Faculty of Cognitive Sciences and Human Development

A COMBINATORIAL OPTIMIZATION TECHNIQUE USING GENETIC ALGORITHM: A CASE STUDY IN MACHINE LAYOUT PROBLEM

Lau Siew Yung

Kota Samarahan
2007

A COMBINATORIAL OPTIMIZATION TECHNIQUE USING GENETIC ALGORITHM: A CASE STUDY IN MACHINE LAYOUT PROBLEM

LAU SIEW YUNG

This project is submitted in partial fulfillment of the requirements for a Bachelor of Science with Honours
(Cognitive Science)

Faculty of Cognitive Sciences and Human Development
UNIVERSITI MALAYSIA SARAWAK
2007

The project entitled A Combinatorial Optimization Techniques Using Genetic Algorithm: A Case Study in Machine Layout Problem was prepared by Lau Siew Yung and submitted to the Faculty of Cognitive Sciences and Human Development in partial fulfillment of the requirements for the Bachelor of Science with Honours (Cognitive Science)

Received for examination by:

_____

(Dr. Teh Chee Siong)

Date:

_____

Grade

ACKNOWLEDGEMENT

I would like to address my high appreciation to my supervisor, Dr. Teh Chee Siong for his guidance and help for this study.

I would also like to express my gratitude to my family and all my friends, for their support and help throughout this study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

A COMBINATORIAL OPTIMIZATION TECHNIQUE USING GENETIC ALGORITHM: A CASE STUDY IN MACHINE LAYOUT PROBLEM

Lau Siew Yung

Solving Combinatorial Optimization Problem is significant as it abounds in our daily lives. However, it is impractical to solve combinatorial optimization problems by exploring all the possible solutions due to combinatorial explosion. Genetic Algorithms (GAs) are a powerful stochastic search in solving optimization problems. However, conventional GAs with binary representation approach cannot be used in solving these kinds of problems. In this study, different crossover and mutation techniques are adapted in GAs so that it suits to combinatorial optimization. In empirical tests, the combinatorial optimization techniques using GAs are able to approximating optimization, which had been justified theoretically in a simple Machine Layout Problem (MLP). Several complex cases of MLP also had been demonstrated and the results of different input parameters are compared.

ABSTRAK

SATU TEKNIK PENGOPTIMUMAN KOMBINATORIAL DENGAN MENGGUNAKAN ALGORITMA GENETIK: SATU KAJIAN KHAS DALAM MASALAH SUSUN ATUR MESIN

Lau Siew Yung

Penyelesaian Masalah Pengoptimuman Kombinatorial adalah penting kerana ianya sering berkaitan dengan kehidupan seharian kita. Namun, menyelesaikan Masalah Pengoptimuman Kombinatorial dengan menjelajah semua penyelesaian berkemumgkinan adalah tidak praktikal disebabkan letupan kombinasi. Algoritma Genetik (GAs) adalah satu teknik pencarian yang amat berkuasa dalam menyelesaikan masalah pengoptimum. Namun, GAs yang konventional dengan pendekatan perwakilan perduaan tidak dapat digunakan dalam menyelesaikan masalah pengoptimum berkombinasi. Dalam projek ini, teknik-teknik penyilangan and mutasi disuaikan dalam GAs supaya ianya bersesuaian dalam menyelesaikan Masalah Pengoptimum Berkombinasi. Dalam experimen empiric, teknik-teknik Pengoptimuman Kombinatorial menggunakan GAs bermampu untuk mencapai pengoptimuman, di mana ianya telah dibukti secara teoritikal dalam satu kajian kes Masalah Susun Atur Mesin (MLP) yang ringkas. Beberapa kes MLP yang lebih komplexs juga telah didemostrasikan dan keputusannya yang menggunakan input parameter yang berlainan dibandingkan.

CHAPTER 1
INTRODUCTION

1.0 Background

Combinatorial Optimization Problems are problems that have finite number of feasible solutions. This kind of problems abounds in our everyday life. Solving combinatorial optimization problems is important particularly in the field of engineering design because it concerns with the use of limited resources and tends to increase the efficiency and productivity (Gen & Cheng, 1997). Thus, the studies on combinatorial optimization problems using various kinds of heuristic search are becoming widespread. Some examples of combinatorial problems are the knapsack problems, quadratic assignment problem, traveling salesman problems (TSP), machine layout problem (MLP), facility layout problem (FLP), and etc.

Theoretically, the optimal solutions to combinatorial optimization problems can be determined by simply examining all the possible combinatorial solutions. However, it is only suitable for simple combinatorial optimization problems. When the problems are getting complex with the increase of the number of possible combination, combinatorial explosion may occur, where there will be an enormous number of feasible solutions. Thus, it is not practical to

solve the combinatorial problems in such a way especially in the field of industry and engineering design which requires high efficiency and rapid production.

Genetic algorithms (GAs) are one of the powerful stochastic optimization techniques, because GAs can find the best solution in a short period of time. GAs are search methods based on principles of natural selection and genetics (Holland, 1975). GAs attempt to mimic the biological evolution process and natural selection for discovering the good solutions. GAs have shown the great potential in solving optimization problems particularly in the industrial engineering areas.

Generally, there are two basic operation in GA, which are genetic operation and evolution operation. Genetic operation includes the reproduction stage, which are crossover and mutation while evolution operation includes the selection stage. However, conventional GAs always deal with binary representation approach, where the solutions are encoded into bit string. Thus, in solving combinatorial optimization problems, this approach will result in invalid solution after undergoing the conventional crossover and mutation operation.

1.1 Problem Statement

Conventional GAs which are binary-coded, encounter with the problems of redundancy when the legal codes are subject to finite discrete set (Herrera, Lozano & Verdegay, 1998). This binary approach may cause the codes out of the domain set. Consider a value $x_i \in S$, where $1 < S_i < 10$. In this case, four bits are needed to code x_i. Thus, the legal range of the binary-coded set is from 0001 to 1001. Then, if crossover and mutation operation are applied to these codes, the codes, for instance 1010 or 1111 would be out of domain set. Thus, conventional GAs are not applicable to combinatorial optimization problems which have finite set of feasible solutions.

Besides, the conventional crossover and mutation techniques such as

one-point crossover, two-point crossover and uniform crossover cannot be applied to GAs in solving combinatorial optimization problems as these operators will generated useless individuals with repeated elements and at the same time missing certain elements in the individuals (Ferreira, 2002). All these situations are not allowed in combinatorial solutions. For example, in traveling salesman problem (TSP), the set of *x* represents the sequence of the cities to be traveled.

Thus, in this case, if the solution consists of repeated cities to be traveled, it would be illegal as each city can only be traveled once, as shown in Figure 1.1. Therefore, conventional crossover and mutation techniques cannot be applied to combinatorial optimization problems.

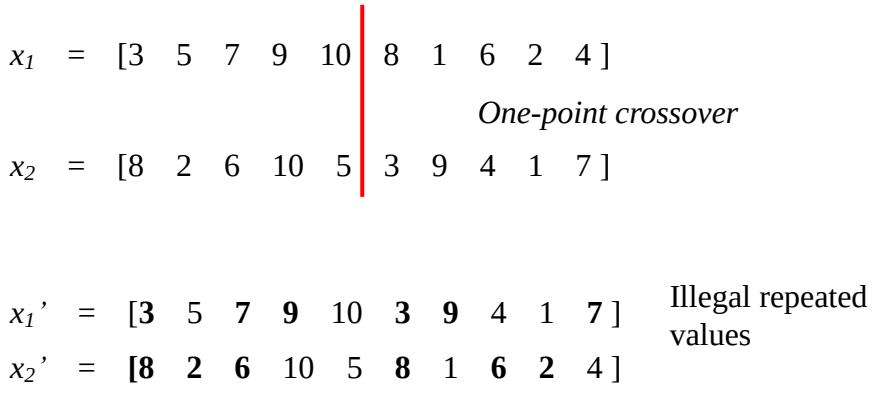


Figure 1.1: Illustration of Illegal repetition using conventional GAs in solving Combinatorial Optimization Problem

1.2 Objectives

1.2.1 General Objective:

The general objectives of this study are to develop a system for solving Machine Layout Problem (MLP) by combinatorial optimization techniques using GA and to conduct empirical study on the combinatorial optimization techniques using GA.

1.2.2 Specific Objectives:

The specific objectives are:

- To study the combinatorial optimization techniques using GA, which is to study various crossover and mutation techniques for combinatorial optimization problems
- To develop a system for Machine Layout Problem (MLP) using Visual Basic 6.0.
- To conduct experiments with various input parameter, such as different population size, number of machine, parent selection techniques, crossover and mutation techniques, crossover probability and mutation probability.

1.3 Methodology

There are five main phases of methodology in this study. These phases are reviewing on combinatorial optimization problems, conventional GAs, crossover and mutation techniques for combinatorial problems, and develop a system for MLP. Finally, a series of experiments will be conducted systematically and the results will be demonstrated.

CHAPTER 2
LITERATURE REVIEW

2.0 Combinatorial Optimization Problems

"*Combinatorial optimization problems are concerned with the efficient allocation of limited resources to meet desired objectives when the values of some or all of the variables are restricted to be integral"* (Hoffman, n.d.).

Combinatorial optimization problems have constraints on basic resources which limit the possible feasible solutions. However, the alternative feasible solutions might be "burst" as the number of combination increases. Thus, solving combinatorial optimization problems is to find the best solutions among all the alternatives (Nemhauser & Wolsey, 1988).

There are many well-known instances of combinatorial optimization problems, such as Knapsack Problem (Martello & Toth, 1990), Quadratic Assignment Problem (Burkard & Bonninger, 1983; Kaku& Thompson, 1986), Traveling Salesman Problem (TSP) (Grefenstette et al., 1985), Film-Copy Deliverer Problem (Cheng & Gen, 1993), etc.

2.1 Examples of Combinatorial optimization problem

2.1.1 0-1 knapsack problem

Generally, knapsack problem can be described as:

> Given items of different values and volumes, find the most valuable set of items that fit in a knapsack of fixed volume.

Mathematically, the 0-1 knapsack problem can be formulated as:

Suppose there are n kinds of item. Each item x_j has the value of p_j and weighted by a weight w_j. The maximum weight that we can carry in the fixed volume of knapsack is C.

$$\text{maximize} \quad \sum_{j=1}^{n} p_j x_j.$$

$$\text{s.t.} \quad \sum_{j=1}^{n} w_j x_j \leq c, \qquad x_j = 0 \text{ or } 1, \quad j = 1, \ldots, n.$$

(Martello, S. & Toth, P., 1990)

2.1.2 Traveling Salesman Problem (TSP)

TSP is a classic combinatorial optimization problem. The description of TSP is as follows (Bui & Colpan, 2005):

Suppose a salesman has to travel to n cities and then return to the initial city. Given the distance between any two cities (the cost of the direct edge connect any two cities), the salesman has to find the shortest path whereby he can only visit every city exactly once in his traveling. The set of edges that the salesman used is called *Hamiltonian tour* and the sum of all the edge cost is called the *tour cost*. TSP aims to find the order of cites such that the *Hamiltonian tour cost* is minimized. TSP can be represented by a complete undirected graph, G = (V, E), with V = vertex which represent the set of cities and E = set of edges connect the vertex V.

TSP also subjected to the representation adopted and the genetic operators that can be applied. Thus, in TSP, efforts are more to:

1. Encoding of a solution to a proper representation
2. Applied proper genetic operators and avoid illegality
3. prevent premature converge

2.2 Computational complexity theory

Combinatorial Optimization Problem is subjected to combinatorial complexity. Computational complexity theory describes the inherit difficulty in providing a scalable algorithms for hard computational problems. The theory also describes the challenge of an algorithm to limit the running time and the memory requirements when the size of the input to an algorithm increases (Wikipedia, 2007). Thus, heuristic approach is important tin searching for the sub-optimal solutions for Combinatorial Optimization Problem.

2.3 Definition of Genetic Algorithms

Genetic algorithms (GAs) are search methods based on principles of natural selection and genetics. (Holland, 1975).

"*Genetic algorithms are based on a biological metaphor: They view learning as a competition among a population of evolving candidate problem solutions. A 'fitness' function evaluates each solution to decide whether it will contribute to the next generation of solutions. Then, through operations analogous to gene transfer in sexual reproduction, the algorithm creates a new population of candidate solutions.*" (George, 2002)

2.4 Origin of Genetic Algorithms

GAs are developed by Bremermann, however, GAs are popularized by

Holland who study adaptation in nature and applied it to GAs (Holland, 1975). GAs are one of the Evolutionary Algorithms. Evolutionary algorithms consists of three main areas, which are GAs, Evolution strategies (ES) and Evolutionary Programming. GAs become popular because of its robustness in searching for the optimal solution in many optimization problems. Over many generations, natural populations evolve according to the principles of natural selection, which is the survival of the fittest. This principle stated clearly by Charles Darwin in *The Origin of Species*. The highlight point of Darwin in the principles of natural selection is as follows (Fang, 1994):

1. Each individual tends to pass its traits to its offspring.
2. However, individuals produced are with different traits by nature.
3. The fittest individuals, which are those with the most favorable traits tend to produces more offspring than those individuals with unfavorable traits. As a result, the whole population converges towards favorable traits.
4. Over many generations, the variation which accumulated produces a population of new species with favorable traits which suit them to the environment.

GAs try to mimic the process of natural selection by Darwin to evolve a favorable solutions to real world optimization problems. The details of GAs will be discussed in Chapter 3.

CHAPTER 3
CONVENTIONAL GENETIC ALGORITHMS

3.0 Introduction to Genetic Algorithms

Genetic Algorithms (GAs) are one of the evolutionary algorithms that are very powerful as an optimization technique to solve various types of optimization problem. GAs are inspired by the theory of evolution by Charles Darwin which simulate the natural process of evolution, such as inheritance, mutation, selection and crossover or recombination. GAs use a "survival of the fittest" techniques and are varied until a good result are obtained. The uniqueness of GAs is that it is a robust search method, which can search effectively in a large or poorly-understood search space by only requiring a little information. In other sense, GAs can achieve the optimization "greedily".

Genetic Algorithms (GAs) represent a set of initial solutions of an optimization problem in *population*. Each solution is represented as a *chromosome*, which is the individual in the population of an optimization problem in a form of chromosome, where all chromosomes are individuals in a population. The chromosomes *evolve* to breed new population through iteration. Each iteration represents a *generation*. The new chromosomes generated are called *offspring* while the chromosomes that generate the offspring are called

parent. In generation of new *population*, the chromosomes undergo two basic genetic operations, which are *crossover* and *mutation.* Crossover involves merging two chromosomes at the same generation while mutation involves altering one *gene* of chromosome. At each generation, the chromosomes are evaluated based on a *fitness function*. The chromosomes with the best fitness value will be selected to breed a new population in next generation. The chromosomes with low fitness value will be eliminated. This is according to the principle of natural selection in evolution by Charles Darwin. Generation after generation, the solutions will converge to an optimum solution.

3.1 How Genetic Algorithms Work?

Generally, there are four basic steps in GAs, which are initialization, evaluation, selection and reproduction. The basic outline in words is as follows:

1. Initialize and encode a random population of chromosomes. This is called the "current population"
2. Evaluate each chromosome's fitness in the current population.
3. Produce an intermediate generation by stochastically selecting current population chromosomes according to fitness. These will be parents of the next generation.
4. Apply crossover and mutations operators to pairs of and/or single chromosomes in the intermediate generation, thus producing a new generation of chromosomes. This is now the current population.
5. Repeat step 2 to 4 until it meets the stopping criteria.

Let P(t) and C(t) be the parents and offspring in the generation t. A basic algorithm of GAs is shown in Figure 3.1 (Gen & Cheng, 1997) and the basic outline is shown in Figure 3.2:

```
Begin
    t ← 0
    initialize P(t);
    evaluate P(t);
    while (not termination condition)
    do
        Reproduction : Crossover and mutation on P(t) to yield C(t);
        evaluate C(t);
        select P(t+1) from P(t) and C(t);
        t ← t + 1;
    end
end
```

Figure 3.1: Basic Algorithm of Genetic Algorithms

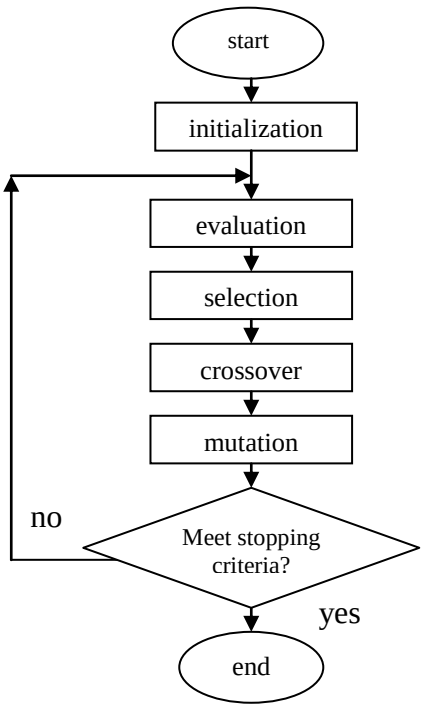
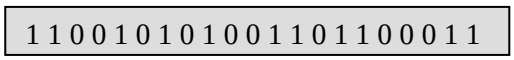


Figure3.2: Basic outline of Genetic Algorithms

3.1.1 Initialization

During initialization, initial population is formed by the randomly generated solutions. The solutions are encoded into a set of chromosomes. A chromosome is represented as a binary string such as



chromosome

Conventional genetic algorithms generate the population randomly by covering the whole range of possible solutions or entire search space. However, the solutions may contain the "seed" where the optimal solutions are likely to be found.

3.1.2 Evaluation

Each individual in the population is evaluated to determine how fit each individual is to a problem. The measure of fitness is problem-specific and is calculated via a fitness function, *f*. The higher the fitness of an individual, the better is the individual as the solution to the problem.

3.1.3 Selection

During selection, the individual solutions are evaluated via a fitness function. The individual solutions with higher fitness value will be selected. In other words, the solutions are said to be fitter. The solutions, which are less fit, will be eliminated. This will keep the population size not to diverse unlimitedly and prevent premature convergence on poor solutions. Roulette wheel selection and Tournament selection are among the popular selection methods.

3.1.3.1 Roulette Wheel Selection (Holland, 1975; Goldberg, 1989)

In this selection mechanism, each individual in the population is given a chance to be selected as the parents in next generation. Their chances to be selected are based on their fitness, which means the fitter the individual is, the