AN APPROACH TO THE FMS TOOL ALLOCATION PROBLEM WITH MATERIAL HANDLING CONSIDERATIONS

by

Deniz Koşucuoğlu

B.S., Industrial Engineering, Boğaziçi University, 2006

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2009

ACKNOWLEDGEMENTS

First of all, I would like to thank Prof. Ümit Bilge for her endless patience and support. Besides offering an unbelievable amount of her time for supervision of this thesis, she has always been helpful and encouraging during my entire graduate study. Secondly, I am also grateful to Assoc. Prof. Ali Tamer Ünal and Assist. Prof. Deniz Aksen for taking part in my thesis jury.

I would like to thank my dear friend Erinç for his insightful comments, he definitely played a crucial role in improvement of this thesis. I also would like to thank my other friends Salim, Umut, İlker, and Gönen for their friendship and support. I will always remember the great times we spent together in BUFAIM.

I don't think I can express how grateful to my parents I am. It was their warmth in my hearth that brought me here. I am sure I would be lost without the guidance, support, and eternal love you offered me. I also would like to thank Derya, for not only being a great brother to me, but also for his friendship. I feel myself very lucky to have such a wonderful family.

At last but not least, I would like to thank Banu. Thank you for loving me, the bliss of being together with you makes everything easier.

The work reported in this thesis is supported by Boğaziçi University Research Fund under Grant No: 08A303.

ABSTRACT

AN APPROACH TO THE FMS TOOL ALLOCATION PROBLEM WITH MATERIAL HANDLING CONSIDERATIONS

The aim of this thesis is to propose solution methods for solving the tool allocation problem with material handling considerations faced in a Flexible Manufacturing System (FMS). The FMS is assumed be capable of handling alternative process plans, which enables to produce the same part type with different operation sequences. Moreover, the tool slot capacity, machine time limitation, tool copy availability and machine-tool compatibility constraints are considered. The problem is tried to be solved with the objective of minimizing the total distance traveled by the parts during their production. In order to achieve that, the distances between machines, on which the tools are to be mounted, are taken into account. Three mathematical model formulations and a genetic algorithm (GA) is proposed for solving this problem. The proposed mathematical model formulations include a mixed integer non linear programming (MINLP) model and two mixed integer programming (MIP) models. The proposed genetic algorithm on the other hand, utilizes optimal solutions of linear programming (LP) models for determination of the fitness values. The performance of these solution methods are tested by conducting extensive numerical experiments on generated problem instances.

ÖZET

ESNEK İMALAT SİSTEMLERİNDE TAKIM ATAMA PROBLEMİNE MALZEME TAŞIMA ODAKLI BİR YAKLAŞIM

Bu tezin amacı, esnek imalat sistemlerinde (EİS) karşılaşılan takım atama problemine malzeme taşıma odaklı bir yaklaşım önermektir. İncelenen EİS'nin parçaları birden fazla alternatif operasyon rotasını kullanarak üretebildiği varsayılmaktadır. Ayrıca önerilen yöntemler; takım kartuşlarının kapasitesi, makina zaman kısıtlamaları ve makina takım uyumluluğu gibi kısıtları da göz önüne almaktadır. Öne sürülen problemin çözümünde amaç fonksiyonu, parçaların sistemde katettiği mesafelerin enazlanması olarak seçilmiştir. Bu amaçla makinaların tesis yerleşim planında bulundukları mevkiler arasındaki mesafeler kullanılmıştır. Çözüm için üç adet matematiksel formülasyon ve bir genetik algoritma önerilmiştir. Önerilen formülasyonlardan ilki, doğrusal olmayan karışık tamsayı, diger ikisi ise karışık tamsayı programlama modelidir. Önerilen genetik algoritma ise her bireyin uygunluğunun belirlenmesinde bir doğrusal programlama modeli çözmektedir. Son olarak, önerilen tüm bu çözüm metodları, detaylı sayısal analizlerle test edilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AGV	Automated Guided Vehicle
CP	Crossover probability
FLP	Facility Layout Problem
FMS	Flexible Manufacturing Systems
GA	Genetic Algorithm
LP	Linear Programming
MINLP	Mixed Integer Nonlinear Programming
MIP	Mixed Integer Programming
MP	Mutation probability
RBS	Rank Based Selection
RWS	Roulette Wheel Selection

1. INTRODUCTION

The enormous speed of change in technology and market requirements force manufacturing companies to have more flexible and adaptive production systems, which enable them to compete with their rivals. The increased importance of product customization and the shortened product lives in today's market makes this necessity even more important. For that reason, manufacturing systems must be designed in a way that they are flexible enough and possess the ability to perform well in dynamic environments, where the product mix changes frequently. On the other hand, the companies must also reduce non-value adding activities in their manufacturing operations to reduce costs. The most significant of these non-value adding activities in a manufacturing system is material handling. It is estimated that 15% to 70% of the manufacturing expenses can be attributed to material handling activities and more interestingly effective facility layout strategies can reduce these costs by at least 10-30% [1]. Thus, the design of manufacturing systems has critical importance: it should not only minimize material handling costs, but also cope with the dynamic changes in the environment.

The minimization of material handling costs mostly depend on the layout design used in facilities. A facility layout is defined as the arrangement of entities in a facility that are needed for production or delivery of services [2]. Due to the importance of designing good layouts, facility layout problem (FLP) is being formally studied since the emergence of the field of operations research, but the requirements and challenges have been changed much since then [2]. The most well-known layout configurations, namely flow-line, functional and cellular configurations and their generation techniques fail to meet the requirements of today's enterprises [3]. The main reason for that is the fact that these traditional techniques are suitable for relatively stable environments, where neither the demand patterns nor the volumes are subject to frequent and serious changes. As the most promising one of these, cellular configuration, which is essentially an implementation of group technology ideas, proposes generation of manufacturing cells which are dedicated to certain product families. However, due to shortened product life cycles of today's market, it is almost impossible to define product families that remain valid sufficiently long. There are many studies in the literature which address the very same issue [3, 4, 5, 6]. Therefore, new layout generation techniques and approaches are needed to cope with the problem.

The most important challenge in FLP is dealing with the dynamic nature of product mixes. Since the product mixes are usually not fixed throughout the planning horizon, determination of an "optimal layout design" is almost impossible. Therefore one approach to this problem is to come up with a design that is robust for different product mixes. The robustness means that the proposed design may not be optimal under any specific product mix but it is ensured that it will perform reasonably" good for a variety of different product mixes [7, 3, 4].

Another approach to dealing with uncertainty in product mix and/or volumes is to re-design the layout whenever a significant change in the system occurs. For instance, in order to minimize the material handling costs in the upcoming period, the locations of the machines can be changed according to the estimated part-flow information. That type of an approach is called the "dynamic approach" and requires the consideration of the trade-off between two costs, namely the material handling and machine re-location costs [8].

On the other hand, the concept of flexible manufacturing concentrates on similar issues. It proposes creation of manufacturing systems that are flexible enough to cope with rapidly changing environment and production needs. The main idea is the use of flexible and versatile machines, which require short setup times and are capable of executing a variety of different operations depending on the tools that are located on their tool magazines [9]. Connecting these machines with an efficient material handling system and creating product designs that allow alternative process routes to be followed are other points indicated to enhance the flexibility in the system [10].

As mentioned above, the machines of an FMS are capable of executing operations depending on the tools located in their tool magazines at that time. In FMS literature, an FMS set-up problem, called FMS Loading Problem" is defined [9, 11, 10]. The aim of FMS loading models in the literature is to allocate tools to the tool magazines of the machines in the system by taking the capacity of the magazines and the time-availability of the machines into account. The solution to the loading problem will eventually determine which operation can be executed at which machines. However, for reaching this aim the proposed FMS loading models in the literature use various different objective functions [12, 13]. Some of these are as follows:

- Maximization of throughput;
- Minimization of tardiness;
- Workload balancing among machines;
- Part movement minimization;
- Minimization of empty tool slots in magazines.

One of these objectives has special importance for the purpose of this thesis, namely the part movement minimization. When the objective of FMS loading problem is selected as part movement minimization, it can be stated that there is an incentive to allocate tools to the same machine, if they can execute operations that appear consecutively in process plans of part types. In addition, from the material handling point of view, it may be also beneficial to allocate this type of tools to the machines which are located close to each other in the layout. Hence, it is possible to reduce material handling effort by intelligently re-changing the stations where operations are executed. Actually, this idea is the same as in the dynamic approach to the FLP. But in this case, instead of re-locating the machines only the tools are re-allocated, which fulfill the same purpose in the context of an FMS. It is also worth to note that the cost of employing the dynamic approach for an FMS is much more cheaper than applying the same idea to a standard manufacturing system, as it is always easier to re-allocate tools instead of changing the physical locations of the machines.

The aim of this thesis is to propose a solution methodology for a special type of FMS loading problem, which takes the actual physical locations of the machines in the system into consideration with the aim of minimizing material handling effort required.

The constraints of the problem environment are the tool slot capacity, machine time limitation, and technological constraints. Furthermore, it is aimed that the proposed methodology will allow use of alternative process plans for parts to be manufacturing, instead of allowing only a single process plan for parts. In that way the routing flexibility of the system under study can be fully exploited.

The rest of the thesis is organized as follows; second chapter is devoted to literature survey mainly about FMS loading problem and the evolution of its solution methodologies. Third chapter presents a concise definition of the tool allocation problem with material handling considerations. The proposed solution methodologies are introduced in Chapters 4 and 5. The performance of the solution methodologies are also tested via numerical experiments, which are presented in Chapter 6. Finally a brief conclusion and recommendations for future studies is made in Chapter 7.

2. LITERATURE REVIEW

Due to the complex nature of the production planning problems faced in an FMS, the FMS set-up problem is said to be intractable in its entirety. For that reason, the researchers tend to decompose this problem into several sub-problems and attack each of them separately. The first decomposition made in Stecke [14] is as follows.

- Part type selection problem: Determination of a subset of part types for production in the next period
- Machine grouping problem: Partition of machines into groups so that each single machine in a certain group is able to perform the same set of operations
- Production ratio problem: Determination of production ratios for the part type selected
- Resource allocation problem: Allocation of limited number of pallets and fixtures among the selected part types
- Loading problem: Allocation of operations and their required tools of selected part types among the machine groups

The solution of these problems constitute a complete system set-up. However, it is worth to note that there are many other problems that may be faced in an FMS, such as routing of parts, release of orders, control of material handling system, etc. But these are related to the on-line control of the system, for that reason they are not treated as a part of production planning problem that arises in an FMS.

FMS loading problem is a well-known one among the short-term planning sub-problems and has received significant attention since its first definition. This review of literature is mostly focused on the approaches and solutions techniques for FMS loading problems. Firstly, a preliminary explanation of concepts in FMS is given. Thereafter, the definition of loading problem and the evolution of its solution methodologies is discussed and finally, a brief review of studies regarding the use of genetic algorithms for FMS loading problems is presented.

2.1. Features of FMS

The most important feature of an FMS is not the high level of automation it possesses, but the manufacturing flexibilities it offers [10]. There are various classifications and definitions of types of manufacturing flexibilities offered in the literature. The first explicit definition of these flexibilities is made in Browne et al. [15]. They defined 8 types of flexibilities, which are critical for a manufacturing system to attain the title FMS. A review on flexibility made by Sethi and Sethi [16] extends the list of manufacturing flexibilities to 11 and gives more detailed information about their purposes, means, valuations and measurements. Another study made by Vokurka and O'Leary-Kelly [17] defines 15 different types of flexibilities. The most frequently encountered types of these manufacturing flexibilities are the following [18]:

- **Machine flexibility:** Machine flexibility is defined as the ability of machines to perform a wide range of operations without large setup times between their execution. Actually, the use of versatile machines in the system makes the use of this flexibility possible, and it is the key element for attaining other type of flexibilities,
- **Operation flexibility:** A typical part to be produced in FMS have most of the time more than one alternative process plan, i.e. operation routes. The operation route for a part type defines the set of operations that are needed to be executed for its completion. This feature of the part types is called the *operation flexibility*,
- **Process flexibility:** The ability of an FMS to produce different part types under a certain setting without the need of set-up is called process flexibility,
- **Routing flexibility:** Routing flexibility is the capability of an FMS to produce parts by using the alternate routes in their process plans. This type of flexibility can be attained only in existence of machine and/or operation flexibilities. It is worth to note that the routing flexibility is a feature of the system itself, whereas the operation flexibility is a feature of the part type's definition.

As for the categorization of FMS, there are various of different approaches, each of which is based on a certain aspect of the system. One of these categorizations is proposed by Shanker and Srinivasulu [19]. They divide FMS into two categories:

- Dedicated FMS: This type of FMS is designed for a small family of part types with known and limited processing requirements,
- Random FMS: Neither the part family nor their processing requirements are known at the design stage of such systems.

According to this classification the FMSs of random type have a wide range of capabilities and the production problems arise in these type are more complicated. Another categorization is made due to the types of machines used in the system:

- Parallel FMS,
- General FMS.

An FMS is said to be parallel if the machines used in the system are identical. This categorization is needed because in some past research efforts, the methodologies cannot be generalized for the cases, where the machines are not identical, namely for the case of general FMS [13].

2.2. Definition of the FMS Loading Problem

A typical FMS consists of computer numerically controlled machines that are capable of executing different operations, depending on the tools mounted on them, with automatic tooling and instruction changeovers. These machines are connected via an automated material handling system, and controlled with a complex computer network. Since the main aim of FMS concept can be summarized as to attain the efficiency and utilization of mass production while having the flexibility of job shops, these systems are capable of producing various part-types efficiently and simultaneously without long setup times in between [9].

As mentioned before, the operational capabilities of the machines in an FMS at a given time depend on the tools they have in their magazines. This property of machines in an FMS is called the "versatility", and it is the key element that allows production of multiple part types while maintaining high machine utilization [12]. For that reason, the allocation of tools to magazines of the machines in the system plays a very crucial role in terms of efficiency.

This allocation problem is named as the FMS loading problem and the first definition is made by Stecke [14] as follows:

> "To allocate operations and the required tools of the selected part types among the machine groups subject to technological and capacity constraints of the FMS."

The machine groups mentioned in this definition refer to group of machines that are identically tooled. The machine groups are formed with the aim of maximizing the expected production in the system. Most of the time, it is assumed that the machine groups are formed prior to solving the loading problem, and at loading level it suffices to determine which tool is assigned to which machine group [20].

The tool magazines of machines have limited number of slots, on which tools can be mounted. Therefore, tool magazine slot availabilities are additional capacity constraints of the FMS from loading point of view along with time availability of machines. On the other hand, the technological constraints refer to the information related to the process plans of parts, machines and tool types. In addition to these constraints, there may be also a limited number of tool copies to be allocated.

The FMS loading problems studied in the literature can be divided into three categories based on their tool management strategies [13]:

1. Batching strategy: In this type of tool management strategy it is assumed that the allocation of tools to magazines of machines remain unchanged for a certain period of time. This period of time can be fixed and pre-defined or it may depend on the on-going production on the shop-floor. For instance, re-allocation of tools may be beneficial if the production of a batch of certain part type is completed,

because some tools loaded on machine may not be needed anymore. Thus, by removing these unnecessary tools, additional capacity for production of new part types can be created.

2. Flexible strategy: If the FMS under study possesses automated tool handling system, it may be possible to change the tool allocation on-line. But if this procedure causes a halt or delay of on-going production operations, its advantage should be analyzed carefully. In these type of systems the tool magazine configurations of the machines are evolved throughout the planning period.
3. Hybrid strategy: There are also systems that can be operated by using the two tooling strategies together.

The tool management strategy to be used in the system highly depends on the system's physical structure and the technology used.

As for the choice of objectives for loading problems, there is a wide range of different selections among the researchers. A complete list of the objectives used in past research efforts is given in the recent paper of Kumar et al.[21] and will not be repeated here. The most preferred ones are as follows:

- Minimization of manufacturing costs,
- Minimization of makespan or maximization of system production rate or maximization of system saturation,
- Maximization of the differences of load among machines,
- Minimization of total overload and underload of machines,
- Minimization of load of workpiece transport system,
- Maximization of alternative routes used,
- Minimization of system unbalance and maximization of throughput.

Since the approach followed in this thesis is also based on the minimization of load of workpiece transport system, it deserves further discussion. The selection of this objective is suitable for systems with the following properties:

- The capacity of the material handling system used is limited, and requires intelligent allocation.
- The travel and loading/unloading times of parts on the shop-floor are comparable or longer than the processing times of parts.

This type of objective is frequently used in the literature and it is also adopted in this study [14, 20, 22, 23, 24, 25, 26, 27].

2.3. Evolution of Loading Methodologies

The methodology followed in the pioneering works on the loading problem is to construct a nonlinear mixed integer problem, and apply some linearization techniques to make it "solvable" [14, 20, 22].

As the first study on the subject, Stecke and Solberg [9] define the loading problem, and propose an MINLP model for its solution. The aim of their work is to explore alternative loading and scheduling strategies for a real-life system. Stecke [14] focuses only on grouping and loading problems by presenting two separate MINLP models, which are to be solved sequentially. Furthermore, several linearization methods for these models are also investigated. Both of the studies emphasize the minimization of part movement in the system. They claim that, it can be more advantageous for a part to remain on a machine for several consecutive operations rather than to move to another machine for the sake of balancing. Hence, if technologically possible, both travel and waiting time may be saved by allocating consecutive operations to the same machine.

The study conducted by Lashkari et al. [22] introduces an extension of the model proposed in Stecke [14]. This extension includes consideration of refixturing and limited tool copy availability issues. Another important issue addressed in Laskari et al. [22] is the detailed effort of incorporating the material handling issues into the proposed model. The objective in their work is to minimize the total distance traveled by the parts, which includes both loaded and unloaded travels made by material handling entities. However, the layout structure they have assumed for the FMS is a special type, where all the parts should be carried first to a central storage location after leaving a machine. In other words, inter-machine part travel is not allowed, which is a very restrictive assumption. An example of this layout structure is given in Figure 2.1. Wilson [23] proposed an alternative formulation of the model in Lashkari et al. [22] with the aim of avoiding the nonlinearities involved.

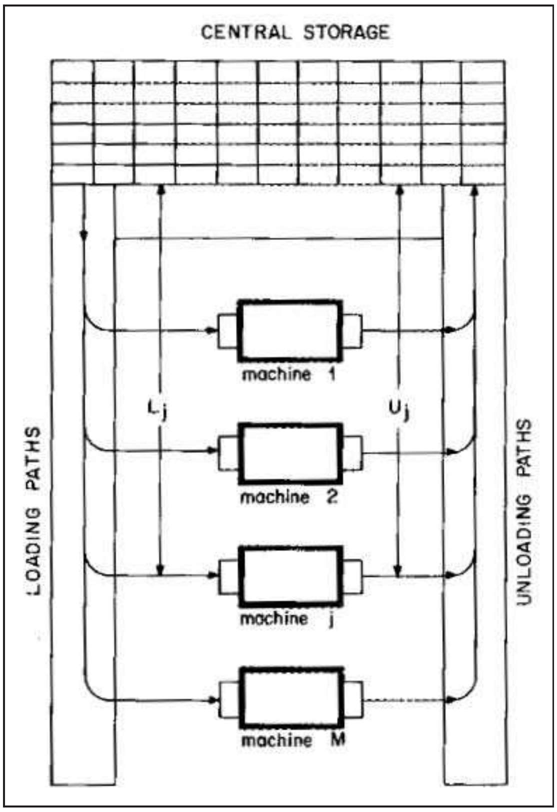

Figure 2.1. Sample layout of the FMS in Lashkari et al.

Shanker and Srinivasulu [12] introduce a branch and backtrack method for the linearized version of the mixed integer nonlinear model proposed in Stecke [14]. They use the maximization of the assigned workload as their objective. Due to the complexity of their exact solution methodology, they also present three heuristic methods with a bi-criterion objective of balancing the workload and maximizing the throughput.

The joint consideration of the closely related problems, namely machine grouping and loading are analyzed in the work of Kumar et al. [25]. They utilize a multistage and multiobjective optimization model with a min-max approach for the solution of the combined problem. The results of their work indicate that the solution of grouping problem does not give always the best results according to loading objectives.

Another joint consideration of two different problems, namely part assignment and tool allocation is studied by Leung et al. [28]. They emphasize the fact that the existing research does not address the material handling issues at the planning level but at the operational level instead. In their work, they deal with material handling problem by incorporating the allocation of parts to the material handling entities in the system. They construct an MIP formulation, which takes the loaded and unloaded trips made by material handling entities into account. The unloaded trips are assumed to be equal to a specific fraction of the loaded travel. The main difference of their work from the current study is that they do not consider alternative operation routes. The objective in their base formulation is cost minimization, but they also propose two alternative formulations to reflect two other objectives, namely time minimization and machine workload balancing. Finally they test their results via numerical simulation with different number of part-carriers.

Denizel and Erenguc [11] present a MIP model for part type selection and loading problems. Firstly, they propose a basic single machine framework and then extend their formulation to the multiple machines case. Their work includes the design of a special branch and bound algorithm for effective solution of the addressed problems.

Gurerrero et al. [10] presents a new approach for loading problem in FMS, which focuses on alternative routes of part types. Hence, their decision include both route selection and tool allocation. Furthermore they make use of routing flexibility to a full extent because their model permits selection of more than one route for each part type. They assume that the part type selection problem is solved beforehand, thus the quantities of each part type to be produced are known. A MIP formulation is constructed for the concurrent determination of routing and tooling decision. Afterwards, they also propose an extension for their formulation that incorporates also the part type selection decisions.

An example use of meta-heuristics for the loading problem can be found in the work of Kumar and Shanker [27]. They design a genetic algorithm (GA) for concurrent determination of both FMS part type selection and machine loading decisions. They assume that there is no machine grouping in the system, so they treat each machine as a single entity. They propose a constraint-based GA that uses integer-coded chromosomes for representing the part type selection and tool allocation decisions.

Tiwari and Vidyarthi [29] consider solving machine loading problems by using genetic algorithms. They categorize the FMS decision problems as pre-release and post-release decisions. The pre-release decisions include all decisions regarding the pre-arrangement of the FMS system before the process begins. Whereas the post-release decisions are related to the sequencing and routing of parts when the system is in operation. They regard the loading problems under the category of post-release decisions, and state that loading decisions serve as links between strategic and operational level of decisions. They emphasize the advantage of using GAs in loading problems as follows:

- Search by GA is not limited to the neighborhood of a single solution but a population of solutions,
- GAs use only the information contained in objective function, not any other axillary information such as derivatives or gradient,
- GAs use probabilistic transition rules, not deterministic rules.

They propose a GA-based heuristic for determination of part type selection and operation allocation problem. The objective function is selected as the minimization of system unbalance and maximization of system throughput.

Another GA-based approach is presented in Kumar et. al. [21]. They extend the simple GA and propose a constraint-based algorithm for loading and part-type selection problems, which works with solutions that are ensured to be feasible with respect to the constraints of the FMS. Similarly they designed the genetic operators (initialization, crossover and mutation) such that they also consider the violations of the various constraints that are typical to the FMS. They use the minimization of system unbalance and maximization of throughput as their objective and their constrained-based GA dominates the solutions generated by using the simple GA approach.

An example for the solution methodology of FMS loading problem with material handling constraints is the work of Goswami and Tiwari [30]. They take the material handling capacity of an FMS into consideration by imposing additional constraints on the maximum number of AGVs used. The determination of part type sequence is made by evaluating the contribution of part type characteristics such as batch size, total processing time and AGV movement. After determination of part-sequence, the operation allocations decisions are made on the enumeration of priority index. An iterative procedure is proposed to reach a solution that minimizes the unbalance in the system and maximizes the system throughput.

A hybrid evolutionary heuristic that combines GA and simulated annealing (SA) is presented in Yogeswaran et al. [31]. Their algorithm aims to solve the FMS loading problem with the bi-criterion objectives of unbalance minimizing and throughput maximization of the system. They also present a comparison of their results with other studies in the literature.

This study aims to attack the FMS tool loading problem with layout considerations. The consideration of the machine layout is handled by selecting the objective as the minimization of the travel distances of the parts along the facility. Therefore, the aim is not only to minimize inter machine part-flow but also the distance traveled by the parts. Moreover, the problem is modeled in such a way that it considers alternative process plans of the parts and it also allows partial use of these process plans. The constraints that are taken into consideration are the limited tool slot and time capacity of the machines and also the technological constraints regarding the compatibility of the tools with the machines. There is not any study in the literature to the knowledge of the authors that takes all of the aforementioned considerations concurrently into account.

3. PROBLEM DEFINITION

Even though the first definition of the loading problem introduced in Stecke [14] is a concise and complete definition of the problem, it is slightly to be modified for the purposes of this study. This chapter is devoted to the definition of the problem analyzed in this thesis. In the following two sections, the characteristics of the hypothetical manufacturing system and the details of the approach to the loading problem with material handling considerations will be given, respectively.

3.1. Characteristics of the System

The hypothetical system under study is a typical FMS, which consists of several machines connected to each other via an automated material handling system. There are several tool types and copies of each of these tool types to be allocated on the magazines of the machines. The aim is to produce the demanded amounts of a certain product mix that is assumed to be determined at a higher-order planning level.

First of all, in this study there is no concept of identically tooled machines. Every machine in the system is assumed to be a single entity, which can be uniquely tooled. Moreover, machines in the system have specific input/output locations on the material handling flow-path. Therefore, the real distances between the machines that are to be traveled by the material handling entities is known beforehand. Furthermore, each of the machines belongs to a certain machine type. The machine types differentiate from each other in terms of tool slot capacities, available machining times, set of mountable tools and executable operations.

Likewise, existence of several tool types in the system is assumed. The type of a certain tool determines the set of executable operations. Furthermore, these sets are disjoint, that is to say every operation can be executed by only a single tool type. However, it is possible to execute more than one operation by using certain tools.

As for the part types, their selection for the upcoming production period is made beforehand. Hence, it is assumed that there is a certain product mix with associated demand quantities for the upcoming period. In other words, the part-selection problem is solved already and the tool management strategy to be followed is the batching strategy, in which the tools allocated to the machines remain unchanged for a predetermined period of time. This period of time can be named as the “loading period” and it is assumed that, the proposed FMS loading problem in this thesis is solved at the beginning of these loading periods.

The part types in the system are assumed to have different process plans. Each of these process plans contains information about the sequence of operations to finish an instance of the associated part type. The existence of these alternative process plans, which are also named as the alternative operation routes, stems from the fact that there may not be only single procedure to produce a part, but a set of alternative procedures. These alternative operation routes, can differ from each other only in the arrangement of the operations or also in the set of necessary operations. It is assumed that the partial fulfillment of total demand by using these alternative process plans is allowed in the system. In other words, the dedicated operation route allocation is not enforced.

The operations that can be executed in the system have deterministic processing times and they do not depend by which machine it is executed, i.e. the processing times do not differ from machine to machine. It is also important to note that, the travel and load/unloading times in the system are assumed to be significant relative to the processing times. Therefore, a great deal of time is spent due to material handling activities.

As for the material handling entities, it is assumed that they travel on a predetermined network of aisles, which is called as the material handling flow-path (see Figure 3.1). These entities are assumed to be automated guided vehicles (AGVs), but they can be any type of agent which is enforced to travel on a flow-path. In calculation of travel distances between machines, the shortest path emerging from a machine’s output point to the other machine's input point is taken. Neither blocking nor traffic jamming issues are taken into consideration.

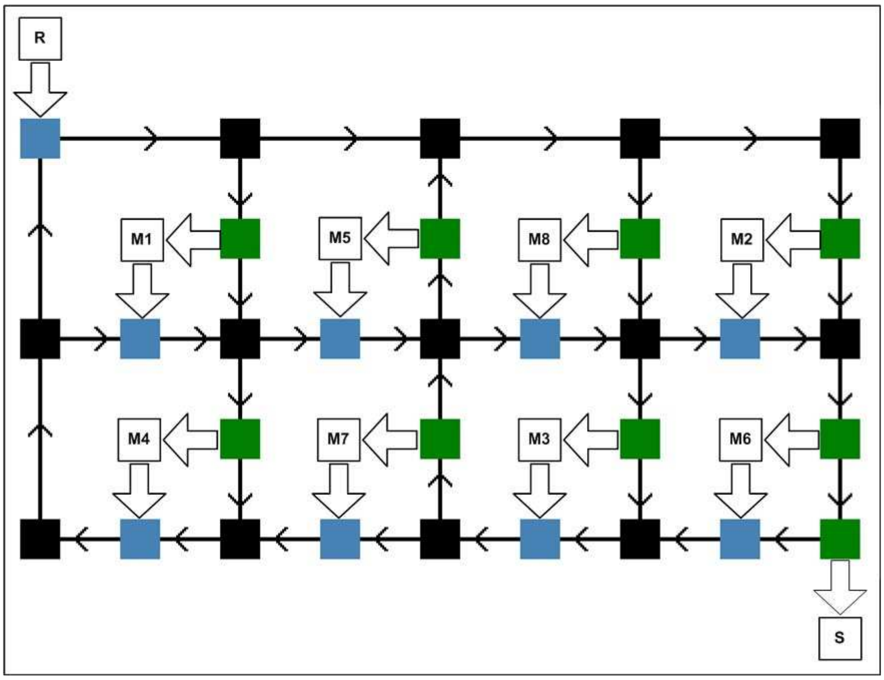

Figure 3.1. A sample flow-path with 8 machines

3.2. Definition of Loading Problem with Material Handling Considerations

The objective of the loading problem is selected as the part movement minimization in this study. As mentioned before, the travel and loading/unloading times are comparable to processing times and therefore minimizing the total part movement is an effective approach for maximizing system throughput.

Travel times in the system are incurred whenever an inter-machine part flow occurs. An inter-machine part flow is needed when the tools of two consecutive operations are allocated to different machines. If this is not the case, i.e. tools are allocated to the same machine, then next operation can be executed immediately without carrying the part. However, achieving this objective for all parts manufactured in the system is not possible due to the limited time and tool slot capacities of the machines, so some parts are required to be carried among machines. Even in the case of inter-machine part movement, the distance traveled during transportation can be minimized by allocating the tool of the next operation to a nearby machine. This concept is a crucial point in the approach adopted in this study. Since the locations of the stations, where the operations are executed depend on the tool allocation in the system, the distance traveled by the parts can be minimized by intelligent allocation of tools. In fact, the very same concept is adopted in the dynamic facility location approach, which aims to minimize the material handling effort by re-locating the machines in the system. However, in the domain of FMS, this aim can be achieved without changing the physical locations of the machines but by reallocating the tools on their magazines.

4. PROPOSED MATHEMATICAL PROGRAMMING FORMULATIONS

This chapter introduces the mathematical programming formulations designed for the solution of the FMS loading problem with material handling considerations. A MINLP model, its linearized MIP version and another alternative MIP model are developed for this purpose. Each of these models has its own advantages and disadvantages with respect to number of variables, number of constraints, and the size of the input matrices.

The critical issues regarding the development of a loading model are as follows;

- Handling of tool allocation decisions: The tool allocation decisions can be represented by a single binary variable for each tool-machine or tool type-machine pair. It is apparent that the latter version will require less number of binary variables, but this representation requires the assumption that the allocation of two tools of the same type to the same machine is not allowed in the system. This assumption is valid since the tool-life issues are not taken into consideration in this study. Furthermore, there is also need of additional constraints to avoid violation of tool availability since the assignment of individual tools are not accounted in the second representation.
- Representation of alternative process plans for parts: There are a number of ways to represent the alternative process plans in the system. There are two different approaches followed in this thesis, each of which will be explained in detail.
- Representation of inter-machine flows: The inter-machine flow amounts result from tool loading and route selection decisions. The representation of these flows can be handled by explicit definition of additional continuous variables or by exploiting the information contained in the alternative process plans. The details of these approaches will be made clear in the upcoming sections.

The next three sections will present the proposed formulations. The models with the exception of MINLP is tested via numerical experiments. The details about these experiments are given in Sections 6.3 and 6.4.

4.1. Nonlinear Mixed Integer Formulation

With the aim of modeling the FMS loading problem with material handling considerations, a MINLP model is formulated. Before presenting the model, the issue of handling of the operation routes will be made clear. The process plans for parts contain the operation sequence information. That is to say, they explicitly define the order of operations for producing the associated part type. However, in loading problem there is not a one-to-one mapping between operations and tools. An operation that requires a certain tool type can be executed by one of the copies of this tool type. For that reason, there is a need for another representation of the alternative routes for part types, which is based on the sequence of distinct tools, instead of operations. As an example, assume that an operation route for a part type is given as follows:

- Operation1 → Operation3 → Operation5

Furthermore, the tool types in the system can execute the following operations:

- ToolType1 : Operation1, Operation2
- ToolType2 : Operation3
- ToolType3 : Operation4, Operation7
- ToolType4 : Operation5, Operation6

Finally, assume that there are two distinct copies for each of these tool types. Then the given operation route may also be represented as follows:

- ToolType1-Tool1 → ToolType2-Tool1 → ToolType4-Tool1
- ToolType1-Tool1 → ToolType2-Tool1 → ToolType4-Tool2
- ToolType1-Tool1 → ToolType2-Tool2 → ToolType4-Tool1

- ToolType1-Tool1 → ToolType2-Tool2 → ToolType4-Tool2
- ToolType1-Tool2 → ToolType2-Tool1 → ToolType4-Tool1
- ToolType1-Tool2 → ToolType2-Tool1 → ToolType4-Tool2
- ToolType1-Tool2 → ToolType2-Tool2 → ToolType4-Tool1
- ToolType1-Tool2 → ToolType2-Tool2 → ToolType4-Tool2

This type of routes will be denoted as "tool routes" to avoid confusion. Since there are three operations in the original operation route and two tool copies for each tool type, there are $2^3 = 8$ tool routes in total. In the MINLP formulation, all the routes will be given as tool routes. This alternative formulation of the routes will enable to define the workload assigned to each tool and also the potential flows between consecutive tools that appear in the tool routes. The term "potential" is used, because the flow may not be incurred if these tools are allocated to the same machine.

Before introducing the MINLP model; indices, variables and parameters are given.

- Indices

Tools	i	$=$	$1, 2, \ldots, N$
Alias of tools	j	$=$	$1, 2, \ldots, N$
Tool types	s	$=$	$1, 2, \ldots, S$
Machines	m	$=$	$1, 2, \ldots, M$
Alias of machines	k	$=$	$1, 2, \ldots, M$
Part types	q	$=$	$1, 2, \ldots, Q$
Tool Routes	r	$=$	$1, 2, \ldots, R_q$

- Sets

$I(s)$	:	Set of tools of tool type s
$R(q)$	:	Set of tool routes of part type q
$M(i)$	:	Set of machines on which tool i can be mounted

- Variables
  - x_{im} : Equal to 1 if tool i is allocated to machine m, 0 otherwise
  - f_{ij} : Amount of flow from tool i to tool j
  - ϕ_{qr} : Proportion of demand satisfied of part q via tool route r
  - β_i : Total workload assigned to tool i in units of time
- Parameters
  - d_{mk} : Distance between machines m and k
  - C_m : Tool slot capacity of machine m
  - W_m : Machine time capacity of machine m
  - D_q : Demand for part type q
  - α_{qri} : Equal to 1 if tool i appears on tool route r of part type q, 0 otherwise
  - t_{iqr} : Total processing time of tool i on tool route r of part type q
  - z_{ijqr} : Number of times tool i precedes tool j on tool route r of part type q
  - F_{ij}^{max} : The maximum amount of flow that may occur from tool i to tool j

Parameters like t_{iqr}, α_{qri}, and z_{ijqr} can be calculated by utilizing the information contained in tool routes. On the other hand, F_{ij}^{max} values denote the maximum attainable values for the flow amount from tool i to tool j. It can be calculated by assuming that for each part type, the route that yields maximum flow between tool i and tool j is selected. If F_{ij}^{max} turns out to be equal to 0 for a specific pair of tools, then that means there may not be any flow from tool i to tool j, so there is no need to include the associated f_{ij} variable in the formulation.

The proposed **MINLP model** is as follows:

$$\min \sum_{i=1}^{N} \sum_{j=1(F_{ij}^{max}>0)}^{N} \sum_{m=1}^{M} \sum_{k=1}^{M} (f_{ij} d_{mk} x_{im} x_{jk}) \tag{4.1}$$

s.t.

$$\sum_{m=1}^{M} x_{im} \leq 1 \qquad \forall i \qquad (4.2)$$

$$\sum_{i \in I(s)} x_{im} \leq 1 \qquad \forall m, s \qquad (4.3)$$

$$\sum_{i=1}^{N} x_{im} \leq C_m \qquad \forall m \qquad (4.4)$$

$$\sum_{i=1}^{N} x_{im}\beta_i \leq W_m \qquad \forall m \qquad (4.5)$$

$$\phi_{qr}\alpha_{qri} \leq \sum_{m=1}^{M} x_{im} \qquad \forall i, q, r \in R(q) \qquad (4.6)$$

$$\beta_i = \sum_{q=1}^{Q}\sum_{r=1}^{R_q} \phi_{qr} D_q t_{iqr} \qquad \forall i \qquad (4.7)$$

$$f_{ij} \geq \sum_{q=1}^{Q}\sum_{r=1}^{R_q} \phi_{qr} D_q z_{ijqr} \qquad \forall i, j \qquad (4.8)$$

$$\sum_{r=1}^{R_q} \phi_{qr} = 1 \qquad \forall q \qquad (4.9)$$

$$x_{im} = 0 \qquad \forall i, m \notin M(i) \qquad (4.10)$$

$$x_{im} \in \{0, 1\} \qquad \forall i, m \qquad (4.11)$$

$$f_{ij}, \phi_{qr}, \beta_i \geq 0 \qquad \forall i, j, q, r, m \qquad (4.12)$$

- (4.1) resembles the well-known objective function of quadratic assignment problems with a crucial difference; f_{ij} variables are not parameters but decision variables. For that reason, the objective function is the multiplication of three decision variables and the distance, which is equal to the total distance traveled due to material handling activities.
- (4.2) ensures that each tool is allocated only once.
- (4.3) enforces that only one copy of a certain tool type may be allocated to a machine.
- (4.4) and (4.5) are the tool slot and time capacities for machines, respectively.
- (4.6) ensures that a route is not used whenever any of the tools appearing in its definition is not mounted on any of the machines.
- (4.7) is the definition of the workload assigned to a certain tool due to the route

allocation decisions. The value of the β_i will be incurred as workload to the machine on which tool i is mounted.

- (4.8) is the set of equations which determines the values of f_{ij} variables. The right-hand-side (RHS) of the equation is the total amount of part flow between tools i and j due to the routing decisions. So, if these tools are not allocated to the same machine then this amount must be multiplied with the distance between the machines, on which they are mounted on. If the rhs is equal to 0, then the value of the associated f_{ij} will also be equal to 0, since these terms appear in the minimizing objective function.
- (4.9) ensures that the total of the fractions of tool route usages add up to 1 for all part types. In other words these constraints enforce demand satisfaction for all part types in the product mix.
- (4.10) represents the technological constraints of the system; tools can only be mounted on certain machines. This constraints are relaxed for systems where all tools can be mounted on every machine.
- The last two constraints, namely (4.11) and (4.12) are integrality and non-negativity constraints, respectively.

The following remarks are worth to point out regarding the above MINLP formulation.

- It is a nonlinear formulation of the problem. Furthermore, the objective function is a non-convex function. Thus, if branch-and-bound method is applied to the problem, then solutions found at the nodes of the tree may not be the global optimum for that node. Hence, pruning decisions may not be reliable. Hence, the solution found at the end may not be a global optimum for the problem.
- For a problem instance with $N = 36$, $M = 12$, $Q = 20$ and $R_q = 81$ $\forall q$; there are 432 binary and 2952 continuous variables. A more detailed analysis of model sizes will be presented in Section 6.1.
- The actual problem data is large in dimension. Number of values for parameters like T_{iqr}, α_{qri} and Z_{ijqr} may be plenty depending on the problem size. For example, in a problem instance with $N = 36$, $Q = 20$ and $R_q = 81$ $\forall q$; the total number of

Z_{ijqr} values equals to 2099520.

The first remark deserves further discussion. Due to the non-convex nature of the formulation, even if good solutions may be found, there may be better ones in the unexplored regions of the feasible solution space. Therefore, the solution obtained by a MINLP solver should be treated as a heuristic solution of the problem. However, if this formulation can be convexified somehow, it can solved by one of the MIP solvers available and absolute optimality can be proved. Next section provides such a convexification based on linearization.

4.2. Linearization of the MINLP Formulation

In order to linearize the formulation given in the previous section, a well-known trick is applied. Additional variables will be introduced to denote non-linear terms. Since the same formulation will be extended only the additional variables will be given before introducing the model.

- Additional Variables

g_{ijmk}	:	Greater than or equal to f_{ij} if tool i is allocated to machine m and tool j is allocated to machine k
b_{im}	:	Greater than or equal to β_i if tool i is allocated to machine m

The linearized version of the MINLP model, namely **MIP1 model** is as follows:

$$\min \sum_{i=1}^{N} \sum_{j=1(F_{ij}^{max}>0)}^{N} \sum_{m=1}^{M} \sum_{k=1(k\neq m)}^{M} (g_{ijmk} d_{mk}) \tag{4.13}$$

s.t.

$$g_{ijmk} \geq f_{ij} - F_{ij}^{max}(2 - x_{im} - x_{jk}) \qquad \forall i, j(F_{ij}^{max} > 0), m, k \qquad (4.14)$$

$$\sum_{m=1}^{M} x_{im} \leq 1 \qquad \forall i \qquad (4.15)$$

$$\sum_{i \in I(s)} x_{im} \leq 1 \qquad \forall m, s \qquad (4.16)$$

$$\sum_{i=1}^{N} x_{im} \leq C_m \qquad \forall m \qquad (4.17)$$

$$\sum_{i=1}^{N} b_{im} \leq W_m \qquad \forall m \qquad (4.18)$$

$$\phi_{qr}\alpha_{qri} \leq \sum_{m=1}^{M} x_{im} \qquad \forall i, q, r \in R(q) \qquad (4.19)$$

$$\beta_i = \sum_{q=1}^{Q}\sum_{r=1}^{R_q} \phi_{qr} D_q t_{iqr} \qquad \forall i \qquad (4.20)$$

$$b_{im} \geq \beta_i - M(1 - x_{im}) \qquad \forall i, m \qquad (4.21)$$

$$f_{ij} \geq \sum_{q=1}^{Q}\sum_{r=1}^{R_q} \phi_{qr} D_q z_{ijqr} \qquad \forall i, j \qquad (4.22)$$

$$(4.23)$$

$$\sum_{r=1}^{R_q} \phi_{qr} = 1 \qquad \forall q \qquad (4.24)$$

$$x_{im} \in \{0, 1\} \qquad \forall i, m \qquad (4.25)$$

$$f_{ij}, \phi_{qr}, \beta_i, b_{im}, g_{ijmk} \geq 0 \qquad \forall i, j, q, r, m, k \qquad (4.26)$$

The new constraints are (4.14) and (4.21), they ensure that the new variables are equal to the multiplication of the old terms. All the terms in the above formulation are linear, therefore any MIP solver can be utilized to solve the problem to optimality. However, the disadvantage of applying such linearization is the increase in the number of decision variables and also constraints. A detailed analysis of model sizes will be presented in Section 6.1.

4.3. Alternative MIP Formulation

In order to overcome the shortcomings of the previous MIP formulation a new model is designed. The main disadvantage of MIP1 was its high number of variables. There are two main reasons behind that:

- MIP1 is the linearized version of MINLP by including additional variables to the formulation.
- In MIP1 a binary variable is required for assignment of every tool-machine pair. Thus, if the number of tool copies increases, the problem size increases directly. This may not be very efficient if there are high number of tool copies in the system.

MIP2 is created with the motivation of decreasing the number of binary variables in the formulation. Actually, the main difference between MIP1 and MIP2 is in the handling of alternative operation routes of parts. MIP1 takes the tool routes as an input, so it is necessary to convert the operation routes to the tool routes. Since the set of machines, on which a specific tool type can be mounted is known beforehand, these tool-routes can be converted into "machine routes". These machine routes contain tool type and machine pairs to be visited for completion. As an example, consider the following operation route of a part type:

- Operation1 → Operation3 → Operation5

Furthermore, assume that the tool types in the system are capable of executing the following operations:

- ToolType1 : Operation1, Operation2
- ToolType2 : Operation3
- ToolType3 : Operation4, Operation7
- ToolType4 : Operation5, Operation6

Besides these information, also the machine on which these tool types can be mounted is given as follows:

- ToolType1 : Machine1, Machine3
- ToolType2 : Machine4, Machine6
- ToolType3 : Machine5, Machine3
- ToolType4 : Machine2

Without loss of generality, it can be assumed that only a single copy of each tool type can be mounted on machines. Hence, it is possible to represent the machine route associated with the given operation route as pairs of tool types and machines:

- ToolType1-Machine1 → ToolType2-Machine4 → ToolType4-Machine2
- ToolType1-Machine3 → ToolType2-Machine4 → ToolType4-Machine2
- ToolType1-Machine1 → ToolType2-Machine6 → ToolType4-Machine2
- ToolType1-Machine3 → ToolType2-Machine6 → ToolType4-Machine2

The main advantage of the above representation is the fact that it contains the machine to machine flow information and also the potential work load assigned to the machines. In other words, if the fraction of the total production allocated to the machine routes is known, then inter-machine flows and also work loads assigned to machine can be calculated. This feature of the representation eliminates the necessity to define flow and work load assignment variables.

The indices, sets, variables and parameters of the model are as follows:

- Indices

Tool types	s	$=$	$1, 2, \ldots, S$
Machines	m	$=$	$1, 2, \ldots, M$
Alias of machines	k	$=$	$1, 2, \ldots, M$
Part types	q	$=$	$1, 2, \ldots, Q$
Machine Routes	r	$=$	$1, 2, \ldots, R_q$

- Sets

  $R(q)$: Set of machine routes of part type q

  $M(s)$: Set of machines on which a copy of tool type s can be mounted

- Variables

  x_{sm} : Equal to 1 if tool type s is allocated to machine m, 0 otherwise

  ϕ_{qr} : Proportion of demand satisfied of part q via machine route r

- Parameters

  A_s : Number of available tool copies of part type s

  d_{mk} : Distance between machines m and k

  C_m : Tool slot capacity of machine m

  W_m : Machine time capacity of machine m

  D_q : Demand for part type q

  t_{smqr} : Total processing time of tool type s mounted on machine m, which appears on machine route r of part type q

  z_{mkqr} : Number of times machine m precedes machine k on machine route r of part type q

The MIP2 formulation is as follows:

$$\min \quad \sum_{m=1}^{M}\sum_{k=1}^{M} d_{mk} \sum_{q=1}^{Q}\sum_{r=1}^{R_q} z_{mkqr} D_q \phi_{qr} \tag{4.27}$$

s.t.

$$\sum_{m=1}^{M} x_{sm} \leq A_s \qquad \forall s \qquad (4.28)$$

$$\sum_{s=1}^{S} x_{sm} \leq C_m \qquad \forall m \qquad (4.29)$$

$$\sum_{q=1}^{Q} \sum_{r=1}^{R_q} \sum_{s=1}^{S} D_q t_{smqr} \phi_{qr} \leq W_m \qquad \forall m \qquad (4.30)$$

$$\sum_{q=1}^{Q} \sum_{r=1}^{R_q} t_{smqr} \phi_{qr} \leq M x_{sm} \qquad \forall s, m \qquad (4.31)$$

$$\sum_{r=1}^{R_q} \phi_{qr} = 1 \qquad \forall q \qquad (4.32)$$

$$x_{sm} = 0 \qquad \forall s, m \notin M(s) \qquad (4.33)$$

$$x_{im} \in \{0, 1\} \qquad \forall s, m \qquad (4.34)$$

$$\phi_{qr} \geq 0 \qquad \forall q, r \qquad (4.35)$$

- (4.27) is equal to the total distance traveled by parts. Firstly, the fraction of usage of machine routes is multiplied with the total demand of the associated part type and the number of times machine m precedes machine k in the machine route r of part type q. This multiplication is equal to number of trips made from machine m to machine k. By multiplying the number of trips with the distance between these machines, total distance traveled for carrying this type of part can be obtained.
- (4.28) ensures that the number of assignments made for a tool type does not exceed the number of copies of this tool type.
- (4.29) and (4.30) are the tool slot and time capacity constraints for machines, respectively.
- (4.31) ensures that a machine route is not used, when any of the tool types in its definition is not assigned to any of the machines.
- (4.32) ensures that the total of the fractions of tool route usages add up to 1 for all part types. In other words these constraints enforces demand satisfaction for all part types in the product mix.
- (4.33) represents the technological constraints of the system; tools can only be mounted on certain machines.

- The last two constraints, namely (4.34) and (4.35) are integrality and non-negativity constraints, respectively.

As for the number of variables in the above definition, consider a problem instance with $S = 12$, $N = 12$, $Q = 20$,and $R_q = 81$ $\forall q$; there are 144 binary and 1620 continuous variables. Even tough these figures appear to be small, in problem instances where all tool types can be mounted on all machines, the number of machine routes increases drastically. Consider such a system where $S = 12$, $N = 12$, $Q = 20$,and every part type has three operations routes, each of which contains five operations. Then R_q turn out to be equal to 746496 for all part types. In this case there are again 144 binary but 14929920 continuous variables. Furthermore there are 2149908480 values for z_{mkqr} and t_{smqr} parameters each. There are two main advantages of the above formulation;

- The number of binary variables are required to represent the same problem is either equal or less than the number required for MIP1.
- The number of binary decision variables is independent of number of tool copies.

A detailed analysis of model sizes for MIP2 formulation will be presented in Section 6.1.

5. PROPOSED GA BASED HEURISTIC

This chapter is devoted to the GA based heuristic proposed for tool allocation problem with material handling considerations. Before giving details about the algorithm some background information about GA will be presented.

5.1. Background About GA

GA's (also known as evolutionary algorithms) are based on the basic mechanisms of natural evolution. These mechanisms include but not limited to natural selection, crossover and mutation. The solutions in GA framework are commonly named as individuals and a set of solutions as populations. The solution of the problem is represented in the chromosome of the individuals by using strings with special encoding mechanisms. Furthermore, the quality of the individuals is measured according to their fitness values. The algorithm is initiated with the creation of an initial population. Thereafter, by means of selection, crossover, and mutation search for better individuals is intended over the generations, namely iterations. By imitating the "survival of the fittest rule" in nature, GA's aim to create better individuals by exploiting the information contained in the population [32].

The process of "selection" in GA framework determines the composition of the population in the next generation by selection of individuals from current population. Usually the selection probabilities of the individuals depend on their fitness values. After selection, the individuals may experience "crossover" and/or "mutation" before they are inserted into the new population. The idea of crossover is to combine the strings of two individuals according to some rules to create an offspring, which carries genotypical properties of its parents. However, if the search mechanism is based solely on crossover the probability of quickly converging to a local optimum solution is high. Therefore, randomly changing some of the properties of the individuals to be included in the new generation may be beneficial for increasing the diversity of the search. This means that, when individuals created over the generations possess some properties, which they did not inherited from their parents, the chance of providing new search direction for the algorithm increases. The process of randomly changing the properties of individuals with certain probabilities is called as mutation in the GA framework.

There are numerous methods applied for improving the performance of evolutionary algorithms by researchers. It is beyond the scope of this study to enumerate these methods, since their design is basically based on to the user's imagination. Interested readers are advised to resort one of the textbooks devoted to evolutionary computation [32, 33, 34, 35, 36].

5.2. Overview of The Algorithm

The proposed genetic algorithm is based on the idea to generate good estimations for the values of the binary variables in the formulations given in Chapter 4. As mentioned before, the proposed formulations aim to make tool allocation and route selection decisions simultaneously. However, the binary decision variables in these formulations represent only the tool allocation decisions. Therefore, if the values of these binary variables can be set somehow and treated as given parameters instead of decision variables, then these formulations reduce to LP models. These LP models can be easily and quickly solved to determine the route selections. The aim of the proposed procedure is to search for a "good" set of values for the binary variables in the MIP formulations by using the GA framework as a tool.

The next six sections are devoted to the algorithmic details of the proposed heuristic procedure. They include information about the representation, initial solution generation, selection, mutation and crossover mechanisms, which are common to all GA applications that are encountered in the literature. The last section however, contains information about the special mechanisms utilized for the improvement of the algorithm. Before presenting the details of the algorithm the general steps are given in Figure 5.1.

Algorithm 1

```
Create an initial population, set as CURRENTPOP
Sort CURRENTPOP
NUMITER ← 0
while NUMITER ≠ MaxIter do
    Create an empty population, set as NEWPOP
    NEWPOPSIZE ← 0
    while NEWPOPSIZE ≠ PopSize do
        Select an individual from CURRENTPOP, set as IND
        Generate a random number, set as RANDNUM
        if RANDNUM ≤ CrossoverPr then
            Select an individual from CURRENTPOP, set as MATE
            Apply crossover to IND and MATE, set offspring as IND
        end if
        Generate a random number, set as RANDNUM
        if RANDNUM ≤ MutationPr then
            Apply mutation for IND
        end if
        Calculate objective value of IND
        Add IND to NEWPOP
        NEWPOPSIZE ← NEWPOPSIZE + 1
    end while
    Sort NEWPOP
    Calculate fitness values for NEWPOP
    Set NEWPOP as CURRENTPOP
    NUMITER ← NUMITER + 1
end while
Report best individual in the CURRENTPOP
```

Figure 5.1. General steps of the proposed GA

5.3. Representation

The chromosomes in GAs are used to represent the information contained in the solutions. The representation scheme preference depends on the problem under study and the applied search mechanism. As a rule of thumb, Goldberg [32] proposes the selection of the smallest alphabet that allows a natural expression of the problem. The representation scheme used in this study is based on encoding the allocation of individual tools to machines. For that purpose, a single gene in the chromosome is reserved for every individual tool in the system and it stores the index of the machine on which the tool is mounted. An example chromosome representation of a solution taken from a problem instance with six machines and four tool types each having four distinct tool copies can be seen in Figure 5.2.

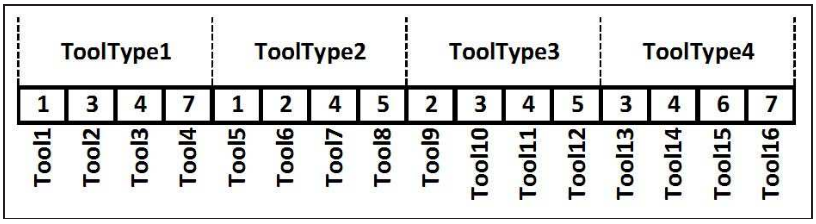

Figure 5.2. A sample chromosome.

According to this solution some of the tool allocations are as follows:

- Tool1 is allocated on Machine1,
- Tool2 is allocated on Machine3,
- Tool3 is allocated on Machine4,
- Tool4 is allocated on Machine7,...

This representation avoids allocating tools to more than one machine, since a single digit is dedicated to every individual tool.

Furthermore, in order to abide tool-machine compatibility issues, it is necessary to assign only valid machine indices for the tools. To achieve this, an eligible index set is kept for each tool type. These eligible machine index lists only contain indices of machines on which the associated tool type can be mounted. Moreover, due to the problem assumptions it is necessary to avoid allocating two tools of the same type to a machine. This is done as follows; whenever a tool is allocated to a machine, the machine's index is removed from the eligible index list of the associated tool type. In that way, mounting more than one copies of a tool type on the same machine can be avoided. However, note that a chromosome can be infeasible with respect to tool slot capacity and machine time limitations. The procedure explained above only avoids creation of solutions which contains assignment of more than one copies of a tool type and the ones that violate the tool-machine compatibility constraints.

5.4. Initial Solution Generation

GA's have the advantage of conducting search not based only a single solution, but a "population" of solutions. Therefore they necessitate creation of an initial population. The information contained in the individuals of this population will constitute the initial gene pool, which will evolve to provide better individuals throughout the generations. It can be stated that, the likelihood of reaching better individuals depend on the richness, namely diversity, of this initial gene pool.

There are various ways of creating an initial population. However, since the diversity is of crucial importance, the creation of the initial population should be based on a randomized process. On the other hand, completely random creation of the chromosomes without considering the validity of the chromosomes, will result in generation of useless encodings. Therefore, in this study a semi-random strategy is followed to create initial populations. The digits in the chromosomes are filled in a random order, by randomly selecting an index from the eligible machine index lists without replacement. In this way, it is ensured that the generated solutions will abide the tool-machine compatibility issues.

Even though the pseudo-random method explained above can avoid generation of invalid chromosomes, it may create solutions that violate tool slot capacity and machine time constraints. Hence, there is a probability, that all the individuals created initially are infeasible with respect to tool slot capacity issues. This case should be avoided since that means there are absolutely no individuals which have good features to pass to their offspring. Therefore, it is ensured that a certain percentage of the initial population is composed of individuals that are feasible with respect to tool slot capacities. The procedure for generating of tool slot capacity feasible individuals is explained in Subsection 5.4.1. On the other hand, it is very likely that a lot of infeasible individuals are created during the initial population generation. Handling of these infeasible individuals is explained in the next section.

5.4.1. Creation of Feasible Individuals

In order to create individuals that are feasible with respect to tool slot capacity, a trial and error based method is used. Like in creation of a random individual, the bits of the chromosome are randomly selected and after each selection an assignment from the eligible machine index list of the associated tool is made. However, in this procedure the number of available slots for each machine is updated whenever an assignment is done. This procedure is repeated until either all of the bits in the chromosome are filled or an assignment for a certain tool cannot be done. In the latter case, all of the previous assignments are discarded and the entire procedure is restarted. Since the bits are randomly selected, it is expected that the procedure terminates with the creation of a tool slot capacity feasible individual in the long term.

5.5. Objective Value Calculation

The chromosomes of the individuals used in this study represent only the tool allocation decisions in the system. There is no information related to the route selection decisions contained in individuals. In order to determine the objective value of the problem represented by the chromosome, it is necessary to find the optimal route selection strategy for the given tool allocation decisions. This can be achieved firstly by converting the chromosome of an individual to binary decision variable values in models presented in Chapter 4. Then, by treating the binary variables as given parameters, the models can be solved to determine the values of other decision variables. Since all the remaining decision variables are of continuous type, the reduced versions of these formulations are LP models and it is possible to solve them quickly to optimality. The crucial point about these models is that their optimal solutions clearly represent the best route selection decisions for the given tool-allocation decisions. Therefore they provide good estimators for the quality of the associated individual.

Before introducing the LP models, it is worth to point that, the objective value calculation scheme explained in this section is used only for individuals that are feasible with respect to tool capacity and machine time constraints. The treatment of infeasible solutions are explained in Section 5.5.3.

Individuals in the population are kept in increasing order of their objective values. Therefore the best individuals occupy the first locations in the list and the worst ones take place at the end. Sorting of the population is made after the assignment of objective values as can be seen in Figure 5.1.

The next two subsections will present the reduced version of two of the proposed models, namely MINLP and MIP2. The reduced version of MIP1 is omitted since it is the linearized version of the MINLP, by means of addition of extra decision variables. The selection of the LP model to be used in the proposed GA is made by conducting numerical experiments and the details about this issue are given in Subsection 6.5.1.

5.5.1. LP Model Based on MINLP

If the binary x_{im} variables of the model given in Section 4.1 are treated as parameter values then the formulation will reduce into the following LP model:

$$\min \quad \sum_{i=1}^{N} \sum_{j=1(F_{ij}^{max}>0)}^{N} \sum_{m=1}^{M} \sum_{k=1}^{M} (f_{ij} d_{mk} x_{im} x_{jk}) \tag{5.1}$$

s.t.

$$\sum_{i=1}^{N} x_{im}\beta_i \leq W_m \qquad \forall m \qquad (5.2)$$

$$\phi_{qr}\alpha_{qri} \leq \sum_{m=1}^{M} x_{im} \qquad \forall i, q, r \in R(q) \qquad (5.3)$$

$$\beta_i = \sum_{q=1}^{Q}\sum_{r=1}^{R_q} \phi_{qr} D_q t_{iqr} \qquad \forall i \qquad (5.4)$$

$$f_{ij} \geq \sum_{q=1}^{Q}\sum_{r=1}^{R_q} \phi_{qr} D_q z_{ijqr} \qquad \forall i, j \qquad (5.5)$$

$$\sum_{r=1}^{R_q} \phi_{qr} = 1 \qquad \forall q \qquad (5.6)$$

$$f_{ij}, \phi_{qr}, \beta_i \geq 0 \qquad \forall i, j, q, r, m \qquad (5.7)$$

5.5.2. LP Model Based on MIP2

If the binary x_{sm} variables of the model given in Section 4.3 are treated as parameter values then the formulation will reduce into the following LP model:

$$\min \quad \sum_{m=1}^{M}\sum_{k=1}^{M} d_{mk} \sum_{q=1}^{Q}\sum_{r=1}^{Rq} z_{mkqr} D_q \phi_{qr} \qquad (5.8)$$

s.t.

$$\sum_{q=1}^{Q}\sum_{r=1}^{Rq}\sum_{s=1}^{S} D_q t_{smqr}\phi_{qr} \leq W_m \qquad \forall m \qquad (5.9)$$

$$\sum_{q=1}^{Q}\sum_{r=1}^{Rq} t_{smqr}\phi_{qr} \leq MX_{sm} \qquad \forall s, m \qquad (5.10)$$

$$\sum_{r=1}^{R_q} \phi_{qr} = 1 \qquad \forall q \qquad (5.11)$$

$$\phi_{qr} \geq 0 \qquad \forall q, r \qquad (5.12)$$

5.5.3. Treatment of Infeasible Individuals

As mentioned before there are two types of infeasibility that can be acquired by the solutions represented in the chromosomes used in this study. These are the violations with respect to the tool slot capacities and machine time limitations.

Firstly, since the LP models presented in previous sections assume that the tool slot capacities are not violated, the tool slot capacity constraints are omitted in their formulation. Hence, there is no need to construct and solve an LP model for individuals that are tool capacity infeasible. Moreover, it is also easy to detect these type of violations by examining its chromosome. This examination simply implies counting the number of tools that are allocated on each machine thereby obtaining the tool slot capacity violation. Such individuals are penalized proportional to the magnitude of their violations.

As for the individuals that violate machine time limitations; it can be stated, that the LP models constructed for such individuals will turn out to be infeasible. If this is the case, the individuals are penalized by assignment of a high value as objective value.

The determination of the value to assign for the objective values of infeasible individuals requires further discussion. Since the problem under study is of minimization type, the number to be assigned to such individuals should be large enough so that every feasible individual has lower values than all of the infeasible ones. On the other hand, setting this value too high cause problems in selection stage of individuals, where their selection probability may depend on their objective values. Actually what is needed is an upperbound for the objective value of the problem instance under study. Such an upperbound can be calculated by assuming that the tools, which are consecutive in the process plans of parts are allocated to their farthest compatible machines. Clearly, this is an upperbound of the problem, because the distance between such machines is aimed to be minimized.

It is also important to note that infeasibility with respect to tool slot capacity is a more serious violation than violating the machine time limitations, in the sense that it is violation of a physical property of the system. Therefore, it is ensured that the objective values assigned to such individuals are always larger than the objective values of other individuals. This is done as follows; the slot capacity violation of an individual is determined and multiplied with 1000, then it is added to the objective value of the individual. In that way, the objective values of individuals that cause more violation of the slot capacity limitations are strictly greater than other individuals.

5.6. Fitness Determination and Scaling

The fitness value of an individual is an indicator of how good the solution represented by its chromosome is. Due to the survival the fittest rule, it is of crucial importance to find a good estimator for the fitness values. The fitness values in this study are determined by using the optimal objective values of the LP models presented in Section 5.5. However, as mentioned in the previous section, the infeasible individuals objective values that are either equal or higher than the upperbound of the problem under study. Therefore the fitness values assigned to these infeasible individuals is equal to 0. The procedure that will be explained in this section is for assigning fitness values only to feasible individuals.

Since the objectives of the LP models are minimization, high fitness values are assigned to the solutions that have the low objective values. Moreover it is ensured that the assigned fitness values are in the interval between 100 and 200. This is made by using the linear scaling of objective values by using the following formulas ([32], [37]):

$$f_i = aOV_i + b \tag{5.13}$$

$$a = 100/(OV_{Best} - OV_{Worst}) \tag{5.14}$$

$$b = 200 - OV_{Best}a \tag{5.15}$$

During the preliminary experiments it is observed that the selection of the interval plays a crucial role in the performance of the selection strategy. The choice of the interval [100, 200] is to avoid assignment of fitness values that have dramatic differences. For instance consider the interval [1, 100]; the fitness of the best individual is equal to 100 times the fitness value of the worst individual. For the interval [100, 200] however, this factor is only 2. The reason for the tendency to not assigning fitness values with such dramatic differences will be made clear in Section 5.7.

The fitness values calculated by the linear scaling scheme above are used in assigning selection probabilities to individuals (Section 5.7) and in calculation of population diversity (see Subsection 5.10.2).

5.7. Selection

The selection mechanism must be designed in a way that it favors the fit individuals of the population, so that the search leads itself to better solutions. This feature is referred as "selection pressure" of the mechanism. The probability of selecting fitter individuals increases with increasing selection pressure. However, exerting too much selection pressure speeds up the convergence of the population, because in this case only a small subset of the population is used to build up the next generation. This two extremes are related to the trade off between intensification and diversification.

If the algorithm presented in Figure 5.1 is analyzed, it can be seen that two selection procedures take place during the course of the algorithm at steps 8 and 11. The first selection aims to select the individual that will be inserted in the new population after crossover and mutation procedures, whereas in the latter one, the second parent (MATE) that will be included in the crossover is selected. In order to balance the trade off between intensification and diversification of the search, the first selection is made randomly and the second one by applying some selection pressure. Making the first selection in a completely random manner helps the search to avoid getting trapped at local optimum. On the other hand, in order to ensure at least one of the parents is one of the fit individuals, the selection probability for the second parent is higher for individuals that have high fitness values. This strategy helps the algorithm to exploit the good properties possessed by fit individuals and helps the search to intensify.

Two alternative selection procedures are proposed for the mate selection in the crossover. These are roulette based selection and rank based selection. Both of these selection procedures exert some selection pressure and both have some advantages and disadvantages. They will be explained in the following two subsections. The determination of which of these procedure to use is made according to the results of the numerical experiments presented in Subsection 6.5.6.

5.7.1. Roulette Wheel Selection

In roulette wheel selection (RWS), selection probabilities of individuals are assigned proportional to their fitness values. Let F denote the total fitness of the individuals in the population, then selection probability of the ith individual is calculated as f_i/F, where f_i represents the fitness of the ith individual. RWS has the disadvantage of assigning very high selection probabilities to super individuals" in the population. The term super individuals is used to refer individuals which have very high fitness values relative to the rest of the population. This situation leads to the dominance of the population by these individuals in a short period of time [33]. In order to overcome this situation, the fitness scaling interval is selected as [100, 200], since this selection avoids dramatic differences between the assigned probabilities more than the interval [1,100].

5.7.2. Rank Based Selection

Rank based selection (RBS) is based on the idea of assigning selection probabilities with respect to the ranks of individuals. The least fit feasible individual gets a rank of 1 and the best individual gets the highest rank. The selection probabilities are calculated by dividing the individual's rank to the total of ranks. In this way, it is ensured that the selection probability decrease when the fitness decreases. Since selection probability assignment scheme does not depend on the magnitude of the fitness values, RBS eliminates the problem of super individuals. Even if the fitness values of the best individuals converge to a very narrow range the selection probabilities assigned by RBS procedure will not be effected [32].

5.8. Crossover

The foundation of the GA is explained by building block hypothesis. According to this hypothesis, the search mechanism of GAs depends on combining small block into larger blocks. That is to say the small pieces of chromosomes which enable individuals to have good fitness values are aimed to be combined to create better individuals. This is the very same idea behind crossover. It aims to combine good features of both of the parents to create the offspring [32].

After randomly selecting the first individual from the population, a random number between 0 and 1 is generated. If this random number is less than the crossover probability (CP) parameter, a second individual is selected for mating. Afterwards a new individual, namely the offspring is generated by using the information contained in the chromosomes of these individuals. The most common types of crossover, encountered in the literature, are as follows [36]:

- One-point crossover: In one point crossover, generation of the chromosome of the offspring is as follows; the bits coming before a randomly chosen point are taken from one parent and the bits coming after are taken from the other parent.
- Two-point crossover: Two point crossover is an enhanced version of one-point crossover in which two location on the chromosome are randomly chosen and the segments defined by these points are swapped to create the offspring.
- Uniform crossover: The offspring chromosome is created bit by bit, when uniform crossover is used. Every bit randomly is taken either from the first parent or the second one.

In this study, uniform crossover is used with a minor difference. During the offspring creation, the probability of selecting the bit from parents is biased toward the fitter one of the parents. That is to say, the generated offspring inherits more features from the fitter one of the parents. The magnitude of this bias is is a parameter that can be modified.

5.9. Mutation

Crossover aims to combine the features of the individuals to create offspring for the next generation. Therefore, since crossover works on the existing gene pool of the population, the addition of new features into the gene pool is impossible. When new features are not added and only the ones generated at initial population creation are used, the search is likely to terminate with premature convergence. Hence, before a new individual is inserted into the new population, it experiences mutation with a certain probability. Mutation involves changing one bit of its chromosome randomly by obeying the validity rules of the chromosome, as explained in Section 5.4. On the other hand, mutation is never performed on an individual if it is better than the best individual in the current population. The reason behind this rule is not to destroy a solution that is already better than the incumbent. Determination of the mutation probability (MP) for individuals that does not have this property is made by conducting numerical experiments, which are presented in Subsection 5.10.4.

5.10. Other Mechanisms used for Improvement

The procedures that are explained in the previous sections in this chapter are common GA mechanisms. In the rest of this chapter more advanced mechanisms designed for improvement of the proposed algorithm are explained.

5.10.1. Elitist strategy

The elitist strategy is defined as the inclusion of the fittest individuals of the current population in the new population without applying mutation or crossover. This could be done by carrying only the best individual or a specified number of individuals to the next generation. In that way, it is ensured that the objective value of the best individual in the current population decreases monotonically throughout iterations. The size of the portion of the current population to be passed on to the new generation is determined through the numerical experiments presented in Subsection 6.5.5.

5.10.2. Diversity Analysis

The premature convergence of the population is the counterpart for getting stuck at the local optimum in the domain of GA. It is a very serious issue which should be avoided by taking some special measures. The first step in avoiding premature convergence is to detect the convergence before it happens. The diversity of the population can be used as an indicator for detecting the convergence of the population. There are two types of diversity measures; phenotypical and genotypical [38]. The phenotypical diversity can be used to estimate the diversity by the phenotypical properties of the individuals, namely the fitness values, whereas the genotypical diversity measure tries to measure diversity by determining the differences in the chromosomes themselves. It is apparent that the genotypical diversity measures provide more accurate estimations about the diversity of the population, but the main disadvantage of them is their high computational effort requirement.

Scaled fitness values are used to calculate the phenotypical diversity of the population, (5.16) is used in this study to estimate the diversity.

$$\epsilon = (f_{max} - \bar{f})/(f_{max} - f_{min}) \tag{5.16}$$

ϵ : Phenotypical diversity,

$\bar{f}$: Average fitness value of feasible individuals,

f_{min} : The fitness of the worst feasible individual,

f_{max} : The fitness of the best feasible individual.

Please note that if the fitness values of the feasible individuals approach to that of incumbent, ϵ decreases. If all of them are equal, ϵ drops to zero level, indicating that the population is converged".

5.10.3. Adjustment of the Population Size

The size of the initial population is very important in order to provide more search directions at this early stage of the search. For this purpose the size of the initial population is determined twice as the nominal population size level.

Apart from starting with a larger population at the beginning, the population size is also adjusted adaptively with respect to the value of the population diversity. Whenever ϵ gets equal to zero, the size of the population is increased. Note that this situation is equivalent to the state where all feasible individuals are the same. When this occurs, the population size is increased up to a level which is higher than the nominal level by a certain percentage. The increase in the population size is achieved by addition of randomly generated individuals. In that way, it is aimed to increase the diversity of the population thereby presenting new search directions to help the procedure escape from local optima.

At every iteration, if the population size is not equal to the nominal level, it is decreased. This is done by removing a certain number of the worst individuals of the population. In this way, it is ensured that the population size is reduced gradually to its nominal level.

5.10.4. Adjustment of MP

Keeping the MP at a fixed level throughout the search may not be a good idea for several reasons. Actually, it is an important parameter which can be tailored for every individual. For instance, it may be beneficial to apply high mutation rates for infeasible individuals and relatively low rates for feasible ones. Furthermore, the level of mutation can also be altered according to the status of the search. The diversity measure can be useful for achieving this aim. Whenever the diversity of the population tends to decrease, it may be a good idea to increase the probability of mutation to avoid premature convergence.

Due to the reasons mentioned above, the following adaptive strategy is adopted for determination of MP:

$$MP_i = MP_f + MP_v^1(1-\epsilon) + MP_v^2(f_{max} - f_i)/(f_{max} - f_{min}) \tag{5.17}$$

MP_i	:	MP of ith individual
MP_f	:	Fixed portion of the probability,
MP_{v1}	:	First variable portion,
MP_{v2}	:	Second variable portion,
ϵ	:	Phenotypical diversity.

Determination of the above parameters is made by conducting several numerical experiments, which will be given in Subsection 6.5.3.

6. NUMERICAL STUDIES

Intensive numerical experiments are conducted to test the effectiveness of the proposed mathematical formulations and the heuristic algorithm. For this purpose, a problem instance generation scheme is presented and several instances are created. After introducing these problem instances, the results of the experiments for the proposed methods will be presented.

Experimentation covers MIP1 and MIP2 solutions and the GA based heuristic. The reason for omitting MINLP model is its non-linear structure.

The proposed MIP formulations and the LP formulation used in GA are solved by using CPLEX 11.0. The software used for experimentation is developed in Microsoft Visual Studio 2008 environment by using C# 3.0. The experimentation platform has an Intel ® Core™ 2 Duo E6555 @ 2.33 GHz processor and 4.00 GB of RAM.

6.1. Generated Problem Instances

In order to test the performance of the proposed methods, a random problem instance generation scheme is utilized. The generated instances are of different sizes and they vary in their difficulty levels. The problem size depends on the number of machines, tools, part types and the number of alternative operation routes of these part types, whereas the imposed technological constraints determine the difficulty level of the generated problem. These technological constraints are the compatibility issues of tool types to the machines. Tool allocation problem encountered in systems where each tool can be mounted on a subset of machines is easier to solve than the one encountered in systems where all of the tools can be mounted on all machines. This reason for this fact is that, the latter have larger feasible regions.

Forty five instances are generated and divided into three categories with different difficulty levels. These categories are labeled with capital letters and ordered from the easiest to hardest. Instances with the same index in each category are equivalent except for the machine-tool compatibility issues. Before giving specific information about these difficulty categories; common data, which is used in all the instances regardless of its category will be presented.

To start with; tool types, the operations that can be executed by these tool types and their processing times are presented in Table A.1. There are 12 tool types and 24 operations in the system. Each tool type can execute two operations. The processing times of these operations are randomly generated by using uniform distributions.

The product range of the hypothetical manufacturing system includes 20 part types, each of which has one to three alternative operation routes. These operation routes are generated by randomly selecting subsets from the 24 operations. The number of operations per route varies from four to six operations. Each operation route commences with "Receiving" and terminates with "Shipping". These special operations are executed by "Receival tool" and "Shipment tool", which are already allocated to machines named "Receival" and "Shipment". The complete definition of these operation routes are given in Tables A.2 and A.3.

The amount to be produced of each of the part types are generated such that the overall shop workload will be around 90 per cent. These required production amounts for part types are given in Table A.4.

The number of machine types is equal to four in all instances. However, the number of copies for each of these machine types varies from instance to instance. Likewise, even tough the number of tool types is equal for all instances, the number of replicates for each of these tool types differs among the instances. Actually, by increasing the number of copies of machine and tool types, the problem size can be modified. Therefore, in order to test the proposed methods, instances with different problem sizes are generated. The problem sizes and difficulty levels of the generated problem instances are given in Table 6.1.

Table 6.1. The problem sizes and difficulty levels of problem instances are given in the below table. The figures in the last three columns indicate the number compatible machines for each tool

	Problem Size				**Difficulty**		
Instances	**Mach.**	**Tools**	**Part Types**	**Routes**	**Set A**	**Set B**	**Set C**
PI-01	4	12	14	21	2	3	4
PI-02	4	12	13	20	2	3	4
PI-03	4	12	16	27	2	3	4
PI-04	4	12	15	24	2	3	4
PI-05	4	12	15	24	2	3	4
PI-06	8	24	15	26	4	6	8
PI-07	8	24	13	24	4	6	8
PI-08	8	24	17	29	4	6	8
PI-09	8	24	15	24	4	6	8
PI-10	8	24	15	26	4	6	8
PI-11	12	36	15	26	6	9	12
PI-12	12	36	12	18	6	9	12
PI-13	12	36	14	24	6	9	12
PI-14	12	36	15	27	6	9	12
PI-15	12	36	14	24	6	9	12

The figures in the last three columns of Table 6.1 denote the number of machines, on which each tool type can be mounted. As mentioned above the difficulty of the problem instance is highly dependent on the tool-machine compatibility issues. More detailed information about machine-tool compatibility issues for each category is given in Tables A.5, A.6, and A.7. It is worth to mention again that, the instances in category C are harder to solve, because more alternative allocations need to be considered in these instances. The number of variables with respect to the generated problem instances for MINLP, MIP1 and MIP2 models are presented in Tables A.9, A.8, and A.10, respectively.

Since the proposed methods aim to solve the tool allocation problem with material handling considerations, the material handling aspect of the hypothetical system should also be considered. It is assumed that the parts are carried by AGVs throughout the system, which are required to travel along the predetermined network of aisles. The parts are carried from the output points to the input points of machines. The parts enter the system through the output point of Receival and leave at the input point of the Shipment. The material handling flow path network for the instances with four machines is given in Figure 6.1. The material handling flow path networks for the systems with eight and 12 machines are given in Figures A.1 and A.2, respectively.

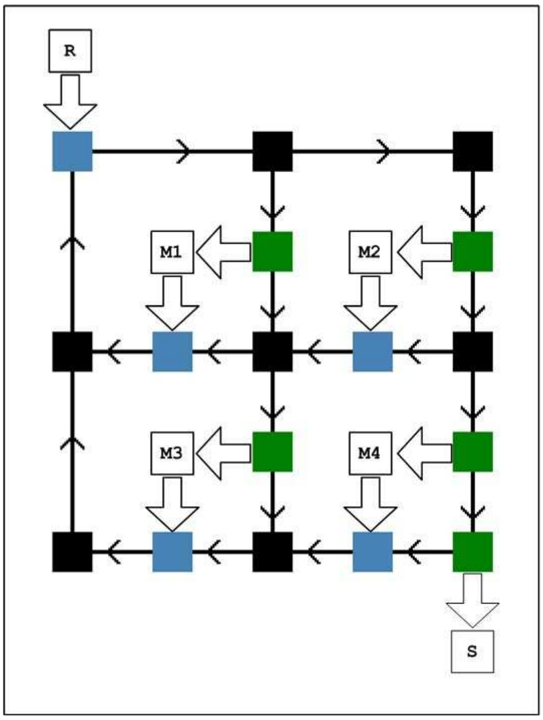

Figure 6.1. Layout for the problem instances with 4 machines

It is obvious that, the shortest travel distances between the machine centers can be calculated by analyzing the material handling flow path network. Actually these distance values are used as the values for the d_{mk} parameters that appear in the objective functions of the proposed mathematical formulations.

6.2. General Information about the MIP Formulation Results

CPLEX 11.0 utilizes a branch and bound method for solving MIP's. The time limit is set as eight hours. The best integer solution found so far (incumbent) and the current lower bound used for pruning is reported at the end. If the gap between incumbent and the lower bound decreases below 0.01 per cent the search terminates by reporting the optimal solution of the problem. For more detailed information about the MIP solution methods uitilized by CPLEX^{TM} 11.0 the user is referred to users manual [39]. The summary of the Cplex parameters are given in Table 6.2.

Table 6.2. CPLEX paremeters for MIP solution

Paremeter	**Value**
Optimality gap	0.01%
Time limit	8 hours
Working memory limit	1500 MB

If the specified memory limit given in Table 6.2 is exceeded during branch & bound procedure the hard drive of the unit is utilized for storage of node information.

6.3. MIP1 Results

The results obtained for 45 problem instances are given in Table 6.3. Total run time for MIP1 experiments is 240 hours, 2 minutes and 39 seconds. The average optimality gap is calculated as 64.7 per cent. Only in 15 instances out of 45, the optimal solution is achieved with a 0.01 per cent gap. Actually, these are the smallest instances. In all other instances, the procedure failed to achieve optimal solution during the eight hours of runtime.

Table 6.3. MIP1 results. (Best: Best integer solutions, LB: Lower bounds, Gap: Per cent deviation of the incumbent solution from the lower bound, Time: Solution generation times in seconds.)

	Set A				Set B				Set C			
PI	**Best**	**LB**	**Gap**	**Time**	**Best**	**LB**	**Gap**	**Time**	**Best**	**LB**	**Gap**	**Time**
01	261157.9	261157.9	0.00	0	243330.0	243322.5	0.00	4	218546.6	218546.6	0.00	18
02	252368.6	252368.6	0.00	1	224506.5	224485.7	0.01	2	223535.8	223519.1	0.01	16
03	269077.7	269077.7	0.00	1	244793.4	244793.4	0.00	4	231924.8	231905.8	0.01	31
04	262223.7	262223.7	0.00	0	242440.0	242440.0	0.00	6	232971.5	232970.0	0.01	35
05	255770.0	255770.0	0.00	1	243431.2	243428.9	0.00	6	232911.5	232897.0	0.01	38
06	351817.1	65877.5	81.28	28800	332191.8	23265.0	93.00	28800	337684.9	0.0	100.00	28800
07	358700.0	63748.0	82.23	28800	343113.4	1413.0	99.59	28800	333717.2	0.0	100.00	28800
08	364571.1	20129.7	94.48	28800	330277.4	964.4	99.71	28800	327332.6	0.0	100.00	28800
09	351090.1	55452.3	84.21	28800	326622.5	0.0	100.00	28800	337150.2	0.0	100.00	28800
10	359797.9	70762.0	80.33	28800	331359.2	2961.0	99.11	28800	336574.3	3048.1	99.09	28800
11	590667.3	0.0	100.00	28800	551111.5	0.0	100.00	28800	554859.9	0.0	100.00	28801
12	623201.1	0.0	100.00	28800	592600.5	0.0	100.00	28800	538470.6	0.0	100.00	28800
13	589551.4	0.0	100.00	28800	541508.5	0.0	100.00	28800	546740.9	0.0	100.00	28795
14	594331.0	0.0	100.00	28800	581618.6	0.0	100.00	28799	566843.8	0.0	100.00	28800
15	583240.3	0.0	100.00	28800	554993.8	0.0	100.00	28799	560980.7	0.0	100.00	28800

6.4. MIP2 Results

The results obtained for 45 problem instances are given in Tables 6.4. Total run time for MIP2 experiments is 107 hours 51 minutes and 19 seconds. The average optimality gap is calculated as 2.05 per cent. In 32 instances out of 45, the optimal solution is achieved with a gap less than or equal to 0.01 per cent.

These results are encouraging, since only in instances with medium and large problem sizes of categories B and C, the procedure fails to achieve 0.01 per cent optimality during the eight hours of run time.

6.5. Experiments for parameter tuning of GA

This section is devoted to results of numerical experiments, which are conducted for parameter tuning of the proposed GA. The performance of the tested experimental settings are compared according to their per cent deviations from MIP2 results. The reason for using MIP2 as benchmark is the fact that it generates the optimal solutions for the most of the problem instances and the optimality gap values obtained for MIP2 is much more better than of MIP1.

Experiments for each problem instance are replicated by using 10 different random number seeds. The best solution among the results obtained in these replications is reported as the incumbent for the associated problem instance. Solution generation times on the other hand, are taken as the total elapsed time during the execution of 10 replications.

Some of the parameters used in the proposed GA are determined during the preliminary experimentation. It is found out that the performance of the proposed GA procedure is not effected due to small variations in the values of these parameters. Therefore, they are set to certain values as given in Table 6.5 prior to main experimentation for more important parameters.

Table 6.4. MIP2 Results

	Set A				Set B				Set C			
PI	**Best**	**LB**	**Gap**	**Time**	**Best**	**LB**	**Gap**	**Time**	**Best**	**LB**	**Gap**	**Time**
01	261157.9	261157.9	0.00	0	243330.0	243330.0	0.00	3	218546.6	218538.1	0.00	16
02	252368.6	252368.6	0.00	1	224506.5	224490.6	0.01	1	223535.8	223535.8	0.00	26
03	269077.7	269077.7	0.00	0	244793.4	244793.4	0.00	3	231924.8	231924.8	0.00	23
04	262223.7	262223.7	0.00	0	242440.0	242440.0	0.00	3	232971.8	232955.9	0.01	23
05	255770.0	255770.0	0.00	0	243431.2	243431.2	0.00	2	232911.5	232905.2	0.00	22
06	351453.6	351453.6	0.00	12	315412.1	315384.7	0.01	499	305899.7	300549.6	1.75	28802
07	358700.0	358700.0	0.00	2	325808.9	325808.9	0.00	69	316056.7	316025.2	0.01	703
08	364571.1	364542.8	0.01	11	318989.8	318959.4	0.01	437	314479.6	305478.1	2.86	28802
09	348651.4	348642.5	0.00	5	314931.8	314900.5	0.01	662	314420.0	295412.4	6.05	28804
10	359797.9	359778.6	0.01	9	312826.6	312826.6	0.00	71	312826.6	309544.4	1.05	28803
11	564755.8	564699.5	0.01	69	508869.1	474011.1	6.85	28803	517915.6	445547.8	13.97	28807
12	618034.6	617973.5	0.01	43	532045.7	496728.0	6.64	28798	524381.0	442770.6	15.56	28811
13	563267.4	563214.1	0.01	399	490719.5	469969.1	4.23	28802	506105.3	457512.5	9.60	28942
14	574937.0	574887.2	0.01	48	488560.8	488512.2	0.01	10074	499122.7	464809.9	6.87	29186
15	545471.0	545443.1	0.01	67	499850.6	475981.6	4.78	28802	514914.4	454422.6	11.75	28813

Table 6.5. Values of parameters that are determined during preliminary experimentation

Parameter	**Value**
Bias to fitter individual during crossover	0.6
Maximum number of generations	1000
MP for infeasible individuals	0.8
Percentage of feasible individuals in the initial population	2%
Percentage increase in population size in case of convergence	25%

Since it is very time consuming to use full factorial experimental design for fine tuning of all parameters, most of the time one factor at a time (OFAT) strategy is followed. This means, while experimenting to find the most suitable value for a parameter, the other parameters are kept fixed. More specifically, in the experiments conducted for determination of crossover and mutation probabilities, RWS is used for MATE selection and the best 5 per cent of the individuals are transferred directly to the new population at each iteration.

6.5.1. Selecting LP Model for Fitness Calculation

Two alternative LP models are proposed in Section 5.5 for objective value determination. Since they are only alternative formulations of the same problem, their optimal solutions are equivalent. However, their solution times are significantly different and it is worth to mention that, these LP models are to be solved very frequently during the operation of the GA. Hence the speed of generating solutions is critically important.

As mentioned before, these LP models are the reduced versions of MINLP and MIP2. Furthermore, the main advantage of MIP2 is that it uses less number of binary variables. The number of continuous variables however, are much higher for most of the instances. The details about the number of variables required for formulation are given in Tables A.8 and A.8. The reduced versions of these problems do not contain any binary decision variables, but variables only of continuous type. Thus, the number of variables of LP2 is higher than of LP1. In the preliminary experimentation, it is observed that the solution generation times are higher when LP2 is utilized in GA. Generally speaking, the reason may be due to the fact that, a model that have less number of decision variables might be solved easier than a model that have more number of variables, in the framework of LP. Another reason for the difference in solution times of LP1 and LP2 may be the time spent when constructing these models in CPLEX® ILOG environment. Since more number of objects are included in LP2, more time is spent before even attempting to solve the model. It is also worth to mention a technical detail in the application; in order to decrease the total solution times, the LP models are not constructed from scratch every time. They are constructed through addition of associated elements to a base model.

Due to the reasons explained above, LP2 is used for determination of objective values in the entire experimentation.

6.5.2. Population size determination

The evolutionary algorithms do not work on a single solution but a group of solutions. The number of solutions to be worked on is determined by the population size. Therefore it is one of the most crucial parameters. Actually as the problem size increases the population size should also be increased, because when working with long strings it is necessary to have more rich and diverse gene pool. Likewise, when the alphabet, in other words the eligible value that the genes of a chromosome can take, gets bigger, a population with more number of individuals is required. Thus, in this study the population size differs according to both problem size and difficulty as given in Table 6.6.

Table 6.6. Population sizes for different problem size and difficulty levels

			Population Size		
Instances	**# of mach.**	**# of tools**	**Set A**	**Set B**	**Set C**
PI 01-05	4	12	100	150	250
PI 06-10	8	24	200	300	500
PI 11-15	12	36	300	450	750

6.5.3. Determination of CP

During the preliminary runs it is observed that one of the most critical parameters of the GA procedure is the CP. For selecting a suitable value for CP a set of experiments are conducted. Since, it is aimed to modify MP adaptively as explained in Subsection 5.10.4, various crossover probabilities are tested by using a set of mutation probabilities. The results of these experiments can be seen in Table 6.7. The figures in this table denote the average percent deviations from MIP2 solutions. They are calculated by taking the average of the deviations for 45 problem instances. Negative deviation values indicate that solutions found are worse than of MIP2's on the average. It is also worth to note that, the mutation probability levels in this table are used only for feasible individuals. For infeasible individuals, a higher level of mutation probability, namely 0.8 is applied.

Table 6.7. Average per cent deviations from the MIP2 solutions with various crossover and mutation probabilities

	Mutation				
Crossover	**0.2**	**0.3**	**0.4**	**0.5**	**Avg**
0.8	-0.34	-0.15	-0.23	-0.25	-0.24
0.9	-0.31	-0.23	-0.27	-0.40	-0.30
1.0	-0.20	-0.48	-0.24	-0.41	-0.33
Avg	-0.28	-0.29	-0.25	-0.35	

More detailed information about these experiments, can be found in Tables B.1 - B.4. As can be seen in the above Table, the average per cent deviation between the solutions of GA and MIP2 are higher than -0.5 per cent. This indicates that the GA solutions are very close, but slightly worse than MIP2 solutions on the average. According to these results, the CP is selected as 0.8 since this selection minimizes the average per cent deviation.

6.5.4. Adaptive MP determination

As described in Section 5.10.4, it is aimed to modify the MP depending on the status of the ongoing search and also on the quality of the individual, to which the mutation will be applied. For this purpose, (5.17) presented in Subsection 6.5.4 is proposed. By using this equation the mutation probability is determined according to the diversity of the population and fitness of the specific individual.

The first set of experiments are conducted to test various levels of P_f and P_v^1. For sake of completeness (5.17) is repeated here.

$$MP_i = MP_f + MP_v^1 \cdot (1 - \epsilon) + MP_v^2 \cdot (f_{max} - f_i)/(f_{max} - f_{min})$$

The results are as presented in Table 6.8. Due to the results obtained in the previous section the CP is decided to be taken as 0.8.

As can be seen in Table 6.8 the best experimental setting is the one, in which MP_f is equal to 0.3, MP_v^1 is equal to $= 0.3$ and MP_v^2 is equal to zero. This results shows that it is not a good idea to add the second portion of the MP, namely the portion that is dependent on the individual's fitness value. This claim can also be supported by analyzing Table 6.8 in detail. The average deviation for experimental settings in which $MP_v^2 = 0$ is equal to -0.18, whereas the same value is -0.23 for the experimental settings in which $MP_v^2 = 0.1$.

Table 6.8. Performance of various adaptive parameter fine tuning parameter values

Name	$\mathbf{MP_f}$	$\mathbf{MP_v^1}$	$\mathbf{MP_v^2}$	**Deviation**
Exp13	0.2	0.3	0.0	-0.17
Exp14	0.2	0.4	0.0	-0.22
Exp15	0.3	0.2	0.0	-0.14
Exp16	0.3	0.3	0.0	-0.09
Exp17	0.4	0.1	0.0	-0.11
Exp18	0.4	0.2	0.0	-0.32
Exp19	0.1	0.3	0.1	-0.26
Exp20	0.1	0.4	0.1	-0.28
Exp21	0.2	0.2	0.1	-0.22
Exp22	0.2	0.3	0.1	-0.23
Exp23	0.3	0.1	0.1	-0.19
Exp24	0.3	0.2	0.1	-0.19

6.5.5. Percentage of individuals to transfer directly to the new population

According to elitist strategy, it is important to preserve good individuals in the population. Otherwise, the search has to be start from scratch at every iteration. Therefore, some portion of the population, which contains the best individuals, should be included in the new population without any mutation and crossover. However, the drawback of this strategy is that it may cause to premature convergence of the population at an early stage. Thus, some experimentation is required to determine the most suitable value. Four values are tested for this purpose. The results can be seen in Table 6.9.

According to the results given in Table 6.9, it is decided to set this parameter to 5 per cent.

Table 6.9. Experiment results for determination of percentage of individuals to be transferred directly to the new population

	Elitist percentage	**Deviation**
Exp25	2.5	-0.36
Exp16	5.0	-0.09
Exp26	7.5	-0.14
Exp27	10.0	-0.14

6.5.6. MATE selection for crossover

RWS is tested as an alternative mechanism for MATE selection in the crossover. As described in Section 5.7, RWS exerts a high selective pressure to the most-fit members of the population, since the selection probability is determined depending on the magnitude of the fitness of the individual. The selection probability is the ratio of the individual's fitness over the total fitness of the feasible individuals in the population. A serious drawback of this method is that it is very biased to the super individuals in the population and may lead quickly to premature convergence. The results of this last experiment are given in Table 6.10.

Table 6.10. Experiments for determination of MATE selection procedure

	MATE Selection	**Deviation**
Exp28	RWS	-0.42
Exp16	RBS	-0.09

The results given in the table above indicate that RBS performs better than RWS for the proposed algorithm, which was an expected outcome due to the aforementioned disadvantage of the RWS procedure.

6.5.7. Increasing the nominal population sizes

A final set of experiments are conducted to see whether increasing the nominal population size levels will improve the solutions or not. For this purpose, the population sizes given in Table 6.6 are multiplied by two. This new experimental setting is named as Exp35 and its population sizes are presented in Table 6.11 .

Table 6.11. Population sizes for Exp33

Instances	Mach.	Tools	Population size		
			Set A	Set B	Set C
PI 01-05	4	12	200	300	500
PI 06-10	8	24	400	600	1000
PI 11-15	12	36	600	900	1500

The results obtained by this new setting and their comparison with results of the setting with original population size levels are presented in Table 6.12.

Table 6.12. Experiments with increased nominal population size levels

Setting	Time	Dev.
Exp29	19302	-0.08
Exp16	12464	-0.09

These results indicate that increasing the population has very little effect on improving the performance of the proposed algorithm. The solution generation times on the other hand, are increased significantly. Therefore, it is decided to keep the population sizes at their original levels.

6.5.8. Final form of the proposed GA and its comparison with MIP2

For the sake of completeness, the selected parameter values along with the final form of the algorithm will be presented in this section. Table 6.13 summarizes the selected values for all parameters. These parameters are chosen since they resulted in minimum deviation from MIP2 results. The final form of the proposed algorithm is presented in Figure 6.2. The MP_i parameter that appears in step 16 is calculated by using the formula given in (5.17). Updating the population size at step 26 is made as explained in Subsection 5.10.3.

Table 6.13. Final values for GA parameters

Parameter	**Value**
Selection method	RBS
CP	0.8
MP for infeasible	0.8
MP_f	0.3
MP_v^1	0.3
MP_v^2	0.0
Fitness scaling interval	[100, 200]
Elitist percentage	5%
Bias to fitter in crossover	0.6
Max. # of generations	1000
% of feasible individuals in initial pop.	2%
% increase in pop. size when converged	25%

A detailed comparison of the solutions generated by using this final form of the GA are given in Table 6.14. Another comparison between these two methods for their solution generation times is given in Table 6.15. The total time spent for generating solutions by using the GA is 3 hours, 24 minutes, and 44 seconds, whereas the duration for generating MIP2 solutions is 107 hours, 51 minutes, and 19 seconds.

Algorithm 2

```
Create an initial population, set as CURRENTPOP
NUMITER ← 0
while (NUMITER ≠ MaxIter) do
    Create an empty population, set as NEWPOP
    NEWPOPSIZE ← 0
    while (NEWPOPSIZE ≠ CURRENTPOPSIZE) do
        Select randomly an individual from CURRENTPOP, set as IND
        Generate a random number, set as RANDNUM
        if (RANDNUM ≤ CP) then
            Select an individual from CURRENTPOP by RBS, set as MATE
            Apply crossover to IND and MATE, set offspring as IND
        end if
        Generate a random number, set as RANDNUM
        if (IND is infeasible ∨ RANDNUM ≤ 0.8) then
            Apply mutation for IND
        else if (IND is feasible ∨ RANDNUM ≤ MP_i) then
            Apply mutation for IND
        end if
        Calculate objective value of IND
        Add IND to NEWPOP
        NEWPOPSIZE ← NEWPOPSIZE + 1
    end while
    Sort CURRENTPOP in increasing order of objective values
    Calculate scaled fitness values and selection probabilities for NEWPOP
    Set NEWPOP as CURRENTPOP
    Update CURRENTPOPSIZE
    NUMITER ← NUMITER + 1
end while
Report best individual in the CURRENTPOP
```

Figure 6.2. Final form of the proposed GA

Table 6.14. Comparison of results obtained by GA and MIP2

	Set A			Set B			Set C		
PI	**MIP2**	**GA**	**Dev.**	**MIP2**	**GA**	**Dev.**	**MIP2**	**GA**	**Dev.**
01	261157.9	261157.9	0.00	243330.0	243330.0	0.00	218546.6	218546.6	0.00
02	252368.6	252368.6	0.00	224506.5	224506.5	0.00	223535.8	223535.8	0.00
03	269077.7	269077.7	0.00	244793.4	244793.4	0.00	231924.8	231924.8	0.00
04	262223.7	262223.7	0.00	242440.0	242440.0	0.00	232971.8	235034.6	-0.89
05	255770.0	255770.0	0.00	243431.2	243431.2	0.00	232911.5	232911.5	0.00
06	351453.6	352486.7	-0.29	315412.1	319382.7	-1.26	305899.7	308975.8	-1.01
07	358700.0	358700.0	0.00	325808.9	326791.7	-0.30	316056.7	321933.5	-1.86
08	364571.1	364571.1	0.00	318989.8	322053.2	-0.96	314479.6	314765.7	-0.09
09	348651.4	351665.7	-0.86	314931.8	321344.0	-2.04	314420.0	315277.1	-0.27
10	359797.9	360129.0	-0.09	312826.6	321391.7	-2.74	312826.6	325655.0	-4.10
11	564755.8	565157.5	-0.07	508869.1	512833.4	-0.78	517915.6	490453.2	5.30
12	618034.6	619551.2	-0.25	532045.7	530779.3	0.24	524381.0	523036.3	0.26
13	563267.4	567981.7	-0.84	490719.5	492433.7	-0.35	506105.3	490326.3	3.12
14	574937.0	574937.0	0.00	488560.8	492599.3	-0.83	499122.7	490315.5	1.76
15	545471.0	546693.1	-0.22	499850.6	496560.7	0.66	514914.4	489726.6	4.89
Avg.	396682.5	397498.07	-0.18	353767.7	355644.7	-0.56	351067.5	347494.6	0.47

Table 6.15. Comparison of solution generation times for GA and MIP2

	Set A			Set B			Set C		
PI	**MIP2**	**GA**	**Dev.**	**Best**	**GA**	**Dev.**	**Best**	**GA**	**Dev.**
01	0	37	37	3	66	63	16	112	96
02	1	37	36	1	61	60	26	114	88
03	0	38	38	3	65	62	23	111	88
04	0	37	37	3	59	56	23	111	88
05	0	42	42	2	64	62	22	117	95
06	12	114	102	499	210	-289	28802	373	-28429
07	2	114	112	69	217	148	703	383	-320
08	11	118	107	437	211	-226	28802	382	-28420
09	5	116	111	662	207	-455	28804	369	-28435
10	9	119	110	71	205	134	28803	375	-28428
11	69	269	200	28803	466	-28337	28807	887	-27920
12	43	256	213	28798	449	-28349	28811	847	-27964
13	399	255	-144	28802	450	-28352	28942	854	-28088
14	48	258	210	10074	446	-9628	29186	857	-28329
15	67	254	187	28802	466	-28336	28813	869	-27944
Avg.	44.4	137.60	93.20	8468.6	242.8	-8225.80	17372.2	450.7	-16921.47

6.6. Experiments for testing the performance of GA

The problem generation scheme explained in Section 6.1 is used again to create a new set of problems only for testing purposes. The reason for using another set of problem instances is to see how the proposed method will perform with the selected set of parameters on different problem instances.

The test problems differ from the first set of problems in terms of location of the machines, processing times of operations, the operation routes of the part types and in the demanded production amounts. The values for this new set of problems are given in Tables C.1, C.2, C.3, and C.4. The sizes and difficulty levels of the new problem instances are given in Table 6.16. The new machine layouts are given in Figures C.1, C.2, and C.3.

Table 6.17 presents the detailed comparison of MIP2 and GA for this new set of problem instances. Likewise, comparison of their solution generation times is given in Table 6.18. According to these results it can be concluded that the proposed algorithm does not perform good only in problem instances, which are used for parameter fine tuning. The total time spent for generating solutions by using the GA is 3 hours 35 minutes and 5 seconds, whereas the duration for generating MIP2 solutions is 134 hours 22 minutes and 36 seconds.

Table 6.16. Problem sizes and difficulty levels of test problem instances

	Problem Size				**Difficulty**		
Instances	**Mach.**	**Tools**	**Part Types**	**Routes**	**Set A**	**Set B**	**Set C**
PI-01	4	12	14	27	2	3	4
PI-02	4	12	15	27	2	3	4
PI-03	4	12	14	27	2	3	4
PI-04	4	12	12	26	2	3	4
PI-05	4	12	15	28	2	3	4
PI-06	8	24	15	29	4	6	8
PI-07	8	24	15	30	4	6	8
PI-08	8	24	15	26	4	6	8
PI-09	8	24	15	31	4	6	8
PI-10	8	24	14	25	4	6	8
PI-11	12	36	14	26	6	9	12
PI-12	12	36	14	28	6	9	12
PI-13	12	36	12	23	6	9	12
PI-14	12	36	12	22	6	9	12
PI-15	12	36	15	27	6	9	12

Table 6.17. Comparison of results obtained by GA and MIP2 for test problem instances

	Set A			Set B			Set C		
PI	**MIP2**	**GA**	**Dev.**	**MIP2**	**GA**	**Dev.**	**MIP2**	**GA**	**Dev.**
01	262059.2	262059.2	0.00	231673.9	231673.9	0.00	227161.1	227161.1	0.00
02	256110.0	256110.0	0.00	229491.3	229491.3	0.00	219442.1	219442.1	0.00
03	260186.0	260186.0	0.00	237088.0	237088.0	0.00	231035.2	231035.2	0.00
04	262077.4	262077.4	0.00	236727.9	236727.9	0.00	225560.0	225560.0	0.00
05	249188.5	249188.5	0.00	218433.9	218433.9	0.00	218433.9	218433.9	0.00
06	351474.2	351474.2	0.00	305718.6	309782.0	-1.33	304762.1	301141.3	1.19
07	338341.5	338341.5	0.00	300758.0	307673.3	-2.30	298916.7	289988.5	2.99
08	370423.5	370423.5	0.00	311989.6	319738.1	-2.48	316707.8	306751.4	3.14
09	376326.9	376326.9	0.00	324027.0	334631.6	-3.27	316631.3	320152.0	-1.11
10	353685.1	353685.1	0.00	312913.4	314932.6	-0.65	312691.8	308841.0	1.23
11	535856.9	535856.9	0.00	520749.4	502105.2	3.58	533109.5	485112.5	9.00
12	549802.5	549802.5	0.00	502104.3	500554.5	0.31	530732.9	497572.5	6.25
13	531825.7	531825.7	0.00	478687.1	475904.9	0.58	500050.0	487225.2	2.56
14	535941.2	535941.2	0.00	493374.2	494124.3	-0.15	502104.6	490676.6	2.28
15	534679.9	534679.9	0.00	492294.3	488070.7	0.86	530841.4	491846.1	7.35
Avg.	384531.9	384531.91	0.00	346402.1	346728.8	-0.32	351212.0	340062.6	2.33

Table 6.18. Comparison of solution generation times for GA and MIP2 for test problem instances

	Set A			Set B			Set C		
PI	**MIP2**	**GA**	**Diff.**	**Best**	**GA**	**Diff.**	**Best**	**GA**	**Diff.**
01	1	37	36	7	66	59	53	112	59
02	0	37	37	10	61	51	40	114	74
03	0	38	38	6	65	59	36	111	75
04	1	37	36	12	59	47	27	111	84
05	1	42	41	13	64	51	35	117	82
06	25	114	89	9319	210	-9109	28803	373	-28430
07	15	114	99	4190	217	-3973	28802	383	-28419
08	19	118	99	7815	211	-7604	28802	382	-28420
09	11	116	105	3486	207	-3279	28803	369	-28434
10	24	119	95	20058	205	-19853	28805	375	-28430
11	749	269	-480	28806	466	-28340	30067	887	-29180
12	2984	256	-2728	28803	449	-28354	28812	847	-27965
13	40	255	215	28804	450	-28354	29124	854	-28270
14	95	258	163	28803	446	-28357	29046	857	-28189
15	545	254	-291	28808	466	-28342	29051	869	-28182
Avg.	300.7	137.60	-163.07	12596.0	242.8	-12353.20	19353.7	450.7	-18903.00

7. CONCLUSIONS AND FUTURE WORK

The aim of this thesis is to propose solution methods for allocation of tools among the machines in FMS. The objective for tool allocation problem is selected as minimizing the total distance traveled by material handling entities. The selection of this type of objective is grounded on the assumption that a great deal of the production lead times are due to the material handling activities in the system. The constraints included in the problem are the tool slot capacity, machine time limitation, tool copy availability and machine-tool compatibility constraints. Part types produced in the system are assumed to have alternative process plans and partial fulfillment of demand by using these alternative routes are allowed.

In order to solve the problem under study, three mathematical models and a GA based heuristic are presented. First of the mathematical models is a MINLP model which has a non-convex structure. Since attaining global optimal is not ensured by using the available MINLP solvers, a linearized version of this first model, namely MIP1, is formulated. Since the new formulation is a linear formulation, its LP relaxation can be quickly solved to optimality. Thus, obtaining an optimal solution by using branch and bound possible with this formulation. Linearization is done by defining additional variables and constraints, which replace the non-linearities of the original model. Thus, the number of variables and constraints in MIP1 are higher than of MINLP. Another shortcoming of these two models is the requirement of explicitly defining binary variables for assignment of each individual tool to be allocated. However, instead of using binary variables as much as the number of tools in the system, it is also possible to define binary variables for assignment of each tool type. This is can be achieved by using additional constraints for ensuring that the number of tools allocated of a certain tool type does not exceed the number of available tools for this tool type. This idea motivated the formulation of a third mathematical model, denoted as MIP2. Even though this formulation requires fewer number of binary variables, the number continuous variables is higher. Still, the numerical studies showed that the solutions obtained by this formulation within the same time period are much better than MIP1 (see Table

6.3 and 6.4).

The conducted numerical experiments show that, for some problem instances, the proposed MIP models fail to generate optimal solutions during the permitted eight hours of run time. Therefore, as an alternative solution method to solving these models through branch and bound, a GA heuristic method is proposed. The most interesting feature of this heuristic method is that it incorporates solution of mathematical models. This is done as follows; the tool allocation decisions are encoded in the chromosomes. Through evolutionary computation techniques the best combination of these allocation decisions is searched. In order to determine the quality of a certain chromosome the information contained in the chromosome is converted into the values of associated binary in the formulation of MIP models. Afterwards, by treating these binary variables as parameters and not decision variables, the resulting LP models can be solved quickly to optimality. By using these optimal objective values the fitness values of individuals are determined. Furthermore, several strategies are adopted to improve the efficiency of the proposed GA. The conducted numerical experiments show that the generated solutions by using the GA are very close to that are obtained by solving the MIP2. Moreover, in most of the instances that cannot be solved to optimality during eight hours by using MIP2, better solutions are obtained by using the proposed GA. Another important result observed during parameter fine tuning experiments is that the proposed GA exhibits a robust behavior. That is to say the performance of the algorithm does not change drastically due to small changes in its parameters.

Finally, it is worth to mention that there are numerous other issues that are not answered within the framework of this research. Some of these issues are listed as recommendations for future research directions:

- A single objective approach is adopted in this research. However, there are also other objectives that can be adopted for improving the efficiency of FMSs as mentioned in the literature survey presented in Chapter 2. Therefore, it may be a good idea to follow a "multiobjective" approach for solving the same problem. For instance, as one of the frequently encountered one in the literature, workload balancing can be aimed for improving the throughput of the system. Multi-objective GA approaches constitute an interesting and growing research area.

- By adopting the same approach for determination of the objective values, other meta-heuristics apart from GA can be proposed for solution of the problem.
- Consideration of tool life issues may be incorporated to the problem.

APPENDIX A: DATA FOR GENERATED PROBLEM INSTANCES

This chapter contains information about the generated problem instances, which are used for parameter fine tuning of the proposed GA.

Table A.1. Tools and operations

Tool Type Name	Executable Operations	Processing Time
ReceivalTool	Receival	1
ShipmentTool	Shipment	1
ToolType1	Operation1	96
	Operation2	109
ToolType2	Operation3	119
	Operation4	86
ToolType3	Operation5	94
	Operation6	73
ToolType4	Operation7	129
	Operation8	121
ToolType5	Operation9	124
	Operation10	137
ToolType6	Operation11	134
	Operation12	160
ToolType7	Operation13	120
	Operation14	120
ToolType8	Operation15	95
	Operation16	116
ToolType9	Operation17	80
	Operation18	92
ToolType10	Operation19	90
	Operation20	78
ToolType11	Operation21	111
	Operation22	83
ToolType12	Operation23	93
	Operation24	95

Table A.2. Operation routes of part types 1 to 10

Job Type	**Route Name**	**Operation Routes**					
JobType1	Route1	Rec.	Op1	Op9	Op18	Ship.	
	Route2	Rec.	Op10	Op17	Op19	Ship.	
	Route3	Rec.	Op2	Op15	Op19	Ship.	
JobType2	Route1	Rec.	Op6	Op10	Op16	Op22	Ship.
	Route2	Rec.	Op5	Op10	Op16	Op20	Ship.
JobType3	Route1	Rec.	Op5	Op8	Op15	Ship.	
JobType4	Route1	Rec.	Op4	Op7	Op13	Op22	Ship.
	Route2	Rec.	Op5	Op7	Op15	Ship.	
JobType5	Route1	Rec.	Op6	Op9	Op16	Ship.	
	Route2	Rec.	Op3	Op16	Op22	Ship.	
JobType6	Route1	Rec.	Op11	Op16	Op19	Ship.	
	Route2	Rec.	Op11	Op16	Op23	Ship.	
JobType7	Route1	Rec.	Op5	Op9	Op15	Op23	Ship.
JobType8	Route1	Rec.	Op4	Op13	Op24	Ship.	
JobType9	Route1	Rec.	Op17	Op24	Ship.		
	Route2	Rec.	Op1	Op17	Op21	Ship.	
JobType10	Route1	Rec.	Op5	Op8	Op18	Op21	Ship.

Table A.3. Operation routes of part types 11 to 20 (Continued)

Job Type	**Route Name**	**Operation Routes**					
JobType11	Route1	Rec.	Op7	Op20	Ship.		
	Route2	Rec.	Op5	Op17	Ship.		
	Route3	Rec.	Op3	Op7	Op24	Ship.	
JobType12	Route1	Rec.	Op1	Op12	Op16	Op23	Ship.
JobType13	Route1	Rec.	Op5	Op11	Op17	Op24	Ship.
	Route2	Rec.	Op5	Op22	Ship.		
JobType14	Route1	Rec.	Op1	Op7	Op15	Op21	Ship.
JobType15	Route1	Rec.	Op11	Op17	Op23	Ship.	
JobType16	Route1	Rec.	Op9	Op14	Op20	Ship.	
	Route2	Rec.	Op2	Op9	Op15	Op19	Ship.
	Route3	Rec.	Op3	Op15	Op19	Ship.	
JobType17	Route1	Rec.	Op10	Op14	Ship.		
	Route2	Rec.	Op1	Op12	Op15	Op20	Ship.
JobType18	Route1	Rec.	Op5	Op11	Op14	Op21	Ship.
JobType19	Route1	Rec.	Op2	Op11	Op15	Ship.	
JobType20	Route1	Rec.	Op6	Op16	Op19	Ship.	
	Route2	Rec.	Op3	Op8	Op16	Op23	Ship.

Table A.4. Demand values for part types

	Part Types																			
PI	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
01	0	134	99	90	0	172	56	108	154	0	159	130	0	99	165	0	143	168	83	0
02	0	98	135	0	52	0	167	0	143	76	0	68	0	185	0	187	191	124	50	189
03	83	185	88	107	181	124	75	120	0	79	0	107	120	160	0	66	58	0	64	81
04	118	0	176	0	132	56	170	119	151	163	0	75	128	70	112	141	54	0	155	0
05	0	146	76	62	164	90	127	66	0	61	0	0	125	144	0	156	102	73	170	141
06	224	235	358	0	127	0	201	235	358	220	332	0	313	272	0	332	145	224	100	0
07	130	315	416	0	290	210	244	0	0	0	324	0	341	311	261	147	315	0	362	0
08	159	135	312	336	218	152	177	0	256	288	0	343	232	159	156	190	124	0	118	124
09	0	319	246	232	191	305	223	0	0	164	159	0	177	214	182	0	437	196	173	223
10	240	0	214	288	248	329	177	0	0	129	333	296	214	251	0	170	159	99	370	0
11	149	415	487	260	287	215	420	453	154	453	0	0	0	143	0	271	509	0	492	536
12	239	0	423	0	515	497	448	577	0	331	0	393	0	589	0	0	165	0	522	423
13	460	0	248	219	467	343	241	0	0	204	0	0	680	453	460	548	212	0	416	467
14	491	0	374	327	216	579	163	0	503	526	473	163	409	321	0	204	432	0	391	0
15	0	461	266	409	253	214	461	350	629	539	526	0	181	214	0	0	642	0	0	331

Table A.5. Machines and their compatible tools for Category A

Machine Type	**Tool Capacity**	**Mountable Tool Types**
Receiving	1	ReceivalTool
Shipping	1	ShipmentTool
MachineType1	3	ToolType1 ToolType3 ToolType5 ToolType8 ToolType9 ToolType11
MachineType2	3	ToolType1 ToolType3 ToolType5 ToolType7 ToolType10 ToolType11
MachineType3	3	ToolType2 ToolType4 ToolType6 ToolType8 ToolType9 ToolType12
MachineType4	3	ToolType2 ToolType4 ToolType6 ToolType7 ToolType10 ToolType12

Table A.6. Machines and their compatible tools for Category B

Machine Type	**Tool Capacity**	**Mountable Tool Types**	
Receiving	1	ReceivalTool	
Shipping	1	ShipmentTool	
MachineType1	3	ToolType1	ToolType6
		ToolType2	ToolType7
		ToolType3	ToolType8
		ToolType4	ToolType10
		ToolType5	
MachineType2	3	ToolType2	ToolType9
		ToolType3	ToolType10
		ToolType4	ToolType11
		ToolType5	ToolType12
		ToolType8	
MachineType3	3	ToolType1	ToolType8
		ToolType2	ToolType9
		ToolType5	ToolType11
		ToolType6	ToolType12
		ToolType7	
MachineType4	3	ToolType1	ToolType9
		ToolType3	ToolType10
		ToolType4	ToolType11
		ToolType6	ToolType12
		ToolType7	

Table A.7. Machines and their compatible tools for category C

Machine Type	**Tool Capacity**	**Mountable Tool Types**
Receiving	1	ReceivalTool
Shipping	1	ShipmentTool
MachineType1	3	ToolType1 ToolType7 ToolType2 ToolType8 ToolType3 ToolType9 ToolType4 ToolType10 ToolType5 ToolType11 ToolType6 ToolType12
MachineType2	3	ToolType1 ToolType7 ToolType2 ToolType8 ToolType3 ToolType9 ToolType4 ToolType10 ToolType5 ToolType11 ToolType6 ToolType12
MachineType3	3	ToolType1 ToolType7 ToolType2 ToolType8 ToolType3 ToolType9 ToolType4 ToolType10 ToolType5 ToolType11 ToolType6 ToolType12
MachineType4	3	ToolType1 ToolType7 ToolType2 ToolType8 ToolType3 ToolType9 ToolType4 ToolType10 ToolType5 ToolType11 ToolType6 ToolType12

Table A.8. The number of variables and constraints for MINLP models(The figures are averages of the respective problem instances).

Instances	**Features**	**Difficulty**		
		Set A	**Set B**	**Set C**
PI 01-05	Binary var.	48	48	48
	Cont. var.	192	192	192
	Constraints	527	527	527
PI 06-10	Binary var.	192	192	192
	Cont. var.	807	807	807
	Constraints	5719	5719	5719
PI 11-15	Binary var.	432	432	432
	Cont. var.	1975	1975	1975
	Constraints	24698	24698	24698

Table A.9. The number of variables and constraints for MIP1 models.

Instances	**Features**	**Difficulty**		
		Set A	**Set B**	**Set C**
PI 01-05	Binary var.	48	48	48
	Cont. var.	2544	2544	2544
	Constraints	2879	2879	2879
PI 06-10	Binary var.	192	192	192
	Cont. var.	37863	37863	37863
	Constraints	42775	42775	42775
PI 11-15	Binary var.	432	432	432
	Cont. var.	189031	189031	189031
	Constraints	211754	211754	211754

Table A.10. The number of variables and constraints for MIP2 models.

Instances	**Features**	**Difficulty**		
		Set A	**Set B**	**Set C**
PI 01-05	Binary var.	48	48	48
	Cont. var.	186	627	1485
	Constraints	79	79	79
PI 06-10	Binary var.	96	96	96
	Cont. var.	1652	5573	13210
	Constraints	131	131	131
PI 11-15	Binary var.	144	144	144
	Cont. var.	5141	17351	41127
	Constraints	182	182	182

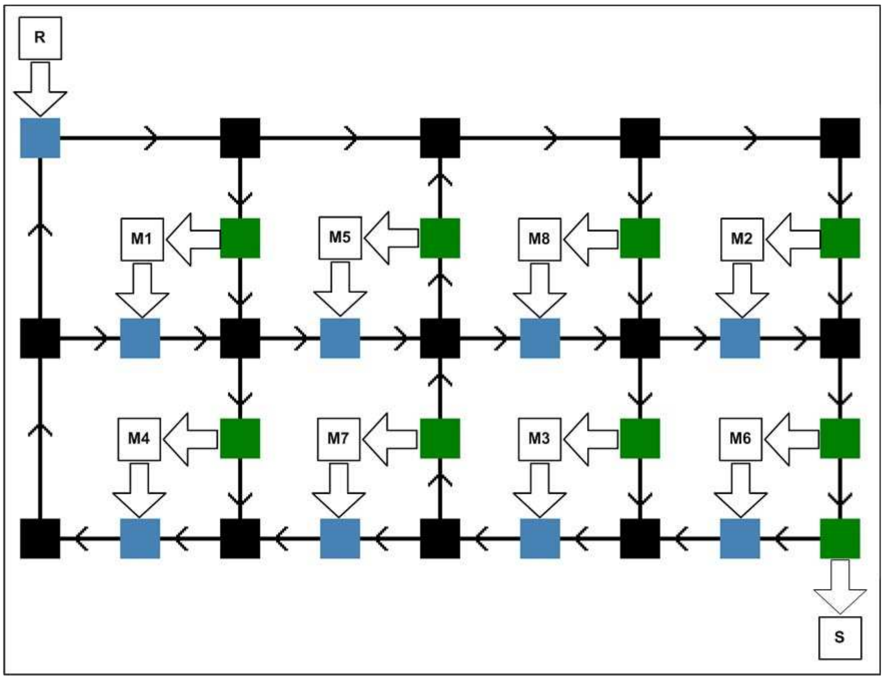

Figure A.1. Layout for the problem instances with 8 machines

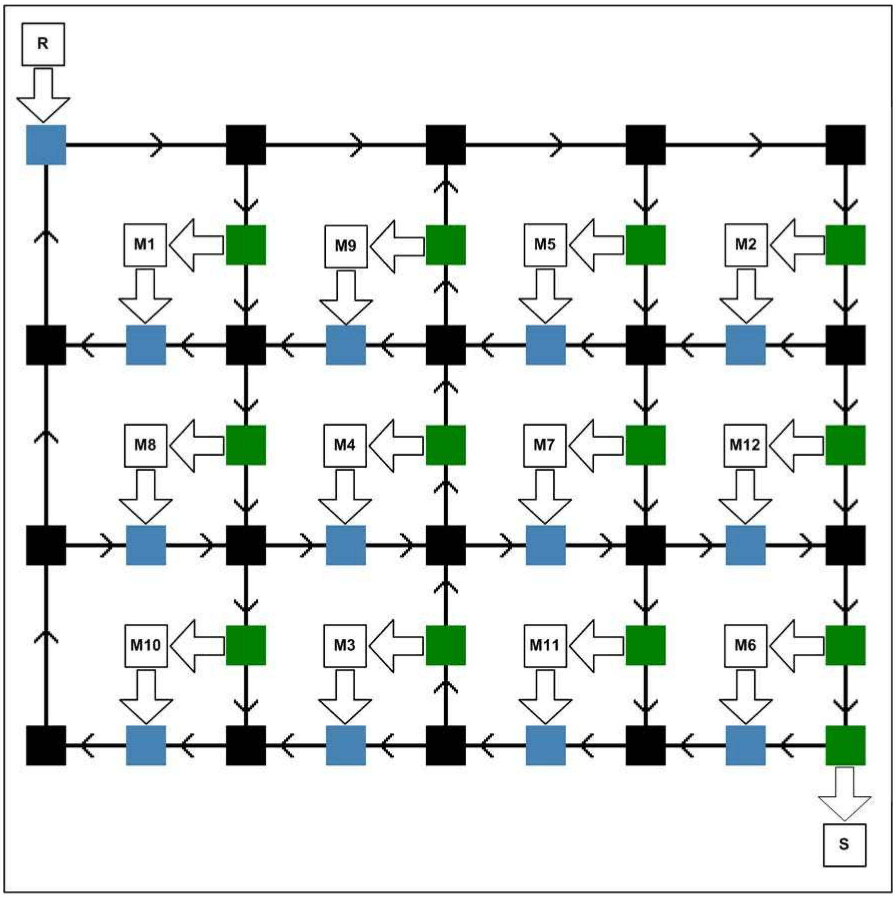

Figure A.2. Layout for the problem instances with 12 machines

APPENDIX B: NUMERICAL EXPERIMENT RESULTS

This chapter contains information about the results of the conducted numerical experiments.

Table B.1. Labeling of experimental configurations with various crossover and mutation probabilities

	Mutation			
Crossover	**0.2**	**0.3**	**0.4**	**0.5**
0.8	Exp01	Exp02	Exp03	Exp04
0.9	Exp05	Exp06	Exp07	Exp08
1.0	Exp09	Exp10	Exp11	Exp12

Table B.2. Objective values obtained in Exp01 - Exp04

	Set A				Set B				Set C			
PI	**Exp01**	**Exp02**	**Exp03**	**Exp04**	**Exp01**	**Exp02**	**Exp03**	**Exp04**	**Exp01**	**Exp02**	**Exp03**	**Exp04**
01	261157.9	261157.9	261157.9	261157.9	243330.0	243330.0	243330.0	243330.0	218546.6	218546.6	218546.6	218546.6
02	252368.6	252368.6	252368.6	252368.6	224506.5	224506.5	224506.5	224506.5	223535.8	223535.8	223535.8	224506.5
03	269077.7	269077.7	269077.7	269077.7	244793.4	244793.4	244793.4	244793.4	231924.8	231924.8	231924.8	231924.8
04	262223.7	262223.7	262223.7	262223.7	242440.0	242440.0	242440.0	242440.0	235034.6	232971.8	235034.6	232971.8
05	255770.0	255770.0	255770.0	255770.0	243431.2	243431.2	243431.2	243431.2	232911.5	232911.5	232911.5	232911.5
06	352486.7	353275.2	352486.7	352486.7	328361.3	320919.4	323520.5	322193.7	316013.4	314830.1	311116.1	313074.3
07	358700.0	358700.0	358700.0	358700.0	327362.5	326791.7	332427.0	327637.8	324452.8	321933.5	324452.8	320018.5
08	364571.1	364571.1	364571.1	364571.1	322599.6	319432.2	319432.2	325007.5	323543.4	314479.6	320419.7	314588.1
09	351665.7	351665.7	351665.7	352792.3	326599.5	321439.2	314931.8	325428.1	315369.5	315326.6	314406.0	314773.6
10	360129.0	360129.0	360129.0	360129.0	316926.9	321391.7	323711.4	323998.1	314305.7	315883.0	316798.6	316564.3
11	564755.8	565157.5	564755.8	565157.5	505132.8	508319.0	503550.6	508143.0	493067.2	491339.2	492233.4	497478.2
12	619551.2	619551.2	619551.2	619460.3	541291.7	544330.3	529469.8	543798.6	529575.7	510575.0	512633.7	524609.0
13	565000.0	564905.9	564905.9	565000.0	503874.9	498038.0	504081.3	501133.8	484566.4	494778.5	491792.0	493062.0
14	577949.6	578008.3	577415.0	576903.4	505060.0	501174.3	512776.5	492068.6	498178.4	495406.0	492166.2	489393.3
15	545887.3	545887.3	548084.1	546693.1	500007.7	495743.4	504186.8	499743.4	479028.0	496071.7	492216.1	490014.8

Table B.3. Objective values obtained in Exp05 - Exp18

	Set A				Set B				Set C			
PI	**Exp05**	**Exp06**	**Exp07**	**Exp08**	**Exp05**	**Exp06**	**Exp07**	**Exp08**	**Exp05**	**Exp06**	**Exp07**	**Exp08**
01	261157.9	261157.9	261157.9	261157.9	243330.0	243330.0	243330.0	243330.0	218546.6	218546.6	218546.6	218546.6
02	252368.6	252368.6	252368.6	252368.6	224506.5	224506.5	224506.5	224506.5	223535.8	224506.5	223535.8	223535.8
03	269077.7	269077.7	269077.7	269077.7	244793.4	244793.4	244793.4	244793.4	231924.8	231924.8	231924.8	231924.8
04	262223.7	262223.7	262223.7	262223.7	242440.0	242440.0	242440.0	242440.0	235034.6	235034.6	236790.0	235034.6
05	255770.0	255770.0	255770.0	255770.0	243431.2	243431.2	243431.2	243431.2	232911.5	232911.5	232911.5	232911.5
06	355253.6	353275.2	355253.6	353275.2	324690.7	324690.7	325681.6	319382.7	314106.8	318009.4	312296.9	312426.4
07	358700.0	358700.0	358700.0	358700.0	333365.1	332427.0	333001.9	333091.1	316056.7	317199.3	316056.7	325919.5
08	364571.1	364571.1	364571.1	364571.1	321498.1	319432.2	320508.1	325007.5	317153.8	314302.2	317281.1	315665.8
09	351665.7	351665.7	351665.7	351665.7	318184.6	321152.0	318184.6	321386.2	316109.8	320648.4	317304.9	319203.3
10	361100.3	360129.0	360129.0	360129.0	316926.9	323711.4	312826.6	312826.6	313708.1	315883.0	312826.6	320287.9
11	565157.5	564755.8	564755.8	564755.8	510550.2	502825.3	511714.7	509488.8	492588.2	494912.0	497714.6	492708.1
12	623152.9	619460.3	622126.7	619350.6	547731.3	538014.9	541749.8	539194.1	523934.2	514610.7	520219.9	527504.8
13	564905.9	564346.8	564905.9	567598.5	501026.9	499905.6	499985.7	502597.4	498146.9	485558.7	492131.9	485940.9
14	575741.5	579653.0	578008.3	580799.0	498448.0	502759.5	505189.7	507542.1	489958.2	490713.9	498346.0	497367.7
15	545887.3	545880.1	552259.5	546810.5	511179.5	509488.4	496560.7	505925.3	489957.8	484240.0	491080.9	494019.0

Table B.4. Objective values obtained in Exp09 - Exp12

	Set A				**Set B**				**Set C**			
PI	**Exp09**	**Exp10**	**Exp11**	**Exp12**	**Exp09**	**Exp10**	**Exp11**	**Exp12**	**Exp09**	**Exp10**	**Exp11**	**Exp12**
01	261157.9	261157.9	261157.9	261157.9	243330.0	243330.0	243330.0	243330.0	218546.6	218546.6	218546.6	218546.6
02	252368.6	252368.6	252368.6	252368.6	224506.5	224506.5	224506.5	224506.5	223535.8	223535.8	223535.8	223535.8
03	269077.7	269077.7	269077.7	269077.7	244793.4	244793.4	244793.4	244793.4	231924.8	231924.8	231924.8	231924.8
04	262223.7	262223.7	262223.7	262223.7	242440.0	242440.0	242440.0	242440.0	235034.6	235034.6	235034.6	232971.8
05	255770.0	255770.0	255770.0	255770.0	243431.2	243431.2	243431.2	243431.2	232911.5	232911.5	232911.5	232911.5
06	355253.6	353275.2	353275.2	355253.6	320938.3	320938.3	320938.3	322193.7	310039.9	317404.5	308568.0	311309.9
07	358700.0	358700.0	358700.0	358700.0	332427.0	331883.1	333091.1	338792.1	321631.1	321933.5	317199.3	322644.8
08	364571.1	366500.0	364571.1	364571.1	320508.1	327724.9	323215.8	322989.1	314765.7	314841.4	316147.2	314659.2
09	352792.3	351665.7	351665.7	351665.7	316189.6	320068.0	314931.8	318184.6	317791.1	315958.1	316853.2	316787.8
10	360129.0	360129.0	360129.0	360129.0	312826.6	327973.6	321391.7	325095.3	313708.1	315883.0	319775.9	315883.0
11	564755.8	565157.5	564755.8	564755.8	512358.3	502125.8	502431.6	511084.9	488141.1	491378.5	503626.8	490397.1
12	622002.6	619551.2	619588.7	619350.6	541773.0	538762.1	533595.5	534048.7	510055.8	528994.5	519824.7	527380.9
13	565000.0	566650.2	564905.9	565000.0	504536.7	504356.7	503576.7	496906.2	498949.3	493211.9	495666.6	494806.7
14	575741.5	578008.3	575741.5	574937.0	497922.7	506118.8	507467.5	512563.7	496953.2	494010.5	496520.9	500024.1
15	545887.3	546744.6	546693.1	545887.3	510562.0	508749.7	502488.1	512313.8	495204.5	497818.9	489103.1	489476.7

Table B.5. Objective values obtained in Exp13 - Exp16

	Set A				Set B				Set C			
PI	**Exp13**	**Exp14**	**Exp15**	**Exp16**	**Exp13**	**Exp14**	**Exp15**	**Exp16**	**Exp13**	**Exp14**	**Exp15**	**Exp16**
01	261157.9	261157.9	261157.9	261157.9	243330.0	243330.0	243330.0	243330.0	218546.6	218546.6	218546.6	218546.6
02	252368.6	252368.6	252368.6	252368.6	224506.5	224506.5	224506.5	224506.5	223535.8	223535.8	225509.6	223535.8
03	269077.7	269077.7	269077.7	269077.7	244793.4	244793.4	244793.4	244793.4	231924.8	231924.8	231924.8	231924.8
04	262223.7	262223.7	262223.7	262223.7	242440.0	242440.0	242440.0	242440.0	235034.6	232971.8	232971.8	235034.6
05	255770.0	255770.0	255770.0	255770.0	243431.2	243431.2	243431.2	243431.2	232911.5	232911.5	232911.5	232911.5
06	352486.7	353275.2	352486.7	352486.7	322402.6	322193.7	320938.3	319382.7	311116.1	310933.6	312658.9	312658.9
07	358700.0	358700.0	358700.0	358700.0	327362.5	329945.5	326791.7	329945.5	321640.6	322678.0	323473.6	324018.5
08	364571.1	364571.1	364571.1	364571.1	322730.6	325007.5	325681.3	319432.2	321052.0	317760.7	314302.2	316048.8
09	351665.7	351665.7	351665.7	351665.7	323457.8	323457.8	319816.9	326643.8	314406.0	316237.0	314668.3	319299.9
10	360129.0	360129.0	360129.0	360129.0	312826.6	323998.1	321391.7	323998.1	316850.0	320942.4	313708.1	315883.0
11	564755.8	564755.8	565157.5	564755.8	507898.4	510938.7	507540.1	503252.9	486689.2	481073.1	502471.4	490077.9
12	619460.3	619350.6	619551.2	619551.2	541748.7	531633.2	537479.6	532510.3	522563.4	504735.0	521608.5	511244.9
13	565000.0	565000.0	564905.9	566338.9	495500.8	500588.1	511643.1	506374.7	487241.9	487221.3	488101.9	481622.2
14	574937.0	578008.3	580664.4	576903.4	491230.8	498625.1	497676.5	509755.5	498475.1	492938.2	497053.7	493784.0
15	545887.3	545880.1	547095.8	545880.1	499133.4	495743.4	509340.0	497568.7	495388.9	497145.8	492250.5	486629.8

Table B.6. Objective values obtained in Exp17 - Exp20

PI	Set A				Set B				Set C			
	Exp17	**Exp18**	**Exp19**	**Exp20**	**Exp17**	**Exp18**	**Exp19**	**Exp20**	**Exp17**	**Exp18**	**Exp19**	**Exp20**
01	261157.9	261157.9	261157.9	261157.9	243330.0	243330.0	243330.0	243330.0	218546.6	218546.6	218546.6	218546.6
02	252368.6	252368.6	252368.6	252368.6	224506.5	224506.5	224506.5	224506.5	223535.8	223535.8	223535.8	223535.8
03	269077.7	269077.7	269077.7	269077.7	244793.4	244793.4	244793.4	244793.4	231924.8	231924.8	231924.8	231924.8
04	262223.7	262223.7	262223.7	262223.7	242440.0	242440.0	242440.0	242440.0	235034.6	235034.6	232971.8	232971.8
05	255770.0	255770.0	255770.0	255770.0	243431.2	243431.2	243431.2	243431.2	232911.5	232911.5	232911.5	232911.5
06	352486.7	353275.2	352486.7	352486.7	319382.7	321769.6	320938.3	322975.3	308975.8	311800.9	312504.7	313491.2
07	358700.0	358700.0	358700.0	358700.0	326791.7	327362.5	327362.5	331570.5	321933.5	321321.0	316056.7	321933.5
08	364571.1	364571.1	364571.1	364571.1	322053.2	321498.1	319432.2	321298.5	314765.7	314302.2	319701.8	314302.2
09	351665.7	352792.3	351665.7	351665.7	321344.0	314931.8	318454.0	320895.3	315277.1	315711.4	318175.5	314406.0
10	360129.0	360129.0	360129.0	360129.0	321391.7	321391.7	321607.9	323711.4	325655.0	319650.9	316553.1	322282.9
11	565157.5	565157.5	564755.8	565157.5	512833.4	513535.7	508224.7	507820.5	490453.2	494614.3	491849.1	492230.8
12	619551.2	619551.2	619350.6	619551.2	530779.3	546768.2	547294.2	547368.1	523036.3	514196.7	517299.0	512422.5
13	567981.7	569669.2	566650.2	565000.0	492433.7	501332.5	502473.1	495689.7	490326.3	490355.2	497780.8	494596.9
14	574937.0	578008.3	574937.0	576903.4	492599.3	491321.4	495958.6	498135.5	490315.5	497144.9	491417.2	492524.9
15	546693.1	546744.6	545887.3	545880.1	496560.7	502383.7	505523.3	495852.6	489726.6	493776.1	493746.2	494681.8

Table B.7. Objective values obtained in Exp21 - Exp24

	Set A				Set B				Set C			
PI	**Exp21**	**Exp22**	**Exp23**	**Exp24**	**Exp21**	**Exp22**	**Exp23**	**Exp24**	**Exp21**	**Exp22**	**Exp23**	**Exp24**
01	261157.9	261157.9	261157.9	261157.9	243330.0	243330.0	243330.0	243330.0	218546.6	218546.6	218546.6	218546.6
02	252368.6	252368.6	252368.6	252368.6	224506.5	224506.5	224506.5	224506.5	223535.8	223535.8	223535.8	223535.8
03	269077.7	269077.7	269077.7	269077.7	244793.4	244793.4	244793.4	244793.4	231924.8	231924.8	231924.8	231924.8
04	262223.7	262223.7	262223.7	262223.7	242440.0	242440.0	242440.0	242440.0	235034.6	235034.6	232971.8	232971.8
05	255770.0	255770.0	255770.0	255770.0	243431.2	243431.2	243431.2	243431.2	232911.5	232911.5	232911.5	232911.5
06	352486.7	352486.7	353275.2	352486.7	325862.1	320938.3	323520.5	319382.7	311510.2	311800.9	317319.6	308568.0
07	358700.0	358700.0	358700.0	358700.0	332837.0	326791.7	332837.0	332427.0	321933.5	316056.7	316056.7	322114.1
08	364571.1	364571.1	364571.1	364571.1	321498.1	319432.2	321498.1	324125.8	317614.4	321450.5	317850.9	320113.7
09	351665.7	351665.7	351665.7	351665.7	317301.8	319739.5	314931.8	321344.0	314406.0	318847.3	315958.1	315277.1
10	360129.0	360129.0	360129.0	360129.0	326640.5	321391.7	323998.1	319126.5	323406.3	318087.3	313708.1	326278.2
11	565424.8	564755.8	564755.8	565157.5	510772.6	513278.0	518615.3	501799.1	492018.7	496162.2	490123.8	495940.0
12	619350.6	619350.6	619350.6	619350.6	531146.6	537965.0	537811.4	532785.9	512048.0	510739.5	508778.5	519938.7
13	569884.2	565000.0	565000.0	569884.2	493190.7	504541.2	510296.3	501695.3	492562.2	490994.5	490750.7	493582.0
14	575741.5	576903.4	574937.0	574937.0	495283.1	498663.7	505229.1	498017.0	488169.5	500805.4	497102.7	496295.7
15	546810.5	546744.6	546744.6	546744.6	499137.6	506972.6	500091.6	509483.6	491117.8	493764.9	487110.2	488578.9

Table B.8. Objective values obtained in Exp25 - Exp27 and Exp16

	Set A				Set B				Set C			
PI	**Exp25**	**Exp16**	**Exp26**	**Exp27**	**Exp25**	**Exp16**	**Exp26**	**Exp27**	**Exp25**	**Exp16**	**Exp26**	**Exp27**
01	261157.9	261157.9	261157.9	261157.9	243330.0	243330.0	243330.0	243330.0	218546.6	218546.6	218546.6	218546.6
02	252368.6	252368.6	252368.6	252368.6	224506.5	224506.5	224506.5	224506.5	223535.8	223535.8	223535.8	223535.8
03	269077.7	269077.7	269077.7	269077.7	244793.4	244793.4	244793.4	244793.4	231924.8	231924.8	231924.8	231924.8
04	262223.7	262223.7	262223.7	262223.7	242440.0	242440.0	242440.0	242440.0	232971.8	235034.6	235034.6	235034.6
05	255770.0	255770.0	255770.0	255770.0	243431.2	243431.2	243431.2	243431.2	232911.5	232911.5	232911.5	232911.5
06	355253.6	352486.7	353275.2	353275.2	322193.7	319382.7	324790.2	325681.6	315313.2	308975.8	313045.4	308356.6
07	358700.0	358700.0	358700.0	358700.0	333211.6	326791.7	327362.5	332656.3	327544.5	321933.5	321746.3	316056.7
08	364571.1	364571.1	364571.1	364571.1	321498.1	322053.2	321498.1	322989.1	318305.5	314765.7	316064.8	314765.7
09	351665.7	351665.7	351665.7	351665.7	320456.9	321344.0	314931.8	323722.0	314805.1	315277.1	319364.8	314931.8
10	360129.0	360129.0	360129.0	360129.0	333052.3	321391.7	323998.1	327353.1	314893.9	325655.0	316798.6	319065.7
11	564755.8	565157.5	564755.8	565157.5	509973.7	512833.4	501722.3	509235.3	491511.9	490453.2	482854.9	490298.7
12	619350.6	619551.2	619350.6	619588.7	540552.3	530779.3	542797.5	532834.2	516927.0	523036.3	508914.0	511279.1
13	565000.0	567981.7	564905.9	570818.2	502420.4	492433.7	501850.3	497409.7	483272.4	490326.3	487100.8	484740.6
14	578008.3	574937.0	579653.0	577415.0	504737.1	492599.3	503168.4	494477.5	486548.0	490315.5	492893.2	489201.1
15	546810.5	546693.1	546693.1	548084.1	503741.3	496560.7	498214.7	501179.9	495087.9	489726.6	496851.3	490906.6

Table B.9. Objective values obtained in Exp28 and Exp16

	Set A		**Set B**		**Set C**	
PI	**Exp28**	**Exp16**	**Exp28**	**Exp16**	**Exp28**	**Exp16**
01	261157.9	261157.9	243330.0	243330.0	218546.6	218546.6
02	252368.6	252368.6	224506.5	224506.5	223535.8	223535.8
03	269077.7	269077.7	244793.4	244793.4	231924.8	231924.8
04	262223.7	262223.7	242440.0	242440.0	235034.6	235034.6
05	255770.0	255770.0	243431.2	243431.2	232911.5	232911.5
06	355253.6	352486.7	320938.3	319382.7	312869.0	308975.8
07	358700.0	358700.0	333091.1	326791.7	321649.9	321933.5
08	366500.0	364571.1	322599.6	322053.2	314659.2	314765.7
09	351665.7	351665.7	317301.8	321344.0	316784.3	315277.1
10	360129.0	360129.0	328515.6	321391.7	316709.6	325655.0
11	564755.8	565157.5	507246.0	512833.4	499368.0	490453.2
12	621310.0	619551.2	531633.2	530779.3	520928.5	523036.3
13	564905.9	567981.7	509214.1	492433.7	497574.4	490326.3
14	575741.5	574937.0	504993.7	492599.3	496236.9	490315.5
15	546693.1	546693.1	511538.5	496560.7	487828.9	489726.6

Table B.10. Objective values obtained in Exp29 and Exp16

	Set A		**Set B**		**Set C**	
PI	**Exp29**	**Exp16**	**Exp29**	**Exp16**	**Exp29**	**Exp16**
01	261157.89	261157.89	243330.0	243330.0	218546.6	218546.6
02	252368.56	252368.56	224506.5	224506.5	223535.8	223535.8
03	269077.74	269077.74	244793.4	244793.4	231924.8	231924.8
04	262223.72	262223.72	242440.0	242440.0	232971.8	235034.6
05	255770.00	255770.00	243431.2	243431.2	232911.5	232911.5
06	353275.22	352486.69	322975.3	319382.7	308975.8	308975.8
07	358700.05	358700.05	333500.5	326791.7	322630.3	321933.5
08	364571.15	364571.15	321498.1	322053.2	319088.2	314765.7
09	351665.75	351665.75	319739.5	321344.0	312051.5	315277.1
10	360129.02	360129.02	320777.0	321391.7	323526.5	325655.0
11	564755.80	565157.48	505575.5	512833.4	488292.2	490453.2
12	619350.59	619551.19	537479.6	530779.3	506822.5	523036.3
13	565000.05	567981.72	500293.5	492433.7	489481.0	490326.3
14	574937.00	574937.00	499033.5	492599.3	491654.5	490315.5
15	545880.08	546693.11	495852.6	496560.7	488150.3	489726.6

APPENDIX C: DATA FOR TEST PROBLEMS

This chapter contains information about problem instances, which are used for performance testing of the proposed GA.

Table C.1. Tools and operations

Tool Type Name	**Executable Operations**	**Processing Time**
ReceivalTool	Receival	1
ShipmentTool	Shipment	1
ToolType1	Operation1	106
	Operation2	94
ToolType2	Operation3	95
	Operation4	70
ToolType3	Operation5	68
	Operation6	68
ToolType4	Operation7	136
	Operation8	142
ToolType5	Operation9	159
	Operation10	125
ToolType6	Operation11	131
	Operation12	136
ToolType7	Operation13	96
	Operation14	82
ToolType8	Operation15	115
	Operation16	84
ToolType9	Operation17	112
	Operation18	94
ToolType10	Operation19	61
	Operation20	99
ToolType11	Operation21	85
	Operation22	99
ToolType12	Operation23	111
	Operation24	112

Table C.2. Operation routes of part types 1 to 10

Job Type	**Route Name**	**Operation Routes**					
JobType1	Route1	Rec.	Op2	Op12	Op16	Ship.	
	Route2	Rec.	Op7	Op15	Ship.		
	Route3	Rec.	Op6	Op9	Op14	Op21	Ship.
JobType2	Route1	Rec.	Op4	Op7	Op18	Op22	Ship.
	Route2	Rec.	Op7	Op14	Op24	Ship.	
JobType3	Route1	Rec.	Op7	Op17	Op19	Ship.	
	Route2	Rec.	Op3	Op17	Op20	Ship.	
JobType4	Route1	Rec.	Op1	Op11	Op22	Ship.	
	Route2	Rec.	Op3	Op12	Op17	Ship.	
JobType5	Route1	Rec.	Op5	Op11	Op19	Ship.	
JobType6	Route1	Rec.	Op2	Op17	Op21	Ship.	
	Route2	Rec.	Op17	Ship.			
JobType7	Route1	Rec.	Op2	Op12	Op17	Op22	Ship.
	Route2	Rec.	Op1	Op7	Op14	Op19	Ship.
JobType8	Route1	Rec.	Op1	Op14	Op20	Ship.	
	Route2	Rec.	Op7	Op16	Ship.		
	Route3	Rec.	Op2	Op17	Op21	Ship.	
JobType9	Route1	Rec.	Op1	Op16	Op20	Ship.	
JobType10	Route1	Rec.	Op5	Op10	Op18	Op21	Ship.
	Route2	Rec.	Op3	Op18	Ship.		

Table C.3. Operation routes of part types 11 to 20 (Continued)

Job Type	**Route Name**	**Operation Routes**					
JobType11	Route1	Rec.	Op6	Op9	Op15	Op22	Ship.
	Route2	Rec.	Op3	Op10	Op16	Op21	Ship.
JobType12	Route1	Rec.	Op4	Op9	Op16	Op22	Ship.
	Route2	Rec.	Op3	Op10	Op18	Op20	Ship.
	Route3	Rec.	Op3	Op10	Op13	Op20	Ship.
JobType13	Route1	Rec.	Op1	Op16	Op21	Ship.	
JobType14	Route1	Rec.	Op3	Op11	Op16	Op24	Ship.
JobType15	Route1	Rec.	Op2	Op12	Op14	Op22	Ship.
	Route2	Rec.	Op8	Op18	Op24	Ship.	
JobType16	Route1	Rec.	Op9	Op18	Op24	Ship.	
	Route2	Rec.	Op6	Op9	Op16	Ship.	
JobType17	Route1	Rec.	Op4	Op12	Op17	Ship.	
	Route2	Rec.	Op4	Op7	Ship.		
	Route3	Rec.	Op4	Op16	Op23	Ship.	
JobType18	Route1	Rec.	Op6	Op12	Op13	Ship.	
	Route2	Rec.	Op2	Op11	Op18	Op21	Ship.
JobType19	Route1	Rec.	Op6	Op7	Op17	Op24	Ship.
	Route2	Rec.	Op1	Op12	Op18	Op21	Ship.
JobType20	Route1	Rec.	Op1	Op9	Op17	Op24	Ship.

Table C.4. Demand values for part types

	Part Types																			
PI	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
01	30	28	35	77	32	82	42	0	0	0	46	0	52	35	0	65	83	56	72	0
02	93	44	25	0	54	45	32	45	30	0	78	0	81	37	76	0	85	0	38	71
03	95	28	33	67	76	88	27	0	99	0	0	99	59	44	0	80	41	0	51	0
04	79	0	82	0	96	27	29	32	0	0	0	92	0	62	91	33	74	0	36	0
05	48	52	34	0	40	41	97	0	0	0	61	50	43	94	94	63	57	0	85	87
06	0	29	50	0	47	60	84	32	0	89	98	46	0	50	76	65	61	0	84	91
07	57	51	52	0	47	46	60	26	0	46	0	66	0	83	0	98	99	99	46	26
08	0	0	65	0	83	35	83	0	35	0	79	35	82	54	38	60	95	84	99	92
09	56	0	29	57	57	38	57	45	55	41	39	77	77	0	0	48	65	0	28	0
10	38	35	0	0	52	40	96	0	36	0	0	64	70	70	62	83	91	0	56	69
11	0	0	49	0	56	0	100	50	0	85	0	92	66	65	29	61	70	31	31	33
12	67	31	98	0	32	31	67	33	0	0	75	0	60	32	96	47	90	0	100	0
13	54	26	80	0	0	58	91	0	73	0	0	0	0	0	34	32	60	86	42	73
14	0	54	93	55	83	0	44	0	0	82	0	96	51	0	0	38	67	0	90	43
15	90	0	36	59	65	34	58	0	90	100	0	97	78	83	0	0	100	67	35	79

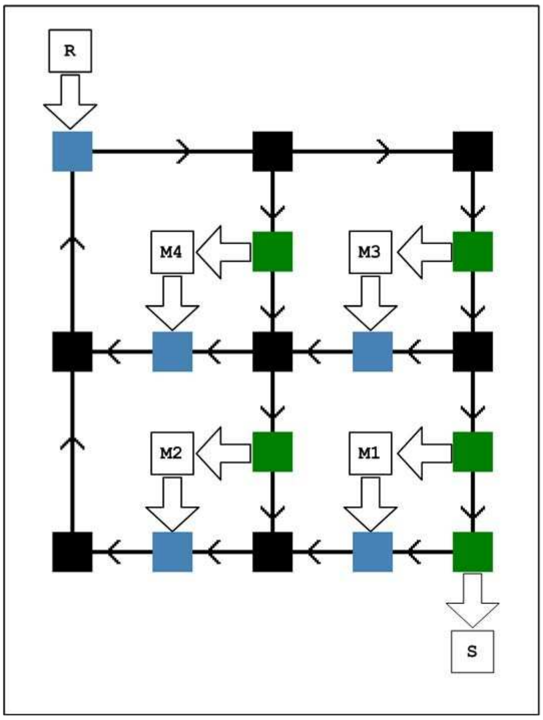

Figure C.1. Layout for the test instances with 4 machines

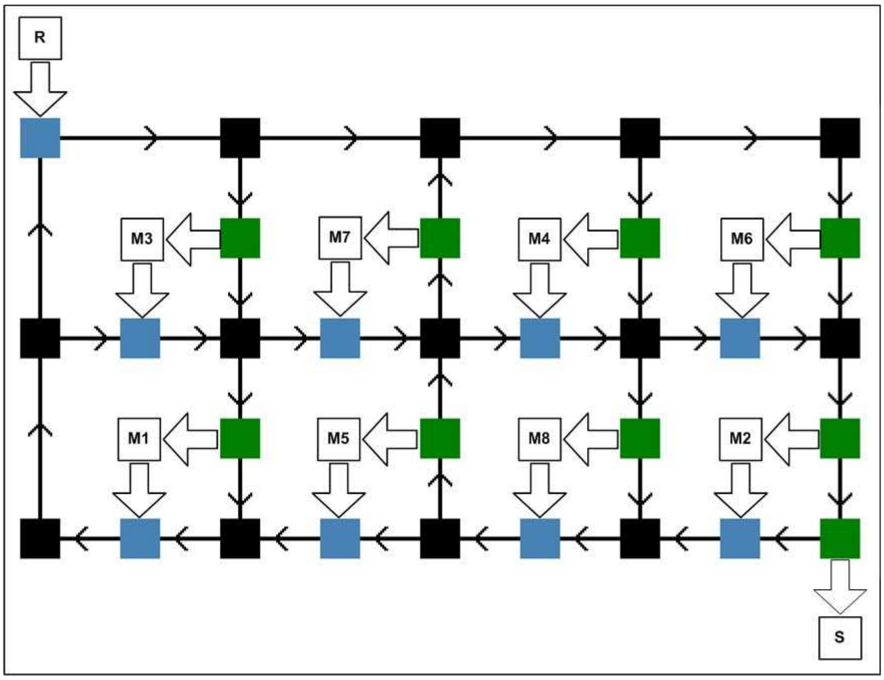

Figure C.2. Layout for the test instances with 8 machines

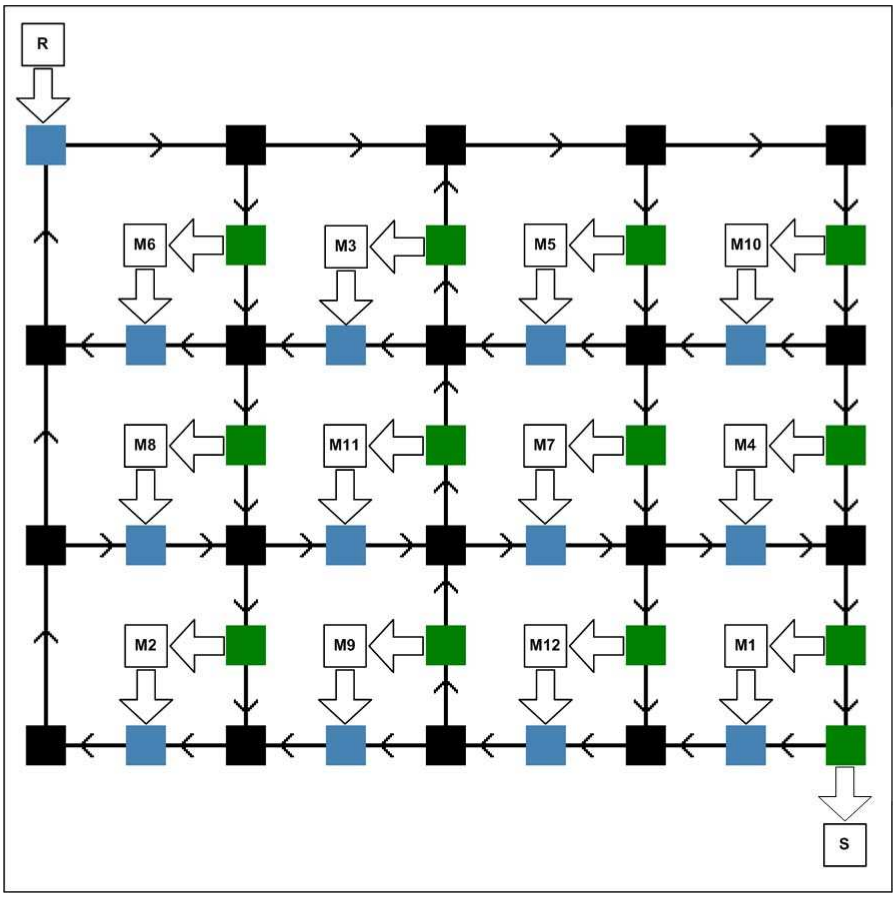

Figure C.3. Layout for the test instances with 12 machines

REFERENCES

1. Tompkins, J. A., J. A. White, Y. A. Bozer, E. H. Frazelle, J. M. A. Tanchoco, and J. Trevino, *Facilities Planning*, Wiley, New York, 1996.

2. Heragu, S. S., *Facilities Design*, Iuniverse Inc, 2006.

3. Benjaafar, S., S. S. Heragu, and S. A. Irani, "Next Generation Factory Layouts: Research Challenges and Recent Progress", *Interfaces*, Vol. 32, No. 6, pp. 58–76, 2002.

4. Yang, T. and B. A. Peters, "Flexible machine layout design for dynamic and uncertain production environments", *European Journal of Operational Research*, Vol. 108, No. 1, pp. 49–64, 1998.

5. Kochhar, J. S. and S. S. Heragu, "Facility layout design in a changing environment", *International Journal of Production Research*, Vol. 37, No. 11, pp. 2429–2446, 1999.

6. Meng, G., S. S. Heragu, and H. Zijm, "Reconfigurable layout problem", *International Journal of Production Research*, Vol. 42, No. 22, pp. 4709–4729, 2004.

7. Benjaafar, S. and M. Sheikhzadeh, "Design of flexible plant layouts", *IIE Transactions*, Vol. 32, No. 4, pp. 309–322, 2000.

8. Balakrishnan, J. and C. H. Cheng, "Dynamic layout algorithms: a state-of-the-art survey", *Omega*, Vol. 26, No. 4, pp. 507–521, 1998.

9. Stecke, K. E. and J. J. Solberg, "Loading and control policies for a flexible manufacturing system", *International Journal of Production Research*, Vol. 19, No. 5, pp. 481–490, 1981.

10. Guerrero, F., "Machine loading and part type selection in flexible manufacturing

systems", *International Journal of Production Research*, Vol. 37, No. 6, pp. 1303–1317, 1999.

11. Denizel, M. and S. S. Erenguc, "Exact solution procedures for certain planning problems in flexible manufacturing systems", *Computers and Operations Research*, Vol. 24, No. 11, pp. 1043–1055, 1997.

12. Shanker, K. and A. K. Agrawal, "Loading problem and resource considerations in FMS: a review", *International Journal of Production Economics*, Vol. 25, pp. 111–119, 1991.

13. Grieco, A., Q. Semeraro, and T. Tolio, "A Review of Different Approaches to the FMS Loading Problem", *International Journal of Flexible Manufacturing Systems*, Vol. 13, No. 4, pp. 361–384, 2001.

14. Stecke, K. E., "Formulation and Solution of Nonlinear Integer Production Planning Problems for Flexible Manufacturing Systems", *Management Science*, pp. 273–288, 1983.

15. Browne, J., D. Dubois, K. Rathmill, S. P. Sethi, and K. E. Stecke, "Classification of flexible manufacturing systems", *The FMS Magazine*, Vol. 2, No. 2, pp. 114–117, 1984.

16. Sethi, A. K. and S. P. Sethi, "Flexibility in manufacturing: A survey", *International Journal of Flexible Manufacturing Systems*, Vol. 2, No. 4, pp. 289–328, 1990.

17. Vokurka, R. J. and S. W. O'Leary-Kelly, "A review of empirical research on manufacturing flexibility", *Journal of Operations Management*, Vol. 18, No. 4, pp. 485–501, 2000.

18. Albey, E., *A hierarchical approach for responsive production planning and control in flexible manufacturing systems*, Master's thesis, Bogazici University, 2006.

19. Shanker, K. and A. Srinivasulu, "Some solution methodologies for loading problems

in a flexible manufacturing system", *International Journal of Production Research*, Vol. 27, No. 6, pp. 1019–1034, 1989.

20. Stecke, K. E., "A Hierarchical approach to solving machine grouping and loading problems of flexible manufacturing systems", *European Journal of Operational Research*, 1986.

21. Kumar, A., M. K. Tiwari, R. Shankar, and A. Baveja, "Solving machine-loading problem of a flexible manufacturing system with constraint-based genetic algorithm", *European Journal of Operational Research*, Vol. 175, No. 2, pp. 1043–1069, 2006.

22. Lashkari, R. S., S. P. Dutta, and A. M. Padhye, "A new formulation of operation allocation problem in flexible manufacturing systems: mathematical modelling and computational experience", *International Journal of Production Research*, Vol. 25, No. 9, pp. 1267–1283, 1987.

23. Wilson, J. M., "An alternative formulation of the operation-allocation problem in flexible manufacturing systems", *International Journal of Production Research*, Vol. 27, No. 8, pp. 1405–1412, 1989.

24. Chen, Y. J. and R. G. Askin, "A multiobjective evaluation of flexible manufacturing system loading heuristics", *International Journal of Production Research*, Vol. 28, No. 5, pp. 895–911, 1990.

25. Kumar, P., N. K. Tewari, and N. Singh, "Joint consideration of grouping and loading problems in a flexible manufacturing system", *International Journal of Production Research*, Vol. 28, No. 7, pp. 1345–1356, 1990.

26. Sodhi, M. S., R. G. Asking, and S. Sen, "Multiperiod tool and production assignment in flexible manufacturing systems", *International Journal of Production Research*, Vol. 32, No. 6, pp. 1281–1294, 1994.

27. Kumar, N. and K. Shanker, "A genetic algorithm for FMS part type selection and machine loading", *International Journal of Production Research*, Vol. 38, No. 16, pp. 3861–3887, 2000.

28. Leung, L. C., S. K. Maheshwari, and W. A. Miller, "Concurrent part assignment and tool allocation in FMS with material handling considerations", *International Journal of Production Research*, Vol. 31, No. 1, pp. 117–138, 1993.

29. Tiwari, M. K. and N. K. Vidyarthi, "Solving machine loading problems in a flexible manufacturing system using a genetic algorithm based heuristic approach", *International Journal of Production Research*, Vol. 38, No. 14, pp. 3357–3384, 2000.

30. Goswami, M. and M. K. Tiwari, "A reallocation-based heuristic to solve a machine loading problem with material handling constraint in a flexible manufacturing system", *International Journal of Production Research*, Vol. 44, No. 3, pp. 569–588, 2006.

31. Yogeswaran, M., "An efficient hybrid evolutionary heuristic using genetic algorithm and simulated annealing algorithm to solve machine loading problem in FMS", *International Journal of Production Research*, Vol. 99999, No. 1, pp. 1–28, 2008.

32. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

33. Bäck, T., D. B. Fogel, and Z. Michalewicz, *Evolutionary Computation 1*, Institute of Physics Publishing, 2000.

34. Bäck, T., *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, USA, 1996.

35. Jong, K. A. D., *Evolutionary computation : a unified approach*, MIT Press, Cambridge, Mass. : MIT Press, 2006.

36. Ashlock, D., *Evolutionary Computation for Modeling and Optimization*, Springer-Verlag New York Inc, 2006.

37. Kreinovich, V., C. Quintana, and O. Fuentes, "Genetic algorithms: What fitness scaling is optimal?", *Cybernetics and Systems*, Vol. 24, No. 1, pp. 9–26, 1993.

38. Neri, F., J. Toivanen, and R. A. E. Mkinen, "An adaptive evolutionary algorithm with intelligent mutation local searchers for designing multidrug therapies for HIV", *Applied Intelligence*, Vol. 27, No. 3, pp. 219–235, 2007.

39. ILOG, S., *CPLEX: ILOG CPLEX 11.0 User's Manual and Reference Manual*, 2007.