the efficiency of the turret. It has been pointed out in the literature that turret setup time can be very time consuming, and if many changes to the turret are required, the setup time can dominate the non-productive time on the machine (e.g. (Marvizadeh & Choobineh, 2013)). In industry, many companies have taken the reduction in punch setup time to an extreme and operate their machines using standard turrets that never vary in their configuration (e.g. (Lowry, 2011)). While Niemi (2003) is successful in demonstrating that a turret can be optimized for a specific production schedule, it has not been shown that a turret can be optimized for a series of varying

production schedules producing the same family of parts as would be necessary to support the implementation of an optimized standard turret.

2.2.2 Influence of Turret Layout on Tool Path Planning

Kumar and Veeramani (1995) described sheet travel time as the second major contributor to inter-hit delay. The reduction in sheet travel time has been studied many times as the optimization of tool path planning for turret punches. As demonstrated by Walas & Askin (1984) and Kumar & Veeramani (1995), the tool path and turret configuration are highly interrelated. A further understanding of the body of research involving tool-path planning allows insight into how a turret configuration can be optimized for nesting.

Research in tool path performance generally focusses on measuring the relative performance of TSP solution heuristics and the practicality of introducing tool precedence constraints. Table 2-1 summarizes the methods used to model the problem and the approaches used for solution. As indicated by the number of columns, a wide range of TSP solution heuristics have been applied. However, it is interesting to note that despite the wide range of solution heuristics applied, the TSP model is by far the most prevalent approach followed.

The solution approaches in Table 2-1 can be further categorized based on which of three part processing assumptions have been adopted: single part programs, tool grouping and tool precedence. It should be noted that these assumptions are not mutually exclusive. In many cases it is assumed that only one single part program is being optimized. This

assumption eliminates the influences of additional programs from the model. Tool grouping is the practice of combining all holes of the same type into a single group, and processing the entire group before proceeding to change the tool. The assumption of tool grouping is often used to simplify the problem by eliminating the decision of when tool changes should occur. Tool precedence refers to the need to process a single tool either before or after another one. For example, in the process of nests, it is often necessary to use a parting or nibbling tool to separate the component pieces from the sheet, as is described by Veeramani & Kumar (1998). This part must be the last tool used in the nest because the parts cannot be further processed after they have been removed from the sheet.

From Table 2-1, it can be seen that the practice of tool grouping is quite prevalent in the field. This trend is the result of a common assumption that tool change time is longer than sheet transfer time between punching locations. A common observation is that removing the tool grouping requirement has been shown to improve the total time required to execute a tool path (e.g. (Svestka, et al., 1981), (Svestka, 1990) , (Cho & Lee, 1993), (Roychoudhury & Muth, 1995)). This observation means that in many cases, the fastest tool path does not punch all holes of the same shape in sequence before changing tools. Instead, dissimilar holes are punched as required in order to complete the part faster. Chauny et al (1987)also came to this conclusion, but identified that changing tools prior to completing a group was most beneficial when the number of tools was high and turret

Table 2-1 Summary of Tool Path Planning Approaches

Source	Model Assumptions			Solution Techniques Used															
	Single Part Program	Tool grouping required	Tool Precedence considered	Assignment base heuristic (GIL)	Farthest Insertion	Nearest Insertion	Nearest Neighbour	Nearest Neighbour with subsequent 2-opt	Nearest Neighbour within tool	Random Configuration with subsequent 2-opt	Random Insertion	Random Insertion with subsequent 2-opt	Closest Unvisited City (CUC)	Manufacturer Proprietary Algorithm	HIOPT, HIMIN,HISUM	3-D Space Filling Curve	Simulated Annealing	Ant Colony	Genetic Algorithm
(Svestka, et al., 1981)	Y	N	N	Y									Y						
(Svestka, 1990)	Y	N	Y	Y									Y						
(Shin, et al., 1990)	Y	Y	Y												Y				
(Chauny, et al., 1987)	Y	N	N													Y			
(Cho & Lee, 1993)	Y	N	Y														Y		
(Roychoudhury & Muth, 1995)	Y	N	N		Y	Y	Y	Y	Y	Y	Y	Y							
(Zhong, et al., 2014)	Y	Y	N																Y
(Ghaiebi & Solimanpur, 2007)	Y	Y	Y															Y	
(Yang, et al., 2011)	Y	Y	Y		Y														

acceleration was fast. These are the conditions that occur when the tool change time does not dominate the sheet transfer time.

For a turret processing nested parts, a similar effect could be achieved by arranging the turret in a way that the tools that are most frequently used together are positioned as close as possible to one another. The tool path planning aspect of the problem could then be a matter of selecting a sequence of tool use that allows for the close proximity of tools to be exploited.

2.3 Modelling Approaches in Similar Fields

From an understanding of nesting and turret punch operation, it can be hypothesized that a heuristic for developing an efficient turret would:

1. determine which tools are used together frequently.
2. allow for flexibility in sequencing the order in which tools are used.
3. consider the randomness that exists in the creation of nests from a production schedule of parts.

These elements are common to three different fields of study outside of turret punch optimization; surface mounted device (SMD) placement, facility layout in cellular manufacturing and the stochastic facility layout problem (SFLP)

2.3.1 SMD Placement

In the production of printed circuit boards (PCB), a robotically controlled head is used to retrieve a component from a part supply location, move it to the installation location on the board and then install the component. This process is generally similar to the turret punch optimization problem if the supply location is considered to be equivalent to a tool shape in the turret, the component installation location is the same as a hole location and the act of installation is likened to punching.

Ayob & Kendall (2008) present a detailed classification of machine types. Using the above comparison, it can be seen that the turret punch machine is particularly similar to the turret-type and the sequential pick and place machines. Although similar in nature, the other classes of machines presented by Ayob & Kendall (2008)are complicated by additional features such as multiple independent delivery heads, multiple stations or the simultaneous installation of multiple components. The extra complexity in these devices makes their machine specific mathematical models irrelevant to the current application. As a framework for describing this field, consider two members of the set of the eight PCB production sub-problems presented by Crama, et al (2002).

1. feeder-allocation (sub-problem 5 by Crama, et al.).
2. component sequencing (sub-problem 6 by Crama, et al.).

In the current research, the sub-problem of feeder allocation is of greatest interest since it is directly applicable to turret layout problem. While component sequencing is not of direct interest in the current research, the combination of the above sub-problems is

highly relevant since the combined solution represents the simultaneous optimization of the feeder layout and feeder sequence under the condition of flexible feeder sequence. This combination is analogous to the optimization of turret tool usage under the condition of flexible tool usage sequence.

The literature shows that these two sub-problems have been studied together many times, using many different approaches. Primarily, these approaches have considered the production of only a single PCB. For the purposes of further classification, they will be categorized as hierarchical, iterative and simultaneous.

2.3.1.1 Hierarchical Approaches

Hierarchical approaches are characterized by the practise of selecting one sub-problem and finding a solution to the sub-problem without consideration of other criteria. The other sub-problems are solved subsequently.

Drezner & Nof (1984) discussed both the assignment of components to bin locations and the sequencing of component installation for the general case of pick and place robots. For a pick and place operation, the head must move from the pick location to the installation location after every pick. The authors exploit this characteristic sequencing of the machine to reduce the feeder assignment problem to a series of single facility location problems. For the case of multiple products assembled from the same configuration, the authors introduce the minimax assignment problem (MAP) which considers the travel time between bin locations and install locations for all components on all assemblies and

seeks to minimize the longest sequence of component installations for any single assembly. After the establishment of bin locations, component sequencing was solved by modelling it as a TSP.

Ball & Magazine (1988) similarly identified movements from the feeder to the installation point as being required movements. In contrast to Drezner & Nof (1984) who used required movements to improve feeder allocation, Ball & Magazine used the minimization of the remaining non-required movements as the basis for the minimization of the component sequence.

De Souza & Lijun (1995) presented a knowledge based system for a chip-shooter style of machine. The allocation of feeder locations was determined by a heuristic that first arranged the components into groups based on component size since turret rotation speed is limited by the size of the largest loaded component. Within each group, the components with the highest frequency of use were grouped together. Sequencing was achieved by using a TSP model that limited available nodes to those that were associated with a feeder location within one feeder slot of the current feeder position.

Kumar & Li (1995) developed a quadratic integer programming model to describe a pick and place machine in which assembly sequences were grouped based on the robot gripping tool required for handling since the time required to change the gripping tool was prohibitively large to allow frequent interchanges. An upper bound to the integer program was shown to consist of two sub-problems; a minimum weight matching

problem, which was first solved to assign feeder locations, and a TSP which was used to determine placement sequences. Results were compared to lower bound solutions found by solving the integer programming model and the gap was found to be less than 2% for instances with 40 or fewer components. The gap was shown to grow to more than 9% when 60 components were considered, although the authors suggested that the gap could be reduced by using more sub-tour elimination constraints in the solution of the TSP.

Yeo, et al. (1996) proposed a rule-based system. Feeder assignments were determined through a proposed one-pitch-incremental-feeder heuristic which seeks to exploit placement movements that can be completed within the time required to index the feeder tray by one position. This concept is similar to the objective that De Souza & Lijun (1995)were trying to achieve by limiting their TSP model to consider components only within one feeder slot of the current location. Component placement sequencing was modelled as a TSP and solved using an NNR heuristic.

2.3.1.2 Iterative Approaches

Iterative approaches are generally similar to hierarchical approaches in that each sub-problem is solved independently. Unlike hierarchical approaches, iterative approaches cycle through sub-problems to determine if any improvements can be made after the original solutions to all sub-problems have been found. Iterations occur for either a set number of cycles or until no further improvements can be found.

Leipälä and Nevalainen (1989) first used an iterative approach to optimize the performance of a pick-and-place machine producing a single style of PCB. In their method, the component placement sequence was modelled as a TSP and the feeder assignments were modelled as a QAP. After an initial construction of a placement sequence and a feeder arrangement, improvements were found by performing a two-opt improvement on the feeder arrangement and then checking for any improvements in the component sequence. Of several proposed feeder arrangement construction approaches, the authors hypothesized that a minimum spanning tree (MST) based approach held the greatest potential.

Foulds & Hamacher (1993) considered the pick-and-place machine. The authors assumed that all components of the same type would be installed in a sequence, much like the tool grouping approach that is common in turret punch tool path planning. This assumption, combined with the mechanics of the machine type, allows the feeder assignment problem to be reduced to a series of one-facility location problems. As with Leipälä and Nevalainen (1989), the placement sequence was modelled as a TSP. The authors noted that the iteration of these approaches would lead to a local optimum. Finding a global optimum would require the incorporation of a tool with the capability of escaping a local minimum.

Bard, et al. (1994) extended the work of Leipälä and Nevalainen by combining the sub-problem of feeder assignment with the additional sub-problem of component retrieval

and modelling them combined as a quadratic integer program. The additional sub-problems were solved simultaneously using a dynamic programming model. To verify their solutions Bard, et al. used a branch and bound model that branched off of feeder allocation. The authors observed that each given feeder assignment corresponded with an optimal placement sequence allowing the branch and bound model to use only the feeder configuration data to represent a complete state.

A computational improvement to Leipälä and Nevalainen (1989) was suggested by Sohn & Park (1996) to calculate only the incremental changes in the sub-problems resulting from iterative 2-opt improvements instead of solving the complete objective function.

Ellis et al. (2001) further adapted the technique of Leipälä and Nevalainen to chip shooter type machines by introducing a construction heuristic that grouped components by size in order to allow for maximum turret rotation speeds. The objective function was also adapted to reflect the more complicated time calculation necessary to describe the concurrent motions in the chip-shooter machine. Iterative improvements were performed through the use of a two-opt algorithm.

Alkaya & Duman (2015) provide a further refinement on the efforts of Ellis et al. by replacing the TSP portion of the Leipälä and Nevalainen model with the sequence dependent traveling salesman problem (SDTSP). In the traditional TSP, the cost of traveling between locations is considered to be fixed. The SDTSP allows the interlocation cost to change depending on the current conditions. In their research, Alkaya & Duman

use this characteristic of the SDTSP to capture variations that occur in turret rotation speed based on the components that are currently held in the turret. Since large components require slower rotation, the content of the turret can significantly change the total assembly time for the PCB.

Among the iterative approaches, it is clear that the approach of solving the component sequence as a TSP and solving the feeder layout as a QAP, first proposed by Leipälä and Nevalainen (1989), has been the most prevalent. Its inclusion in the recent work of Alkaya and Duman (2015) reaffirms its continued relevance to the combined sub-problems.

2.3.1.3 Simultaneous Approaches

Unlike iterative and simultaneous approaches which consider sub-problems independently, simultaneous approaches attempt to find solutions for all sub-problems concurrently. The techniques used are often metaheuristic in nature.

Leu, et al. (1993) developed a genetic algorithm (GA) that could be used for several types of PCB assembly machines. Most notably, they included a model for the chip-shooter type machine, for which they named the problem the moving-board-with-time-delay (MBTD) problem to describe the potential for delay to occur by whichever of the concurrent operations required the greatest time. The chromosome in the GA consisted of two links. The first link provided a sequence for component insertion. The second link described the assignment of components to feeder locations.

Ong & Khoo (1999) used a GA that also featured a two-link chromosome structure to improve the performance of a pick-and-place type machine. Their approach allowed for a component to be stored in either single or multiple feeder locations. Compared to the performance of the pick –and-place heuristic of Leu, et al. (1993), the minimum travel distance necessary was decreased by 7.4%.

Khoo & Loh (2000) used a similar GA approach to model a chip shooter type of machine. Their model included rules requiring identical components to be placed in a single path and also to ensure that the component sequence progressed from smallest components to largest.

Ong & Tan (2002)addressed the MBTD problem using a hybrid genetic algorithm approach. Like Leu, et al. (1993)and Khoo & Loh (2000), Ong & Tan represented both the feeder assignment and the component sequence in the GA chromosome. As a difference, Ong & Tan only applied the genetic operators to one portion of the chromosome at a time. First, component sequence portions of two chromosomes would be joined to form an offspring and the feeder portion of the chromosome from parent 2 would be passed directly to the offspring. Second, the feeder portions of the chromosomes would be joined using genetic operators and the component sequence information was passed directly to the offspring. The most viable offspring were selected to participate in the next generation. The results of this approach were compared to those

of Leu, et al. (1993) and found to show significant improvement when compared over equal numbers of iterations.

Ho & Ji (2003) presented a hybrid genetic algorithm using the same two-link chromosome. After the application of typical genetic operators to form new offspring, three improvement heuristics were applied to the new members of the population. First, the NNR heuristic was applied to the component sequence portion of the chromosome. Second, a 2-opt local search was applied to the feeder configuration portion. Third, a new heuristic described as the iterated swap procedure (ISP) was applied to the component sequence portion of the chromosome. Using the sample problem provided by Leu, et al. (1993), the HGA method was found to produce a result of 26s after 323 iterations which was a significant improvement over the result of 51.5s found after 1750 iterations by Leu, et al.. Ho & Ji (2004) used a similar approach to study the pick-and-place style of machine. Similar improvements in results were achieved. Ho & Ji (2006) extended the work of Ho & Ji (2003) by incorporating the component retrieval problem. They found that by allowing a single component to be drawn from multiple feeder locations, the performance of the machine could be improved.

Chyu & Chang (2008) presented a hybrid genetic algorithm with a new approach to the GA chromosome. In their structure, components were grouped into categories by weight so that the turret rotation speed could be maximized based on the size of the largest component currently loaded in the turret. Each weight category was given a link within

the chromosome, and the order of components within the link represented the arrangement of the components in the feeder. The component placement sequence was not stored in the chromosome, and was found using a nearest neighbour solution with two-opt improvement during the decoding process. The effect of duplicate feeder locations was also studied.

Kumar & Luo (2003) found feeder assignments and placement sequence simultaneously by modelling the chip-shooter machine as a TSP. To make the TSP model work, it was necessary to assume that all components of the same type would be installed in a sequence before the feeder movement occurred. It was also assumed that when feeder movement was necessary, it was much slower than the concurrent board movement and turret rotations. Final solution of the problem was found using UKTSP, a heuristic solver dedicated to the TSP.

To better understand the suitability of solution metaheuristics for PCB assembly, Nelson & Wille (1995) and Duman & Or (2007) separately compared the performance of different methods. Duman & Or (2007) focussed their efforts on the QAP. In a comparison of alt-op, TS, SA and guided evolutionary simulated annealing (GESA) they found that SA yielded the best results. It is important to note that neither a plain GA nor an IHGA using TS for local search were considered in the study. The finding of SA as the preferred approach, particularly over TS, is inconsistent with the findings of Taillard (1995) and Misevicius (2012).

In contrast, Nelson & Wille (1995) studied the component placement sequence problem and compared SA, EP and GA. They also found that the best results were produced by SA but only if allowed to run for a long number of iterations. The EP and GA techniques were also found to be effective.

The solution methods describe above provide useful insight into solution structures that can be used for a problem consisting of both layout and sequencing components. While some modelling approaches used in SMD placement, such as consideration of component size, are not important for turret punch layout, it is interesting to see that a number of solution approaches are common between the two problems. For example, in SMD placement, several authors have assumed that all components of a single type will be placed in succession. This is similar to the tool grouping approach often used in turret punch tool path planning. The use of Chebyshev distance metrics to capture the greatest of turret rotation/feeder movement, X-translation and Y-translation is also common between the two problems.

The approaches discussed above have focussed on the optimisation of machine setup for the production of a single part only. This tendency is also similar to the existing body of literature covering turret punch performance. Of more interest to the current research problem are the handful of works that have studied the assignment of components to feeders when multiple types of PCBs must be assembled using a single setup.

2.3.1.4 Production of Multiple Boards

Most research into process planning for PCB assembly is focussed on the production of single assemblies with known installation locations. In the problem of nested parts, it is necessary for a machine to be able to produce many parts without any setup changes. Within the body of PCB literature, there have been a few approaches that have considered the feeder assignment problem for multiple assemblies using a single machine configuration. Usually the interest arises from the group setup, or family setup, approach that groups similar boards for production on a single machine to decrease setup time.

A common idea in assigning feeder locations for multiple parts is to determine which components are used the most frequently across all of the products to be produced, and then assign these components the most desirable positions in the feeder, typically nearest to the pick-up location or in the middle of the feeder rack. This method results in the most frequently used components being placed in close proximity to each other, and also adjacent to a location that is easily accessed by the component retrieval mechanism. The expected result is that the consolidation of high frequency components will decrease the amount of travel that the feeder rack must complete. Variations of this approach were implemented by Carmon, et al. (1989) and Su & Srihari (1996).

Dikos, et al. (1997) also considered a weighted sum of components used across all components, but further included the distance between feeder locations required for the production of each PCB. The resulting objective function minimized the feeder distance

travelled during the production of all included PCB's. The problem was solved through a GA that was designed to allow feeders of multiple widths to change positions as required. This approach has limited applicability to the current research problem since it requires the sequence of component installations to be known prior to the assignment of components to the feeder.

Another approach to feeder optimization is to maximize the number of so called 'free' feeder movements. These are movements, typically on one slot in distance that can be completed faster than the required movement of the X-Y table carrying the PCB. The move is 'free' in the sense that the required time does not change the makespan of the product since the movement was dominated by the other concurrent movements. Crama, et al. (1997) presented a heuristic to maximize these types of movements before the determination of component sequence by using a fast estimation technique. In their approach, a TSP was solved for all component installation locations for each pair of components and stored for use in the estimation process. Since the movement between adjacent feeder slots was 'free' the authors reasoned that the feeder could move between the two adjacent positions without causing a delay. For a set of n component types, this resulted in an n x n matrix of distances. After an arbitrary assignment of feeders to rack locations, the feeder arrangement was evaluated using the authors' estimation approach. The arrangement was then improved and evaluated iteratively until convergence occurred. This approach was later applied in Klomp, et al. (2000) and Crama, et al.

(2002). While the estimation technique used is interesting, this method is not transferable to the nested parts problem because the punch hole locations are not known in advance.

Wu & Ji (2010) modelled a pick-and-place type of machine using a non-linear integer model and solved using CPLEX, a linear programming solver. The model was additionally decomposed into two sections; one for feeder arrangement and one for component installation sequence. These two sub-problems were solved iteratively and compared to the results from the exact solution. The iterative solution was faster, but was found to be less capable of finding the optimal solution. Wu, et al. (2009) solved the same problem by using a multi-link genetic algorithm in which the first link in the chromosome represented the feeder sequence and each additional link represented the component placement sequence for one of the included assemblies. Results from this approach were near optimal, but the CPU time was 58 minutes for a relatively small problem involving only 20 types of components and four different board types. This is much smaller than the problem scale that must be addressed in the current research.

A two stage ant colony algorithm (ACA) for group setup was proposed by Chen & Chyu (2002). Before implementing their approach, the placement locations of all boards were superimposed on one another to create a single entity showing all locations, which they refer to as a composite board. The first stage of the ant-colony arranges the feeder sequence for the composite board. In the evaluation of each feeder sequence, a second

stage ACA is applied to find a good placement sequence. The overall results is found from the combination of the two searches.

The concept of neighbourhood strength was introduced by Grunow, et al. (2004) to describe the number of connections between components. To arrive at neighbourhood strength, a complete graph was constructed in which each installation point became a node and all nodes were connected with an arc weight equivalent to the minimum time (turret rotation, x-movement or y-movement) required to move between the two points. A minimum spanning tree (MST) was then found to connect the nodes. The neighbourhood strength was defined as the number of arcs connecting any pair of component types within the graph. If each component was only used once, the neighbourhood strength would be one for all pairs connected in the MST. An increased neighbourhood score only occurs when components are used multiple times within a part and the same pair of component types is connected at different locations within the MST.

For the production of a family of parts, the neighbourhood strength for all PCBs to be produced was combined through volume-weighted addition into a single neighbourhood strength matrix. This matrix was then used to populate the feeder by placing the components with the highest neighbourhood strength closest together. After the placement of the initial pair, the component with the strongest neighbourhood strength between itself and either member of the original pair was then assigned an adjacent location. This process was repeated until all components had been placed. Improvement

opportunities were then sought out using a two-opt neighbourhood search approach. This heuristic was also applied in Yilmaz, et al. (2007) and Yilmaz (Yilmaz, 2008).

The concept of using neighbourhood strength as a basis of feeder arrangement is an interesting improvement on the frequency based approach. Since the ultimate goal is to minimize feeder movement, it is logical to locate parts expected to be regularly used in succession to be located near one another. Like most of the approaches in this section, the neighbourhood search technique relies on previous knowledge of where the component locations are going to be. In a nested part assembly model, the locations are not known since any part could appear anywhere on the sheet and in any orientation.

In the PCB production problem the sequence of visiting component locations is used to determine the locations for component storage. This is similar to the operation of a turret punch, but in developing a heuristic for an efficient turret layout for the production of nested parts, we are more interested in using the sequence of component usage to determine locations for component storage.

2.3.2 Flexibility in Cellular Manufacturing

Cellular manufacturing is the application of group technology to manufacturing environments in an attempt to combine the benefits of continuous flow manufacturing with the flexibility of a job shop. Cellular manufacturing can be divided into three critical steps: cell formation, cell layout and intracellular layout (Chang, et al., 2013). Within cellular manufacturing, examples addressing alternative sequences can most frequently

be drawn from the cell formation problem or from research spanning cell formation in combination with other aspects of cellular manufacturing.

As previously noted, research into optimization of SMD placement machines for the production of multiple PCBs emerged from group setup or family setup approaches. These setup approaches are characteristic of cellular manufacturing and often emerge from the group formation process. For both the turret punch and the SMD placement machine, the placement of the punching tools, or feeders, is analogous to the arrangement of workstations in a cellular manufacturing environment. A review of cellular manufacturing layout problems involving flexible sequencing of operations can provide more insight into solution approaches.

In the context of cellular manufacturing, it is important to differentiate between sequence flexibility and routing flexibility. For a general punched part, the absence of precedence constraints allows a part with n operations to be manufactured in $n!$ unique operation sequences, where n is the number of operations (or punch tools) required, simply by reorganizing the order in which the operations occur. Sethi & Sethi (1990) included this property in their definition of routing flexibility, but it has been more specifically described as flexible sequencing (Lin & Solberg, 1991). Routing flexibility appears in the literature much more frequently than sequence flexibility, but typically refers to the ability of a part to have the same operation performed by multiple machines of the same type. Nonetheless, the mathematical handling of routing flexibility can provide insight

into methods that can be used for handling a finite number of combinations with the context of a larger integer programming model.

With flexible sequencing, the flow matrix can take on many different values depending on the particular permutation that is assigned to each part. While finding the best layout for a known flow matrix is a well-known problem, finding the flow matrix that produces the best layout is an additional challenge that is sometimes considered in the context of cellular manufacturing.

In cell formation, Kusiak (1987) introduced the generalized group technology concept as an improved method for finding part families and machine groups by considering alternative routings for each part. In this method, a part machine incidence matrix was constructed with each part being represented by multiple columns. Large negative numbers were used to prevent the similarity coefficient approach from identifying multiple routings for the same part as being similar.

More recent approaches to the cell formation problem have adopted an approach of considering each routing individually by applying a subscript to the part. For example, if part n has j possible routings, each combination of n_i where $i=1\ldots j$ is considered. This type of approach has been adapted to similarity coefficient techniques (Yin & Yasuda, 2002) Tabu Search (Chung, et al., 2011) Genetic Algorithm (Lee, et al., 1997) and grouping genetic algorithm (Vin & Delchambre, 2014). While the process of listing and enumerating all process routings and sequences has merit when the number of

alternatives is small, its suitability for the current problem is limited due to the high number of alternative routings that are possible. Consider that a part with a completely flexible sequence of n operations offers $n!$ different sequences. The consideration of m parts into a single cell than results in $(n!)^m$ combinations. For many industrial applications, the size of the problem quickly becomes excessively large for enumeration.

As an alternative to the complete enumeration approach, Zhao & Wu (2000)extended the GA based heuristic that encodes a chromosome to represent a grouping of machines into cells introduced by Venugopal & Narendran (1992) to include a second stage that incorporates alternative routings. Each chromosome was evaluated as the sum of the optimal routings for each component part. Optimal process paths and optimal routings were identified by considering a directed acyclic graph for each routing and selecting the routing and path resulting in the minimum cost for each part. Koşucuoğlu & Bilge (2012)used a similar GA method to solve the FMS loading problem with flexible routings, except a mixed integer model was used to find the minimum material handling costs given the layout specified by the chromosome.

In the processes described above, all alternative routings are considered during the evaluation of a proposed cell configuration. Rather than evaluate all flexible routing options, Caux et al. (2000) used a branch and bound technique to evaluate potential configurations produced by simulated annealing. By considering the lower limits on each branch, some branches did not need to be evaluated.

Nsakanda et al. (2006) identified that many of the methods used for solving the cell formation problem with alternative routings were not suited to the solution of large scale problems. As a solution, the authors presented a two stage heuristic that uses a genetic algorithm in which the cellular configuration of the machines is represented as a chromosomal string as the first stage. In the second stage, the total processing time for each chromosome was then evaluated by modelling it as a combination of P subproblems and solving each subproblem as a min cost network flow problem using Dantzig-Wolfe decomposition. In addition to solving the problem of alternative routings and alternative process paths as had been done in previous works, Nsakanda et al (2006)also allowed for the possibility of alternative operation sequences within a routing.

Zhao & Wu (2000)and Nsakanda et al (2006) both used a second stage as a means of evaluating the chromosome in the genetic algorithm. If these techniques were further adapted so that the result of the second stage was used to improve the chromosome, rather than just evaluate it, the approach would be very similar to the hybrid genetic algorithms presented by Misevicius (2004)and Drezner (2008).

2.4 Demand Variations in Facility Layout

The previous techniques used to develop layouts, either for turret punches, SMD placement machines or cellular manufacturing facilities mostly rely on known levels of demand. In reality these demand levels are stochastic. For a turret punch processing nested sheets, an additional element of variation is introduced since the content of a

single sheet is a randomly selected subset of the total demand which is in itself stochastic. The challenge of optimizing a layout for unknown demand has been studied as layout flexibility within the context of the facility layout problem.

In most cases, layout flexibility seeks to develop a layout that can operate efficiently when the *F* matrix is subjected to stochastic variation. When the concept was introduced by Shore & Tompkins (1980), the variation in *F* was presented as changes in demand that could occur based on market conditions. The study of flexible layouts has evolved since the work of Shore & Tompkins (1980) and, as described by Kulturel-Konak (2007), can be generally classified as the stochastic facility layout problem (SFLP). As an extension of the SFLP, Palekar, et al. (1992) considered a multiple time period problem in which each period is characterized by unique stochastic demands. This problem was referred to as the stochastic dynamic facility layout problem, of SDFLP. The dynamic facility layout problem (DFLP) is a related problem that seeks to reduce material handling costs, including the costs of reconfiguration, over several time periods where each period is characterized by unique levels of demand. A good review of this problem can be found in Balakrishnan & Cheng (1998). In the current research, only the SFLP and its variants are explored since the research problem has been simplified to address the demand for a part over a single time period.

2.4.1 Stability, Robustness and the Stochastic Facility Layout Problem (SFLP)

Robustness was introduced together with stability by Gupta and Rosenhead (1968) as measures of flexibility in investment plans. Their original definition for robustness was as the ratio of good end states that remain available to good end states that were originally considered. These ideas were expanded upon by Rosenhead et al. (1972).

Braglia et al. (2005) clarified the difference between robust and stable layouts. In their work, they defined that a stable layout is characterized by low *variation* in material handling cost when subjected to stochastic demand. In contrast, a robust layout is characterized by a low *average* material handling cost when subjected to the same demands. The performance of a stable layout is very similar to the risk minimized layout described by Krishnan et al. (2009)and modified by Jithavech & Krishnan (2010) to use a simulation based approach. Although some work (Benjaafar, 2002) has found that a stable layout improves performance by making material flows more consistent, the robust approach has been adopted much more frequently in the solution of the SFLP.

2.4.1.1 <u>Discrete Approaches to the SFLP</u>

Within the SFLP, the most pervasive approach to measuring robustness has centred on comparing the performance of different layouts under different operating conditions. A layout that is capable of performing well under alternative conditions is said to be robust. As will be shown below, a common approach has been to consider a discrete set of operating conditions that layouts should be compared over.

In perhaps the earliest application of this approach, Shore & Tompkins (1980) used ‘flexibility’ as a measure of layout quality and assessed it using a penalty method. Discrete combinations of potential demand levels were formed and assigned probabilistic weightings. The optimum layout for each scenario was found. The total penalty cost was derived by evaluating the operating cost of each candidate layout under each of the alternative operation conditions and weighting it by the probability of occurring. The single layout with the lowest penalty cost was deemed to be the most flexible.

The first use of robustness as an SFLP solution descriptor is believed to be Rosenblatt & Lee (1987). The authors defined a series of possible demand scenarios. A robustness was measured by comparing the performance of a candidate layout to the best possible layout for a given demand scenario. The preferred solution performed within X% of the best possible solution the highest number of times.

In many applications, the number of discrete demand combinations and discrete layouts to be considered is too large to allow the above-described discrete methods to be applied since the number of combinations to be considered is too great.

To reduce the number of combinations, Rosenblatt & Kropp (1992) present a method in which the flow matrix is developed as the weighted sum of all flow matrices for all considered demand states. The QAP is then solved using this weighted flow matrix to find a robust solution. The solution found using this method was compared to the penalty cost method of Shore & Tompkins (1980) and results were found to be equivalent.

Kouvelis et al. (1992) applied the Rosenblatt & Lee (1987) model of robustness to both the single period and multiperiod stochastic FLPs. In their approach, models within *p%* of optimal were found for each possible deterministic demand scenarios. Scenarios meeting the *p%* requirement for all scenarios were identified. Kouvelis et al described a multiplicity of good solutions and ultimately relaxed the p% criteria to be the best 400 solutions for each scenario. For larger problems the authors suggested that the flat topology of the QAP could be exploited to find near optimal candidate locations for each scenario quickly by using a suitable heuristic.

Kouvelis & Kiran (1991)use dominance and efficiency to solve single period and multi-period problems with alternative operation sequences and varying scenarios of product mix. In the proposed algorithm, layouts absolutely dominated by other layouts are not considered as eligible solutions. Layouts that are non-dominated, meaning they are sometimes better and sometimes worse than alternative candidates, are considered as efficient possible solutions. The solution offering the lowest cost relative to the best cost for all product mixes is selected as the best layout.

Despite the improvements offered by Kouvelis et al. (1992)and Rosenblatt & Kropp (1992), the necessity of defining discrete demand states is impractical for the scale of problem defined by the task of turret optimization.

2.4.1.2 Continuous Approaches to the SFLP

An important finding of the SFLP is the existence of an optimum solution. While the measurement approach varies between researchers, including ideas such as flexibility (e.g (Shore & Tompkins, 1980)) robustness (e.g. (Rosenblatt & Lee, 1987) (Norman & Smith, 2006)), and risk minimization (e.g. (Krishnan, et al., 2009) (Jithavech & Krishnan, 2010)), researchers agree that there is often a single layout that best satisfies the design objective when demand is stochastic.

Braglia et al. (2003) presented the Layout Cost Robustness Index (LCRI) that can be used to estimate the probability that a given layout has a better cost function than all alternative configurations. The model was developed using a single row machine layout problem with normally distributed stochastic demands. The LCRI model proposes that the cost of operating each layout curve can be characterized by a normal distribution. A similar continuous distribution curve is developed separately by Norman & Smith (2006).

While Norman & Smith (2006) used the distribution curve to compare costs of specific layouts over a range of expected demand, Braglia et al. (2003) collected the curves formed for the population of layouts to develop a normally distributed random variable that describes the mean costs of layout configurations. The probability that a layout will be more efficient than the other layouts described by the distribution is presented as:

$$LCRI = \varphi\left(\frac{\overline{M(l)} - M(l^*)}{\sqrt{\left[\overline{S(l)} + S(l^*)\right]}}\right)$$

$$0 \leq LCRI(l) \leq 1$$

where:

$$\overline{M(l)} = \frac{\sum_{l=1}^{N!} M(l)}{N!}$$

$$\overline{S(l)} = \frac{\sum_{l=1}^{N!} S(l)}{N!}$$

In this formulation, $\overline{M(l)}$ is the mean material handling cost of the entire population of layouts, and $M(l^*)$ is the mean layout cost associated with the specific layout being tested for robustness. Similarly, $\overline{S(l)}$ is the mean variance of layout cost for the population, and $S(l^*)$ is the variance specific to the layout being tested.

To demonstrate the strength of the measurement, Braglia et al. (2003) compared the results of their probabilistic approach to a total penalty approach and found that solutions with a high LCRI correlated strongly with solutions characterized by low total penalty costs when the layout problem robustness indicator, LPRI, suggested that a robust solution would work. LPRI was a separate indicator that was proposed to determine if a particular problem was better served by an agile or a robust solution approach.

Braglia et al. (2003)use the expected demand for an entire time period as the basis for measuring robustness, allowing material handling costs to vary with the stochastic nature of demand. In the current research demand is fixed and variation in material handling requirements is introduced by considering different nested combinations of the same set of parts.

2.5 Summary

There has been interest in optimizing the performance of turret punches for several decades. These efforts have focussed on either turret layout (e.g. (Niemi, 2004)), tool path planning, or both problems concurrently (e.g. (Walas & Askin, 1984), (Kumar & Veeramani, 1995)). However, these efforts have focussed primarily on the production of single parts and are not relevant to the modern practise of producing nested parts.

An interesting characteristic of sheet metal punching is that the sequence in which the punches can be performed is largely free of precedence constraints. This characteristic has been referred to before as sequence flexibility (1991) . Techniques for optimizing layout for products demonstrating sequence flexibility have been demonstrated in both cellular manufacturing and the optimization of SMD placement machines. Although a number of modelling techniques have been used, the most prevalent approaches have used a QAP model to develop a layout, and a TSP model to refine sequence. Solution techniques have also varied, but of the heuristic approaches, the GA and its variants have been the most popular. In addition, the GA, and specifically the IHGA, has been shown

separately to be an effective heuristic for solving the QAP when the QAP is derived from real life problems.

By randomly combining parts orders into nests based on competing factors such as demand levels, urgency of demand and minimization of sheet metal waste, the number and location of punch holes in each nest is variable. Within the FLP, the SFLP variant provides methods for developing layouts that are robust to changes in demand. The body of literature has also produced a technique for determining if a given layout is robust for a specified set of demand (Braglia, et al., 2003).

Elements from these related research areas have the potential to be recombined into a methodology that will produce turret layouts capable of producing randomly generated nests efficiently based on a single set of expected part demand levels.

CHAPTER 3: METHODS

3.1 Overview

As stated in the research objectives, the goal of this work is to develop a robust, standard turret configuration for use in a nesting environment. One easily understood approach to solving this problem would be to consider all possible nests, and then determine the layout configuration that can produce all nests using the least amount of turret rotations. Such an approach would be similar to the early techniques used in the solution of the SFLP.

For nests, even when only a handful of parts are produced, the number of nests that can be considered is far too large to reasonably consider enumeration. As an alternative, the current research considers a surrogate problem in which the objective is to minimize the total turret rotation required to produce the expected demand levels of all parts individually using a common turret configuration. Since the sequence of tool usage in the production of a nest has great flexibility, each part in the surrogate problem is allowed to have complete tool sequence flexibility when it is considered individually.

The methods used are presented in four sections. First, a mathematical formulation of the surrogate problem is presented. Second, assumptions used in the development of the solution method are explained. Third, a heuristic method for implementing the solution approach is developed by exploring and combining three progressive search approaches:

1. An iterative local search heuristic (LSH) applied to random start points

2. LSH applied to a modified neighbourhood strength start point (MNS)
3. Integration of LSH as the local search method in a hybrid genetic algorithm

Finally, a method for adapting the layout cost robustness indicator (LCRI) to a nesting simulation is described and applied in order to test the ability of the surrogate solution to achieve the research objectives.

3.2 Mathematical Formulation

$$min: Z\left(x_{ik}, x_{jl}, y_{pab}, y_{p(a+1)c}\right)$$

$$= \sum_{p=1}^{P} \sum_{a=1}^{s_p-1} \sum_{b=1}^{s_p} \sum_{c=1}^{s_p} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} v_p t_{pbi} y_{pab} y_{p(a+1)c} t_{pcj} d_{kl} x_{ik} x_{jl}$$

subject to:

(1) $$\sum_{a=1}^{s_p} y_{pab} = 1, \qquad p = 1, \dots, P, \qquad b = 1, \dots, s_p$$

(2) $$\sum_{b=1}^{s_p} y_{pab} = 1, \qquad p = 1, \dots, P, \qquad a = 1, \dots, s_p$$

(3) $$y_{pab} = \begin{cases} 1 & if\ tool\ a\ of\ part\ p\ is\ assigned\ to\ sequence\ b \\ 0 & otherwise \end{cases}$$

(4) $$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \dots, n$$

(5) $$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1, \dots, n$$

(6) $x_{ij} = \begin{cases} 1 & if\ tool\ i\ is\ assigned\ to\ turret\ location\ j \\ 0 & otherwise \end{cases}$

Indices

a – index identifying the tool required within part p.

b – index indicating the operation on part p to which tool a has been assigned

i – index indicating current tool

j – index indicating following tool

k – index indicating candidate turret tool holder location for current tool

l – index indicating candidate turret tool holder location for following tool

p – indicator of part number

Parameters

d_{kl} – distance between turret tool holder locations k and l

v_p – Expected production volume of part p

t_{pbi} - An initial assignment of tool i to sequence number b of part p. This variable is referred to as *base sequence* and it represents an arbitrarily established tool use sequence for part p to which permutations can be applied.

In the following discussion of the method, lower case letters are used to denote matrix elements. The corresponding upper case letters are used below to describe the matrix in its entirety.

$Z(X,Y)$ is the total amount of turret rotation required to produce the forecasted demand level of all parts using a single turret layout.

Constraints (1) and (2) ensure that Y_p is a permutation matrix representing the sequence of tool use that occurs within part p. As a permutation matrix, each tool can be assigned to only one tool use sequence position and each tool use sequence position can have only one tool. Constraint (3) restricts the elements of Y_p to binary values.

Constraints (4) and (5) ensure that X is a permutation matrix representing the assignment of tools to turret tool holder locations. Each tool can be assigned to only one tool holder location and each location can only hold one tool. Constraint (6) restricts the elements of X to binary values.

In comparison to the QAP, a subset of the above formulation replaces the typical representation of F. Consider:

$$f_{ij} = \sum_{p=1}^{P} \sum_{a=1}^{s_p-1} \sum_{b=1}^{s_p} \sum_{c=1}^{s_p} v_p t_{pbi} y_{pab} y_{p(a+1)c} t_{pcj}$$

In this structure, T_p is a rectangular matrix representing the assignment of tool use sequences to tools for part p. Y_p is a permutation matrix for the sequence of tool use that occurs within part p relative to the base sequence defined by T_p. The multiplication of y_{pab} x $y_{p(a+1)c}$ generates flow. If tool b is assigned to tool use sequence step a, and tool c is

assigned to tool use sequence step *(a+1)* the result is 1, or an indication that turret rotation occurs between tools *b* and *c* because they are positioned at successive steps in the tool use sequence. v_p represents the expected demand volume for part *p* and is applied as a weight factor to the entire flow matrix. When considered over the summation of p=1:P, the result is a weighted flow matrix that includes a unique sequence permutation for all parts. Each element in the flow matrix indicates the frequency of turret rotation between tools *i* and *j* during the production of all parts, *P*.

3.3 Solution Assumptions

We begin by noting two observances in the literature review. Firstly, in the study of turret punch tool path planning, several authors adopted an approach of tool grouping. Recall that this means that all holes of a single type are punched prior to changing tools. Secondly, within the study of SMD placement assembly, particularly the MBTD problem, a concept of free moves is often applied to describe moves in which sheet movement time dominates feeder movement time. Applying this concept to a turret punch means that turret indexing times that occur within the minimum possible table movement time have no impact on the overall cycle time.

3.3.1 Default tool path relies on tool grouping

To simplify the current problem, we assume that the default programming approach is to apply tool grouping techniques. In addition to this approach being justified by its prevalence in the literature, this approach is also necessary because the alternative,

determining an exact tool path for each randomly generated nest, would be computationally intractable. In practice, this assumption will cause tools that are frequently used in combination to be located close to one another and potentially allow for more free movements to occur.

3.3.2 No tool precedence constraints

For simplicity, we assume that there are no tool precedence constraints. It is recognized that these constraints exist in practice and approaches for handling these constraints will be discussed after the robustness of the method has been demonstrated.

3.4 A local search heuristic

For the current research problem, the complete solution space is very large since it encompasses all permutations of turret layouts, as well as all combinations of tool sequences for all parts. A heuristic method is required to search the solution space efficiently.

The proposed heuristic presents an iterative two step approach for solving the FLP under the condition of flexible tool use sequences, given an initial layout permutation. The concept of dividing the problem into QAP and TSP sections and solving iteratively was first proposed for the turret punch by Walas & Askin (1984) and was made popular for the SMD placement problem by Leipala & Nevalainen (1989).

In their solution approaches, both Walas & Askin (1984) and Leipala & Nevalainen (1989) not only considered the sequence of use, but also the location of the punched hole (or placed component) within the 2-D plane of the part. Considering the placement location in the part introduces the time travel of the sheet or board. The current model is different because instead of considering the sequence of tool use in the plane of the sheet, it considers only the sequence of tool use within the turret. In this way, the current research focusses on the performance of the turret with the expectation that improving turret efficiency will also allow for the efficient part processing that was explicitly found in the earlier works. This heuristic is used as local search heuristic and will be referred to as LSH.

In the first stage of the heuristic, the optimal operation sequence (i.e. order of tool use) is identified for each part using an initial layout configuration. The layout configuration can be selected either randomly, as is tested in this section, or strategically as will be demonstrated in section 3.5. In the second stage of the heuristic, the flows resulting from the optimal sequences are used as an input to the QAP, and a revised layout is achieved. Stages one and two are repeated iteratively until convergence is achieved or a predefined limit on iterations has been met. A template for this routine is shown in Figure 3-1. Details of the approach used and an example of its implementation are provided in Section 3.4.3.

```
function improve_layout(π,Distance,Costs)
        repeat
                for i:=1 to n_parts
                        F_i=optimal flow for part n
                        F_tot=F_tot+F_i
                end // for //
                if flows have converged then begin
                   π=Solve_QAP(F_tot,Distance, Costs)
                end //if//
        until flows have converged
        return π
end //improve_layout
```

Figure 3-1 Template for LSH heuristic

3.4.1 Stage One: Sequence optimization Through TSP model

The optimal tool use sequence for each part is found using the current layout. This method assumes that turret rotational speed is constant, allowing rotational distance to be a surrogate measure of processing time.

The tool use sequence is modelled as a Messenger problem (Menger, 1932) in which each machine is equivalent to a city in the classic definition of the problem and the travel cost between machines is equal to the distance between them. Solving the messenger problem then yields the operation sequence with the lowest possible travel distance, given a predetermined layout.

In the traditional TSP, the complete solution begins and ends at the same city. To model the messenger problem as a TSP, an additional tool is added to problem. The additional tool is used as the start and end points of a Hamiltonian cycle, but the distance from the additional tool to all other points is set equal to zero. This approach is described by Applegate, et al. (2006).

Since the number of unique tools used by any single part is usually low, the DP method (Bellman, 1962), (Held & Karp, 1962) is applied using the implementation technique presented by Smith (1991).

3.4.2 Stage Two: Layout Improvement through QAP solution

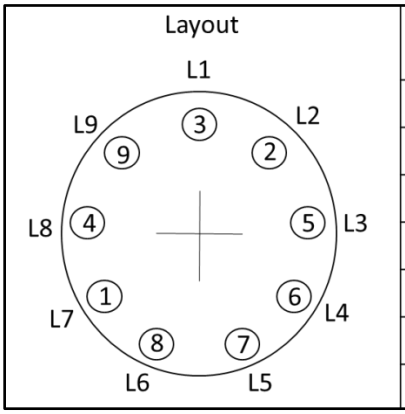
Since the tool holder locations are known, fixed and equal in size, the assignment of tools to locations can be achieved by modelling the problem and solving as a QAP. Solution of the QAP yields an improved layout based on the optimal sequences for each part.

Two QAP solution approaches were compared in the evaluation of the LSH. First, Gilmore-Lawler Bounds ((Lawler, 1963) (Gilmore, 1962)), implemented using the branch and bound method detailed by Francis and White (1974), has been selected as an exact approach. Second, IHGA (Misevicius, 2004)has been used as a heuristic method because of its demonstrated speed and accuracy on both real-life and simulated problems of size up to 256 locations.

3.4.3 Example of LSH Procedure

As a demonstration of the LSH procedure, a sample problem involving the solution of a turret layout for a part set consisting of six unique parts (Part Set 1 in Appendix A) and a randomly generated turret configuration.

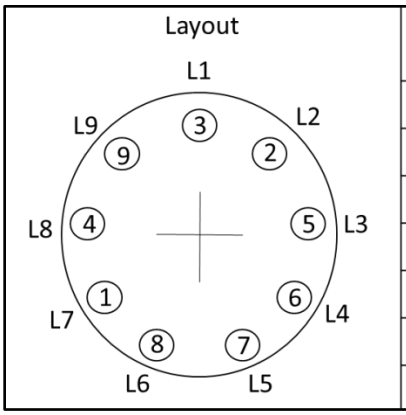
Step 1: The initial turret layout is randomly generated as shown in Figure 3-2. No tool sequences are assigned to any parts, and a measurement of total turret rotation cannot be found.



Part	Tool Use Sequence	Increments of Rotation
1	N/A	N/A
2	N/A	N/A
3	N/A	N/A
4	N/A	N/A
5	N/A	N/A
6	N/A	N/A
	Total	N/A

Figure 3-2 Step 1-Initial turret layout is established

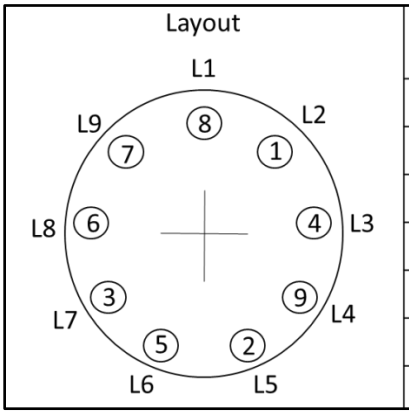
Step 2: An initial tool sequence is found for each part, as shown in Figure 3-3, and the total turret rotation required to produce each part is found. The total turret rotation required to produce all parts using the initial layout and the current tool use sequences is 23 increments of rotation.



Part	Tool Use Sequence	Increments of Rotation
1	4,9,2	3
2	8,3	4
3	7,5,3	4
4	3,6,7,1	6
5	7,8,1	2
6	8,7,6,2	4
	Total	23

Figure 3-3 Step 2-Initial tool sequences are determined

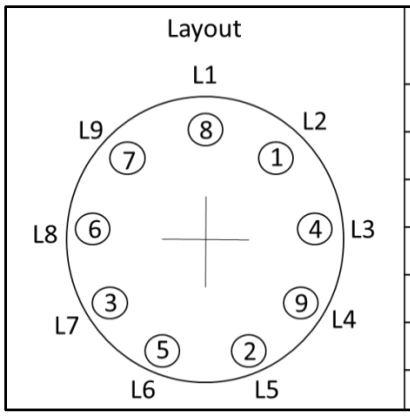
Step 3: A flow matrix, F, is developed for the QAP subproblem based on the current tool use sequences. The QAP is solved to find a new turret layout which reduces the total turret rotation to 20 increments. The improved turret layout is shown in Figure 3-4



Part	Tool Use Sequence	Increments of Rotation
1	4,9,2	**2**
2	8,3	**3**
3	7,5,3	4
4	3,6,7,1	**4**
5	7,8,1	2
6	8,7,6,2	**5**
	Total	20

Figure 3-4 Step 3 - Layout is revised base on initial tool use sequences

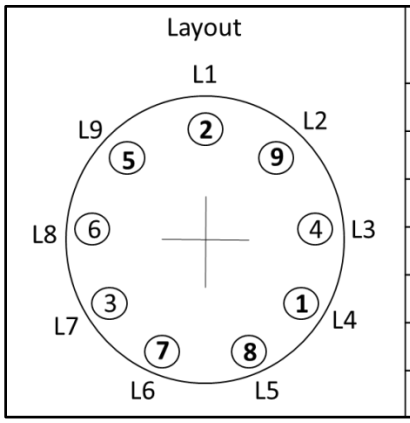
Step 4: Using the distances between tool holder locations in the revised layout, optimal tool use sequences are again found for each part. An improvement of one increment can be found for Part 3, decreasing the total turret rotation to 19 increments. The improved tool use sequences are shown in Figure 3-5.



Part	Tool Use Sequence	Increments of Rotation
1	4,9,2	2
2	8,3	3
3	**7,3,5**	**3**
4	3,6,7,1	4
5	7,8,1	2
6	8,7,6,2	5
	Total	19

Figure 3-5 Step 4- Tool use sequences are revisited

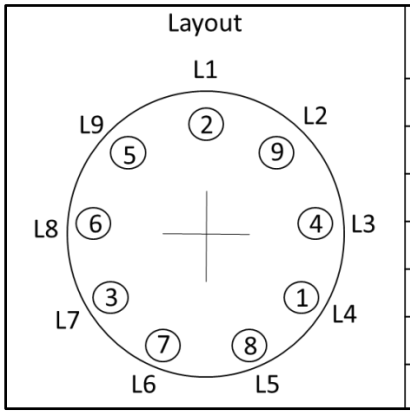
Step 5: Since improvements are still being found, the heuristic returns to the QAP sub-problem and seeks to reduce the total number of turret rotation increments by further improving the layout. A new layout is found, as shown in Figure 3-6, but no reductions in turret rotation are achieved. Note that in the new layout, the turret rotation required to produce part 2 decreases by one increment while the rotation required to produce part 4 increases by the same amount.



Part	Tool Use Sequence	Increments of Rotation
1	4,9,2	2
2	8,3	**2**
3	7,3,5	3
4	3,6,7,1	**5**
5	7,8,1	2
6	8,7,6,2	5
	Total	19

Figure 3-6 Step 5 - Turret layout is revisited

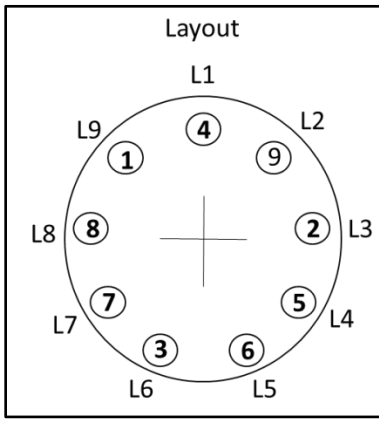
Step 6: Revisiting tool sequences finds an improved tool sequence for part 4, highlighted in bold in Figure 3-7, which reduces the total turret rotation required to produce all parts to 18 rotational increments.



Part	Tool Use Sequence	Increments of Rotation
1	4,9,2	2
2	8,3	2
3	7,3,5	3
4	**6,3,7,1**	**4**
5	7,8,1	2
6	8,7,6,2	5
	Total	18

Figure 3-7 Step 6 - Tool use sequences are revisited

Step 7: A final solution of the QAP subproblem finds a new layout, shown in Figure 3-8, that offers no improvement. Note that the new layout is the same as the previous layout, but reversed (clockwise to counter clockwise) keeping tool 9 in tool holder location 2. With no real changes to the layout, there are no changes to the tool use sequences and convergence has occurred. The iterative LSH heuristic is stopped at this point.



Part	Tool Use Sequence	Increments of Rotation
1	4,9,2	2
2	8,3	2
3	7,3,5	3
4	6,3,7,1	4
5	7,8,1	2
6	8,7,6,2	5
	Total	18

Figure 3-8 Step 7 - Convergence is achieved

3.4.4 Test Method for LSH

The testing approach used for validating the LSH method is shown below in Figure 3-9. The methods used to perform these tasks are elaborated below.

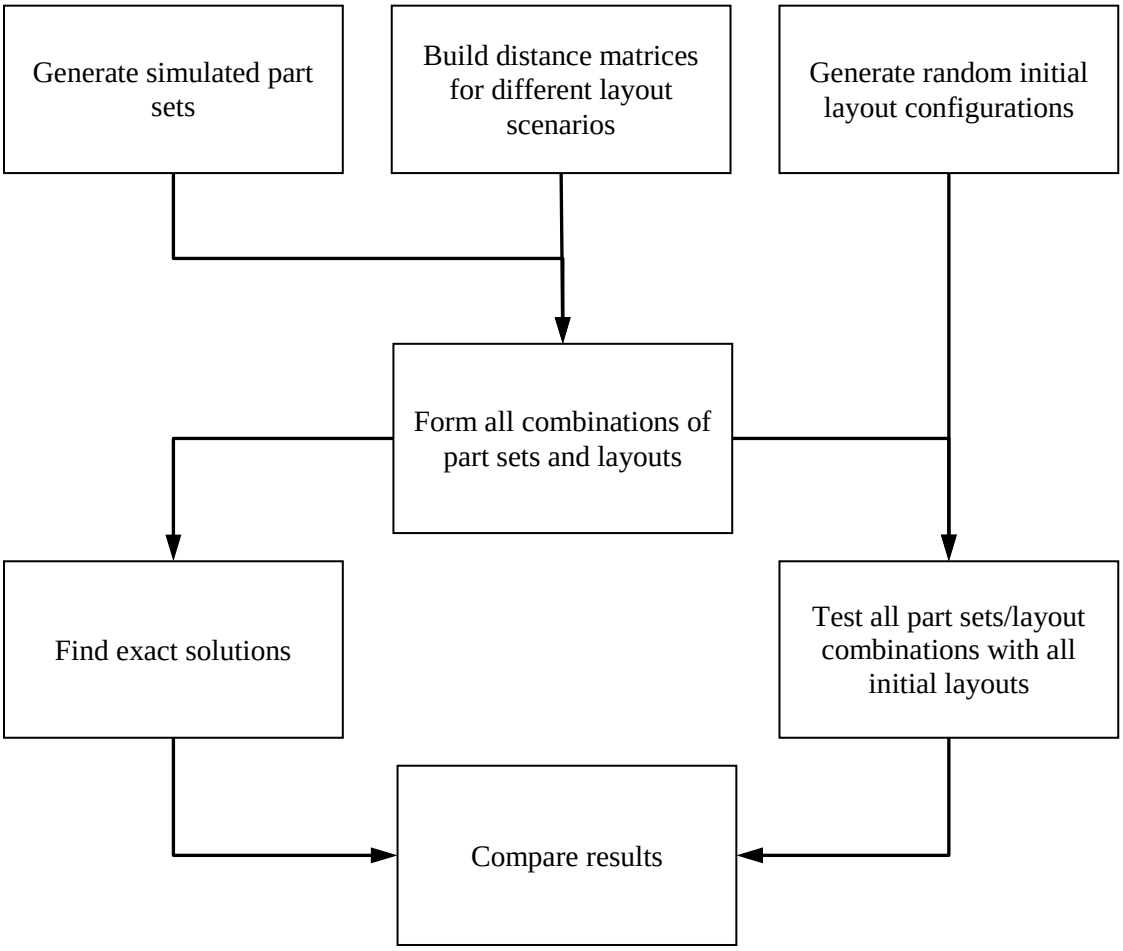


Figure 3-9 Overview of LSH test method

3.4.4.1 Generation of Simulated Part Sets

In order to accurately simulate the challenges of turret punch optimization, 10 unique data sets were randomly generated for testing and comparing the heuristic methods. These data sets are best described as 'part sets' since they represent a group of parts that could be produced by a factory. Within each part set, each part is characterized by a set of required tools and a forecasted demand level.

To make comparison to an exact solution feasible, it was necessary to implement a limit on the number of combined permutations. Within a given set of parts, each part requires n_p number of tools and can be arranged in $n_p!$ different permutations. If symmetric solutions are eliminated, the number of permutations can be reduced to $n_p!/2$. If there are p parts within the group, the number of combined permutations, can be expressed as:

$$Combined\ Permuations: \prod_{p=1}^{P} \frac{n_p!}{2} \leq 20000$$

Preliminary testing found that groups of parts with approximately 20000 combined permutations and a nine location layout required 24-72 hours to reach a solution using a simple branch and bound solution method on a 3.6GHz processor. In order permit a greater number of samples to be run, 20000 was used an upper limit on the number of combined permutations within a single set of test parts.

A sequence of numbers was selected randomly from the uniform distribution spanning the range [2:5], representing a minimum of two tools per part and a maximum of five. The random string was then truncated when the limit for combined permutations described above was achieved.

For each part within the set, the number of tools identified for the part was then selected randomly from the uniform distribution [1:9] representing the nine tools available. Repetition was not permitted since the order of operations is flexible. If a tool was

required for two operations on one part and the order of operations is flexible, moving the part away from the tool between operations would incur unnecessary costs.

In an industrial setting, it is unrealistic to expect all parts to be produced in equal proportions. To reflect this variance, a demand volume for each part has been randomly selected from the uniform distribution [1:100].

Summary statistics for the sets of randomly generated parts are shown in Table 3-1 below. Full details can be found in Appendix A.

Table 3-1 Summary of Randomly Generated Part Sets

Part Set	Number of Parts	Average Tools per Part	Number of Permutation Combinations
1	6	3.17	3888
2	2	4.50	720
3	2	5	3600
4	4	3.75	8640
5	3	3.67	432
6	3	4.33	10800
7	5	3.6	15552
8	5	3.4	10800
9	8	3	19440
10	4	3.5	2160

3.4.4.2 Construction of Distance Matrices

Tool holders within a turret are generally considered to be a circular layout. In practice, there are many variations of tool holder configurations, such as multi-tools, which may require non-circular representations. Consequently, the ability of the heuristic to solve alternative (i.e. non circular) layouts is of interest because strong performance over a range of layouts allows the heuristic to be applied to customized turret configurations. To test the performance of the heuristic over a range of different turret layout possibilities, three distance matrices were prepared corresponding to three common QAP configurations:

a. Linear - Distances are equal to the Euclidean distance between the center points of locations as shown in Figure 3-10. Locations are assumed to be immediately adjacent and of equal width.

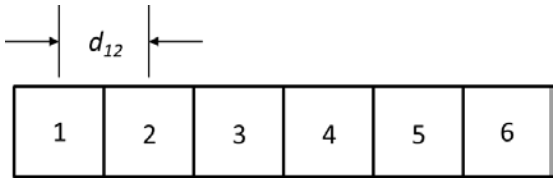


Figure 3-10 Derivation of linear distances

b. Square - Distances between locations were based on the rectilinear distance between the center points of locations. Locations are assumed to be square units of equal size and arranged in an equally spaced grid as shown in Figure 3-11.

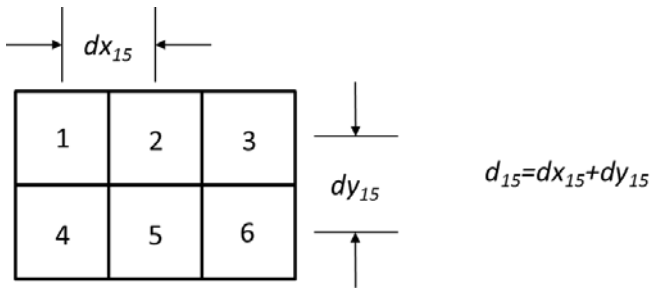


Figure 3-11 Derivation of square distances

c. Circular - The circular turret layout is assumed to be continuous and bidirectional so that the distance between consecutive locations is one rotational increment. Distances are expressed in terms of rotational increments, where a rotational increment is equal to $2\pi/n_{\text{machines}}$. Locations are assumed to be equally distributed around a single ring as shown in Figure 3-12.

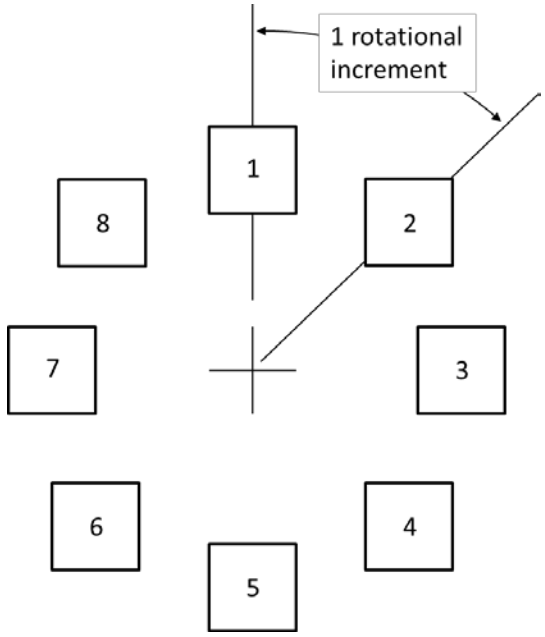


Figure 3-12 Derivation of circular distances

3.4.4.3 Generation of Random Initial Layouts

Ten random turret layouts were generated using the *randperm* function in MATLAB. This function creates a randomized permutation of the numbers 1:*n* using the pseudorandom number generator in MATLAB. Each layout was interpreted as a turret layout configuration by assigning the tool number shown in the nth position to the nth location in the turret layout.

3.4.4.4 Exact Solution Method

Each trial was evaluated using an enumeration approach that considered all combinations of operation sequences for all parts. A schematic for this method is shown in Figure 3-13. This means that each combination of part sequence permutations was considered, with each combination resulting in a unique flow matrix, *F*. For each flow matrix, the resulting QAP was evaluated using the Branch and Bound approach described by Francis & White (1974) based on Gilmore-Lawler Bounds ((Gilmore, 1962), (Lawler, 1963)).

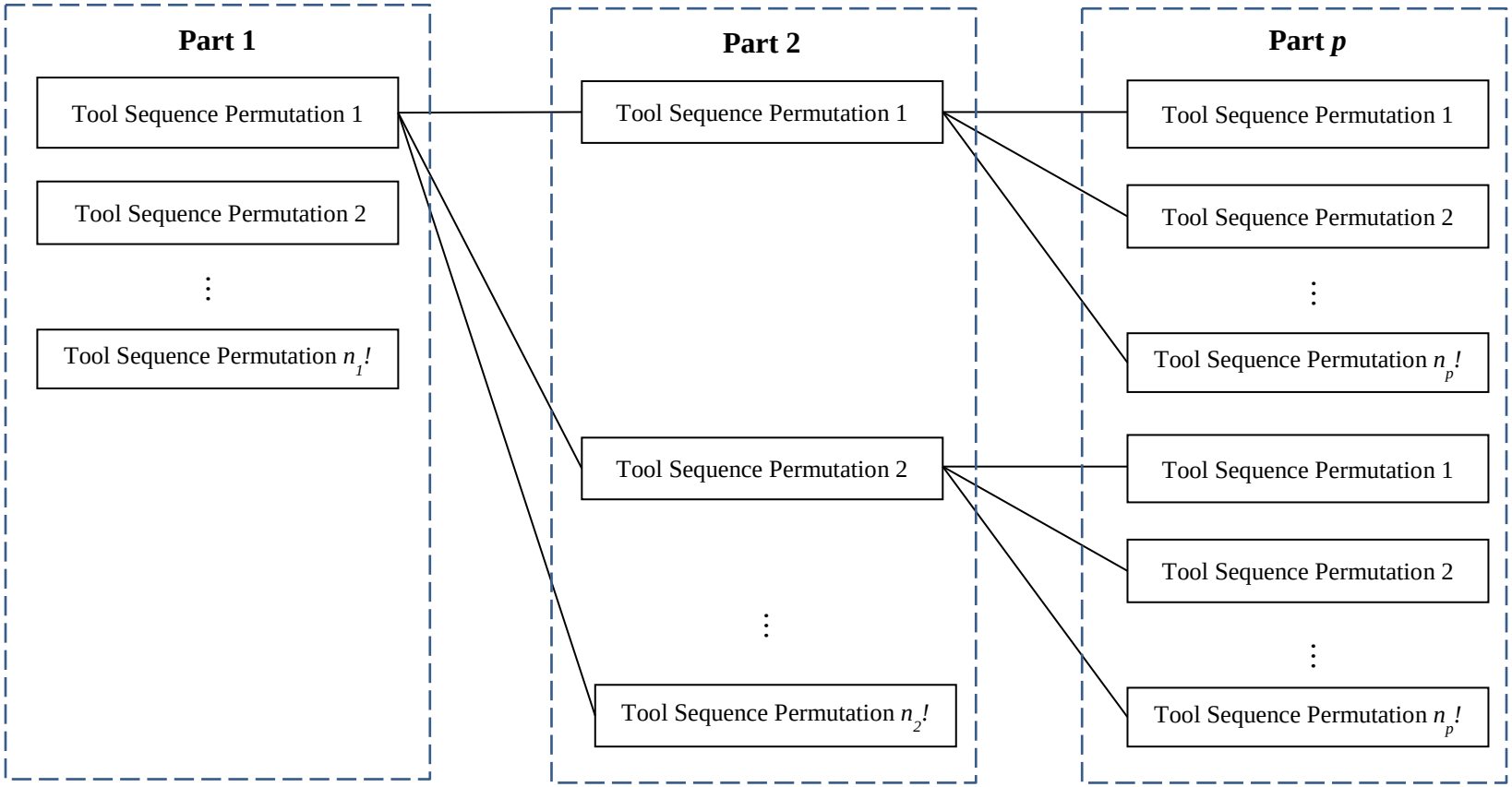


Figure 3-13 Enumeration approach for exact solution

3.4.4.5 Test Plan

A full factorial design was designed as shown in Table 3-2. Each initial layout was applied to each set of test parts and each of the three layouts in a full factorial design. The number of LSH iterations used before ending the algorithm was tested at two levels: two iterations and unlimited. The solution method used for the QAP sub-problem was tested at two levels: the exact solution described above and IHGA.

Table 3-2 Full factorial design for LSH experiment

Factor	Level 1	Level 2	Level 3
QAP Solution Method	Exact Solution	IHGA	
Iteration Limit	Two iterations	Unlimited	
Layout	Linear	Square	Circular
Part Set	Each of ten randomly generated part sets		
Initial Layout	Each of ten randomly generated initial layouts		

Results were analyzed as a general linear model. QAP Algorithm, local search iteration limit, layout and problem set were used as factors. Percentage error (relative to exact minimum) and solution time were considered as responses. Note that solution time was

only compared between variations of the heuristic and not between the heuristic and the exact solution.

3.5 Constructed Start Point: A Neighbourhood Strength Approach

Section 3.4 described a method for improving randomly selected turret layouts. In an attempt to improve the LSH method, this section presents a procedure for finding a strategic start point based on the tool information available. It is hoped that improved results will be achieved following the application of the LSH method.

Grunow et al. (2004) proposed a method of finding an efficient feeder assembly for the production of multiple PCBs using a single setup. In their approach, the locations of the installations were used to form a minimum spanning tree (MST) for each part. A measurement referred to as *neighbourhood strength* was calculated by finding the number of connections between component types within the MST. The adaptation described below is a modified version of the neighbourhood strength method and will be referred to as MNS.

Since hole locations within a nest are never known in advance, the current research only considers the sequence of tools used for each part. By further using the assumption of tool grouping, the MST for a sequence of tools becomes a straight line of n tools with $(n-1)$ connections. Under the tool grouping assumption, the neighbourhood strength for all connections is either one or zero, depending on the sequence since no tool will be used more than once. Complicating matters, the MST will present a different sequence of tools

depending on the assigned locations in the turret. In the heuristic outlined below, the concept of neighbourhood strength introduced by Grunow et al. (2004) is adapted so that neighbourhood strength reflects the potential for two tools to be used in sequence. The neighbourhood strength is then used as the flow matrix input to a QAP.

3.5.1 Development of Neighbourhood Strength

Consider the complete undirected graph of two parts shown in Figure 3-14, one requiring tools ABCD and the second requiring tools ACEF. Each graph contains (n-1)! arcs.

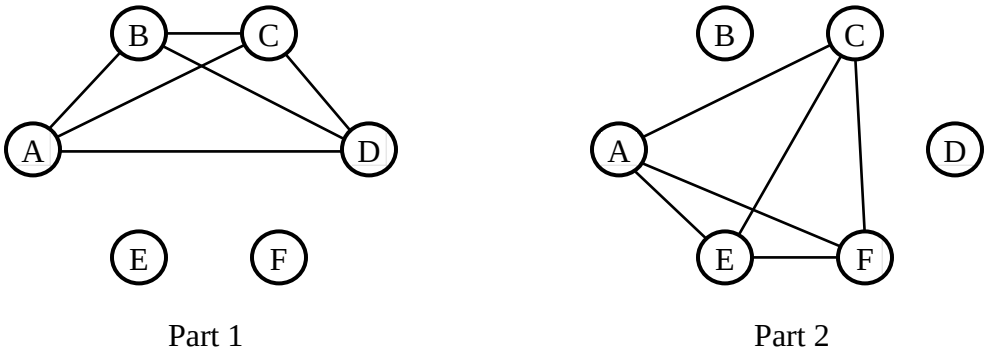


Figure 3-14 Complete Graphs

Each arc is translated into a value of one in a neighbourhood strength matrix corresponding to the part. The two graphs shown in Figure 3-14 have been presented as modified neighbourhood strength matrices in Figure 3-15. A total neighbourhood strength matrix can be calculated by summing the component MNS matrices. If the expected demand volumes are not equal, the flows for each part are multiplied by the

expected volume of the part requiring the flow component. An initial layout is then formed by solving the QAP using the total MNS matrix in place of a flow matrix.

	A	B	C	D	E	F
A	0	1	1	1	0	0
B	1	0	1	1	0	0
C	1	1	0	1	0	0
D	1	1	1	0	0	0
E	0	0	0	0	0	0
F	0	0	0	0	0	0

Part 1

+

	A	B	C	D	E	F
A	0	0	1	0	1	1
B	0	0	0	0	0	0
C	1	0	0	0	1	1
D	0	0	0	0	0	0
E	1	0	1	0	0	1
F	1	0	1	0	1	0

Part 2

=

	A	B	C	D	E	F
A	0	1	2	1	1	1
B	1	0	1	1	0	0
C	2	1	0	1	1	1
D	1	1	1	0	0	0
E	1	0	1	0	0	1
F	1	0	1	0	1	0

Total MNS

Figure 3-15 Formulation of the total MNS matrix

The iterative search procedure is then completed as in section 3.4, replacing the randomly generated start point with the MNS generated start point.

3.5.2 Test Method for MNS

The testing approach used for validating the MNS method is shown below in Figure 3-16. The methods used to perform these tasks are elaborated in the following sections.

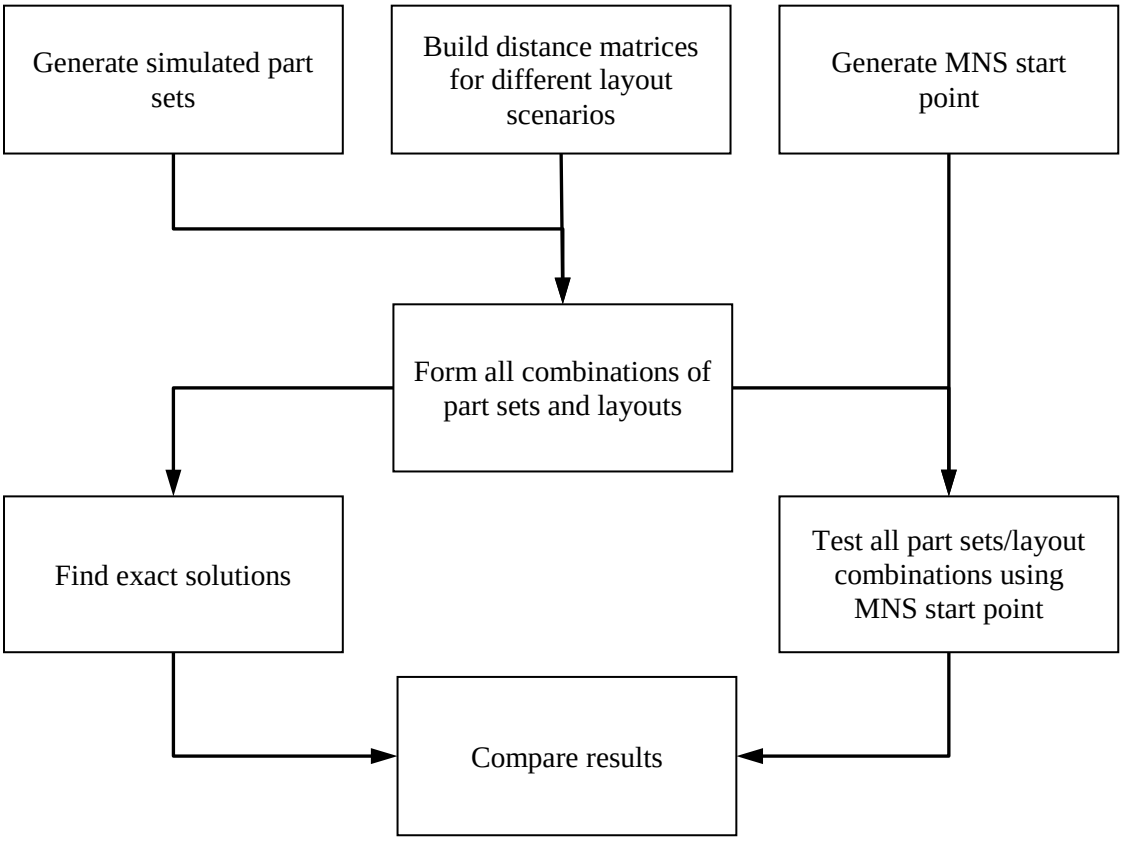


Figure 3-16 Overview of MNS test method

3.5.2.1 Generation of Simulated Part Sets

To allow comparison between methods the part sets created in section 3.4.4.1were used.

3.5.2.2 Construction of Distance Matrices

Distance matrices were used as created for the LSH method in section 3.4.4.2.

3.5.2.3 Exact Solution Method

Exact solutions were the same as found in section 3.4.4.4.

3.5.2.4 Test Plan

The MNS matrix was found for each set of test parts and used to establish an initial layout. The LSH was applied to every combination of part set and layout. Within the LSH method, only the exact QAP solution method was used since time was not critical. Since there are no random elements in the MNS heuristic, and exact QAP and TSP methods were used, the same result will be achieved for each part set. Consequently, only one replicate was completed at each test point. Results were measured as a percentage of the known optimum.

3.6 Hybrid Genetic Algorithm

The LSH method appears to converge to local minima based on the initial layout that is considered. Selecting a strategic initial layout, as is done in the MNS heuristic, achieves better results but still converges to local minima. Good search heuristics feature a mechanism to escape local minima and further explore the solution space in search of a global minimum. This characteristic can be provided by combining the LSH with a genetic algorithm.

Among the hybrid genetic algorithm approaches for the QAP, tabu search has been used most frequently as the local improvement tool. The current approach has been loosely modelled after the IHGA presented by Misevicius (2004)with the iterated tabu search portions replaced by the LSH method presented above. This substitution of local search

techniques is necessitated by the length of string that would be required to represent all tool sequences permutations for all parts in a single string.

Through the progression of the HGA, only the layout is retained in the chromosome. Each layout should correspond to a unique combination of operation sequences for the component parts. A similar observation was used by Bard, et al. (1994) in the solution of the SMD placement problem. Since the size of the solution space is proportional to the length of the chromosome, the truncation of sequence information decreases the area that must be explored.

3.6.1 HGA Parameters

The performance of the HGA method is dependent on the selection and configuration of the genetic operators used. The methods used and the levels considered are described below.

3.6.1.1 Population creation

Population samples are randomly created such that each member is a permutation of the turret layout. Chromosomes were structured such that the first character represents the turret layout location to which tool one is assigned. Similarly, the second character indicates the turret layout location to which tool two has been assigned. This pattern is continued until the nth character which indicates the turret layout location that tool n has been assigned to. Following creation, each sample was improved using the LSH method.

3.6.1.2 Selection

Pairs of population members were randomly selected from a ranked list for reproduction using the rank selection method used by Tate & Smith (1995).

3.6.1.3 Crossover Operation

Tate & Smith (1995) presented a crossover scheme that operated in three main steps:

i. Locations common to both parents were transferred to the offspring as shown below in Figure 3-17.

Parent 1	2	1	4	6	5	3
Parent 2	2	3	4	1	5	6
Child	2		4		5	

Figure 3-17 Step 1 of the crossover scheme

ii. Randomly select a gene from one of the parents at each index location in the chromosome moving from left to right. In Figure 3-18 below, this type of assignment occurs in the second position where the value of 3 is selected randomly from the set [1,3]. Similarly, in the fourth position, the value of 6 is randomly selected from the set [1,6]. If the selected gene has already been assigned, the gene in the offspring remains unassigned.

Parent 1	2	1	4	6	5	3
Parent 2	2	3	4	1	5	6
Child	2	3	4	6	5	

Figure 3-18 Step 2 of the crossover scheme

iii. Randomly assign unassigned locations to unassigned genes to complete the chromosome. For the example shown in Figure 3-19, the value of 1 is assigned to the sixth position in the chromosome since it is the only unassigned location.

Parent 1	2	1	4	6	5	3
Parent 2	2	3	4	1	5	6
Child	2	3	4	6	5	1

Figure 3-19 Step 3 of the crossover scheme

A variation of the optimized crossover (OX) scheme suggested by Misevicius (2004) is used for the creation of offspring. In the OX scheme, the method of Tate & Smith (1995),

also known as uniform like crossover (ULX) is repeated a number of times with different random selections and the chromosome yielding the best result is selected for further improvement. Misevicius used a range of 0.5*n* to 2*n* repetitions of ULX, where *n* is the problem size, before proceeding to improvement.

During preliminary testing, no correlation was found between the quality of the start location and the quality of the final solution using the local search heuristic. Consequently, in the proposed crossover scheme the repetitions of ULX as described above are limited, but the solution from each repetition is advanced to the local search heuristic. The best result from the candidate children is submitted for consideration as part of the complete population.

3.6.1.4 Mutation Operation

Typically, in a genetic algorithm mutations are applied to a small percentage of the population during each generation. No such mutation operator was used since the random alteration result normally achieved through a mutation operator was introduced through the crossover operation.

A second type of mutation, a shift mutation, was applied during the restart procedure as described in Section 3.6.1.6.

3.6.1.5 Pruning

After each generation, all members of the population were ranked based on the corresponding value of the objective function. The best members of the population were retained such that the original population size was maintained.

3.6.1.6 Restart

Sivanandam & Deepa (2008)describe that the success of a GA is a balance between exploitation and exploration in which good candidate solutions must be exploited to determine their full potential, but not to the extent that exploration efforts in other areas of the solution space are unable to succeed.

To ensure that the search continues to explore new areas of the solution space, Misevicius (2004) used a check for entropy. If the entropy of the population was close to zero (E<= 0.1) all individuals, except the top ranked individual, underwent a simple mutation operator referred to by Misevicius (2004) as a shift mutation. This feature has been retained in the current implementation.

3.6.2 Test Method for HGA

The test method used for the HGA method is depicted below in Figure 3-20.

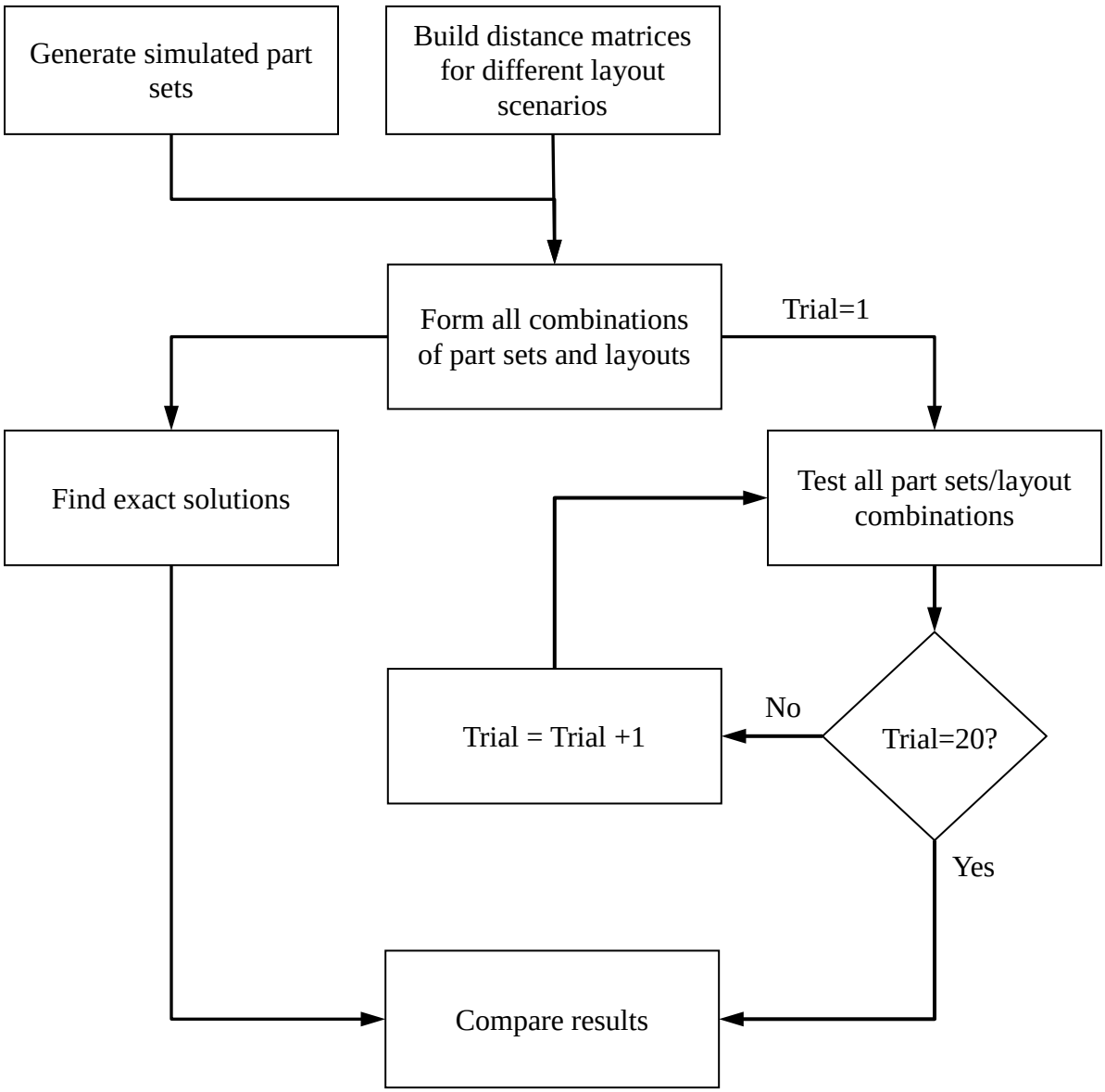


Figure 3-20 Overview of HGA Test Method

3.6.2.1 Generation of Simulated Part Sets

To allow comparison between methods the part sets created in section 3.4.4.1 were used.

3.6.2.2 Construction of Distance Matrices

Distance matrices were used as created for the LSH method in section 3.4.4.2.

3.6.2.3 Exact Solution Method

Exact solutions were the same as found in section 3.4.4.4.

3.6.2.4 Test Plan

In the HGA procedure, a major contributor to processing time is the repeated solution of the QAP required in the local search heuristic. Four variables present themselves as obvious factors in influencing the number of QAP solutions that must occur during the solution of the problem:

- Population Size – since crossovers are scaled to population size, the size of the population determines the number of crossovers that occur each generation.
- Number of Generations – each generation executes the same number of crossover operations. Increasing the number of generations increases the number of crossovers that occur.
- Local Search Iterations – Each iteration of the local search heuristic requires at least one solution of the QAP to improve the solution. The minimum number of iterations required is two. The first iteration finds the best routes for the initial

layout and then improves the layout. The second iteration evaluates the revised layout.

- ULX Crossovers – Each ULX iteration produced by the OX crossover approach requires a complete run of the local search heuristic.

Two levels of each variable were considered, and a full factorial design was executed. The levels of each variable used are presented in Table 3-3. Test points were referred to using 4 character sequences where each character represented either A or B as shown in Table 3-3.

In a review of heuristic evaluation practices, Rardin & Uzsoy (2001) suggest that replicates must be completed because of the inherent uncertainty in heuristics containing random search elements. This recommendation is consistent with the approach used in several recent evaluations of QAP heuristics, such as Stützle (2006), Misevicius (2012), and Drezner & Misevicius (2013). In the current research, 20 replicates were completed for each of the 16 test combinations applied to 10 test problems, each considered on three layouts. The average error and average solution time over the 20 repetitions was used in a general full factorial design to determine the significance of the different variables with respect to both percentage error and solution time.

Table 3-3 Summary of Levels used in HGA ANOVA

	Character Position (Left to Right)	Level	
		A	B
Population Size	1	$2\sqrt{n}$	$4\sqrt{n}$
Number of Generations	2	*n*	*n/2*
Local Search Iterations	3	2	25
ULX Crossovers	4	1	3

3.7 Measurement of Robustness

As described in the overview of this chapter, the earlier sections in this chapter describe heuristic techniques that can be used to solve the surrogate problem of minimizing the total turret rotation for the production of all individual parts. In this section, a method is described which will be applied to solutions attained using the heuristic method in order to determine if the surrogate problem is an appropriate approximation of the research objective.

The following approach focusses on the notion of robustness, which has been used in the study of the facility layout problem to describe a single layout that can perform efficiently, without changes, under varying demand conditions. Historically, the variation in demand has originated in inaccuracies in production forecasts. In the current research, the variation arises from the flexibility that can occur in the formulation of dynamic nests. For any given level of production demand, many different combinations

of nests can be formed depending on the priority that is given to different orders. To measure robustness under these conditions, the LCRI metric developed by Braglia et al. (2003) is adapted to fit the current problem and is then applied to three different large scale test problems.

3.7.1 Approximation of Distributions

The calculation of LCRI as shown in section 2.4.1.2 requires the evaluation of *N!* layout combinations which Braglia et al. (2003) suggest can be reduced to *N!*/2 because of symmetry. For a realistically sized turret punch layout problem with 30-50 tool locations, *N!*/2 is too large for such an enumeration approach. In order to accommodate the large problem size, the values of $M(l)$, $\overline{M(l)}$, $S(l)$ and $\overline{S(l)}$ can be approximated by sampling a significantly sized subset of the population of all layouts.

This approach can be expressed as:

$$\overline{m(l)} = \frac{\sum_{l=1}^{L} M(l)}{L} \tag{1}$$

$$\overline{s(l)} = \frac{\sum_{l=1}^{L} S(l)}{L} \tag{2}$$

where:

$\overline{m(l)}$ is the estimate of $\overline{M(l)}$ based on estimated values of $m(l)$

$\overline{s(l)}$is the estimate of $\overline{S(l)}$, the mean variance across all samples, based on estimated values of $s(l)$

L is the number of layouts sampled.

In the following development, the lower case will be used to represent an estimate of the true value, and uppercase represents the true value. For example $m(l)$ is an estimate of $M(l)$.

While this approximation is simple, the accuracy of the estimates must be considered to ensure their usability. Recognizing the formulas for $\overline{M(l)}$ and $\overline{S(l)}$ as being representations of the mean value, and recalling that the uncertainty in means can be estimated using Student's t-distribution as (Levine, et al., 2001):

$$\overline{M(l)} = \overline{m(l)} \pm t_{L-1} \frac{\sqrt{s(m(l))}}{\sqrt{L}} \tag{3}$$

$$\overline{S(l)} = \overline{s(l)} \pm t_{L-1} \frac{\sqrt{s(s(l))}}{\sqrt{L}} \tag{4}$$

Within each layout, the mean material handling cost across all nest variations of the order set, $M(l)$, can be similarly estimated as:

$$M(l) = m(l) \pm t_{N-1} \frac{\sqrt{s(l)}}{\sqrt{N}} \tag{5}$$

Error in the estimate of $s(l)$ does not need to be considered in the calculation on $M(l)$ since the Student t-distribution is formulated on the premise that the population standard deviation, σ, is not known, and can only be represented by the sample standard deviation (Levine, et al., 2001).

Returning to equation (3) and substituting,

$$\overline{M(l)} = \frac{\sum_{l=1}^{L} m(l)}{L} \pm \left(\frac{\sum_{l=1}^{L} t_{N-1} \frac{\sqrt{s(l)}}{\sqrt{N}}}{L} + t_{n-1} \frac{s(m(l))}{\sqrt{n}} \right) \qquad (6)$$

In a worst case evaluation of LCRI, it would be imperative to use the lower limit of $\overline{M(l)}$ since this represents the minimum distance between the mean of the candidate layout and the mean of all layouts. Simultaneously considering the upper limit of $\overline{S(l)}$ will lower the number of standard deviations between the means and result in a pessimistic measure of the probability that the candidate layout consistently outperforms an average layout.

Considering these formulations in the context of the current experiment,a conservative estimate of LCRI can be found as:

$$LCRI = \varphi \left(\frac{\overline{M(l)} - M(l^*)}{\sqrt{\left[\overline{S(l)} + S(l^*)\right]}} \right) \qquad (7)$$

Where $\overline{M(l)}$ and $\overline{S(l)}$ are as described as above, $M(l^*)$ and $S(l^*)$ are the mean and standard deviations of the performance of a specific solution layout over a range of production schedules.

3.7.2 Test Method for Robustness

The robustness of a layout was tested through following the steps:

i. Generate Problem set

ii. Use the HGA heuristic for solving layout problems with flexible sequencing presented in Chapter 2.

iii. Select a demand set

iv. Randomly formulate the demand into *N* sets of nested parts

v. Randomly generate *L* layouts.

vi. Find the cost of producing each set of nested parts for each of the *L* layouts, as well as for the heuristic solution layout.

3.7.2.1 Generation of Simulated Part Sets

Three problem sets were generated. Two problem sets represented a model of 50 different parts being processed on a 30 tool turret. The third problem set represented 100 parts being processed on a 50 tool turret. Simulated part sets were generated using the procedure described in Section 3.4.4.1. The settings used for each of the three data sets for the robustness tests are shown in Table 3-4. Full details for each part set can be found in Appendix B.

Table 3-4 Settings for generation of large part sets for robustness testing

Part Set	Number of parts	Minimum number of tools per part	Maximum number of tools per part	Number of tools in the turret	Minimum demand quantity	Maximum demand quantity
1	50	2	8	30	1	500
2	50	2	8	30	1	500
3	100	2	10	50	1	500

3.7.2.2 Assignment of Areas

In order to better simulate nesting, an area was randomly assigned to each part. The flat pattern area was randomly selected from the normal distribution centred about an average area, $A_{ave,}$ with standard deviation $A_{ave}/3$. Values less than the minimum area, $A_{min,}$ were replaced with A_{min}, and values greater than the maximum area, $A_{max,}$ were replaced with A_{max}.

Three problems were solved using variations of the hybrid genetic algorithm presented in chapter 2. The BBAB variation of the HGA was followed because earlier testing suggested that it may have a lower maximum error level. For each solution one member of the initial population was seeded as the solution from the MNS method presented in chapter 2. The properties used for each problem set solution are summarized in Table 3-5 below:

Table 3-5: HGA parameters used to find optimal layouts

	n=30, Problem 1	**n=30, Problem 2**	**n=50, Problem 1**
Number of HGA solutions	5	5	1
Population Size	22	22	28
Generations	15	15	25
Local Search Iterations	2	2	2
ULX Iterations	3	3	3

3.7.3 Scaling of Demand Set

The demand for each problem was scaled from the total demand that was used to find the solution layout. Using the entire demand resulted in too many nests and required large amounts of computational time to solve. To compensate, the total demand was scaled to between 20% and 30% of its original value, resulting in a list of ~250 nests per recombination of the demand.

3.7.4 Generation of Simulated Nests

In industrial applications, scheduling requirements usually dictate the preferred order of completion for production jobs. To simulate this factor, a list replicating a scheduled sequence of the demand was created by applying a probability to each part, and then drawing parts from the pool of all demand until all parts had been included. The probability assigned to each part was:

$$P(p) = \frac{d_p}{\sum_{i=1}^{P} d_i}$$

The probability was adjusted after each selection to reflect the remaining demand in the pool.

The number of parts was always required to be an integer. In only a few instances, the expected demand for a part was decreased to zero by the demand scaling process. These parts were left at a level of zero demand.

Simulated nests were formed by drawing from the order priority list. The part at the top of the list was applied to the sheet first. The area remaining on the sheet was adjusted to reflect the area consumed by the first part. The remainder of the sheet was filled by selecting the next part in the order sequence, checking for available space and adding it if possible. Each part in the order priority was checked for fit until the sheet was full or the bottom of the list was reached. This process was repeated until all parts had been assigned to nests. The group of nests was considered to be a simulated production schedule.

3.7.5 Test Plan

The measurement of robustness using population sampling requires enough samples to reduce errors to acceptable levels while still remaining computationally feasible. For the study of robustness, replication occurred in two dimensions: number of test layouts and number of simulated production schedules.

100 unique test layouts were considered, in addition to the specific layouts that were found using the HGA solutions. During analysis, each specific layout was considered in combination with the 100 random layouts separately, creating a sample set of 101 units. This separation was used because a set of five exceptional results within a population of 105 would likely skew the results.

For each layout, the simulated demand was recombined into 30 different simulated production schedules. All production schedules represented the production of the same quantities of parts, but differed in the composition of the individual nests. The same set of 30 simulated production schedules was used to evaluate each of the 100+ unique test layouts.

The minimum cost of production for each nest was found by changing the sequence of tool use for each part using a messenger type problem approach. This was similar to the solution done within the search heuristic, however since the number of tools used by a nest is much more than the number used by a single part, the implementation in the MATLAB optimisation toolbox was used to solve the TSP portion of the problem. The remainder of the robustness calculations were executed using the approximation of LCRI described in earlier in this chapter.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 LSH Method

4.1.1 LSH Results

Figure 4-1 is a histogram capturing the amount of improvement achieved on random start trials. Improvement was measured by comparing the minimum cost achieved by optimizing part sequences using the initial layout (i.e. at the end of the first TSP iteration) to the minimum cost achieved using the final layout.

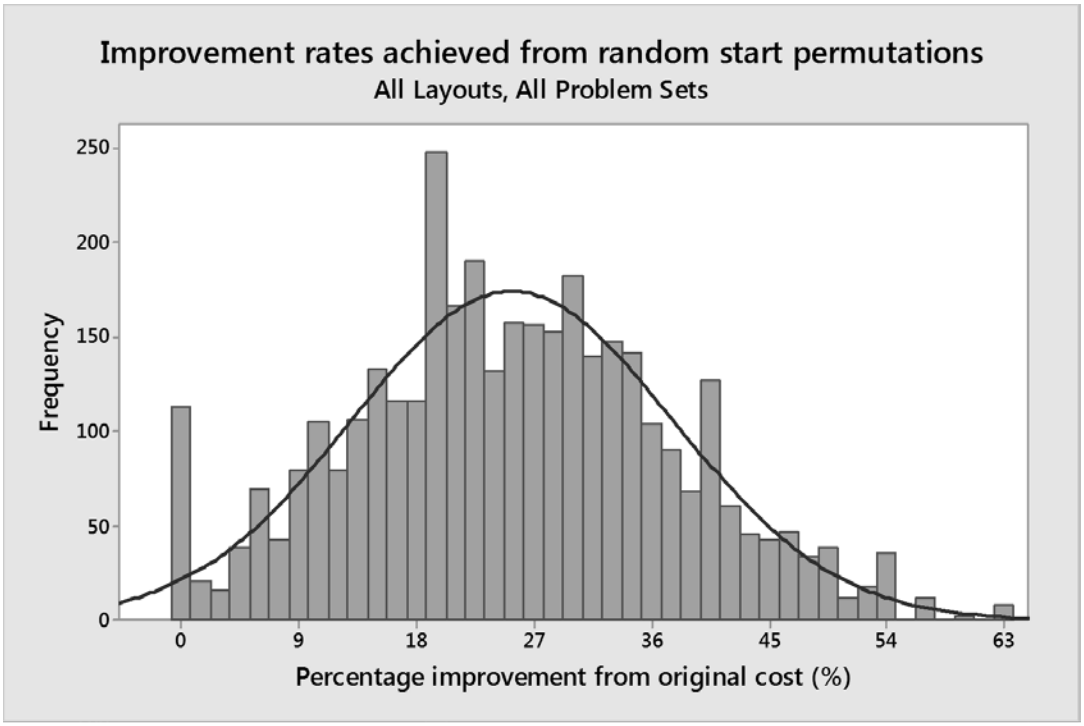


Figure 4-1 Percentage improvement in random starts

Table 4-1 compares the performance differences in both time and accuracy caused by varying the levels for iteration limit and QAP solution tool. The differences were observed in the application of the LSH method to a series of random start points using different QAP solution techniques and different numbers of iterations.

Table 4-1 Comparison of means - LSH

	Exact Solution		**IHGA**	
Iteration	**Mean**	**Mean**	**Mean Time**	**Mean**
One Iteration	9.01	15.691	0.260	15.585
Unlimited	12.71	14.756	0.446	14.708
Change	**3.70**	**-0.935**	**0.186**	**-0.877**

Table 4-2 provides statistical details of the general linear model that was used to study the problem set. Significant factors and interactions are shown in bold. Interactions that were not found to be significant with respect to either percentage error or time have been omitted from the table.

Table 4-2 ANOVA Results - Improvement Heuristic

Factors	***Degrees of Freedom (Model Total =174)***	***Response: % Error***		***Response: Time***	
		F-Value	***P-Value***	***F-Value***	***P-Value***
Algorithm	1	0.07	0.788	316.65	**0.000**
Iteration Limit	1	10.16	**0.001**	87.16	**0.000**
Layout	2	815.78	**0.000**	128.95	**0.000**
Random Start	9	38.82	**0.000**	2.05	**0.031**
2-Way Interactions (Only significant interactions shown)					
Algorithm*Iteration Limit	1	0.01	0.919	83.87	0.000
Algorithm*Layout	2	0.08	0.924	256.35	0.000
Algorithm*PartSet	9	0.12	0.999	62.79	0.000
IterationLimit*Layout	2	0.80	0.449	6.98	0.000
Layout*PartSet	18	34.54	0.000	15.83	0.000

Table 4-3 summarizes the number of correct solutions that were found across all layouts. The number of correct solutions is categorized by the QAP solution method used and the iteration limit that was applied.

Table 4-3 Number of correct solutions by algorithm and iteration limit

	Iteration Limit	
QAP Solution Algorithm	**Two Iterations**	**Unlimited Iterations**
Branch and Bound (Exact Method)	114	125
IHGA	119	129

4.1.2 LSH Discussion

As shown in Figure 4-1, the LSH method, when started from a random configuration, can find an improved layout in which material handling costs have been decreased by an average of approximately 25%. This graph demonstrates the improvement power of the search heuristic. However, as shown in Table 4-1, the average error of the final solutions is still approximately 15%. This demonstrates that despite the improvement power of LSH, it does not guarantee convergence to the global minimum.

Using the IHGA algorithm as the QAP solution tool was found to be statistically significant (α=0.05) in decreasing the processing time, but not significant with respect to percentage error. While it was known that that the IHGA algorithm was dramatically

faster, this result confirms that the IHGA algorithm is acceptably accurate for this application.

The suitability of the IHGA algorithm is reinforced by comparing the total number of correct solutions found by each of the QAP solution algorithms across all layouts. For the IHGA, a total of 248 exact solutions were found, as compared to 239 exact solutions using the branch and bound technique. When compared using a chi-squared test for the differences among proportions, the difference between groups was not significant ($\chi^2(1, N=3600)=0.1923$, $p=0.66$) confirming that the proportion of successes between the QAP solution techniques is similar.

In the analysis of percentage error, layout and iteration limit were both found to be significant. However, a review of the table of means, Table 4-1, shows that the difference caused by the iteration limit is less than 1%. This small cost in accuracy is likely worth the benefit in performance time when used in combination with a genetic algorithm. Layout is likely significant because of the high degree of similarity between the circular and linear layouts, which makes the square layout appear quite different.

With respect to time, all factors and all but one second level interaction were found to be significant. These interactions should be expected. For example, a difficult to solve layout, such as linear, is likely to require many iterations to converge. The time difference between the two iteration levels is going to be much greater for the linear layout than it would be for a square layout which is solved quickly even under an

unlimited iteration scenario. This difference causes the IterationLimit*Layout interaction to be significant. This same scenario also causes the Algorithm*Layout interaction to be significant with respect to time since the difference in time between algorithms increases as the number of required iterations increases. Since the number of iterations required is likely independent of the algorithm used, the time difference will be greater on difficult layouts that require more iterations.

Interactions involving part set occur because the different part sets offer different levels of complexity. While the levels of complexity are important for demonstrating the accuracy range of the heuristic, they also unduly influence time by requiring additional iterations. As in the above explanation of layout interactions, the complexity differences of the part sets are amplified when considered in combination with algorithm and layout.

To summarize, the local search heuristic has been shown to effectively find local minimums, but does not guarantee global minimums. When compared to the exact solution approach, use of the IHGA approach as a QAP solution method did not cause any significant changes in the error suggesting that it is an acceptable alternative. The use of IHGA also decreased the average computational time from 9.01s to 0.26s when the processing was also limited two TSP iterations. The practice of limiting iterations was shown to increase mean error by approximately 1%, but was also shown to decrease computational time 42% from 0.45s to 0.26s when used in combination with the IHGA. The LSH method is best used in combination with the IHGA algorithm for the QAP

solution and a two iteration limit on the LSH process. This combination has been shown to provide accurate results quickly.

4.2 MNS Method

4.2.1 MNS Results

Table 4-4provides a summary of the results from applying the MNS heuristic.

A paired t-test was constructed in which the first distribution was formed by the average of all random starts for each problem set/layout combination. The second distribution was formed by the constructed start point solutions. The results from using the constructed start point were found to be significantly better than the random start averages (α=0.05).

Table 4-5 is a contingency table comparing the proportion of successes between the MNS results and the LSH results for the LSH variation that used the IHGA algorithm and the two-iteration limit. Results shown are aggregated across all part/layout combinations.

Table 4-4 Performance of local search heuristic with constructed start point

Part Set	Linear Layout			Square Layout			Circular Layout		
	Exact Solution	MNS Solution	%Error	Exact Solution	MNS Solution	%Error	Exact Solution	MNS Solution	%Error
1	868	868	0	768	872	13.5%	868	868	0
2	560	560	0	560	560	0	560	560	0
3	408	408	0	408	408	0	408	408	0
4	875	892	1.9%	772	832	7.8%	875	892	1.9%
5	219	219	0	198	198	0	219	219	0
6	542	542	0	504	504	0	542	580	7.0%
7	855	855	0	738	765	3.7%	852	855	0.4%
8	930	977	5.1%	785	808	2.9%	930	977	5.1%
9	1502	1682	12.0%	1160	1224	5.5%	1381	1381	0
10	422	460	9.0%	377	377	0	422	460	9.0%
Mean Error			2.8%			3.3%			2.3%
Max Error			12.0%			13.5%			9.0%
Min Error			0%			0%			0%
Mean Error Random Starts Method			20.8%			7.3%			17.5%

Table 4-5 Contingency table comparing MNS and LSH methods across all layouts

	Heuristic	
	MNS	**LSH**
Exact Solutions	16	119
Sub-optimal solutions	14	781
Total Samples	30	900

4.2.2 MNS Discussion

The use of the MNS heuristic was shown to decrease the mean percentage error decreasing for all three different layouts. While the decrease in percentage error compared to the mean error from the random start points is notable, the range in error that occurred with the MNS constructed start point was large, ranging from 0% to 13.5%. This indicates that the constructed start point offers a good solution but like LSH, it does not guarantee that it will identify the global minimum.

The paired t-test comparison between the MNS and LSH results suggests that the use of the MNS start point is better than the use of a random start point. However, since the MNS data points were represented by the average of a set of random start points, it does not guarantee that the MNS start point yields a better result than a random start point. Instead, it suggests that given an individual random start point, the MNS start point is only more likely to yield an improved solution. It is very possible that a random start point will be generated that will outperform the MNS result.

To further explore the relative performance of LHS and MNS, a chi-squared test for differences among proportions was conducted using the data shown in Table 4-5. The MNS method was found to be significantly more likely to match the exact solution when compared to the LSH method. ($\chi^2(1, N=930)=37.64, p<0.001$)

The MNS method is an improvement over the LSH method and should be used whenever possible. The MNS method has been shown to achieve the exact solution a greater percentage of the time. The results also suggest that an inexact solution produced by the MNS method is likely to have a lower percentage error than an equivalent solution found using the LSH approach.

4.3 HGA Method

4.3.1 HGA Results

Results from the HGA tests are shown graphically in Figure 4-2, Figure 4-3, Figure 4-4 and Figure 4-5. In these figures, the vertical axis represents the degree of error in the solution (either average error or maximum error) and the horizontal axis represents the solution time. As in most heuristic solutions, there are trade-offs between accuracy and processing time. A desirable solution is able to deliver acceptable accuracy in a reasonable time. In the figures below, this represents a position near the origin.

Figure 4-2 includes the data for all solution sets, and Figure 4-3, Figure 4-4 and

Figure 4-5 show the relationship categorized into the linear, square and circular layouts respectively. The results shown are encoded by the factor levels used for the set of samples. Average values shown represent the average result of 20 samples measured using the specified combinations of factor levels. Maximum error is reported as the largest error measured for the combination of factor levels over 20 repetitions. Note that only test conditions where the third character is ‘A’ are shown. This is because the factor in the third character, iteration limit, was found to be insignificant with respect to error and the level corresponding to ‘A’ offered faster time performance.

An alternative method for comparing performance is to consider the number of exact solutions that were achieved out of the 20 repetitions that were completed for each combination of part set and layout. Figure 4-6 shows the aggregated total of exact solutions across all iterations and all combinations of part set and layout. Detailed data by layout type are presented in Table 4-6, Table 4-7 and Table 4-8 for linear, square and circular layouts respectively.

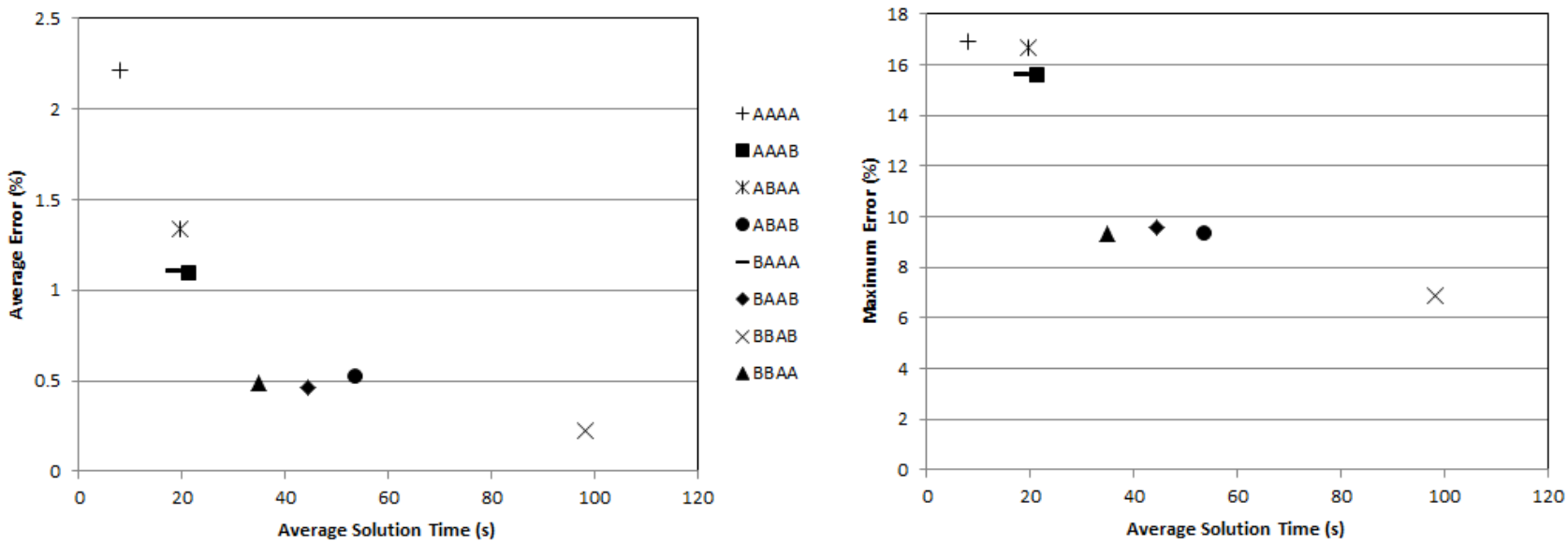


Figure 4-2 Solution time vs error for all HGA trials, all layouts

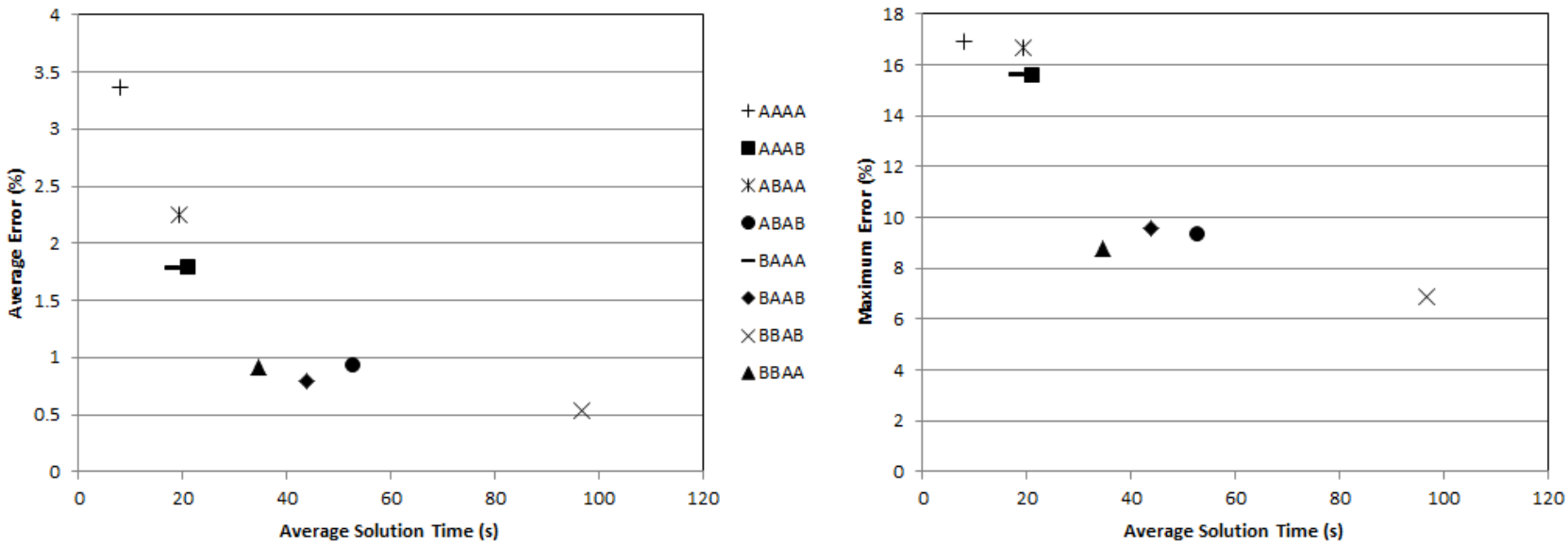


Figure 4-3 Solution time vs error for all HGA trials using linear layout

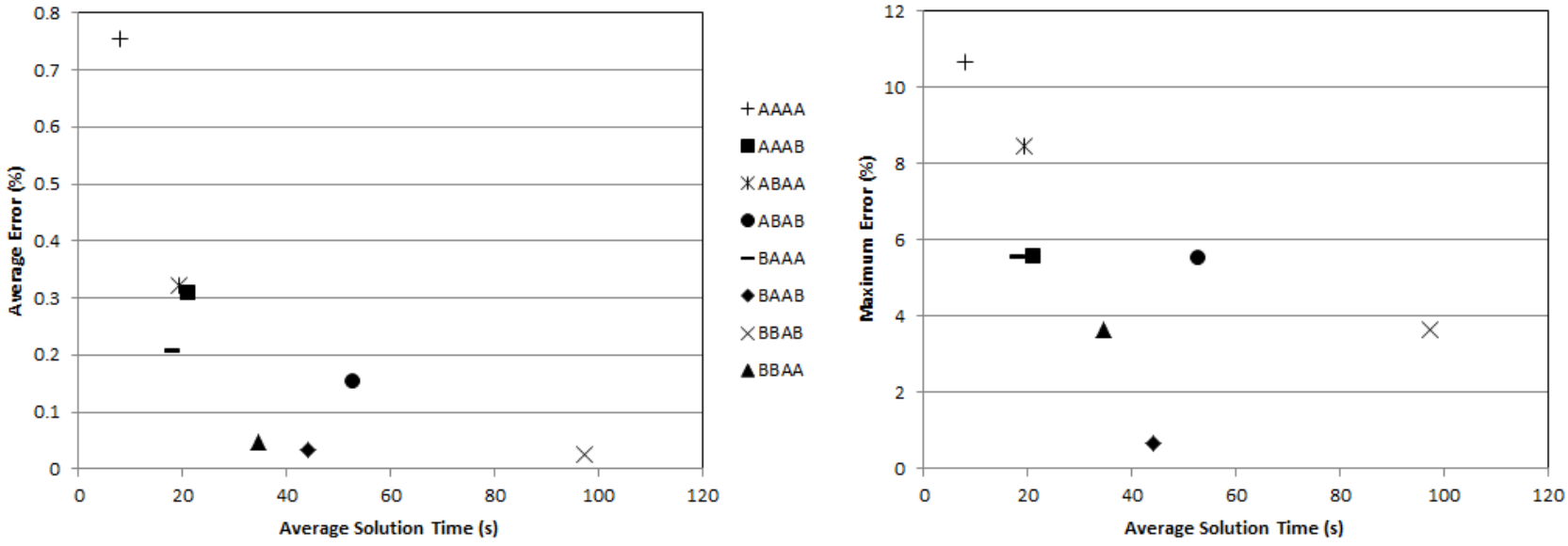


Figure 4-4 Solution time vs error for all HGA trials using square layout

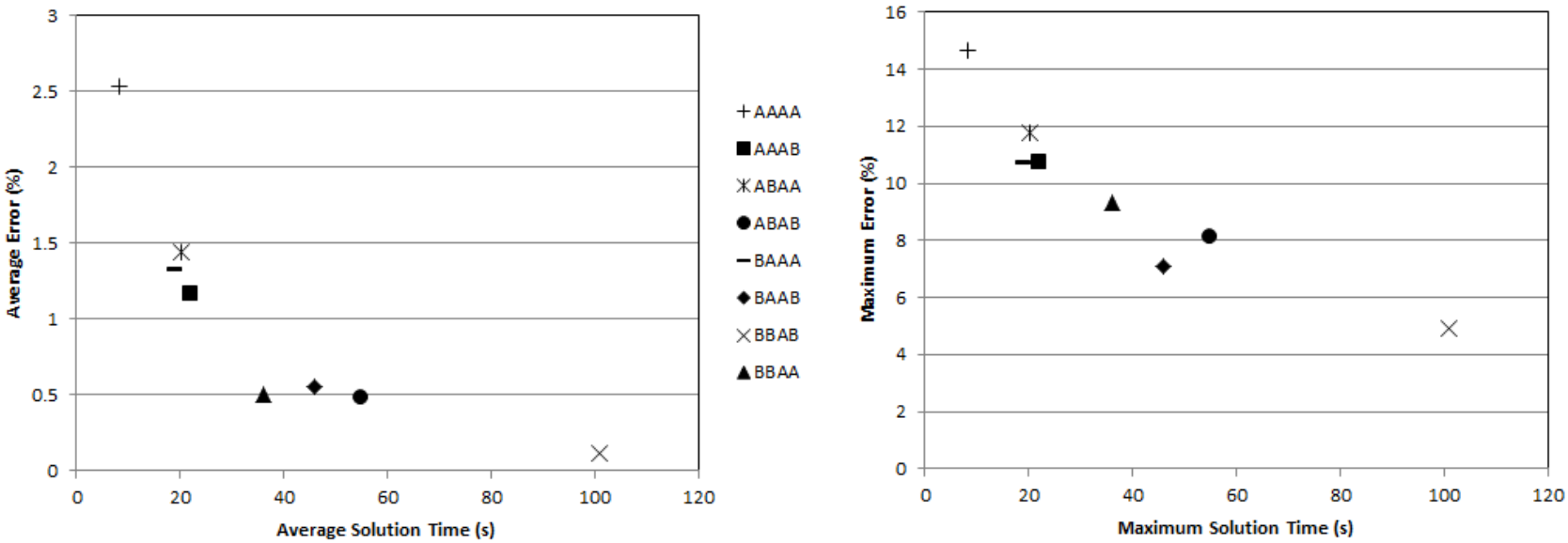


Figure 4-5 Solution time vs error for all HGA trials using circular layout

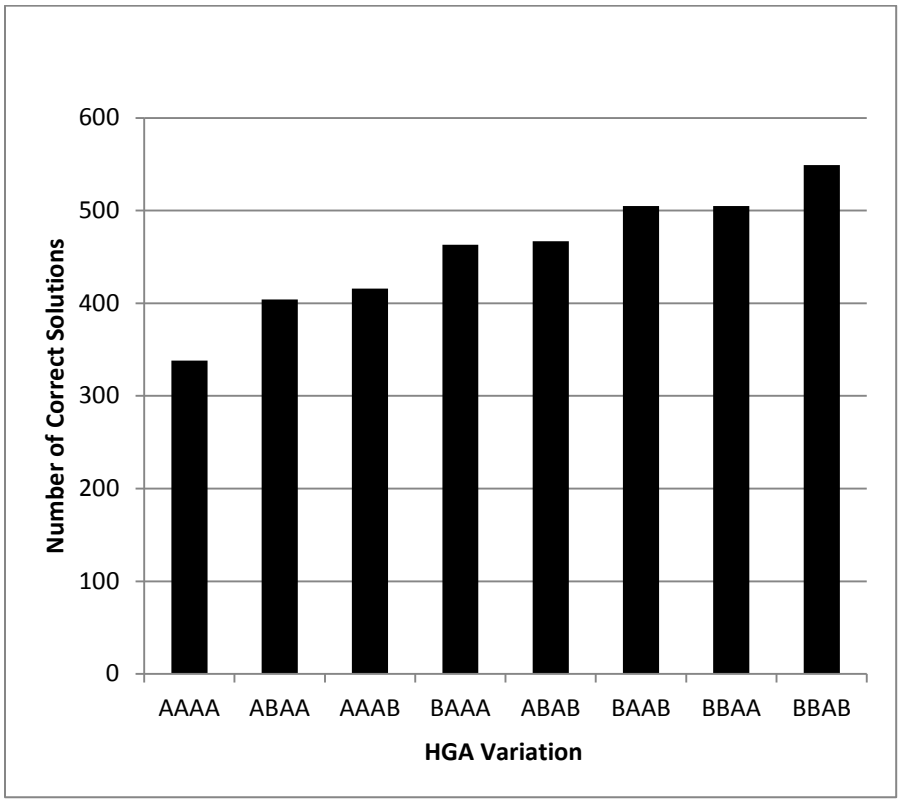


Figure 4-6 Total number of exact solutions across all layouts

Table 4-9 shows the results of the HGA ANOVA with respect to percentage error. Table 4-10 Summary of HGA ANOVA results with computation time as response shows a similar ANOVA with time as a response.

Table 4-6 Number of exact solutions out of 20 - linear layout

Layout	Trial	PartSet										Total
		1	2	3	4	5	6	7	8	9	10	
Linear	AAAA	7	17	19	2	13	10	12	1	6	2	89
	AAAB	12	20	19	8	18	13	13	1	11	2	117
	AABA	10	20	17	5	12	8	7	2	6	2	89
	AABB	15	20	18	4	12	7	15	3	12	1	107
	ABAA	12	20	18	2	15	11	11	3	8	4	104
	ABAB	14	20	19	13	19	15	19	8	15	2	144
	ABBA	15	20	19	0	15	12	17	2	10	3	113
	ABBB	17	20	20	11	20	16	17	5	14	7	147
	BAAA	17	20	20	2	16	12	17	1	16	2	123
	BAAB	17	20	20	14	16	16	20	6	16	8	153
	BABA	19	20	18	5	16	12	15	3	10	7	125
	BABB	19	20	20	11	20	13	20	4	16	4	147
	BBAA	18	20	20	12	19	15	20	5	15	2	146
	BBAB	19	20	20	17	20	17	19	13	16	5	166
	BBBA	15	20	20	11	20	14	17	9	14	7	147
	BBBB	20	20	20	16	20	19	20	9	16	5	165

Table 4-7 Number of exact solutions out of 20 - square layout

Layout	Trial	PartSet										Total
		1	2	3	4	5	6	7	8	9	10	
Square	AAAA	5	20	20	17	20	20	11	11	10	20	154
	AAAB	9	20	20	19	20	20	14	17	15	20	174
	AABA	6	20	20	19	20	20	9	14	8	20	156
	AABB	9	20	20	20	20	20	17	18	17	20	181
	ABAA	6	20	20	19	20	20	16	16	12	20	169
	ABAB	15	20	20	20	20	20	19	18	15	20	187
	ABBA	12	20	20	20	20	20	18	16	18	20	184
	ABBB	13	20	20	20	20	20	18	18	12	20	181
	BAAA	10	20	20	19	20	20	17	16	19	20	181
	BAAB	10	20	20	20	20	20	20	20	20	20	190
	BABA	11	20	20	20	20	20	16	19	16	20	182
	BABB	13	20	20	20	20	20	19	20	17	20	189
	BBAA	17	20	20	20	20	20	19	19	19	20	194
	BBAB	20	20	20	20	20	20	19	20	18	20	197
	BBBA	15	20	20	20	20	20	19	20	17	20	191
	BBBB	20	20	20	20	20	20	20	20	20	20	200

Table 4-8 Number of exact solutions out of 20 - circular layout

Layout	Trial	PartSet										Total
		1	2	3	4	5	6	7	8	9	10	
Circular	AAAA	5	20	17	1	13	10	9	1	15	4	95
	ABAA	14	20	20	2	19	10	10	6	16	8	125
	AABA	12	18	14	3	10	9	8	1	15	7	97
	AABB	16	20	17	3	16	15	17	2	17	9	132
	BAAA	13	20	20	6	16	8	15	6	16	11	131
	AAAB	17	20	18	4	17	18	12	2	19	9	136
	ABBA	11	20	18	2	17	13	10	2	18	12	123
	ABBB	19	20	20	9	20	14	18	3	19	13	155
	ABAB	19	20	20	10	20	13	18	5	20	14	159
	BAAB	19	20	20	6	20	17	17	10	19	14	162
	BABA	14	19	18	3	18	16	15	1	17	11	132
	BABB	19	20	20	3	19	17	17	7	20	17	159
	BBAA	17	20	19	10	20	17	19	8	20	15	165
	BBAB	20	20	20	15	20	20	20	12	20	19	186
	BBBA	20	20	20	8	20	18	17	4	19	15	161
	BBBB	20	20	20	12	20	18	20	11	20	17	178

Table 4-9 Summary of HGA ANOVA results with percentage error as response

Factor		Linear		Square		Circular	
	DOF	**F-Value**	**P-Value**	**F-Value**	**P-Value**	**F-Value**	**P-Value**
Population Size	1	70.99	**0.000**	32.09	**0.000**	75.18	**0.000**
Generations	1	45.69	**0.000**	11.60	**0.001**	70.61	**0.000**
Local Search Iterations	1	0.10	0.757	1.53	0.218	2.54	0.113
ULX Iterations	1	58.55	**0.000**	10.56	**0.001**	88.08	**0.000**
Interactions							
PopulationSize*Generations	1	4.33	**0.039**	2.57	0.111	2.97	0.087
PopulationSize*LocalSearchIterations	1	0.05	0.826	0.86	0.355	0.22	0.637
PopulationSize*ULXIterations	1	4.70	**0.032**	1.88	0.172	10.24	**0.002**
Generations*LocalSearchIterations	1	0.27	0.603	0.42	0.517	0.22	0.643
Generations*ULXIterations	1	2.06	0.153	7.25	**0.008**	5.16	**0.025**
LocalSearchIterations*ULXIterations	1	0.86	0.356	1.21	0.273	0.08	0.774

Table 4-10 Summary of HGA ANOVA results with computation time as response

Factor		Linear		Square		Circular	
	DOF	**F-Value**	**P-Value**	**F-Value**	**P-Value**	**F-Value**	**P-Value**
Population Size	1	3141.7	**0.000**	12223.9	**0.000**	2071.9	**0.000**
Generations	1	3731.8	**0.000**	14091.6	**0.000**	2437.5	**0.000**
Local Search Iterations	1	510.9	**0.000**	4675.9	**0.000**	508.9	**0.000**
ULX Iterations	1	6087.6	**0.000**	23363.7	**0.000**	3991.9	**0.000**
Interactions							
PopulationSize*Generations	1	293.7	**0.000**	1150.8	**0.000**	197.9	**0.000**
PopulationSize*LocalSearchIterations	1	80.3	**0.000**	666.9	**0.000**	75.1	**0.000**
PopulationSize*ULXIterations	1	525.0	**0.000**	2045.1	**0.000**	349.9	**0.000**
Generations*LocalSearchIterations	1	12.4	**0.001**	242.7	**0.000**	20.8	**0.000**
Generations*ULXIterations	1	821.2	**0.000**	3136.8	**0.000**	541.1	**0.000**
LocalSearchIterations*ULXIterations	1	80.2	**0.000**	842	**0.000**	86.9	**0.000**

4.3.2 HGA Discussion

Results of the ANOVA with respect to error, Table 4-9, show that the only factor that does not have a significant impact (α=0.05) on error is the limit on local search iterations. Since this factor is significant with respect to processing time, it should be set to the fastest setting, which corresponds to all tested trial with 'A' in the third character of the trial indicator. It is interesting to note that in the testing of the LSH method independent of the HGA, iteration limit was found to be statistically significant with respect to error, but only reduced the percentage error by a small amount. Perhaps repeated attempts at improving variants of the original layout in each generation of the HGA compensate for the lack of iterations at each independent stage.

Reviewing the remaining results shows three factors and three interactions that are significant with respect to error as well as three factors and six interactions that are significant with respect to time. The high number of interactions that are significant with respect to time should be expected since all factors selected were known to have obvious influences on time. In most cases, the factors independently have a linear increase on time. Combined, they can lead to an exponential relation to time causing the interaction to be significant. With respect to error, the interactions inconsistently appear as significant. For example, PopulationSize*Generations is significant for Linear layouts, but not circular or square. This could be a reflection of the relative difficulty of solving the different layouts.

To help determine the optimum settings for the heuristic, consider the error vs. time plots shown in Figure 4-2, Figure 4-3, Figure 4-4 and Figure 4-5. These plots clearly show that improvements in time performance require an increase in the acceptable amount of error. For this situation, the point closest to the origin should offer the best blend of performance time and quality. The operating conditions that best embody this point are found in BBAA, a set characterized by a large population, large number of generations, limited local search iterations and only one ULX replication per crossover.

As a further check of the suitability of these conditions, consider the maximum error graph in combination with the average error graph. The maximum error graphs represent the single worst trial with respect to percentage error of all trials completed using a given combination of algorithm and layout. In the square and circular layouts (Figure 4-4, and Figure 4-5), it can be seen that test point BAAB was shown to have better maximum error values than BBAA. Since BBAA and BAAB also have similar levels of average error, the slightly slower BAAB should be used in order to decrease the maximum error.

Figure 4-6 reinforces the close relationship between variations BBAA and BAAB as both were able to find the exact solution a similar number of times. Again, the only HGA variation that achieved noticeably better performance with respect to this metric was the time intensive BBAB. The similarities in performance between BBAA and BAAB extend to the number of successes within the individual layouts as can be seen in Table 4-6, Table 4-7 and Table 4-8.

It is important to recall that the HGA method was implemented using a randomly generated initial population. Experimentally, this approach was used to ensure that the performance of the HGA was not artificially influenced by performance of the MNS heuristic. In Figure 4-7 the best MNS results are superimposed on the HGA results to approximate the best performance if the population had been seeded with the layout created by the MNS heuristic. The approximated results represent the worst possible performance over the simulated data sets because they assume that the HGA was not able to further improve upon the seeded MNS result. From Figure 4-7, it is clear that the introduction of the MNS result as a member of the initial population would lead to decreases in both average error and maximum error.

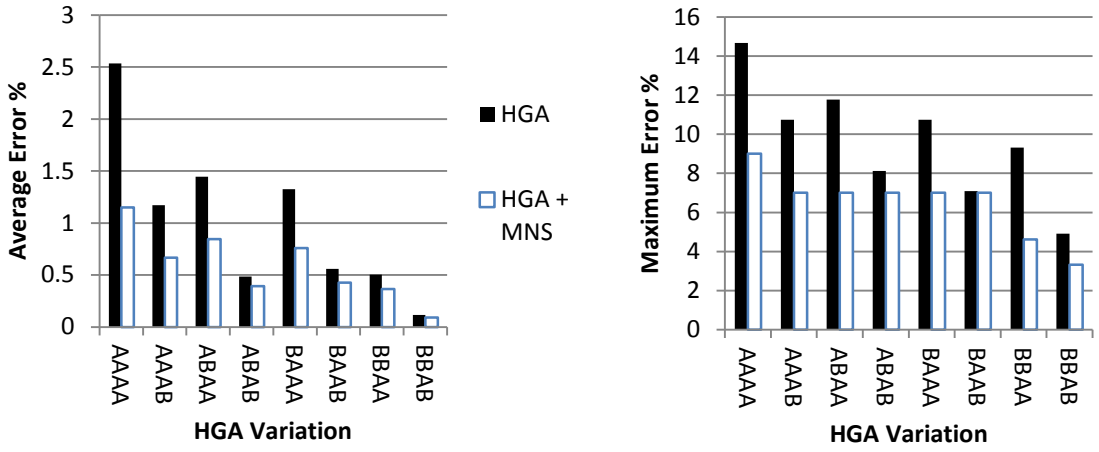


Figure 4-7 Average and maximum error over 20 trials for all HGA variations solving circular layouts before and after combination with MNS

4.4 Solution Robustness

4.4.1 Robustness Results

Three test scenarios were randomly generated as described in Chapter 3 and solved using the HGA heuristic. The problems with n=30 were sampled 5 times each. The problem with n=50 was only sampled once due to the increased computation time required for the larger scaled problem. Results for the three problems are shown in Table 4-10.

The optimum layout solutions found for these scenarios demonstrate differences in the structure of the problems. Consider the solutions found for n=30-p1. Five solutions were generated with equal values for the objective function. Solutions 1 and 3 are identical results in both value and permutation, but the remaining four permutations all appear to be unique with no common planes of symmetry or reflection resulting in the same permutation manifesting itself differently. In contrast, n=30-p2 produced five different solutions, each with a unique minimum value for the objective function. For n=50-p1, the solution from the MNS heuristic dominated the solution and remained as the minimum value after all iterations.

Distribution plots for each of the three problems are shown in Figure 4-8, Figure 4-9, and Figure 4-10. The distribution of the HGA produced layouts have been separated from the randomly generated layouts and appear superimposed on the same set of axes. This method prevents the number of good permutations found through the HGA from skewing the main distribution, while also allowing the difference between the means to be

highlighted. Since the results of the HGA produced permutations are in fact part of the same population of layouts, there position along the x-axis emphasizes their status as outliers.

LCRI Values corresponding to each solution of the test scenarios are presented in Table 4-12

Table 4-11 Layout candidates for robustness testing

Problem	HGA Trial	Min Rotation (HGA Result)	Permutation
n=30,P1	1	154281	13 7 16 5 19 4 28 27 10 22 11 24 1 29 9 20 17 12 15 14 6 21 8 26 25 30 2 18 3 23
	2	154281	4 10 1 12 28 13 19 20 7 25 6 23 16 18 8 27 30 5 2 3 11 26 9 21 22 17 15 29 14 24
	3	154281	13 7 16 5 19 4 28 27 10 22 11 24 1 29 9 20 17 12 15 14 6 21 8 26 25 30 2 18 3 23
	4	154281	10 16 7 18 4 19 25 26 13 1 12 29 22 24 14 3 6 11 8 9 17 2 15 27 28 23 21 5 20 30
	5	154281	8 14 5 16 2 17 23 24 11 29 10 27 20 22 12 1 4 9 6 7 15 30 13 25 26 21 19 3 18 28
n=30,P2	1	159257	17 20 30 2 27 23 22 14 4 28 12 3 21 25 26 8 24 18 10 29 9 16 5 19 15 7 1 11 6 13
	2	155994	22 15 5 1 6 14 13 24 7 16 19 11 26 28 8 2 27 21 17 18 12 20 9 4 23 29 30 10 3 25
	3	160451	21 27 19 14 17 11 23 20 5 1 28 3 18 15 8 12 13 25 26 30 2 24 6 7 29 10 16 4 9 22
	4	159214	28 4 19 22 24 25 30 27 12 8 5 10 26 23 15 21 20 3 29 7 9 1 13 14 6 18 17 11 16 2
	5	158105	4 30 2 15 12 11 7 28 22 9 25 5 8 27 13 21 20 26 3 6 24 1 14 23 29 18 16 10 17 19
n=50	1	620438	13 35 22 38 50 14 44 42 39 25 45 47 1 26 40 17 2 48 20 7 4 33 30 19 9 10 32 21 29 18 12 27 23 49 15 5 16 8 34 43 24 36 46 41 28 3 6 31 37 11

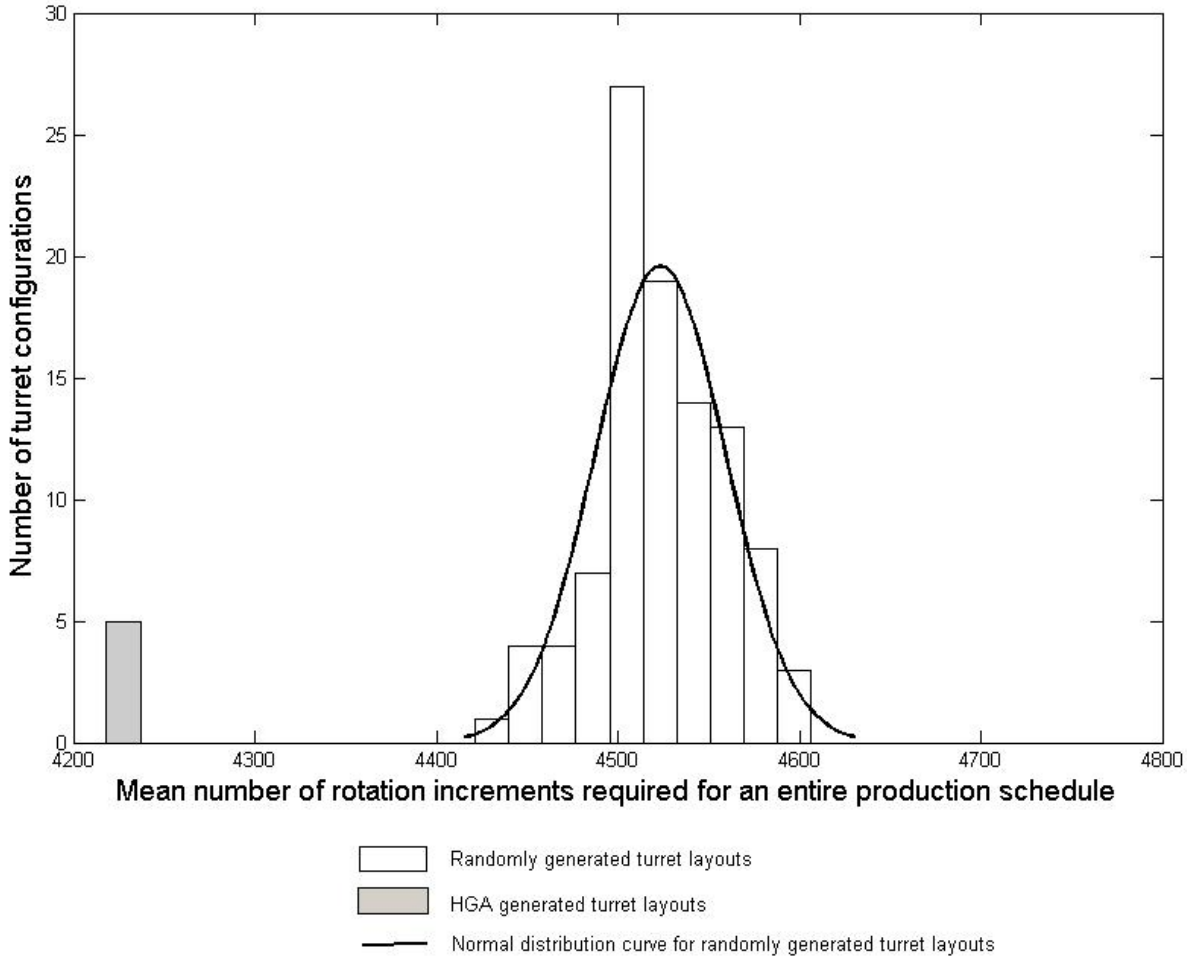


Figure 4-8 Distribution of random layouts as compared to distribution of HGA produced layouts for n=30,p1

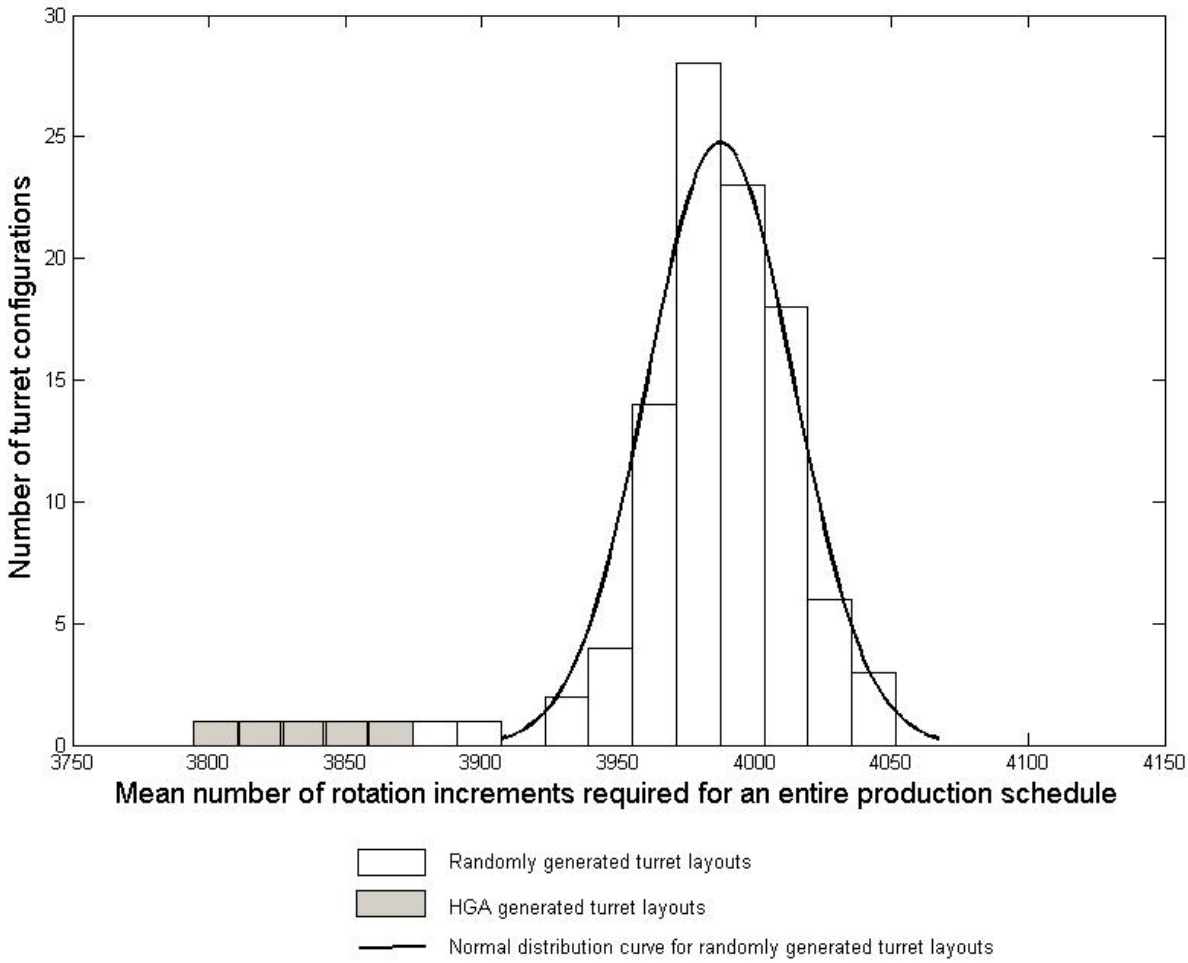


Figure 4-9 Distribution of random layouts as compared to distribution of HGA produced layouts for n=30,p2

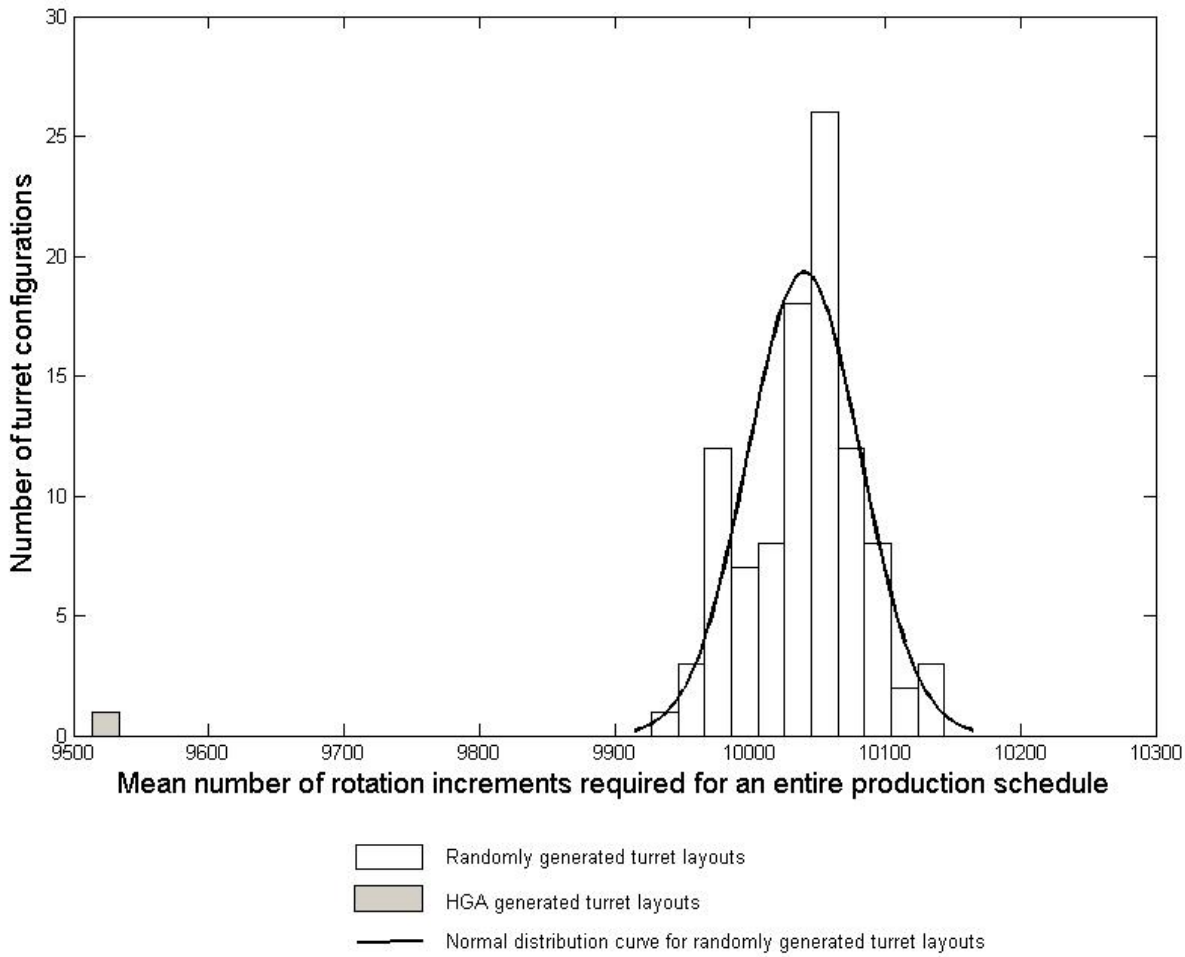


Figure 4-10 Distribution of random layouts as compared to distribution of HGA produced layouts for n=50,p1

Table 4-12 Results of robustness testing

Problem	Solution	Min Value	Z	LCRI
n=30,P1	1	154281	5.64	1
	2	154281	5.64	1
	3	154281	5.64	1
	4	154281	5.64	1
	5	154281	5.64	1
n=30,P2	1	159257	3.35	0.99960
	2	155994	3.11	0.99903
	3	160451	4.39	1
	4	159214	3.52	0.99978
	5	158105	2.80	0.9974
n=50,P1	1	620438	8.41	1

4.4.2 Robustness Discussion

In the LCRI values, it is interesting to see that the solutions to n=30-p1 that all had equal objective function values also resulted in equal LCRI values for all four of the unique permutations produced by the HGA. In contrast, each permutation solution to the n=30-p2 problem also produced a robust value, but each of the five robustness values was different. It is interesting that for n=30-p2, the best LCRI score does not correspond with the best objective function value for minimizing layout. In fact, it is related to the largest of the five HGA values which is equal to the initial seeded value that was found using the

MNS heuristic. This shows that for the other sample sets, it is possible that the robust solution found is not the *most* robust solution in the population of possible layouts.

It is positive to note that all LCRI values found were very close to 1.0. Braglia et al. (Braglia, et al., 2003) proposed LCRI=0.5 as a threshold for identifying a robust solution. By repeatedly exceeding the target of 0.5, it appears that the best solution found through the HGA heuristic described in Chapter 3 is also a robust solution.

Also from the distribution graphs, the improvement between the mean value and the mean of the robust solutions can be easily seen as the distance between means along the x-axis. From the graphs, it appears that an average improvement of ~5% across a random set of nests can be expected by applying the HGA metaheuristic described in Chapter 3 to the layout of a turret punch. This means that an operator applying this heuristic could see a 5% reduction in total turret rotation time. This does not translate to a 5% improvement in productivity since the conversion to productivity requires the turret rotation to be the slowest of the concurrent operations. Furthermore, the 5% difference is not intended to be representative of all turret punch operations, but only the simulated data sets used in this research. Further testing should be completed using actual data from a sheet metal parts manufacturer.

4.5 Implementation Issues and Future Research Opportunities

Through the course of this research, several practical topics pertaining to turret punches have been identified. This section proposes methods to accommodate these issues.

4.5.1 The impact of free moves

As shown in the robustness results, a turret layout developed using the HGA heuristic developed through this research is likely to decrease the overall turret rotation time by approximately 5%. The impact of this reduction on the overall performance of a turret punch is unknown.

One possibility is that the reduction in turret rotation leads to a negligible decrease in machine production time because the large turret movements are already occurring during long sheet travels. For these moves, the sheet travel time is dominant, and no reduction in overall time will be observed because the processing time was not influenced by the turret movement.

A second possibility is that the results will extend directly to the turret punch, with all time delays caused by turret dominance reduced by 5%.

A third possibility is that the improved layout of the turret will enable the CAM software to change tools more frequently. This will in turn allow the CAM software to pursue more efficient tool paths, such as ones that frequently change tools during short travels. If this change enables the operator to abandon tool grouping and choose tool paths that eliminate entire cycles around the material stock, the processing time will be improved greatly.

Future experiments should be conducted using actual industrial data and incorporating the CAM software used for nesting and tool path planning.

4.5.2 Tool Precedence

Tool Precedence constraints can be of two types: relative constraints and absolute constraints. An example of an absolute constraint is that the parting tool must be used last. This type of constraint can be very easily implemented by forcing the tool to be at the end of the TSP. Conversely, if a particular tool must be used first, it can be placed at the start. The procedure for sequencing the remaining tools does not change.

Relative constraints are a more difficult challenge. Assuming that the precedence constraint is associated with a part, it must propagate through the nest. One method to propagate this type of constraint from a part level to a nest level would be to modify the DP to eliminate branches that require an unusable sequence.

It is possible that the incorporation of tool precedence constraints will decrease any observed improvements in performance since they could force a part to be produced in an operation sequence that is not well suited to the layout. An interesting experiment would incorporate the impact of tool precedence constraints on the achievability of a robust turret layout.

4.5.3 Turret Station –Tool Compatibility

In many turret punches, the turret consists of several different sizes of stations for holding tools. Tools themselves are characterized by a physical diameter that restricts which stations they can be used in. A large tool cannot be assigned to a small station. However, a small tool can be assigned to a large station by equipping it with a sleeve. To prevent the infeasible assignment of tools to stations, it is suggested to use the cost matrix associated with the QAP. Recall that the QAP can be expressed in matrix form as:

$$TC = F \cdot XDX^T + CX$$

If the C matrix is populated with a large value in the cells corresponding to the assignment of tool *i* to location *j*, where tool *i* is incompatible with location *j*, the assignment will be effectively prohibited. All QAP heuristics tested in this research can easily accommodate a non-zero C matrix.

The incorporation of tool location constraints will likely increase the total amount of turret rotation that is required. However, if the randomly generated layouts are also forced to adhere to the same constraints, it is possible that the improvement observed in the current research will be maintained since any long turret movements required by immovable tools will be required equally by both the randomly generated layout and the heuristic solution layout.

The experiments in this research could be repeated with location constraints to determine the degree to which location constraints change the amount of improvement that is possible.

4.5.4 Multi-Tools

An increasingly common sight in sheet metal factories is the use of a multi-tool within an indexing station of the turret. The multi-tool is a small turret that is positioned within a large turret that effectively allows several small tools to operate at a single turret position. Modelling these locations can be easily accommodated using the distance matrix. From locations outside of the multi-tool to locations within the multi-tool, the distance can be approximated as being the same as the rotational distance required to move from the start location to the tool station holding the multi-tool. The distance between multi-tool positions can be modelled as either zero or a small value estimated based on the rotational speed of the multi-tool. Due to the relatively small size compared to the main turret, multi-tool rotational speed is typically much faster than the rotational speed of the main turret.

CHAPTER 5: CONCLUSIONS

5.1 Conclusions

The production of nested parts out of sheet metal using a CNC turret punch was studied and modelled as a facility layout problem with sequence flexibility. In order to solve the research objective problem of creating a robust turret for the production of all nests, the problem was first simplified to one of finding a single turret layout that could be used to efficiently produce the expected demand levels of all required parts given that the tool use sequence within each part is flexible.

Three heuristics used for similar problems in PCB manufacturing, turret punch layout and cellular manufacturing were adapted and combined to create an effective heuristic for solving the reduced problem. The resulting heuristic was a hybrid genetic algorithm in which candidate layout solutions were found using standard GA operators and then improved through iterative solution of TSP and QAP subproblems. The improved layout solutions were then added to the population instead of their unimproved starting points.

Based on the combination of good solution time, good average accuracy, low maximum error and high number of successful solutions, the HGA variant with parameter settings shown in Table 5-1 below was found to produce the best results. Where appropriate, parameter levels were set relative to *n*, the number of tool locations in the turret layout problem.

Table 5-1 Best performing parameter combination for HGA method

	Level
Population Size	$4\sqrt{n}$
Number of Generations	n
Local Search Iterations	2
ULX Crossovers	3

To improve performance, the initial population was seeded with a layout created using modified neighbourhood strength heuristic adapted from a method used by Grunow, et al. (2004).

The suitability of using the reduced problem to approximate the research objective was tested by adapting the LCRI metric to measure the robustness of heuristic achieved solutions for three large scale test problems built on simulated data. LCRI results conclusively showed that a turret layout found by reducing the problem and applying the HGA method described above produced turret layouts that could consistently produce simulated production schedules using fewer total turret rotations than would be expected from a range of randomly generated turret layouts. For the simulated part sets, the improvement was seen to be in the range of 5%. This result shows the validity of the solution approach.

5.2 Contributions

The completion of this work resulted in the following contributions to the field of operations research:

1. The introduction of a modified neighbourhood strength heuristic that can be used for finding common flows among groups of processes that are individually characterized by multiple, equivalent minimum spanning trees.
2. The implementation of a hybrid genetic algorithm for finding the optimum layout for a group of elements when the sequence of elements is highly flexible.
3. The application and adaptation of LCRI, a robustness measurement tool, as a measurement tool for the suitability of turret punch layouts.

5.3 Future Research

Considerable efforts were taken during the completion of this work to reduce the computational time required for each element of the heuristic. However, the processing time is still considerable. Since the heuristic is intended as a design tool, the lengthy computational time can be accepted for moderately sized problems. Further work should be completed to explore performance improvements that may occur by only applying the iterative TSP/QAP improvement method to a portion of the population during the execution of the HGA.

In the production of single parts, it is easy to estimate the total production time since all required machine operations are known. For nested parts, the total production time cannot

be estimated until the nest is created immediately prior to production. Consequently, it is impossible to know the true benefit that can be realized by reducing total turret rotation. Future research should measure the actual effects of reducing turret rotation in a real production environment.

The simulated part sets used in the testing of the heuristic do not necessarily represent the characteristics of all sheet metal parts produced by all manufacturers. The projected benefit from implementing a robust turret for different types of products should be explored.

REFERENCES

Alkaya, A. F. & Duman, E., 2015. Combining and solving sequence dependent traveling salesman and quadratic assignment problems in PCB assembly. *Discrete Applied Mathematics*, Volume 192, pp. 2-16.

Amada GmbH, 2011. *Turret punch press AE-255 NT AE-2510 NT*. [Online] Available at: http://www.amada.de [Accessed 25 October 2014].

Applegate, D. L., Bixby, R. E., Chvatal, V. & Cook, W. J., 2006. *The Traveling Salesman Problem: A Computational Study*. Princeton, NJ: Princeton University Press.

Armour, G. C. & Buffa, E. S., 1963. A heuristic algorithm and simulation approach to relative location of facilities.. *Manag Sci*, 9(1), pp. 294-309.

Ayob, M. & Kendall, G., 2008. A survey of surface mount device placement machine optimisation: Machine classification. *European Journal of Operational Research*, Volume 186, pp. 893-914.

Balakrishnan, J. & Cheng, C. H., 1998. Dynamic Layout Algorithms : A State-of-the-art Survey. *Omega*, 26(4), pp. 507-521.

Ball, M. & Magazine, M. J., 1988. Sequencing of Insertions in Printed Circuit Board Assembly. *Operations Research,* 36(2), pp. 192-201.

Bard, J. F., Clayton, R. W. & Feo, T. A., 1994. Machine Setup and Component Placement in Printed Circuit Board Assembly. *The International Journal of Flexible Manufacturing Systems,* Volume 6, pp. 5-31.

Bellman, R., 1962. Dynamic Programming Treatment of the Travelling Salesman Problem. *Journal of the ACM,* 9(1), pp. 61-63.

Benjaafar, S., 2002. Modeling and analysis of congestion in the design of facility layouts. *Management Science,* 48(5), pp. 679-704.

Binder, G., 2008. Pushing nesting efficiency for the punch press. *The Fabricator*, 13 May.

Braglia, M., Zanoni, S. & Zavanella, L., 2003. Layout design in dynamic environments: Strateges and quantitative indices. *International Journal of Production Research,* 41(5), pp. 995-1016.

Braglia, M., Zanoni, S. & Zavanella, L., 2005. Robust versus stable layout design in stochastic environments. *Production Planning & Control,* 16(1), pp. 71-80.

Brixius, N., 2000. *Solving Large Scaled Quadratic Assignment Problems, Ph.D Thesis.* Iowa City: The University of Iowa.

Buffa, E. S., Armour, G. C. & Vollmann, T. E., 1964. Allocating Facilities with CRAFT. *Harvard Business Review,* 42(2), pp. 136-158.

Burkard, R., Dell'Amico, M. & Martello, S., 2009. *Assignment Problems.* Philadelphia: Society for Industrial and Applied Mathematics.

Carmon, T. F., Maimon, O. Z. & Ezey, M. D.-E., 1989. Group set-up for printed circuit board assembly. *International Journal of Production Research,* 27(10), pp. 1795-1810.

Caux, C., Bruniaux, R. & Pierreval, H., 2000. Cell formation with alternative process plans and machine capacity constraints: A new combined approach. *International Journal of Production Economics,* Volume 64, pp. 279-284.

Chang, C. C., Tai-Hsi, W. & Chien-Wei, W., 2013. An efficient approach to determine cell formation cell layout and intracellular machine sequence in cellular manufacturing systems. *Computers & Industrial Engineering,* Volume 66, pp. 438-450.

Chauny, F., Haurie, A., Wagneur, E. & Loulou, R., 1987. Sequencing Punch Operations in a Flexible Manufacturing Cell a Three Dimensional Space-filling Curve Approach. *INFOR*, 25(1), pp. 26-45.

Chaves, A., Lorena, L., Senne, E. & Resende, M., 2016. Hybrid method with CS and BRKGA applied to the minimization of tool switches problem. *Computers & Operations Research*, Volume 67, pp. 174-183.

Chen, W.-S. & Chyu, C.-C., 2002. Reduction of printed circuit board group assembly time through the consideration of efficieny loss of placement time. *Assembly Automation*, 22(4), pp. 360-370.

Cho, K. H. & Lee, K., 1993. Tool Path Optimization for NC Turret Operation using Simulated Annealing. *American Society of Mechanical Engineers, Production Engineering Division (Publication) PED*, Volume 64, pp. 315-321.

Chung, S.-H., Wu, T.-H. & Chang, C.-C., 2011. An efficient tabu search algorithm to the cell formation problem with alternative routings and machine reliability considerations. *Computers & Industrial Engineering*, Volume 60, pp. 7-15.

Chyu, C.-C. & Chang, W.-S., 2008. A genetic-based algorithm for the operational sequence of a high speed chip placement machine. *International Journal of Advanced Manufacturing Technology*, Volume 36, pp. 918-926.

Crama, Y., Flippo, O. E., van de Klundert, J. & Spieksma, F. C., 1997. The assembly of printed circuit boards: A case with multiple machines and multiple board types. *European Journal of Operational Research*, Volume 98, pp. 457-472.

Crama, Y., Moonen, L. S., Spieksma, F. C. & Talloen, E., 2007. The tool switching problem revisited. *European Journal of Operational Research,* Volume 182, pp. 952-957.

Crama, Y., van de Klundert, J. & Spieksma, F. C., 2002. Production planning problems in printed circuit board assembly. *Discrete Applied Mathematics,* Volume 123, pp. 339-361.

De Souza, R. & Lijun, W., 1995. Intelligent optimization of component onsertion in multi-head concurrent operation PCBA machines. *Journal of Intelligent Manufacturing,* Volume 6, pp. 235-243.

Dereli, T. & Filiz, I. H., 2000. Allocating optimal index positions on tool magazines using genetic algorithms. *Robotics and Autonomous Systems,* Volume 33, pp. 155-167.

Dikos, A., Nelson, P. C., Tirpak, T. M. & Wang, W., 1997. Optimization of high-mix printed circuit card assembly using genetic algorithms. *Annals of Operations Research,* Volume 75, pp. 303-324.

Drezner, A. & Misevicius, A., 2013. Enhancing the performance of hybrid genetic algorithms by differential improvement. *Computers & operations Research,* Volume 40, pp. 1038-1046.

Drezner, A. & Nof, S. Y., 1984. On Optimizing Bin Picking and INsertion Plans for Assembly Robots. *IIE Transactions,* 16(3), pp. 262-270.

Drezner, Z., 2003. A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing,* 15(3), pp. 320-330.

Drezner, Z., 2008. Extensive experiments with hybrid genetic algorithms for the solution of the quadratic assignment problem. *Computers & Operations Research,* Volume 35, pp. 717-736.

Drira, A., Perreval, H. & Hajri-Gabouj, S., 2007. Facility layout problems: A survey. *Annual Reviews in Control,* Volume 31, pp. 255-267.

Duman, E. & Or, I., 2007. The quadratic assignment problem in the context of the printed circuit board assembly process. *Computers & Operations Research,* Volume 34, pp. 163-179.

Ellis, K. P., Vittes, F. J. & Kobza, J. E., 2001. Optimizing the Performance of a Surface Mount Placement Machine. *IEEE Transactions on Electronics Packaging Manufacturing,* 24(3), pp. 160-170.

Fleurent, C. & Ferland, J., 1994. Genetic hybrids for the quadratic assignment problem. In: P. Pardalos & H. Wolkowicz, eds. *Quadratic Assignment and Related Problems DIMACS Series in Discrete Mathemeatics and Theoretical Computer Science, vol.16.* Providence: AMS, pp. 173-188.

Foulds, L. & Hamacher, H., 1993. Optimal bin location and sequencing in printed circuit board assembly. *European Journal of Operational Research,* Volume 66, pp. 279-290.

Francis, R. L. & White, J. A., 1974. *Facility Layout and Location - an Analytical Approach.* 1st ed. Englewood Cliffs, NJ: Prentice-Hall Inc..

Ghaiebi, H. & Solimanpur, M., 2007. An ant algorithm for optimization of hole-making operations. *Computers & Industrial Engineering,* Volume 52, pp. 308-319.

Gilmore, P., 1962. Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem. *Journal of the Society for Industrial and Applied Mathematics,* 10(2), pp. 305-313.

Grunow, M., Gunther, H.-O., Schleusener, M. & Yilmaz, I. O., 2004. Operations planning for collect-and-place machines in PCB assembly. *Computers & Industrial Engineering,* Volume 47, pp. 409-429.

Gupta, S. K. & Rosenhead, J., 1968. Robustness in Sequential Investment Decisions. *Management Science,* 15(2), p. B18.

Gutin, G. & Punnen, A. P., 2007. *The Traveling Salesman Problem: Applications, Formulations and Variations.* New York, New York: Springer Science+Business Media, LLC.

Hahn, P. & Anjos, M., 2011. *QAPLIB - A Quadratic Assignment Problem Library*. [Online]
Available at: http://anjos.mgi.polymtl.ca/qaplib/ [Accessed 19 02 2016].

Held, M. & Karp, R. M., 1962. A Dynamic Programming Approach to Sequencing Problems. *Journal of the Society for Industrial and Applied Mathematics,* 10(1), pp. 196-210.

Herrmann, J. W. & Delalio, D. R., 2001. Algorithms for Sheet Metal Nesting. *IEEE Transactions on Robotics and Automation,* 17(2), pp. 183-190.

Ho, W. & Ji, P., 2003. Component scheduling for chip shooter machines: a hybrid genetic algorithm approach. *Computers & Operations Research,* Volume 30, pp. 2175-2189.

Ho, W. & Ji, P., 2004. A hybrid genetic algorthm for component sequencing and feeder arrangement. *Journal of Intelligent Manufacturing,* Volume 15, pp. 307-315.

Ho, W. & Ji, P., 2006. A genetic algorithm approach to optimising component placement and retrieval sequence for chip shooter machines. *International Journal of Advanced Manufacturing Technology,* Volume 28, pp. 556-560.

Jithavech, I. & Krishnan, K. K., 2010. A simulation based approach for risk assessment of facility layout designs under stochastic product demands. *Int J Adv Manuf Technol*, Volume 49, pp. 27-40.

Khoo, L. & Loh, K., 2000. A Genetic Algorithms Enhanced Planning System for Surface Mount PCB Assembly. *The International Journal of Advanced Manufacturing Technology*, Volume 16, pp. 289-296.

Klomp, C., van de Klundert, J., Spieksma, F. C. & Voogt, S., 2000. The feeder rack assignment problem in PCB assembly: A case study. *International Journal of Production Economics*, pp. 399-407.

Konak, A., Kulturel-Konak, S. & Azizoglu, M., 2008. Minimizing the number of tool switching instants in Flexible Manufacturing Systems. *International Journal of Production Economics*, Volume 116, pp. 298-307.

Koopmans, T. & Beckmann, M., 1957. Assignment Problems and the Location of Economic Activities. *Econometrica*, 25(1), pp. 53-76.

Koşucuoğlu, D. & Bilge, Ü., 2012. Material handling considerations in the FMS loading problem with full routing flexibility. *International Journal of Production Research*, 50(22), pp. 6530-6552.

Kouvelis, P. & Kiran, A. S., 1991. Single and multiple period layout models for automated manufacturing systems. *European Journal of Operational Research,* Volume 52, pp. 300-314.

Kouvelis, P., Kurawarwala, A. A. & Gutierrez, G. J., 1992. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research,* Volume 63, pp. 287-303.

Krishna, A. G. & Rao, K. M., 2006. Optimal allocation of index positions on tool magazines using an ant colony algorithm. *Int J Adv Manuf Technol,* Volume 30, pp. 717-721.

Krishnan, K. K., Jithavech, I. & Liao, H., 2009. Mitigation of risk in facility layout design for single and multi-period problems. *International Journal of Production Research,* 47(21), pp. 5911-5940.

Kulturel-Konak, S., 2007. Approaches to uncertainties in facility layout problems: Perspectives at the beginning of the 21st Century. *Journal of Intelligent Manufacturing,* 18(2), pp. 273-284.

Kumar, R. & Li, H., 1995. Integer Programming Approach to Printed Cricuit Board Assembly Time Optimization. *IEEE Transactions on Components, Packaging and Manufacturing Technology-Part B,* 18(4), pp. 720-727.

Kumar, R. & Luo, Z., 2003. Optimizing the Operation Sequence of a Chip Placement Machine Using TSP Model. *IEEE Tranactions on Electronics Packaging Manufacturing,* 26(1), pp. 14-21.

Kumar, S. & Veeramani, D., 1995. Determination of turret configuration and hit sequence for optimization of NC turret punch press optimization. *International Journal of Production Research,* 33(4), pp. 1071-1089.

Kusiak, A., 1987. The generalized group technology concept. *International Journal of Production Research,* 25(4), pp. 561-569.

Kusiak, A. & Heragu, S. S., 1987. The facility layout problem. *European Journal of Operational Research,* Volume 29, pp. 229-251.

Lawler, E. L., 1963. The Quadratic Assignment Problem. *Management Science,* 9(4), pp. 586-599.

Lee, M. K., Luong, H. & Abhary, K., 1997. A genetic algorithm based cell design considering alternative routing. *Computer Integrated Manufacturing Systems,* 10(2), pp. 93-107.

Leipälä, T. & Nevalainen, O., 1989. Optimization of the movements of a component placement machine. *European Journal of Operational Research,* Volume 38, pp. 167-177.

Leu, M., Wong, H. & Ji, Z., 1993. Planning of Component Placement/Insertion Sequence and Feeder Setup in PCB Assembly Using Genetic Algorithm. *Transactions of the ASME,* Volume 115, pp. 424-432.

Levine, D. M., Ramsey, P. P. & Smidt, R. K., 2001. *Applied Statistics for Engineers and Scientists.* New Jersey: Prentice Hall.

Lin, E. Y.-H., 1998. On the global optimum of the numerical controlled punch press too location and hit sequencing problem. *Engineering Optimization,* 31(1), pp. 53-64.

Lin, G. Y.-J. & Solberg, J. J., 1991. Effectiveness of Flexible Routing Control. *The International Journal of Flexible Manufacturing Systems,* Volume 3, pp. 189-211.

Loiola, E. et al., 2006. A survey for the quadratic assignment problem. *European Journal of Operational Research,* Volume 176, pp. 657-690.

Lowry, D., 2011. Leaning on the turret punch press. *The Fabricator,* May, pp. 50-53.

Marvizadeh, S. Z. & Choobineh, F. F., 2013. Reducing the number of setups for CNC punch presses. *Omega,* 41(2), pp. 226-235.

Meller, R. D. & Gau, K.-Y., 1996. The facility layout problem: Recent and emerging trends and perspectives. *Journal of Manufacturing Systems,* 15(5), pp. 351-366.

Menger, K., 1932. Das botenproblem. *Ergebnisse Eines Mathematischen Kolloquiums,* Volume 2, pp. 11-12.

Misevicius, A., 2004. An improved hybrid genetic algorithm: new results for the quadratic assignment problem. *Knowledge-Based Systems,* Volume 17, pp. 65-73.

Misevicius, A., 2005. A Tabu Search Algorithm for the Quadratic Assignment Problem. *Computational Optimization and Applications,* Volume 30, pp. 95-111.

Misevicius, A., 2012. An implementation of the iterated tabu search algorithm for the quadratic assignment problem. *OR Spectrum,* Volume 34, pp. 665-690.

Misevicius, A. & Guogis, E., 2012. Computational Study of Four Genetic Algorithm Variants for Solving the Quadratic Assignment Problem. In: T. Skersys, R. Butleris & R. Butkiene, eds. *Information and Software Technologies 18th International Conference, ICIST 2012, Kaunas, Lithuania, September 13-14, 2012. Proceedings.* Berlin: Springer Berlin Heidelberg, pp. 24-37.

Murata Machinery, Ltd., 2014. *Motorum 2048TS 2044TS.* [Online] Available at: http://www.turretpunchpresses.com [Accessed 25 October 2014].

Nayanzin, N. & Romanov, V. F., 1988. Optimum arrangement of tools in a machining centre magazine. *Soviet Engineering Research,* 8(1), pp. 81-83.

Nelson, K. & Wille, L., 1995. *Comparative Study of Heuristics for Optimal Printed Circuit Board Assembly.* Fort Lauderdale, IEEE, pp. 322-327.

Niemi, E., 2003. Optimal tool magazine operation. Part 1: Rotating magazines with unbuffered tool change. *Int J Adv Manuf Technol,* Volume 22, pp. 720-726.

Niemi, E., 2004. Optimal tool magazine operation. Part 2: rotating magazines with buffered tool change. *Int J Adv Manuf Technol,* Volume 24, pp. 567-572.

Norman, B. A. & Smith, A. E., 2006. A continuous approach to considering uncertainty in facility design. *Computers & Operations Research,* Volume 33, pp. 1760-1775.

Nsakanda, A. L., Diaby, M. & Price, W. L., 2006. Hybrid genetic approach for solving large-scale capacitated cell formation problems with multiple routings. *European Journal of Operational Research,* Volume 171, pp. 1051-1070.

Ong, N.-S. & Khoo, L. P., 1999. Genetic algorithm approach in PCB assembly. *Integrated Manufacturing Systems,* 10(5), pp. 256-265.

Ong, N.-S. & Tan, W.-C., 2002. Sequence placement planning for high-speed PCB assembly machine. *Integrated Manufacturing Systems,* 13(1), pp. 35-46.

Palekar, U., Batta, R., Bosch, R. & Elhence, S., 1992. Modeling uncertainties in plant layout problems. *European Journal of Operational Research,* 63(2), pp. 347-359.

Punnen, A. P., 2007. The Traveling Salesman Problem: Applications, Formulations and Variations. In: A. P. Punnen & G. Gutin, eds. *The Traveling Salesman Problem and its Variations.* New York: Springer Science+Business LLC, pp. 1-28.

Rardin, R. L. & Uzsoy, R., 2001. Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics,* Volume 7, pp. 261-304.

Ripka, J., 2014. Maximum sheet metal punching, minimal distortion. *The Fabricator*, 17 December.

Rosenblatt, M. J. & Kropp, D. H., 1992. The Single Period Stochastic Plant Layout Problem. *IIE Transactions,* 24(2), pp. 169-176.

Rosenblatt, M. J. & Lee, L. H., 1987. A robustness approach to facilities design. *International Journal of Production Research,* 25(4), pp. 479-486.

Rosenhead, J., Elton, M. & Gupta, S. K., 1972. Robustness and Optimality as Criteria for Strategic Decisions. *Operational Research Quarterly,* 23(4), pp. 413-431.

Roychoudhury, B. & Muth, J. F., 1995. Tool Path Optimization Procedures for Machine Tools. *Computers & Industrial Engineering,* 28(2), pp. 367-377.

Sahni, S. & Gonzales, T., 1976. P-complete approximation problems. *Journal of the Association for Computing Machinery,* pp. 555-565.

Sethi, A. K. & Sethi, S. P., 1990. Flexibility in Manufacturing: A Survey. *The International Journal of Flexible Manufacturing Systems,* Volume 2, pp. 289-328.

Shin, W. A., Bullington, S. F. & Causey, D. W., 1990. A procedure for scheduling punch presses with restrictions on the sequence of punches. *International Journal of Production Research,* 28(4), pp. 723-734.

Shore, R. H. & Tompkins, J. A., 1980. Flexible Facilities Design. *AIIE Transactions,* 12(2), pp. 200-205.

Sivanandam, S. & Deepa, S., 2008. *Introduction to Genetic Algorithms.* Berlin Heidelberg New York: Springer.

Smith, D. K., 1991. *Dynamic Programming A Practical Introduction.* New York, NY: Ellis Horwood.

Sohn, J. & Park, S., 1996. Efficient operation of a surface mounting machine with a multihead turret. *International Journal of Production Research,* Volume 34, pp. 1131-1143.

Steinberg, L., 1967. The backboard wiring problem: A placement algorithm. *SIAM Review,* Volume 3, pp. 37-50.

Stützle, T., 2006. Iterated local search for the quadratic assignment problem. *Discrete Optimization,* Volume 174, pp. 1519-1539.

Su, Y.-Y. & Srihari, K., 1996. Placement Sequence Identification Using Artifical Neural Networks in Surface Mount PCB Assembly. *The International Journal of Advanced Manufacturing Technology,* Volume 11, pp. 285-299.

Svestka, 1990. Imposing precedence constraints on NC punch press sequences. *International Journal of PRoduction Research,* 28(12), pp. 2309-2319.

Svestka, J. A., Dornfeld, D. A. & Gil, R. A., 1981. On improving the productivity of numerically controlled punch presses. *International Journal of Production Research,* 19(5), pp. 471-480.

Taillard, E., 1991. Robust taboo search for the quadratic assignment problem. *Parallel Computing,* Volume 17, pp. 443-455.

Taillard, E. D., 1995. Comparison of Iterative Seraches for the Quadratic Assignment Problem. *Location Science,* 3(2), pp. 87-105.

Tate, D. & Smith, A., 1995. A genetic approach to the quadratic assignment problem. *Computers & Operations Research,* 22(1), pp. 73-83.

Tlusty, G., 2000. *Manufacturing Processes and Equipment.* Upper Saddle River, NJ: Prentice Hall.

Veeramani, D. & Kumar, S., 1998. Optimization of the nibbling operation on an NC turret punch press. *International Journal of Production Research,* 36(7), pp. 1901-1916.

Vin, E. & Delchambre, A., 2014. Generalized cell formation: iterative versus simultaneous resolution with grouping genetic algorithm. *Journal of Intelligent Manufacturing,* Volume 25, pp. 1113-1124.

Vollmann, T. W., Nugent, C. E. & Zartler, R. L., 1968. A Computerized Model for Office Layout. *The Journal of Industrial Engineering,* 19(7), pp. 321-329.

Walas, R. A. & Askin, R. G., 1984. An Algorithm for NC Turret Punch Press Tool Location and Hit Sequencing. *IIE Transactions,* 16(3), pp. 280-287.

Wang, G. & Xie, S., 2005. Optimal process planning for a combined punch-and-laser cutting machine using ant colony optimization. *International Journal of Production Research,* 43(11), pp. 2195-2216.

Wu, Y. & Ji, P., 2010. *Optimizing Feeder Arrangement for a PCB Assembly Machine for Multiple Boards.* Macao, IEEE, pp. 2343-2347.

Wu, Y., Ji, P. & Ho, W., 2009. Optimizing PCB assembly for family setup strategy. *Assembly Automation*, pp. 61-67.

Xie, S. S. & Tu, Y., 2011. Optimal Process Planning for Compund Laser Cutting and Punching Using Genetic Algorithm. In: *Rapid One-of-a-kind Product Developement.* London: Springer, pp. 309-326.

Yang, Z., Shirwaiker, R. A. & Wysk, R. A., 2011. Optimization of Manufacturing Operations Using a New Interactive Computer-Aided Process Planning Program. In: T. Doolen & E. Van Aken, eds. *Proceedings of the 2011 Industrial Engineering Research Conference.* s.l.:s.n.

Yeo, S., C.W., L. & K.H.Yong, 1996. A Rule-Based Frame System for Concurrent Assembly Machines. *The International Journal of Advanced Manufacturing Technology*, Volume 12, pp. 370-376.

Yilmaz, I., Grunow, M., Gunther, H.-O. & Yapan, C., 2007. Development of group setup strategies for makespan minimisation in PCB assembly. *International Journal of Production Research*, 45(4), pp. 871-897.

Yilmaz, I. O., 2008. Development of Group Strategies. In: *Development and Evaluation of Setup Strategied in Printed Circuit Board Assembly.* Wiesbaden: Gabler, pp. 75-122.

Yin, Y. & Yasuda, K., 2002. Manufacturing cells' design in consideration of various production factors. *International Journal of Production Research,* 40(4), pp. 885-906.

Zhao, C. & Wu, Z., 2000. A genetic algorithm for manufacturing cell formation with multiple routes and multiple objectives. *International Journal of Production Research,* 38(2), pp. 385-395.

Zhong, P., Yu, Y., Xin, H. & Lu, Z., 2014. Method of Holes Machining Path Optimization for NC Turret Punch Press. *Advanced Materials Research* , Volume 842, pp. 553-557.

APPENDIX A: PART SETS USED FOR TESTING HEURISTIC METHODS

Part Set 1

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tool 6***	***Permutations***
1	83	4	9	2			3	3
2	99	8	3				2	1
3	14	7	3	5			3	3
4	95	6	1	3	7		4	12
5	77	1	8	7			3	3
6	9	7	2	6	8		4	12
							Combined Permutations:	3888

Part Set 2

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	77	1	3	9	6	8	5	60
2	84	5	8	7	1		4	12
							Combined Permutations:	720

Part Set 3

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	38	4	9	5	8	7	5	60
2	64	5	9	2	1	3	5	60
							Combined Permutations	3600

Part Set 4

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	77	3	5	9	7		4	12
2	88	1	6	4	8	9	5	60
3	60	6	3				2	1
4	43	1	5	6	9		4	12
5	12	9	6	3			3	3
							Combined Permutations	25920

Part Set 5

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	30	6	3	2	1		4	12
2	21	1	3	5			3	3
3	22	5	9	2	7		4	12
4	91	2	9	4	6	3	5	60
							Combined Permutations	25920

Part Set 6

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	38	6	2	5	1	8	5	60
2	76	5	3	1			3	3
3	50	2	3	8	4	1	5	60
							Combined Permutations	10800

Part Set 7

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	27	1	8	3			3	3
2	71	8	4	5	3		4	12
3	30	7	5	1	3		4	12
4	85	5	6	2	3		4	12
5	63	1	4	9			3	3
							Combined Permutations	15552

Part Set 8

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	23	4	9				2	1
2	100	8	9				2	1
3	41	4	7	6			3	3
4	78	2	8	5	4	6	5	60
5	67	6	2	7	8	9	5	60
							Combined Permutations	10800

Part Set 9

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	97	1	2				2	1
2	66	1	6	8			3	3
3	91	5	4				2	1
4	89	5	1	4			3	3
5	64	2	6	7	8		4	12
6	89	4	2	3	9	7	5	60
7	96	2	6				2	1
8	9	6	2	4			3	3
							Combined Permutations	19440

Part Set 10

Part	***Demand***	***Tool 1***	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tools***	***Permutations***
1	60	4	1				2	1
2	31	9	4	3	8	6	5	60
3	14	5	8	7			3	3
4	55	4	2	8	6		4	12
5	73	2	1	8	3		4	12
							Combined Permutations	25920

APPENDIX B: PARTS USED FOR ROBUSTNESS MEASUREMENTS

Part Set n=30,p1 – Parts 1-25

Part	*Area (m^2)*	*Demand*	*Tools Used*							
			Tool 1	*Tool 2*	*Tool 3*	*Tool 4*	*Tool 5*	*Tool 6*	*Tool 7*	*Tool 8*
1	1.179	247	14	9	15	24	23	8		
2	1.611	191	25	1	27	6				
3	0.247	406	17	22						
4	1.287	414	21	30	13					
5	1.106	497	26	8	10	12				
6	0.564	283	3	7	26	5				
7	0.855	328	29	27	4	7	25	13	8	
8	1.114	362	24	22	7					
9	2.193	252	9	11	15	19	17			
10	1.923	340	30	18	29	11	13	2		
11	0.550	104	29	14	4					
12	2.012	365	28	3	26	15	9	7	21	23
13	1.242	239	1	11						
14	0.979	350	20	1	28					
15	1.238	250	9	4	15	18	20	17	11	
16	0.932	217	25	27	29	4				
17	0.959	60	20	3	25	19	26			
18	1.497	350	24	28	13					
19	1.470	453	11	30	16	22	10			
20	1.472	182	16	7	2	12				
21	1.224	144	4	15	26	28	25	11		
22	0.598	396	2	6	23	17	28	29	1	18
23	1.239	217	17	10	13	14	8	28		
24	1.543	27	26	28	7	19	10	23		
25	1.163	172	18	27	22	13	14	25		

Part Set n=30,p1 – Parts 26-50

Part	Area (m^2)	Demand	Tools Used							
			Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	Tool 8
26	1.345	145	17	6						
27	1.242	121	18	8						
28	0.899	493	22	10	6	26	24			
29	1.098	3	29	10	28	15	1	24	16	
30	0.738	130	5	1						
31	1.296	105	16	10	4	29	21	17	24	18
32	0.618	324	27	26	7	14	29	6	13	
33	0.644	338	14	26	23	18				
34	0.730	50	25	3	9	27	1	14		
35	0.050	215	16	28	12	9	3	1	19	
36	1.479	497	17	12	10	25	30	24	16	
37	1.108	136	5	21	18	2	23			
38	0.748	95	2	24	10	3	21			
39	1.457	480	21	6	8	29	16	2	24	23
40	0.429	268	21	25	3	18	9	28	10	1
41	0.966	161	1	28	5	18	19	25		
42	0.920	131	14	9						
43	1.106	253	15	2						
44	1.104	497	28	24	30	25	14	8	12	7
45	0.712	131	27	24						
46	0.990	381	12	7						
47	0.945	475	11	27	13	9	6	4	21	
48	1.209	322	21	25	10	30	5			
49	1.364	300	2	7	13	25				
50	1.370	59	4	26	24	10	5	15		

Part Set n=30,p2 – Parts 1-25

Part	Area (m^2)	Demand	Tools Used							
			Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	Tool 8
1	0.712	4	18	7	24	27	20	14	13	17
2	1.026	77	8	25	27	2	3	28		
3	0.595	376	10	5	6					
4	0.629	137	13	17	22	16	27	3	23	8
5	0.998	387	10	7	21	2	30	19		
6	1.511	249	12	27	16	15	21	23	5	
7	0.743	378	13	23	6	12	22	28	30	
8	1.124	403	25	22						
9	0.925	319	7	1	12	14	5	16		
10	1.372	32	27	12	28	23	5	9	26	
11	0.637	388	27	17	9	23	4	14	22	5
12	1.011	261	3	24	29					
13	1.184	158	12	20						
14	1.367	80	5	15	22	20	1	7		
15	1.515	83	8	9	12	16	28	24		
16	1.029	80	3	20	4	9	16	29	19	
17	0.503	105	6	11	4	15	18	24		
18	0.753	421	27	26	29					
19	0.646	426	11	18	22	2	19			
20	1.783	155	16	22	17	27	5			
21	0.795	247	5	23						
22	1.249	354	20	3	7	5	22			
23	0.936	392	14	24	15	8				
24	1.296	164	7	6						
25	0.745	95	25	17	24					

Part Set n=30,p2 – Parts 26-50

Part	Area (m^2)	Demand	Tools Used							
			Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	Tool 8
26	0.533	148	12	5						
27	0.526	247	19	12						
28	1.163	485	20	2	22	12	10	25		
29	0.941	463	5	13	27	1	19	2	8	3
30	0.935	310	6	13						
31	1.473	241	20	29	13	23	15	1	18	2
32	1.097	354	1	10	25	13	9	17	24	
33	1.066	95	22	14	17	3	24	8	1	12
34	1.529	451	22	7	14	2				
35	0.732	63	3	18	2	11	12	16	15	21
36	1.232	291	16	15	6	26	17			
37	1.278	399	12	19	2	28	21	23	9	24
38	0.919	447	9	3	12					
39	1.072	377	8	25						
40	0.611	334	10	4	17	30	19			
41	0.617	359	10	15						
42	1.035	66	30	24						
43	1.241	161	23	15	2	24	5	14	17	
44	1.862	172	2	10						
45	0.778	168	23	11	22	26	19	27		
46	1.062	439	8	1	22	30	18			
47	0.973	256	22	1	25	18	19	11		
48	0.356	242	30	13	8	1	6	16	14	21
49	0.854	332	7	27	30	5	4			
50	0.402	204	2	25	27	14	4	30	17	

Part Set n=50,p1 – Parts 1-20

Part	Area (m^2)	Demand	Tools Used									
			Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	Tool 8	Tool 9	Tool 10
1	1.179	122	45	48								
2	1.611	460	33	10	15	41						
3	0.247	292	15	40	44	23	45	42	27	48	29	9
4	1.287	51	12	5	1							
5	1.106	467	26	36	47	13	6	20	43	40		
6	0.564	208	44	15								
7	0.855	299	47	50	39	20	30	34				
8	1.114	171	4	2								
9	2.193	173	15	31	49	44	1	13	39	25	19	50
10	1.923	491	19	37	14	50	6	3	32			
11	0.550	128	36	43	45	19	39					
12	2.012	28	4	11	2	23	14					
13	1.242	290	30	43	33	37	36	9	44	8		
14	0.979	50	4	19	48	2	17					
15	1.238	239	16	31	14	26	35	27	15	39		
16	0.932	360	21	44	20	12	13	42	9	17	5	18
17	0.959	232	48	23	29	24	28	12	35			
18	1.497	342	32	3								
19	1.470	464	13	20	40	11	18	5				
20	1.472	427	28	47	48	10	3	15	49			

Part Set n=50,p1 – Parts 21-40

Part	Area (m^2)	Demand	Tools Used									
			Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	Tool 8	Tool 9	Tool 10
21	1.224	247	6	3	20	34	37					
22	0.598	173	28	34	36	47	19	20	37	46	5	
23	1.239	434	32	41	40							
24	1.543	289	49	45	29	37	32	27	22	48		
25	1.163	324	44	20								
26	1.345	39	24	7	43	13	25	17				
27	1.242	40	15	18	22	30	3	26	17			
28	0.899	301	42	2	19	39	8	5	17	22	47	27
29	1.098	56	34	44	45	46	42	8	12			
30	0.738	258	9	47	22	23						
31	1.296	495	7	46	48	40	9	34	22			
32	0.618	466	15	14	21							
33	0.644	143	6	41	11	24	18	47				
34	0.730	381	13	18	24	39	23	34				
35	0.050	261	41	16	26	24	21	5	33	27		
36	1.479	71	18	43								
37	1.108	435	27	12	22							
38	0.748	470	31	16	17	47	20	1	46	50	38	35
39	1.457	62	17	3								
40	0.429	432	2	39	44	16	22	9	48			

Part Set n=50,p1 – Parts 41-60

Part	***Area (m^2)***	***Demand***	***Tools Used***									
			Tool 1	***Tool 2***	***Tool 3***	***Tool 4***	***Tool 5***	***Tool 6***	***Tool 7***	***Tool 8***	***Tool 9***	***Tool 10***
41	0.966	389	43	12	40	35	44	6	23	7		
42	0.920	93	28	38	16	5	22					
43	1.106	105	49	17	18	3	42					
44	1.104	392	8	7								
45	0.712	31	38	30	45	5	22	13	39			
46	0.990	439	50	23								
47	0.945	451	44	32	28	22	46	49				
48	1.209	126	28	9								
49	1.364	105	46	7	19	39	2	32	42	25		
50	1.370	218	15	50	34	17	4	40	43			
51	0.712	472	15	7	50							
52	1.026	249	7	9	3	30						
53	0.595	367	26	21								
54	0.629	233	7	38	45	5	1	35	25			
55	0.998	47	24	27	2	11	25	15	14	26	7	9
56	1.511	413	17	7	38	44	18	39	5	12	15	
57	0.743	12	15	33	32	43	10					
58	1.124	207	50	18	2	34	39	15	43	10	9	
59	0.925	45	48	6	25	45	47	31				
60	1.372	132	26	42	24	37	49	14	11	6	30	

Part Set n=50,p1 – Parts 61-80

Part	Area (m^2)	Demand	Tools Used									
			Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	Tool 8	Tool 9	Tool 10
61	0.637	15	27	9	6	42	14					
62	1.011	474	34	49	50	26	11	13	27	17	46	
63	1.184	376	18	15	36	22	43	27	32	49		
64	1.367	444	29	22	23	49	32					
65	1.515	197	36	14	42	32						
66	1.029	265	17	27	35							
67	0.503	438	38	15	26	7	12	25	47	18	43	
68	0.753	464	16	18	26	22	23	19	3			
69	0.646	112	47	38	10	39						
70	1.783	213	23	30	37	40	9	34	21	15		
71	0.795	224	11	10	27	48	47	40	4	46	45	25
72	1.249	203	5	1	45	48	32	25	34	28	26	
73	0.936	332	8	18	36	21	25					
74	1.296	472	43	45	14	11	7	29				
75	0.745	40	27	20	33	42	3	1				
76	0.533	242	36	44	42	14	9	38	6	31		
77	0.526	128	21	15	2	46	17	42				
78	1.163	357	23	41	10	9	12	14				
79	0.941	369	20	43	37	28	29	17				
80	0.935	371	19	5	34	15	20	37	24	16	31	18

Part Set n=50,p1 – Parts 81-100

Part	Area (m^2)	Demand	Tools Used									
			Tool 1	Tool 2	Tool 3	Tool 4	Tool 5	Tool 6	Tool 7	Tool 8	Tool 9	Tool 10
81	1.473	444	8	48								
82	1.097	395	31	37	18	38	12	17	46			
83	1.066	42	33	3	36	45	20	23	34	40		
84	1.529	326	36	46	44	17	12	49	21	43	47	
85	0.732	399	34	18	2	32						
86	1.232	485	24	3	41	16						
87	1.278	447	24	45	10							
88	0.919	488	18	4	8	5	34	12	11	43	9	49
89	1.072	150	39	6	12	30	35					
90	0.611	139	14	42	13	8	9	43	2			
91	0.617	318	47	19	39	45	40	14	42			
92	1.035	77	24	34								
93	1.241	321	40	35	38	23	4					
94	1.862	42	11	14	30	36	50	5	3	35		
95	0.778	405	6	22	1	44	20	11				
96	1.062	174	5	39	48	3	33	27	19			
97	0.973	395	21	2	22	5	47	25	28			
98	0.356	289	46	8	40							
99	0.854	382	17	22	43							
100	0.402	205	44	49	18	34	42	3	36	15	33	