THE OPTIMUM COMBINATION OF LOCAL SEARCHES FOR GENETIC OPERATORS IN MEMETIC ALGORITHM FOR THE SPACE ALLOCATION PROBLEM

SIVARENUKA DEVI A/P CHANDRAKAISAN

UNIVERSITI SAINS MALAYSIA

MAY 2008

THE OPTIMUM COMBINATION OF LOCAL SEARCHES FOR GENETIC OPERATORS IN MEMETIC ALGORITHM FOR THE SPACE ALLOCATION PROBLEM

by

SIVARENUKA DEVI A/P CHANDRAKAISAN

**Thesis submitted in fulfillment of the requirements
for the degree of
Master of Science**

MAY 2008

ACKNOWLEDGEMENTS

“To learn is to change. Education is a process that changes the learner” – Anonymous.

Thank you God, for the strength and courage to complete this challenging journey. Thanks for holding and walking with me through the difficult times and lifting me up when I needed it the most.

To my husband and my soul mate, Ganesh – thanks for your support and encouragement. You have always been there for me, with unconditional love and continuous motivation to ensure that I never give up. If it wasn’t for you, this would not have happened. I cannot thank you enough.

To my dearest parents, Mr. & Mrs. Chandrakaisan – Kausalya Devi, thank you for your support, prayer and blessings. I am indeed very grateful to both of you.

Special thanks to my supervisor, Associate Professor Dr.Ahmad Tajudin bin Khader, for his guidance. You had led me through, in this entire journey. Your guidance is very much appreciated.

Thanks to Universiti Sains Malaysia for the sample data which was very valuable for the experiments. I would also like to extend my appreciation to the committee of EURO WATT (Working Group on Automated Timetabling) and ASAP (Automated Scheduling, Optimization and Planning) research group of University of Nottingham for the exposure in the latest research information in automated timetabling and scheduling methods and issues.

My appreciation to Farizah and all my friends who shared loads of valuable information that helped me in my research studies. Last but not least, thanks to each and everyone of you who had continuously asked “How is your thesis going on?” - You have no idea how that had encouraged me and kept me going.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

LIST OF PUBLICATIONS & SEMINARS

1 Sivarenuka Devi Chandrakaisan (2004). *Approaches to Automate Space Allocation in Academic Institution* Computer Science Postgraduate Colloquium 25-26th May 2004 Universiti Sains Malaysia, Penang, Malaysia.

2 Sivarenuka Devi Chandrakaisan, Ahmad Tajudin Khader (2008). The Optimum Combination of Local Searches for Genetic Operators in Memetic Algorithm for the Space Allocation Problem. In *Journal of Memetic Computing, Springer, September 2008* (submitted).

3 Sivarenuka Devi Chandrakaisan, Ahmad Tajudin Khader (2008). *The Decomposition Concept in Handling the Space Allocation Problem*. 3rd International Symposium on Information Technology 2008 (ITSim08) (submitted).

4 Sivarenuka Devi Chandrakaisan, Ahmad Tajudin Khader (2008). The Optimum Genetic Operators' Selection Method in Genetic Algorithm for the Space Allocation Problem. In *International Journal of Granular Computing, Rough Sets and Intelligent Systems (IJGCRSIS), Inderscience Publishers* (submitted).

PENGABUNGAN GELITARAN TEMPATAN YANG OPTIMUM UNTUK OPERATOR GENETIK DALAM ALGORITMA *MEMETIC* UNTUK MENANGANI MASALAH PENGAGIHAN RUANG

ABSTRAK

Dalam tesis ini, kami membuat penyelidikan mengenai pengagihan ruang di universiti. Kajian ini memfokus kepada pengagihan ruang dalam penyediaan jadual waktu. Pengagihan ruang perlulah mematuhi kesemua syarat yang wajib and keperluan lain sebanyak mungkin serta memastikan ruang yang sedia ada digunakan secara optimum.

Untuk memudahkan masalah yang rumit ini, kami menggunakan kaedah penguraian dengan memberi penumpuan terhadap konsep *tuple* dan *bin.* Dalam kajian ini, kami mendapati bahawa Algoritma *Memetic* memberi keputusan yang lebih baik berbanding dengan Algoritma Genetik dalam mengatasi masalah pengagihan ruang. Kami menggabungkan tiga gelitaran tempatan secara optimum dalam Algoritma *Memetic*. Gelitaran tempatan tersebut ialah *Hill Climbing*, Simulasi Sepuhan Lindap dan Gelitaran Tabu. Cara ini merupakan pendekatan yang baharu kerana sebelum ini hanya satu gelitaran tempatan digunakan dalam Algoritma *Memetic*. Kajian ini turut menyelidik tentang keberkesanan penggunaan kaedah pemilihan bagi memilih individu yang sesuai untuk tujuan pembiakan dan genetik lintas silang. Kaedah pemilihan yang disiasat adalah pemilihan rawak, berketentuan dan roda *roulette*. Kaedah pemilihan yang pelbagai untuk memilih individu dalam Algoritma *Memetic* juga merupakan pendekatan yang berbeza. Dalam kajian sebelum ini, hanya satu kaedah pemilihan digunakan. Kajian mutasi kami dalam Algoritma *Memetic* menunjukkan bahawa pencapaian yang optimum diperolehi apabila peratus pemilihan mutasi yang rendah dan bilangan pergerakan yang sedikit digunakan.

THE OPTIMUM COMBINATION OF LOCAL SEARCHES FOR GENETIC OPERATORS IN MEMETIC ALGORITHM FOR THE SPACE ALLOCATION PROBLEM

ABSTRACT

This thesis investigates the university space allocation problem, which focuses on the distribution of events among the available venues, without violating any hard constraints while satisfying as many soft constraints as possible and ensure optimum space utilization.

To simplify this complex problem, we applied the decomposition method by using the concepts of tuple and bin. In this research, we have shown that the Memetic Algorithm produces a better result compared to the Genetic Algorithm for the space allocation problem. We combined the optimum local searches for the genetic operators in Memetic Algorithm. These local searches are Hill Climbing, Tabu Search and Simulated Annealing. This is a different approach from the past, where only one type of local search is used in the Memetic Algorithm. We also investigated the optimum selection method to choose individuals for the reproduction and crossover genetic operators. The application of several selection methods to choose individuals in Memetic Algorithm is also a different approach, as in the past, only one type of selection method is used. Our investigation on the mutation performance shows that lower mutation selection percentage and smaller movement count in the Memetic Algorithm produce better results.

CHAPTER 1
INTRODUCTION

1.0 Introduction

Space optimization and utilization are important aspects in almost all companies and organization regardless of the size of the institutions. Since the available space is often limited, the efficiency of an institution depends on having a good distribution of the limited space. A good distribution must ensure that all events are given the required space and the space itself is utilized as efficiently as possible. It must also ensure that the additional constraints are satisfied as much as possible. An efficient utilization of space means that no event is given too much space, which is classified as space wastage, and no event is given less space than it minimally requires, which is space overuse (Landa Silva, 2003).

Space allocation problem arises in many institutions, raising the need to develop effective and efficient automated solution methods. However, this problem, as with many other combinatorial optimization problems, is difficult to tackle with computer algorithms. Among the characteristics that make these problems very difficult are (Burke & Landa Silva, 2004):

- Huge search space – The size of the search space increases dramatically with the size of the problem, making it impossible to explore all solutions except for very small problems.
- Highly constrained – Usually a considerable number of constraints exist in this problem. Constraints limit the possible ways in which a schedule can be constructed.
- Difficult to represent – Often it is difficult to find a representation that captures all the details of the problem including the complete set of constraints. Thus, most of the time, the problem is simplified.
- Time-consuming fitness evaluation – Computing the fitness of the solutions in the problem usually consumes time, mainly due to the existence of many constraints.

This thesis investigates the space allocation problem with the Memetic Algorithm approach. We selected the university space allocation as our research area. Space allocation is an important issue in the university sector (Burke & Varley, 1998a). As we know, the Memetic Algorithm is a combination of Genetic Algorithm and local search. The Memetic Algorithm is a very powerful algorithm that can be applied to a wide variety of problems and has been gaining popularity among researchers and practitioners. This is because the Memetic Algorithm provides simple formula that allows combining a robust global search technique, with powerful domain specific local searches (Krasnogor et al., 2006). In this research, three local searches for genetic operators are combined in the Memetic Algorithm, to find the optimum combination for the space allocation problem. These local searches are Hill Climbing, Simulated Annealing and Tabu Search – the three most popular local searches (Aarts & Lenstra, 1997).

1.1 Space Allocation Problem - Background

The space allocation problem can be viewed as a problem of distributing the available space among the demanding entities in such a way that the space utilization is optimized (Landa Silva, 2003).

The space allocation problem is a combinatorial optimization problem involving multi-criteria decision process (Landa Silva, 2003). It is a problem where the amount of space or area or capacity that is available has to be distributed among a set of items, satisfying specific requirements and constraints (Landa Silva, 2001). Multi-criteria or multi-objective means several criteria must be taken into consideration when evaluating the quality of the proposed solution, and some of these objectives are conflicting with each other and in fact incommensurable (Landa Silva et al., 2004).

Examples of space allocation problem include bin packing problem, knapsack problem, space planning, academic resource allocation and others. All these problems have one important condition, that is – the available space and events are fixed and not subject to modification. In the knapsack problem for example, there are a number of items of given sizes and a number of knapsacks of given capacities. Each item has an associated profit and an associated weight assigned to it. The goal of this problem is to fill each knapsack with a subset of the items without exceeding the capacity of the knapsack and by maximizing the total profit (Khuri et al., 1993). However, there are no additional constraints exist in the traditional knapsack problem. In academic timetabling, the problem is to accommodate a set of timetable events into the set of available timeslots and satisfy the additional constraints. In the bin packing problem, a number of items with different sizes must be packed into a number of bins with different sizes in such a way that no bins overflow or a number of items with different sizes must be packed into a number of bins with identical size in such a way that the number of bins used is minimized (Reeves, 1996).

Space allocation in academic institution has close relationship with academic timetabling, either course or examination timetabling since it is actually a part of academic timetabling problem. In 1996, Wren has defined scheduling as arrangement of objects into a pattern of time or space in such a way that the goals are achieved or nearly achieved, and the constraints of the objects are satisfied or nearly satisfied. Wren has also clearly defined that a timetable shows when particular events are to take place. It does not necessarily imply an allocation of resources. Hence, there is distinction between timetabling and space allocation even though they are interrelated (Wren, 1996). For timetabling, the main concern is *when* the events are to take place whereas for space allocation; it's *where* they take place (Newall, 1999).

In the context of academic institutions, the space allocation problem is defined as the allocation of events to area of space such as rooms, optimally and satisfying as many requirements and constraints as possible (Burke & Varley, 1998a; 1998b).

1.2 Problem Statement

Space allocation in academic institution is a complex, difficult and time consuming task, often carried out manually or semi-automatically by the officers involved. This university space allocation problem emphasizes on allocating timetable events to area of space, which differs from the usual space allocation in academic institution (Landa Silva, 2001); (Burke et al., 2001a; 2001b; 2001c; 2001d); (Landa Silva, 2003); (Burke & Varley, 1998a; 1998b); (Ritzman, 1980); (Benjamin, 1992); (Diminnie, 1986); (Burke et al., 2000), whereby those studies emphasize allocation of rooms among teachers or lecturers, researchers, and non-academic staffs. Our research differs from theirs as we are focusing on the course space allocation problem.

Generally, timetable generating method varies from one education institution to another. At this institution, the course timetabling process is carried out at two levels: school level, and university central unit level. At the school level, each school prepares the basic timetable. This basic timetable consists of: arrangement of subjects (class) with the time of the class, lecturer assigned as well as the size of the class (the number of students registered or expected to register for that subject). Each school then submits the basic timetable to the university central unit, where the second level of timetabling, which is the space allocation takes place.

Since the schools generate and provide the basic timetable to the university central unit, the timetable fundamental constraints are resolved. Therefore, we no longer need to take the following constraints into consideration:

- Conflicts of classes for students,
- Conflicts of classes for lecturers,
- Total lecture hours that must be covered per subject, and
- Type of classes (e.g.: 3 single periods, 1 double & 2 single period, etc).

This means that the main concern of timetabling, *when* has been answered, and the space allocation concern of *where* is what we are interested in, to solve.

There are two fundamental constraints that are universal to all timetabling and space allocation problems in general, and that no feasible timetable may violate. These are:

- No resource can be in more than one location at any one time
- For each time period, there should be sufficient resources available for all the events that have been scheduled for that time period (Burke et al., 2001a).

However, there are other constraints to be considered. Constraints can be any of the following types – soft constraints or hard constraints. Hard constraints are conditions that cannot be violated at all. On the other hand, soft constraints are rules that desired to be satisfied but not essential therefore can be broken but penalized (Landa Silva, 2003). Following are the constraints that need to be considered and fulfilled in the space allocation problem:

- Sharing restrictions

  Strictly no sharing among the rooms is allowed. Therefore, only one class can be assigned to one room at any one time.
- Proximity or distance requirement

  For two or more consequent classes, it is compulsory to use the same room.
- Resource specific location.

Certain classes can only be held in the specific rooms or specific sets of rooms. For example, Chemistry class must be held in Chemistry labs only. There are also classes that prefer to have certain facilities but it is not mandatory.

- Requirements and limits for wastage and overuse of space.

  The room capacity must be appropriate with the size of the class. There should not be too much of space wasted or overused.

- Pre-booking.

  Some rooms are pre-booked by certain schools. These pre-booking should not be changed in the end result.

- Priorities or preference.

  Some schools are given priorities at certain hours of the day to minimize friction between them.

- All classes assigned.

  All classes must be assigned to rooms.

Thus, the space allocation here refers to the distribution of the available space with different capacities and specifications, among sets of timetable events with different requirements and sizes, without violating any mandatory condition, satisfying as many other requirements as well as constraints as possible, and ensure optimum space utilization.

1.3 Research Objectives

This research is conducted with several objectives. We aim to demonstrate the capability of Memetic Algorithm in handling highly constraint optimization problem such as this problem. We selected Memetic Algorithm as it has been recognized as a powerful algorithm in evolutionary computing (Wu, 2001); (Burke & Landa Silva, 2004).

The core objective is to find the optimum local searches for the genetic operators and combine them in Memetic Algorithm, for the space allocation problem; among Hill Climbing, Tabu Search and Simulated Annealing. As mentioned earlier in this chapter, these three local searches are selected as they are the most popular local search.

In addition to that, we are interested in finding the optimum selection method to select individuals during the reproduction and crossover genetic operations. The investigated selection methods are Random, Deterministic and Roulette Wheel selection.

Besides, we would also like to find the optimum mutation selection percentage and movement count for Memetic Algorithm. Mutation selection percentage determines the size of the subset selected from an individual to perform mutation. On the other hand, movement count determines the number of movement that takes place within the selected subset, during the mutation process.

1.4 Contributions of the Thesis

The contributions of this thesis are summarized as follows:

- Application of Memetic Algorithm in the university space allocation problem. The performance of Memetic Algorithm in handling this problem was compared against Genetic Algorithm. We found that both single local search Memetic Algorithm and combined local search Memetic Algorithm performed far better than the Genetic Algorithm.
- Application of three local searches separately after each genetic operation in Memetic Algorithm. In the past only one type of local search is used in the Memetic Algorithm (Digalakis & Margaritis, 2004); (Rossi-Doria & Paechter, 2004); (Burke et al., 2001e). We found that by combining the local searches optimally, a better result can be obtained, compared to the single local search Memetic Algorithm.

- Application of several selection methods to choose individuals during reproduction and crossover. In the past, only one type of selection method is used in the Memetic Algorithm (Burke et al., 1996, 2001e);(Rossi-Doria & Paechter, 2004), (Digalakis & Margaritis, 2004). We found that the optimum selection method for reproduction is different from the crossover's optimum selection method. In fact, to obtain an optimum result for crossover, two different selection methods were applied. Random selection method produced the optimum result in reproduction. On the other hand, in crossover, combination of Random and Roulette Wheel selection method, where one parent is selected randomly and another is selected by using Roulette Wheel method, produced the optimum result.

1.5 Overview of the Thesis

This thesis has six chapters including this first chapter which is the introduction chapter. This chapter gives an overview and later describes the background of the space allocation problem. The problem statements were explained as well. The objectives of the research were also spelled out. It also covers the overview of the thesis, briefly explaining the content of each chapter.

Chapter two reviews the literature from two perspectives – how the university space allocation problem was tackled in the past, and the effect of Memetic Algorithm in solving problems in the past. Goal Programming, Hill Climbing, Simulated Annealing, Genetic Algorithm, Metaheuristic and Iterative Improvement Algorithm are among the methods used in the past to solve the space allocation problem. Besides, basic introduction to Memetic Algorithm is covered as well. This provides us some idea about the problem discussed. It also helps to understand the advantages and disadvantages of Memetic Algorithm and the local searches by knowing their capabilities and weakness.

Chapter three covers the problem representation and the solution method. This chapter begins by introducing the concept of decomposition (also known as divide and conquer), tuple and bin. Then, the problem is represented by applying these concepts. The search and decision making process is discussed as well. The equation of the problem is also presented. This is followed by elaboration of each solution method in detail.

Chapter four is about the methodology where the method of implementation is discussed in detail. The overall algorithm is spelled out in detail. This is followed by explanation about each genetic operation, selection method and local search. The prototype used in the experimentation is also introduced in this chapter.

Chapter five discusses the series of experiments carried out by using the prototype. We have two phases of experiments and total of 10 sets of experiments, each fulfilling the aspects that meet our objectives. The experiments, results and findings are presented and discussed in this chapter.

Chapter six describes the conclusions made from the experiments' results and observations. Recommendation and future directions are presented at the end of this final chapter.

CHAPTER 2
LITERATURE REVIEW

2.0 Introduction

In this chapter, previous research work on the university space allocation problem is reviewed and analyzed. Over the years, various approaches were taken to solve the space allocation problem. Goal Programming, Hill Climbing, Simulated Annealing, Genetic Algorithm, Metaheuristic and Iterative Improvement Algorithm are among a few of them. These works are categorized into Mathematical approach and Metaheuristic approach. Besides, brief introduction to Memetic Algorithm is also given. Previous researches on Memetic Algorithm approach in handling similar problems are also reviewed, giving some idea on the possible approaches to tackle the university space allocation problem. We found that the Memetic Algorithm produces better results compared to other algorithms, in solving the optimization problem.

2.1 Previous Research on University Space Allocation Problem

2.1.1 Mathematical Approach

2.1.1.1 Mixed Integer Goal Programming

In 1978, Ritzman presented research on the automated planning of academic facilities (Ritzman et al., 1980). His studies were on reassignment of 144 offices to 289 members in 6 academic departments within the Ohio State University. The objective of the research was to make the reassignment of offices as fair as possible, but some conflicting objectives were identified. Among them were minimizing the distances between the rooms assigned to each department and its administrative office, and ensure that each department obtains a fair share of the available high quality offices. Besides, the number of reassignments must be minimized.

The author calculated the constraints of each goal separately and used the Mixed-Integer Goal-Programming model to formulate the problem. An interactive computer program was developed to evaluate the performance of each solution with respect to the studies' objectives. The program allowed the decision-makers to obtain and compare different alternative layouts before producing a final solution. The goal programming showed that either more good space was required or some departments would have to lower their expectation levels. From the computational experience, it was found that Mixed Integer Goal Programming model was rather complex for this type of problem and a standard Linear Programming was sufficient.

2.1.1.2 Linear Goal Programming

In 1987, a new computer integrated manufacturing laboratory was constructed where 15 sections would be allocated at the University of Missouri-Rolla (Benjamin & Omurtag, 1992). In addition to the allocation of space to each section, several goals were specified, whereby some of them were conflicting with each other. This includes developing new courses relying on the laboratory facilities, increase the students' use of the laboratory facilities and stimulate the graduate-level and funded research.

Benjamin and his team applied the Linear Goal Programming approach to this problem. Before applying the algorithm, the goals were prioritized. This requires a substantial amount of time and knowledge from the decision-makers. To help the facilities planners prioritize the conflicting goals, they used the analytic hierarchy process (AHP). The AHP is a multiobjective decision making technique. It employs pair wise comparisons to rank the alternatives of problems in hierarchical structure. The AHP methodology provides a systematic method to determine the weights and priorities for a set of objectives. The basic assumption was that the objectives of a problem can be represented in a hierarchical structure. Then, this priority structure was incorporated into Linear Goal

Programming model that determines the optimum resource allocation. The AHP methodology was found to be effective in eliciting the judgement. The model was applied to plan the laboratory and the lab fulfilled its teaching, research and extension objectives. The drawback of Benjamin's research was that no comparison was made with any other algorithm. Therefore the effectiveness of the methodology could not be precisely judged.

2.1.1.3 Pre-emptive Goal Programming

In 1995, Giannikos solved the university space allocation problem by using the Integer Goal Programming (Giannikos et al., 1995). The problem was reorganizing the distribution of the academic space in six major sites at the University of Westminster, United Kingdom. Among the main objectives include assigning enough and adequate type of offices to each school and each office is assigned to only one school. Other objectives were minimizing the distance between the rooms assigned to a school to its administrative centre and minimizing the number of people that have to be relocated.

Since the university found it easier to rank the research objectives according to their importance, they decided to use the Pre-emptive Goal Programming to obtain a satisfactory solution. The Pre-emptive Goal Programming starts by finding a solution that is as close as possible to meet the highest priority goal.

Comparing the actual distribution of offices with the one produced with the automated method, it was found that in the former, the office space was used in an inefficient way. The authors highlighted that their ultimate goal was to provide the managers with a decision support tool to evaluate the current space distribution and explore alternative allocations and the objectives were met. They also showed that it is possible to find a solution that satisfies most of the objectives and the model can be used to obtain a satisfactory allocation of office space in any similar problem. However, just like Benjamin, Giannikos failed to precisely prove the effectiveness of the solution. He did not compare the integer goal-programming with other algorithms.

2.1.2 Metaheuristic Approach

Burke and his team from the ASAP, Automated Scheduling and Planning Group of University of Nottingham have shown deep interest in space allocation automation for higher academic institutions. The problem studied was departmental room allocation for non-residential space in the universities in United Kingdom. These rooms include lecture halls, labs, lecture rooms and staff rooms (Landa Silva, 2001); (Burke et al., 2001a; 2001b; 2001c; 2001d); (Landa Silva, 2003); (Burke & Varley, 1998a); (Burke et al., 2000); (Burke & Petrovic, 2002); (Giannikos, 1995). In October 1996, questionnaires on the university space allocation subject were sent to the estate managers of 96 British universities. In most of the surveyed universities, the room allocation process was carried out manually and only a few British universities used some kind of automated tools.

There are two different levels of problem in space allocation: space utilization level and constraint satisfaction or optimization level. The problem was complicated by the fact that not all events are capable of sharing rooms with other events and majority of them require their own rooms. The problem was then to maximize the utilization of the rooms without violating any of the sharing limitation.

By grouping the adjacent rooms and storing the information regarding the distance of these groups from each other, the amount of information was reduced. The subset grouping method is also applied to the resources requiring allocation. Knowing which rooms are adjacent to each other, allows the adjacency constraints to be easily satisfied. The generalized penalty function was used in order to ascertain the quality of the space allocation solution. Generalized penalty function was a result of the multi-objective problem linearization into a single overall objective problem. Refer to *Section 3.2.5* for further information on the generalized penalty function.

2.1.2.1 Hill Climbing, Simulated Annealing and Genetic Algorithm

In 1998, Burke and Varley used three different optimization methodologies to automatically generate solutions to the space allocation problem (Burke & Varley, 1998a, 1998b). The methodologies were Hill Climbing, Simulate Annealing and Genetic Algorithm. Hill Climbing was applied in two ways: random selection of rooms (also called as random fit) and selection of room with the lowest penalty (best fit). The Genetic Algorithm used roulette wheel method in the selection process. The Genetic Algorithm was tested with various population sizes and various initial populations. It was tested with the random fit Hill Climbing (random selection of rooms), best fit Hill Climbing (selection of room with the lowest penalty) and Simulated Annealing initialized population.

The conclusion of this research was Simulated Annealing performed the best and random fit Hill Climbing performed the worst. However, the results were offset by the amount of time taken by each method – Simulated Annealing took far longer time compared to the Hill Climbing method.

In a separate research, Burke and his team worked on improving the existing distribution of rooms in academic institution in United Kingdom (Burke et al., 2000). The team compared Hill Climbing, Simulated Annealing and Genetic Algorithm in improving an existing allocation. The authors concluded that in attempting to optimize an existing allocation, Hill Climbing algorithm produced the best results.

2.1.2.2 Variation of Hill Climbing, Simulated Annealing and Genetic Algorithm

Later, Burke used twelve different variants from the same three optimization methodologies (Hill Climbing, Simulated Annealing and Genetic Algorithm) on the above mentioned space allocation problem (Burke et al., 2001a).

The authors observed that the Hill Climbing variants produced the best results when applied to optimization problems. For example, when there was an existing allocation and it needs to be improved. Simulated Annealing and Hill Climbing variants produced the best results for the reorganizing allocation problem. The reason why these strategies have good performance in optimizing and reorganizing problems could be for the fact that the most conflicting resources were already allocated. Thus, the improvement of these solutions can be accomplished by using these local search strategies. Genetic Algorithm had good performance in reorganizing problems if there are only basic constraints. In constructing a completely new allocation, Hill Climbing and Simulated Annealing variants constructed good solutions but did not match the quality of the manually constructed allocation. However, Genetic Algorithm was capable of producing acceptable results in terms of time in constructing complete allocations, but neither provided a better solution than the one obtained manually. Genetic Algorithm produced a set of solutions that can be improved using a local search heuristics.

2.1.2.3 Hybrid Metaheuristic

In 2001, Burke's team used the Hybrid Population-based Metaheuristic (Burke et al., 2001b; 2001d, 2004, 2005) to solve the space allocation problem in academic institutions in United Kingdom. The author investigated the application of Hill Climbing, Simulated Annealing, Tabu Search and Genetic Algorithm. The approach incorporated the best characteristics of each technique and made an automatic selection of the parameters according to the problem characteristics. This approach incorporated local heuristics (such as Hill Climbing), adaptive cool schedules in Simulated Annealing, tabu list (list of favorable) move and mutation operators.

The authors compared the Hybrid Metaheuristic with standard Simulated Annealing implementation as well as manual solution. On space utilization, both Hybrid Metaheuristic and Simulated Annealing achieved good results comparable with the manual solution. The differences between the performances of both approaches were mostly in constraint satisfaction. Even the worse solutions produced by the Hybrid Metaheuristic were better than those obtained with the standard techniques (Hill Climbing, Simulated Annealing, Tabu Search and Genetic Algorithm). The Hybrid Metaheuristic results in solutions with total penalty value as low as the manually constructed solutions. This methodology is capable of producing one single high quality solution or a population of high-quality allocations.

2.1.2.4 Metaheuristic

Landa Silva presented an investigation on the application of Metaheuristic technique to solve the space allocation problem in the academic institution (Landa Silva, 2003). He proposed and compared a range of heuristics for the initialization of solutions. The space allocation problem was approached as a single objective optimization problem as well as from multiobjective perspective using the Pareto optimization.

Experiments were also carried out to compare the performance of four Metaheuristic - Iterative Improvement, Simulated Annealing, Tabu Search and Genetic Algorithm. It was observed that these Metaheuristic approaches offer a good alternative for automating the academic space allocation process in a shorter time. Among these approaches, the author concluded that the Iterative Improvement and Tabu Search were able to produce the best results, however still do not match the quality of the manually constructed allocation when the problem is highly constrained. Overall, Iterative Improvement and Tabu Search were the best performers, Simulated Annealing produced good results and the Genetic Algorithm was the worst performer.

The author then investigated single-solution Hybrid Metaheuristic as well as population based Hybrid Metaheuristic approach for the same problem. The author assessed the performance of the proposed hybrid approach and compared against three single solution Metaheuristics - Iterative Improvement, Simulated Annealing and Tabu Search. It was found that the Hybrid Metaheuristic outperforms the other three algorithms and it was capable of finding better solutions. Besides, the performance of the Hybrid Metaheuristic was found to be more robust compared to the rest. In terms of space utilization, the experimental results showed that all the solutions obtained with the four algorithms were comparable with the manual solution. However, the single-solution Hybrid Metaheuristic produced better quality solutions because it was capable of finding solutions with less violation to soft constraints than the solutions produced by the other three algorithms.

Later, the single-solution Hybrid Metaheuristic was extended to population-based Hybrid Metaheuristic to experiment its performance. The author observed that the population-based algorithm produced solutions that were very competitive compared with those obtained by the single solution approach. However, in terms of the quality of the solutions, the results produced by the single-solution approach were better that those obtained with the population-based variant. The population-based algorithm also produced more diverse sets of solutions, which is an important goal in the multi-criteria decision-making and multi-objective optimization.

The author also assessed the suitability of the single Hybrid Metaheuristic algorithm and population-based Hybrid Metaheuristic algorithm for Pareto optimization of the space allocation problem. From the obtained results, it was clear that the population-based Hybrid Metaheuristic algorithm produced the best sets of non-dominated solutions.

2.1.2.5 Iterative Improvement Algorithm

In 2006, Abdullah used a randomized Iterative Improvement algorithm with composite neighborhood for the university space allocation problem (Abdullah et al., 2006). It is referred as university course timetabling problem in the literature. This problem deals with the assignment of a set of courses to specific timeslots and rooms within a working week. At the same time, students and teachers were assigned to courses so that the events can take place.

The author presented a composite neighborhood structure with a randomized iterative improvement algorithm. This algorithm always accepted an improved solution and a worse solution was accepted with a certain probability. A composite neighborhood structure consists of two or more neighborhood structures. The advantage of combining several neighborhood structures was, it helps to compensate the ineffectiveness of each type of structure when used in isolation.

The experiment results showed that this approach was capable of producing high quality solutions and particularly effective on smaller problems. In the case of medium problems, good solutions were obtained at the expense of significantly high computational time.

2.2 Previous Research on Memetic Algorithm in Solving Optimization Problems

Memetic Algorithm was inspired by Richard Dawkin's concept of meme, which represents a unit of cultural evolution that could exhibit local refinement. The concept of Memetic Algorithm was first introduced by Moscato and Norman to describe the Evolutionary Algorithm in which local search is used to a large extent. It was later formalized by Radcliffe and Surrey (Radcliffe & Survey, 1994). "Memetic Algorithm is an Evolutionary Algorithm that includes one or more local search phases within its evolutionary cycle" – Krasnogor, 2002.

2.2.1 E.K. Burke's Research

2.2.1.1 Exam Scheduling

In 1996, Burke and his team used the Memetic Algorithm approach in the university examinations scheduling in selected universities in UK (Burke et al., 1996). The main technique used in the algorithm was combination of both light and heavy mutation followed by Hill Climbing. The initial population was generated by using the roulette wheel selection method to choose which period to place each examination. This selection was made based on the examinations which were already placed. A mix of random and heuristic assignment was chosen in order to produce a higher quality initial population. A random operator was used to perform light mutation on the population. This was followed by the application of Hill Climbing algorithm. On the other hand, the heavy mutation operator preserved well-constructed periods in a timetable while randomly rescheduling the examinations in the remaining periods to find new higher quality solutions.

The algorithm was tested on real data and the results showed that the addition of Hill Climbing local search after each mutation operator greatly increased the speed and better solutions were obtained compared to the evolutionary operators alone. Although the initial results were promising, the algorithm did not perform well on the more highly constrained problems compared to other methods.

2.2.1.2 Timetabling

Burke and Newall, 1999 presented the multi-stage evolutionary algorithm for the timetabling problem. Multi-stage algorithm is a method of decomposing larger problems into smaller components - a size that the evolutionary algorithm can effectively handle. This is an extended research of Memetic Algorithm. This is because, while the Memetic approach showed promising results for timetabling problems, the time involved in optimizing large problems is much longer. It is preferred to produce timetables in a matter of minutes and to achieve this, the original Memetic Algorithm was used but only applied to a subset of the total events at a time. The algorithm was able to fix the events in the timetable before considering the next subset of events. The multi-stage evolutionary algorithm not only drastically reduced the amount of time to find that solution, but also considerably improved the quality of that solution.

2.2.1.3 Thermal Generator Maintenance Scheduling

Later, Burke and Smith presented Memetic Algorithm approach in the maintenance-scheduling problem (Burke & Smith, 1997; 1999; 2000). The thermal generator maintenance-scheduling problem is about scheduling essential maintenance over a fixed planning horizon for a number of thermal generator units while minimizing the maintenance costs and providing enough capacity to meet the anticipated demand. The problem is classified as a deterministic cost-minimization problem. In earlier work, the authors found that using the local search alone produced good results. The authors compared the Simulated Annealing, Tabu Search and Genetic Algorithm for this problem. The results showed that Tabu Search performed well compared to other algorithms and Genetic Algorithm performed badly. Combining Tabu Search and Simulated Annealing into a single hybrid algorithm produced better results than Simulated Annealing alone. However, it is still not better than the results achieved by Tabu Search.

The authors tested several initializations including random and heuristic techniques. The results suggested that a good initial population was less significant for Memetic Algorithm compared to the Genetic Algorithm. This is because, after the first application of local search, the fitness of each individual was improved tremendously regardless of its initial population starting point. Tournament selection method was found to produce better results compared to roulette wheel selection.

To conclude, in terms of time, Memetic Algorithm was slower than non-Memetic algorithm. Tabu Search was capable of finding better solutions, even though at the expense of longer running time. Tabu Search ran twice longer than Simulated Annealing. Using Hill Climbing in Memetic Algorithm improved the speed of the solution but the results were not of a good quality. Simulated Annealing on the other hand, found better solutions than Hill Climbing in a shorter time than Tabu Search. The fastest local optimizer was the Hill Climbing, followed by Simulated Annealing and Tabu Search. The results of the experiments also showed that for very small problems, the Memetic Algorithm produced similar solutions as Simulated Annealing and Tabu Search. But for larger scale problems, the Memetic Algorithm was highly successful.

2.2.1.4 Nurse Rostering

In 2001, E.K. Burke and his team deal the nurse rostering problem in the Belgian hospitals with the Memetic Algorithm approach (Burke et al, 2001e). In Belgian hospitals, the personnel prefer ‘ad hoc’ schedules more than the rigid practices of regular three-shift schedules that rotate weekly. Moreover, the requirements of the hospitals demand broader variety of services than just morning, day and night shifts. Thus, rise the need of scheduling nurses to suit both the hospital requirements and personnel preferences.

In this case, the authors experiment small rostering problem using the Tabu Search heuristics and Memetic Algorithm. Three types of initialization strategies were identified: the current schedule, the previous planning schedule and random initialization. Several variants of Memetic Algorithm with different recombination mechanisms were tested. The automated results reduced the scheduling effort and calculation time considerably compared to the manual approach. Hybrid Tabu Search could quickly find reasonably good schedules in response to the events such as staff absenteeism. The Memetic Algorithm approach was able to produce excellent solutions when more time was available. The hybrid Memetic Algorithm, which combined the basic approach with the hybrid Tabu Search provided good solutions and the solutions were significantly better than the best Tabu Search solution and they were unaffected by the initialization and parameter changes.

2.2.2 Others' Research

2.2.2.1 University Course Timetabling

Alkan and Ozcan presented a variety of new operators that can be applied in Memetic Algorithm for the university course timetabling problems (Alkan & Ozcan, 2003). Operators include violation directed mutations, crossovers and violation directed hierarchical Hill Climbing method. Tests were performed on a small portion of a real data obtained from the Faculty of Engineering and Architecture (FEA), Yeditepe University (YU). Their earlier experiments on the same research using the Genetic Algorithm showed that the individuals tend to become similar, causing premature convergence unavoidable.

After random initialization was applied, the population was passed through the Hill Climbing local search. A random, low probability mutation was applied. An additional mutation was also implemented to guide the search, and penalty values were adjusted by a factor. Besides, one point crossover and uniform crossover were also applied to the population. New crossover selection was created using the ranking strategy.

The violation directed Hill Climbing refers to the application of Hill Climbing method to each type of constraint and combine them under a single Hill Climbing. Steady-state and trans-generational approaches were also implemented. Steady-state approach requires two offspring to be produced, whereas, the trans-generational requires creation of an offspring pool and replacement occurs on the old generation and the offspring.

The experimental results confirmed that the best crossover operator was the traditional uniform crossover operator and the best mutation operator was the violation directed operator that was applied on a subset rather than the whole individual. Experiments also demonstrated that genetic search combined with Hill Climbing achieved the best performance. Besides, trans-generational Memetic Algorithm yielded better results than the steady state Memetic Algorithm.

In 2004, Rossi-Doria and Paechter approached the university course timetabling by using the Memetic Algorithm (Rossi-Doria & Paechter, 2004). The research was a simulation of actual timetabling problem at Napier University in Edinburgh. The timetable was represented in the form of an integer matrix. They used the steady-state evolution model, where only one child solution was generated from two parents at each generation.

Initial population was generated by mixture of constructive heuristics and semi-random manner to ensure diversity. On top of that, each individual was improved with local search before the evolution start. Parents were selected using the tournament selection whereby two randomly selected individuals were compared and the best among them was selected to be the parent. Simple crossover and 80% mutation rate were applied. Replacement strategy was also applied, where at each generation, the child replaced the worst number of population, as long as it is not identical to either parent in order to avoid early convergence.

This Memetic Algorithm was compared against the random restart local search (RRLS). The RRLS iterates the same local search used by Memetic Algorithm from the starting solution built with the same initialization constructive strategy, and stores the best solution found. The conclusion was, comparing Memetic Algorithm with the RRLS, the later failed to find good results. The Memetic Algorithm provided a far better result. The use of the effective local search was also found to be an important element of the Memetic Algorithm's good performance.

2.3 Summary

Goal Programming, Hill Climbing, Simulated Annealing, Genetic Algorithm, Metaheuristics and Iterative Improvement Algorithm are among the approaches taken in the past to solve the space allocation problem. As the Memetic Algorithm approach is gaining popularity and has proven to perform better compared to other algorithms, we take this opportunity to investigate the effectiveness of Memetic Algorithm in handling such problem and compare it with other proven solution such as Genetic Algorithm.

The literature also showed that over the years, among the popular algorithm used in optimization problems are Hill Climbing, Simulated Annealing, Genetic Algorithm, and Tabu Search. In fact, Metaheuristic and Memetic Algorithm are derived from these algorithms. So, we will continue to use these algorithms (Hill Climbing, Simulated Annealing, Genetic Algorithm, and Tabu Search) in this research, as Memetic Algorithm.

However, we attempt to further improve the approach by finding the optimum local search for each genetic operator and combine them in Memetic Algorithm.