AN EFFECTIVE PARAMETERS SETTING FOR GENETIC ALGORITHM WITH LOCAL SEARCH IN JOB SHOP SCHEDULING PROBLEM

By

INAAM JABBAR KADHIM
(1430211403)

A thesis submitted in fulfilment of the requirements for the degree of Master of Science in Computer Engineering

School of Computer and Communication Engineering
UNIVERSITI MALAYSIA PERLIS

2016

ACKNOWLEDGMENT

In the name of Allah, I would like to praise Allah S.W.T., for providing me with his generosity and blessing to finish this research.

I wish to express my sincere thanks and appreciation to my, Main Supervisor, Dr. Shahrul Nizam Bin Yaacob and Co-Supervisor, Prof. Sabira Khatun for their valuable advices, guidance and supervision during the progress of this research.

I would like to extend my thanks to Professor Dr. R. Badlishah Ahmad (Dean of the School) and Dr. Phaklen Ehkan, and School Computer and Communication Engineering, University Malaysia Perlis (UniMAP), for their help and support me in overcoming the difficulties that I face to achieve this work.

Last but not least, I would like to offer warm and sincere thanks to my parents, and all my family, for their great support, encouragement and unshakable faith in me. Also, my own special thanks to my loving husband, who has been very supportive and encouraged me every step of the way. Thanks also to all postgraduate friends and all of those who supported me in any respect throughout this research.

Inaam Jabbar Kadhim

Universiti Malaysia Perlis (UniMAP)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACBN	Active Critical Block Neighbourhood
AS	Adjacent Swapping
B&B	Branch and Bound
BDR	Basic Dispatching Rules
CB	Critical Block
DR	Dispatching Rules
EDD	Earliest Due Date First
FCFS	First Come First Serve, first
FIFO	First In First Out
GA	Genetic Algorithm
GLS	Genetic Local Search algorithm
GOX	Generalised Order Crossover
GPMX	Generalized Partially Mapped Crossover
GT	Giffler and Thompson
JSSP	Job Shop Scheduling Problem
LPT	Longest Processing Time
MSMF	Multi-Step Mutation Fusion
MSXF	Multi-Step crossover Fusion
NS	Nowicki and Smutnick
PPX	Partially Preservative Crossover
SA	Simulated Annealing
SB	Shifting Bottleneck
SGA	Simple Genetic Algorithm
SIRO	Service in Random Order
SPT	Shortest Processing Time First
TS	Tabu Search
TSP	Travelling Salesman Problem

LIST OF SYMBOLS

J_i	Job $i.\,(\,i\,=\,1,2,\ldots.n\,)$
M_k	Machine $k\,(\,k\,=\,1,2,\ldots,m)$
O_{iqk}	q-th operation for job i
S_{iqk}	Starting time forO_{iqk}
r_{iqk}	Ready time for O_{iqk}
P_{iqk}	Processing times for O_{iqk}
C_{iqk}	The completion time (makespan)
C_{Max}	the minimization of completion time (makespan) values
T	Metropolis criterion
$n \times m$	n number of jobs need to be processed on m number of machines
ABZ5	Benchmarks Problems 10 jobs and 10 number machines
ABZ6	Benchmarks Problems 10 jobs and 10 number machines
LA01	Benchmarks Problems 10 jobs and 5 number machines
LA02	Benchmarks Problems 10 jobs and 5 number machines
LA03	Benchmarks Problems 10 jobs and 5 number machines
LA06	Benchmarks Problems 15 jobs and 5 number machines
LA07	Benchmarks Problems 15 jobs and 5 number machines
FT06	Benchmarks Problems 6 jobs and 6 number machines
FT10	Benchmarks Problems 10 jobs and 10 number machines

PENETAPAN PARAMETER BERKESAN UNTUK
ALGORITMA GENETIK MENGGUNAKAN KAEDAH CARIAN
TEMPATAN BAGI MASALAH PENJADUALAN KERJA-KEDAI

ABSTRAK

Tesis ini bertujuan untuk menyelesaikan masalah Penjadualan Kerja-Kedai (JSSP) dengan objektif utama untuk meminimumkan nilai *makespan*. Keberkesan penetapan parameter dan kejiranan struktur bagi algoritma genetik (GAS) telah dianalisa dan dikenal pasti menggunakan teknik Penghibridan Genetik (GLS) algoritma untuk menyelesaikan JSSP Pencarian Dalaman (LS). Algoritma yang dicadangkan terdiri daripada tiga bahagian utama. Pertama, perwakilan tidak langsung menggabungkan pembina jadual yang melaksanakan LS dicadangkan dan digunakan untuk memecahkan kod kromosom ke dalam kaedah yang dikenali sebagai jadual jadual aktif. Kromosom kemudian dinyahkod ke dalam jadual ini untuk meningkatkan dengan ketara kebarangkalian mendapatkan penyelesaian yang optimum yang hampir. Dalam bahagian kedua, *Multi-Step Crossover Fusion* (MSXF) dengan penjanaan rawak muda sebagai penghubung kepada teknik yang digunakan dalam GLS. MSXF adalah versi panjang bagi LS, menggunakan struktur kejiranan dan ukuran jarak di dalam algirothmnya. Dalam kajian ini juga, teknik Nowicki dan Smutnicki (NS) untuk tujuan carian kejiranan telah digunakan untuk meningkatkan keberkesanan algoritma dalam menghasilkan penyelesaian yang lebih baik. Dengan menggunakan jenis ini, penyelesaian yang dihasilkan di antara kedua-dua induk menggunakan laluan carian penyelesaian yang bersesuaian. Bahagian ketiga dilaksanakan dengan menggunakan pelbagai langkah mutasi fusion sebagai carian tambahan untuk meningkatkan prestasi GLS. Eksperimen pengiraan yang dijalankan ke atas algoritma yang dicadangkan, yang menghasilkan keputusan yang lebih baik untuk masalah penanda aras yang paling berbanding JSSP diperolehi daripada kesusasteraan. Keputusan eksperimen menunjukkan bahawa satu set tetapan parameter GLS berkesan (saiz penduduk = 10; rolet pemilihan roda; MSXF crossover dan kebarangkalian crossover = 0.5 dengan operator mutasi) wujud. Gabungan tetapan ini adalah sinergi, iaitu, mereka secara kolektif membolehkan prosedur untuk menjadi cepat dan mantap. Sementara itu, NS carian kejiranan ketara mengurangkan jumlah panjang makespan dan mengurangkan masa pengiraan. Menyediakan suasana parameter yang berkesan memudahkan penggunaan GLS, menjadikan ia sesuai untuk masa yang optima, aplikasi praktikal dalam bidang sains komputer, industri, dan semua bidang lain yang memerlukan pengurangan kos, kelewatan, dan lain-lain.

AN EFFECTIVE PARAMETERS SETTING FOR GENETIC ALGORITHM WITH LOCAL SEARCH IN JOB SHOP SCHEDULING PROBLEM

ABSTRACT

It is well known that GAs are not well suited for fine-tuning structures that are very close to optimal solutions and that it is essential to incorporate local search methods, such as neighborhood search, into GAs. In addition, in solving combinatorial problems such as job-shop-scheduling problem (JSSP) by GAs, efficient parameter settings that provide optimal solutions are usually difficult to construct. This thesis presents an improved approach to Genetic Algorithm (GA) with Local search techniques (LS) in the hybridization Genetic Local Search (GLS) algorithm. GLS attempts to solve the JSSP with the objective of minimizing the makespan value, this can be achieved using effective parameter setting and ideal neighbourhood structure, in GLS algorithm framework. The proposed algorithm consists of three main parts. In the first one, an indirect representation incorporating a schedule builder that performs simple LS is proposed to decode the chromosome into a legal schedule called active schedule. The chromosomes are then decoded into active schedules to significantly increase the probability of obtaining near or optimal solution. In the second part, Multi-Step Crossover Fusion (MSXF) with the intuition of generating a child from the path re-linking technique is used. MSXF, an extended version of LS, utilizes a neighborhood structure and a distance measure in its procedure. In this study, Nowicki and Smutnicki (NS) neighborhood search is used to increase the effectiveness of the algorithm in producing better solutions. By using this type of crossover, a solution or child is generated between both parents using through a search path joining parent solutions. The third part is accomplished by using multi-step mutation fusion as the supplementary search to improve GLS performance. Computational experiments are carried out on the proposed algorithm, which yields better results for most benchmark problems compared with JSSP obtained from the literature. Experiment results indicate that a set of effective GLS parameter settings (population size = 10; roulette wheel selection; MSXF crossover; and probability of crossover = 0.5 with mutation operator) exists. The combination of these settings is synergistic, i.e., they collectively enable the procedure to be fast and robust. Meanwhile, NS neighborhood search significantly reduces the total length of the makespan and decrease the computational time. Providing an effective parameter setting facilitates the use of GLS, rendering it suitable for wide-ranging real-time, practical applications in computer science, industries, and all other fields requiring the minimization of costs, delays, etc.

CHAPTER 1

INTRODUCTION

1.1 Overview

The increasingly fierce competition in today's business world has motivated manufacturing practitioners to shorten their production time in order to improve their profitability, customer demands, and market survival (Ong, 2013). These improvements can be achieved by optimizing resource allocation through effective scheduling. Effective scheduling plays an important role in reducing production processing times without incurring additional costs, such as those for machines and labour in the production line. Effective production scheduling mechanisms have been recognized to increase productivity and machine utilization (Roshanaei & ElMaraghy, 2012).

Scheduling mechanism is broadly defined by Pinedo as the process of assigning a set of tasks to resources over time to meet certain objectives while complying with a set of constraints (Yang & Gu, 2014). The difficulty of the assignment is increased when the production line is producing variable products. Poor scheduling in this type of production line is not time efficient because of ineffective resource allocation. This allocation of resources is important because a proper allocation enables companies to optimize their objectives and achieve their goals.

Recently, research has been focused on investigating machine scheduling problems in the manufacturing and service environment, where jobs represent activities, machines represent resources, and each process can be assigned to one job at a time. In this type of environment, products are made to order. Usually, these orders differ in terms of processing requirements, material needs, processing times, processing sequences, and

setup times. In this type of low environment system, a scheduling problem is referred to as a job shop scheduling problem (JSSP).

The JSSP is well-known for being one of the most difficult NP-hard combinatorial optimization problems in practice (Yang & Gu, 2014). It becomes complicated to solve when the size of the problems increases. The size of the problems refers to the total number of operation tasks and the total number of machines that are involved in the process. The main objective in solving the JSSP is to find the sequence for each operation on each machine that optimizes the objective function. The most common objective function that has been used in scheduling JSSP is the minimization of the makespan value or the time to complete all jobs.

As an integrated component of computerized and flexible manufacturing systems, the JSSP is widely encountered in many industrial contexts, such as in manufacturing industries, production, transportation, distribution, and information processing and communication. Common scheduling problems include online problems, bus schedules, university timetables, and construction activities. One common feature of these problems is the lack of efficient methods that derives optimum solutions in polynomial time. "Efficient methods for solving JSSP are important for increasing production efficiency, reducing cost and improving product quality" (Sun, Cheng, & Liang, 2010).

As a result of the complexity of the JSSP, the way of solving scheduling problems has shifted from finding exact solutions to using an enumerative algorithm. However, feasible solutions to a wide range of problems cannot always be derived with this technique, the use of which thus becomes limited (S. Jayasankari, 2013).

Only by the end of the 1980s, when new techniques from the field of artificial intelligence emerged, did the approximation method receive significant attention. Since this period, other innovative algorithms have been formulated; examples include shifting bottleneck (Wenqi & Aihua, 2004) , Tabu search (TS) (ChaoYong Zhang, Li, Guan, & Rao, 2007), simulated annealing (SA) (Elmi, Solimanpur, Topaloglu, & Elmi, 2011), and genetic algorithms (GAs)(Sun et al., 2010); (Maghfiroh, Darmawan, & Yu, 2013) , hybrid genetic tabu searches (Banharnsakun, Sirinaovakul, & Achalakul, 2012; Meeran & Morshed, 2012; Zhang, Rao, & Li, 2008; Zhang et al., 2013).

Different from other approximation procedures, GAs can be uniquely characterized by their population-based search strategy and their operators: selection, crossover, and mutation (Yamada, 2003). Unlike other procedures, GAs search from a population point and not from a single point. If the search is conducted from a single point, then the system will likely become trapped at the local optima. In the use of GAs, little information is needed, and these algorithms can be easily adapted to a given problem. Owing to the robustness of GAs when applied to problems with high complexity, they have been widely applied in the fields of artificial intelligence, numeric and combinatorial optimization, business, management, medicine, computer science, engineering, and so on (Ranjini & Zoraida, 2013). On the basis of these advantages, the JSSP is explored in the present work using a GA framework.

GA maintains a population of individuals, each of which represents a potential solution to a given problem. Each solution is evaluated to provide some measure of its "fitness". Then, the fittest individuals are selected to form a new population for the next iteration. In the reproduction process, some selected members endure amendment by means of crossover and mutation to form a new solution. A crossover operator combines

some parts from two individuals to form a new individual. On the contrary, a mutation operator creates a new individual simply by modifying a single individual. These genetic parameter setting, include size of population, selection, crossover and mutation operators, control the precise operation of GAs.

In solving combinatorial problems such as JSSPs by GAs, efficient parameter settings that provide optimal solutions are usually difficult to construct. The quality of the solutions derived with GAs relies on the choice of the best parameters to prevent premature convergence and to ensure diversity in the search space (Essafi, Mati, & Dauzère-Pérès, 2008). Therefore, the performance of the most promising parameter settings efficiency need to be examined before it is applied in GA, for different JSSP benchmark problems.

Furthermore, optimal solutions cannot always be achieved with GAs. As a result, various GA strategies have been introduced to increase the efficiency of GAs in finding optimal or near optimal solutions for JSSPs. Among the GA strategies, the hybridization of GAs with other methods or local search methods has been found to provide good problem-solving results. The strength of GAs in incorporating local search options to locate optimal or near optimal solutions is maximized with such strategies. Specifically, the local search procedure of Nowicki and Smutnicki (2005) is embedded into GAs because of its effectiveness and capability of increasing the performance of GAs (Guan, 2008).

Additionally, the structure of GAs can be modified and enhanced to reduce problems often encountered in GA optimization. In the work of (Park, Choi, & Kim, 2003), the premature convergence in GA was retarded by using parallelization of GA (PGA) to find the near optimal solutions. In the work of (Watanabe, Ida, & Gen, 2005),

a GA with search area adaption and a modified crossover operator was proposed such that the algorithm can be adapted to the structure of the solutions space. In the work of (Ripon, Siddique, & Torresen, 2011), a heuristic method was embedded into crossover functions to reduce the tail redundancy of chromosomes when implementing crossover operations.

As indicated in the literature survey (Godinho Filho, Barco, Neto, Fernandes, & Neto, 2014), the abilities of GAs are enhanced by modifying the structure of these algorithms. Hence, GAs is not restricted to a single procedure and can perform well when their structures are modified or when hybridization is implemented with local search to increase the accuracy of identifying solutions. Such inherent flexibility in its structure has encouraged researchers to use and test GAs in combination with different strategies.

A hybrid optimization method (GLS) based on GA with local search is proposed in the present study. GLS is built on the basis of the multi-step crossover fusion (MSXF) method proposed by Yamada (2003). MSXF is usually used to solve combinatorial problems. With this method, neighbourhood structure and distance in the search space are utilized. MSXF is especially useful while being implemented with local search because it exploits a good starting point for the subsequent local search. The structure of neighbourhood is very important while working with MSXF since local search always known consuming large amount of time. Stand for that reason; reducing the size of neighbourhood possibly could decrease the amount of computational time. In the present work, neighbourhood structure modifications have been made with intention to increase the effectiveness of the algorithm in producing good solutions. Furthermore, the large solution search space problem encountered by the GA is reduced by applying an iterative

scheduling method. The simulations' results show the sustainability of this GA in solving JSSP.

1.2 Problem Statement

Previous studies that implemented GAs with local search frame works have failed to focus on the size of JSSPs and their relation to selected parameters (Omar, 2008). As already known, numerous parameters are involved in the application of GAs; examples include the selected population size, selection operators, probability of implementing crossover, and importance of mutation operators. The quality of the solutions obtained with GAs is dependent on the selection of the best parameters (Suguna & Thanushkodi, 2010) to prevent premature convergence and to ensure diversity in the search space. Hence, studying the behaviour of these parameters setting in various problem sizes is necessary, for developing efficient algorithms to solve JSSPs.

Neighbourhood search, which is a local search technique, is embedded in the GA structure to handle the problem of premature convergence and to escape from the local optima to find superior solutions. However, the application of local search with a GA initially requires the definition of a neighbourhood structure on the set of feasible solutions. The set of neighbours of a solution is defined as a set of solutions that differ only by small changes. Defining the neighbourhood structure is very important because it determines the way through which the solution space is navigated and the computational time for finding the best solution.

According to Hurink (1998), the computational time for finding the best neighbour or an improving neighbour is proportional to the size of the neighbourhood. Nevertheless, in a large neighbourhood, a considerable number of possibilities can be tapped into to

change current solutions. Under this condition, reaching a high quality solution becomes highly likely. In practice, alternatives that lead to better results are only achievable in computational tests. Therefore, the comparison of two different types of neighbourhood structures, namely, the active critical block neighbourhood (ACBN) and the structure proposed by Nowicki and Smutnicki (NS), is very necessary in the GLS algorithm.

1.3 Objectives

The main objectives of this study are as follows:

1) To propose a hybrid method (GLS) by combining GA and local search, so as to increase the efficiency of the GA in searching for optimal solutions for JSSPs. The methods include the following:
   I. Diversification of the recombination methods by MSXF that utilizes a neighborhood structure and the distance in the search space.
   II. NS Neighborhood search procedure on a critical path in schedule that acts as an exploitation mechanism in searching for the best solutions.
2) To analyse and identify effective parameter settings (population size, selection operator, crossover probability, and mutation) and the ideal neighbourhood structure for GLS that could find high quality solutions within a reasonable time for different sizes of JSSPs.
3) To evaluate the capability of GLS in reducing the total makespan time of jobs using JSSP benchmarks as references, and to compare the results with other JSSP strategies.

1.4 Thesis Scopes

This thesis is devoted mainly to the analysis of the performance of parameter settings for GLS in deterministic and static job shop problems. To this end, developing appropriate parameters and finding a suitable set of parameters that are applicable in GLS are necessary. Therefore, MSXF and the NS neighbourhood structure in the GLS algorithm framework are used in this study. Different problem sizes are used as test problems so that the resulting performance can be derived for a variety of problem sizes. The results of this study can benefit practitioners in other science fields such as management and mathematics.

1.5 Thesis Organization

This thesis covers the effective parameter settings for GLS in the JSSP. The rest of the thesis is organized as follows:

The literature on the JSSP and the related basic concepts, such as notations, formulations, objective functions, and representation, are presented in Chapter 2. The related literature on the different techniques for solving the JSSP is also reviewed here, but a special focus is given on GAs and local search. The GA structure and local search embedded in the GA are introduced. In this chapter, the problem and the algorithms and procedures built for the problem are identified.

The methodology of implementing local search in the GA framework is explained in Chapter 3. The main procedures used and the other partial procedures that need to be performed with them are also analysed here.

The results and the outcome of the comparison are discussed and analysed in Chapter 4. The summary and conclusion of the research are provided in Chapter 5. Future works and research directions are suggested.

CHAPTER 2

LITERATURE REVIEWS

2.1 Introduction

In this chapter, the literature on JSSPs is introduced. Manufacturing terminologies, such as job, operation, machine, processing time, and task, are used to express the conditions and requirements for the problem.

The ideas for solving the JSSP from other researchers are briefly presented. The literature is reviewed in three different sections. An overview of the JSSP is offered in the first section. The methods available for solving the JSSP are discussed in the second section. The work dedicated to GAs is presented in the final section.

2.2 Background of Job Shop Scheduling Problem

Scheduling may be described as sequencing in order to arrange the activities into a schedule. Alternatively, the JSSP is a work scheduling problem that can be described by one or several jobs that must be accomplished by one or several sources. Several operations are involved in each job, and these operations must be performed several times without interruption and using specific sources. (Kumar, S.A., & Suresh, 2009) classified the production systems which include the job shop problem in scheduling and controlling production activities. Entities which pass through the shop are called jobs (products) and the work conducted on them on a machine (resource) is called an operation (task). Where it is applicable, the required technological ordering of the operations on each job is called a routing.

Traditionally, scheduling problems have been viewed as problems in optimization subject to constraints. A variety of useful techniques for solving scheduling problems are involved in scheduling theory. The selection of an appropriate technique is dependent on the complexity of the given problem, the nature of the model, and the choice of a criterion, as well as other factors. The configuration of resources and the behaviour of tasks must be characterized while classifying major scheduling models. For example, if a set of tasks available for scheduling does not change over time, then the system is called static. By contrast, if a set of tasks arises over time, then the system is called dynamic (Gao, Li, Wen, Lu, & Wen, 2015). As stated by the author, static models are traditionally more tractable than dynamic models and have thus been extensively studied.

Other problem classifications are deterministic and stochastic problems. A problem is deterministic when all the parameters, such as processing time, are known and fixed. By contrast, a problem is stochastic when parameters such as processing time are uncertain. All practical scheduling problems are dynamic and stochastic because predicting exactly the arrival of jobs or the breakdown of machines is impossible. Even though most of the problems can be identified as dynamic and stochastic, most existing research is focused on static and deterministic types.

In the work of French (1982), problems in which any randomness is quite clearly insignificant are found to exist. For example, the uncertainty in various quantities is several orders of magnitude less than the quantities themselves. According to the author, the study of dynamic and stochastic problems cannot be accomplished until static and deterministic problems are understood.

2.3 Problem Definition and Notations

Before previous works are reviewed, the criteria used in the study are presented. The notations subject to the JSSP are shown in Table 2.1.

Table 2.1: Notations

Symbol	Description
J_i	Job $i.\,(i = 1,2,\ldots.n)$
M_k	Machine $k\ (k = 1,2,\ldots,m)$
O_{iqk}	q-th operation for job i , J_i, which should be processed on machine k, M_k .
S_{iqk}	Starting time forO_{iqk}.
r_{iqk}	Ready time for O_{iqk}.
P_{iqk}	Processing times for O_{iqk} .
C_{iqk}	The completion time (makespan) of O_{iqk}. $C_{iqk} = S_{iqk} + P_{iqk}$
$n \times m$	n number of jobs need to be processed on m number of machines where n = 1,2,... and m = 1,2,...

In a job shop problem, n number of jobs J, $\{J_i\}_{i=1}^{n}$, need to be processed on m number of machines M, $\{M_k\}_{k=1}^{m}$. Each job J_i consists of an O number of operations that must be processed on each machine M_k in a specific route or order. O_{iqk} is the q-th operation for job i, J_i, which should be processed on machine k, M_k for an uninterrupted processing time period P_{iqk}. The objective is to identify an operating job sequence for each machine while considering the precedence constraints and time to optimally determine a feasible schedule that minimizes the makespan (C_{Max}) (i.e., the cumulative time to complete all operations on all machines). The supposed constraints on static scheduling (Qiu & Lau, 2014) are listed as follows:-

1) Each machine is assigned with a job only once.

2) No machine can hold a job and start another one until the previous job is finished.

3) One operation at a time can be processed on every machine.

4) The delay time for the transfer of a job to a machine is neglected, and the operation allocation for the machine is predefined.

5) No pre-emption is allowed between operations.

6) The operations of different jobs must not have precedence constraints.

7) Neither release times nor due dates are specified.

For further analysis, some assumptions on the structure of the static scheduling problem are listed below:

1) No cancellation is allowed. Each job must be processed to completion.

2) Machines may be idle.

3) Only one operation at a time can be processed by a machine.

4) Machines can never break down and are available throughout the scheduling period.

5) The technological constraints are known in advance and are immutable.

6) No randomness exists. In particular;

   a) The numbers of jobs are known and fixed.

   b) The numbers of machines are known and fixed.

   c) The processing times (duration times) are known and fixed.

Assumption 4th and assumption 6th confine attention to non-random problems that is problem with all the numerical quantities are known and fixed in advance. There is no uncertainty.