

A Genetic Algorithm for Finding Minimal Wavelength on WDM Ring

Der-Rong Din

Department of Computer Science and Information Management
Hung-Kuang Institute of Technology
Taichung 433, Taiwan R.O.C.
deron@sunrise.hkc.edu.tw

Meng-Kai Tsai and S. S. Tseng

Department of Computer and Information Science
National Chiao-Tung University
Hsinchu 300, Taiwan R.O.C.
mktsai.cis90g@nctu.edu.tw, sstseng@cis.nctu.edu.tw

Abstract

We consider the routing and wavelength assignment (RWA) problem on WDM ring networks without wavelength conversion. When the physical network and required connections are given, RWA is the problem to select a suitable path and wavelength among the many possible choices for each connection such that not any paths using the same wavelength pass through the same link. This problem has been proven to be the NP-hard problem. In the paper, a genetic algorithm (GA) is proposed to solve it. Experimental results indicate that GA is robust for this problem.

Keyword: genetic algorithm, routing and wavelength assignment, integer programming, WDM ring.

1 Introduction

All Optical network based on wavelength division multiplexing (WDM) using the wavelength routing technique is considered as a very promising approach for the realization of future large bandwidth networks [15], [17]. WDM technology is used to accommodate several wavelength channels on a fiber. This technology could enhance the line capacity of networks. Moreover, since the large bandwidth network requires not only transmission line capacity enhancement but also cross-connect node processing capability enhancement, WDM should be used in combination with wavelength routing [10][9].

In wavelength routing, data signals are carried on a unique wavelength from a source node to a destination node passing through nodes where the signals are optically routed and switched without regeneration in the electrical domain. When a physical network is given and connections among the nodes in the network are required, we must establish an optical path

(light path) with a dedicated wavelength for each required connection. The routing and wavelength assignment (RWA) problem is to select suitable paths and wavelengths among the many possible choices for the required connections. To avoid collision, no two paths using the same wavelength pass through the same link. Generally, the traffic pattern can be considered in two categories. One is static traffic the other is dynamic traffic. In the case of static traffic, all the connection requirements are given and the objective is typically to minimize the number of used wavelengths [3][7][11][16][17][9].

In the case of dynamic traffic, connection requirements arrive and depart one by one in a random manner and the objective is typically to minimize the call blocking probability [3][5][15]. By practical limitations on the transmission technology, the number of available wavelengths on a fiber is restricted. So, a good solution to RWA problem is important to increase the efficiency of WDM networks. Many studies related to the problem RWA have been performed. On the topological aspects, many studies concentrated on mesh network [1][2][3][8][10][12][13][16]. The key advantage of mesh network is the ability to utilize network resources in the most efficient manner. The ring networks may not be as efficient, but they may have many other advantages: simple routing policy, simple control and management of network resources, simple hardware system, and simple protection from network failures. Moreover, ring networks are the predominant topology for current MAN/interoffice networks, and are expected to be the first topology to be used for WDM networks in real world. Recently, several studies on ring networks have been performed [5][7][11][15][17]. But the researchers have concentrated on the development of heuristic algorithms. Wavelength conversion is the ability to convert the data on one wavelength to another wavelength.

Moreover, optical wavelength converters are still laboratory curiosities [14]. In ring networks, RWA without wavelength conversion is known to be NP-hard [4]. In this paper, we consider RWA on ring networks without wavelength conversion and with static traffic. Following the integer-programming formulation proposed by [9], a genetic algorithm is proposed to solve this problem.

Since finding an optimal solution to this problem is NP-hard [4], and that an exact search for optimal solutions is impractical due to exponential growth in execution time. Moreover, traditional heuristic methods and greedy approaches should trap in local optima. Genetic algorithms (GA) have been trusted as a class of general-purpose search strategies that strike a reasonable balance between exploration and exploitation. Genetic algorithm proposed by John Holland [6] has been constructed as robust stochastic search algorithms for various optimization problems. GA searches by exploiting information sampled from different regions of the solution space. The combination of crossover and mutations helps GA escape from local optima. These properties of GA provide a good global search methodology for the RWA problem. In this paper, we propose a genetic algorithm for optimal finding the minimal number of required wavelengths on ring network without wavelength conversion.

This paper is organized as follows. In Section 2, we describe the integer programming formulation of the RWA problem. The background of Genetic Algorithm is discussed in Section 3. In Sections 4 and 5, we describe our genetic algorithm for the optimal solution of the RWA problem. In Section 6, we give our experimental results. Finally, conclusions are given in Section.

2 Problem Formulation

Consider a ring network G , which has n nodes, indexed from 1 to n in the clockwise direction. Two nodes may require some connections between them, and the connections are established by paths (*lightpaths*) using specific wavelengths. When all the required connections are given, RWA is the problem to select paths on G for all required connections with the wavelength assignment such that no two paths using the same wavelength pass through the same link. The objective is to minimize the number of required wavelengths.

First, we introduce some notations. Let V be the set of nodes, $V = \{1, 2, \dots, n\}$; E be the set of undirected links, $E = \{(1, 2), (2, 3), \dots, (n-1, n), (n, 1)\}$; W be the set of available wavelengths; and M be the set of selected pairs of nodes, $M = \{(s_1, d_1), (s_2, d_2), \dots, (s_{|M|}, d_{|M|})\}$. The connection between two nodes (s_k and d_k in V) of a selected pair is denoted as $c_k = (s_k, d_k)$, where c_k in M and $s_k < d_k$.

The number of connections required between s_k and d_k for c_k in M is denoted as r_k , assumed to be a positive integer. In the ring network, the routing path of a connection (s_k, d_k) from source node s_k to d_k has exactly two paths: the one is the clockwise and the other is the counter-clockwise. Let p_k^c be the path for connection $c_k = (s_k, d_k)$ having clockwise-direction and p_k^r be the path for connection $c_k = (s_k, d_k)$ having counterclockwise direction. The set of possible paths for all c_k in M is denoted by $P = \cup_{c_k \in M} \{p_k^r, p_k^c\}$. Then, the set of connections whose clockwise and counterclockwise direction paths pass edge $e \in E$ are denoted by M_e^c and M_e^r [9].

With the above notation, the RWA problem can be formulated as the following integer program [9]:

$$\text{Objective : } \min \sum_{w \in W} y_w \quad (1)$$

s. t.

$$\sum_{w \in W} (x_{wc}^k + x_{wr}^k) = r_k, \text{ for all } c_k \in M \quad (2)$$

$$\sum_{w \in M_e^c} x_{wc}^k + \sum_{w \in M_e^r} x_{wr}^k \leq y_w, \text{ for all } w \in W, e \in E \quad (3)$$

$$x_{wc}^k, x_{wr}^k \in \{0, 1\}, \text{ for all } c_k \in M \quad (4)$$

$$y_w \in \{0, 1\}, \text{ for all } w \in W \quad (5)$$

The binary decision variable $x_{wc}^k = 1$ if p_k^c using wavelength w is selected to establish clockwise connection k , $x_{wc}^k = 0$ otherwise. Similar, the binary decision variable $x_{wr}^k = 1$ if p_k^r using wavelength w is selected to establish counterclockwise connection c_k , $x_{wr}^k = 0$ otherwise. The binary decision variable $y_w = 1$ if wavelength is used, $y_w = 0$ otherwise. The objective is to minimize the number of wavelengths used. Constraints (2) mean that every required connections must be established. Constraints (3) ensure that the paths using wavelength can be selected only when wavelength is used and at most one path using wavelength passes edge. We assume that the number of available wavelengths is large enough to establish all the required connections.

Example 1. Consider the graph shown in Fig. 1. There are five nodes and links in ring network G ; each fiber between two nodes can provides 8 wavelengths, i.e., $W=8$. Twelve pairs of nodes are requested to be established ($|M| = 12$). A possible assignment of Fig. 1 for $r_k = 1$ is shown in Fig. 2. Observation from Fig. 2 shown that six wavelengths are used to assign connections.

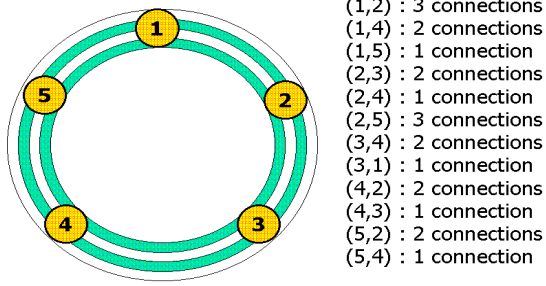


Figure 1: Ring network with five nodes and twelve connections.

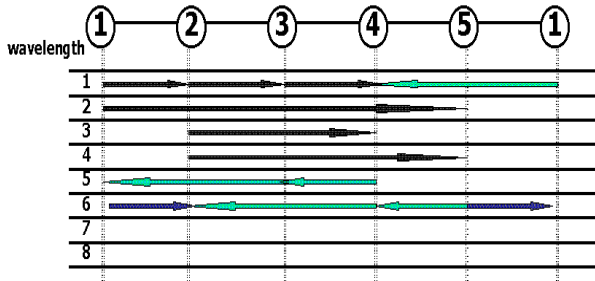


Figure 2: A possible assignment of Example 1 for $r_k=1$.

3 Concept of GA

The search space in GA is composed of possible solutions to the problem. A solution in the search space is represented by a sequence of 0s and 1s. This solution string is referred as a chromosome in the search space. Each chromosome has an associated objective function called the *fitness*. A good chromosome is the one that has a high/low fitness value, depending upon the nature of the problem (maximization/minimization). The strength of a chromosome is represented by its *fitness value*. Fitness values indicate which chromosomes are to be carried to the next generation. A set of chromosomes and associated fitness values is called the *population*. This population at a given stage of GA is referred to as a *generation*. The general GA proceeds as follows:

Genetic Algorithm()

```

Begin
  Initialize population;
  while (not terminal condition) do
    Begin
      choose parents from population; /* Selection */
      construct offspring by combining parents; /* Crossover */
      optimize (offspring); /* Mutation */
    if suited (offspring) then

```

```

      replace worst fit (population) with better offspring;
      /*Survival of the fittest */
    End;
  End.

```

There are three main processes in the while loop for GA:

- (1) The process of selecting good strings from the current generation to be carried to the next generation. This process is called *selection/reproduction*.
- (2) The process of shuffling two randomly selected strings to generate new offspring is called *crossover*. Sometimes, one or more bits of a chromosome are complemented to generate a new offspring. This process of complementation is called *mutation*.
- (3) The process of replacing the worst performing chromosomes based on the fitness value.

The population size is finite in each generation of GA, which implies that only relatively fit chromosomes in generation (i) will be carried to the next generation ($i + 1$). The power of GA comes from the fact that the algorithm terminates rapidly to an optimal or near optimal solution. The iterative process terminates when the solution reaches the optimum value. The three genetic operators, namely, selection, crossover and mutation, are discussed in the next section.

3.1 Selection / Reproduction

Since the population size in each generation is limited, only a finite number of good chromosomes will be copied in the *mating pool* depending on the fitness value. Chromosomes with higher fitness values contribute more copies to the mating pool than do those with lower fitness values. This can be achieved by assigning proportionately a higher probability of copying a chromosome that has a higher fitness value[?]. Selection/reproduction uses the fitness values of the chromosome obtained after evaluating the objective function. It uses a biased roulette wheel[?] to select chromosomes, which are to be taken in the mating pool. It ensures that highly fit chromosomes (with high fitness value) will have a higher number of offspring in the mating pool. Each chromosome (i) in the current generation is allotted a roulette wheel slot sized in proportion (p_i) to its fitness value. This proportion p_i can be defined as follows. Let Of_i be the actual fitness value of a chromosome (i) in generation (j) of g chromosomes, $Sum_j = \sum_{i=1}^g Of_i$ be the sum of the fitness values of all the chromosomes in generation j , and let $p_i = Of_i / Sum_j$.

When the roulette wheel is spun, there is a greater chance that a better chromosome will be copied into the mating pool because a good chromosome occupies a larger area on the roulette wheel.

3.2 Crossover

This phase involves two steps: first, from the mating pool, two chromosomes are selected at random for mating, and second, crossover site c is selected uniformly at random in the interval $[1, n]$. Two new chromosomes, called *offspring*, are then obtained by swapping all the characters between positions $c + 1$ and n . This can be shown using two chromosomes, say P and Q . each of length $n = 6$ bit positions

chromosome P: 111|000;
 chromosome Q: 000|111.

Let the crossover site be 3. Two substrings between 4 and 6 are swapped, and two substrings between 1 and 3 remain unchanged; then, the two offspring can be obtained as follows:

chromosome R: 111|111;
 chromosome S: 000|000.

3.3 Mutation

Combining the reproduction and crossover operations may sometimes result in losing potentially useful information in the chromosome. To overcome this problem, mutation is introduced. It is implemented by complementing a bit (0 to 1 and vice versa) at random. This ensures that good chromosomes will not be permanently lost.

4 Genetic Algorithm for RWA Problem with $r_k = 1$

In this section, we discuss the details of genetic algorithm developed to solve the RWA problem on ring network without wavelength conversion. First, we design the genetic algorithm for the special case with $r_k = 1$; then the algorithm will be extended to solve the case with $r_k \geq 1$ in Section 5. The development of genetic algorithm requires: (1) an encoding scheme, (2) genetic crossover operators, (3) mutation operators, (4) a fitness function definition, (5) a replacement strategy, and (6) termination rules.

4.1 Encoding

Since our problem involves representing of relations between connections and wavelengths, we employ a coding scheme that uses integer numbers. Two chromosomes which are one-dimension arrays are introduced to represent the usage of wavelengths and the assignments of connections. They are *wavelength chromosome* Y and *connection chromosome* C . In the wavelength chromosome Y , $y_i, i=1,2,\dots,W$, the chromosome structure is shown in Fig. 3(a); where $y_i = 1$ if the i^{th} wavelength is used; $y_i = 0$, otherwise. For example,

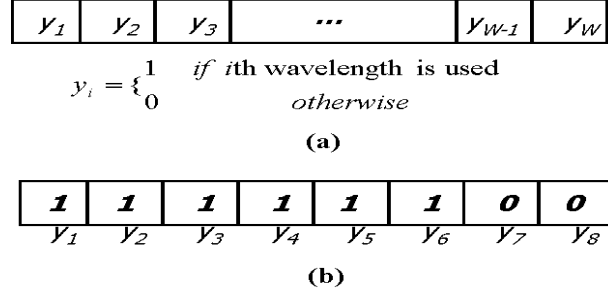


Figure 3: Wavelength chromosome.

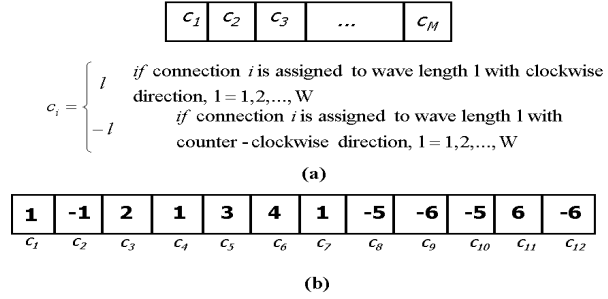


Figure 4: Connection chromosome.

the chromosome of the possible assignment of the Example 1 with $r_k=1$ in Fig. 2 is shown in Fig. 3(b). In the connection chromosome C , $c_j, j=1,2,\dots,|M|$, the chromosome structure is shown in Fig. 4(a); if $c_j = l$ ($1 \leq l \leq W$), the j^{th} connection is assigned to wavelength l in clockwise direction; if $c_j = -l$ ($1 \leq l \leq W$), the j^{th} connection is assigned to wavelength l in counter-clockwise direction. For example, the connection chromosome of the possible assignment of the Example 1 in Fig. 2 is shown in Fig. 4(b).

It is worth noting that if there exist a connection c_i in C such $|c_i| = j$, then $y_j = 1$, for $i = 1, 2, \dots, M$. For example, the relationships of wavelength and connection chromosomes of the possible assignment with $r_k = 1$ of the Example 1 in Fig. 2 is shown in Fig. 5.

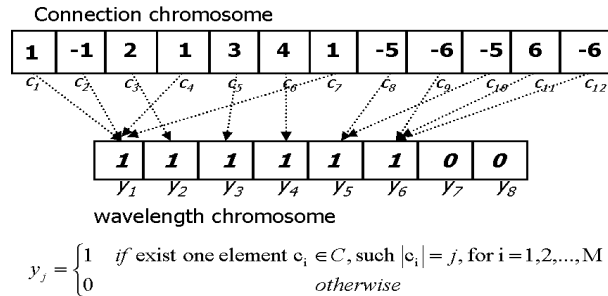


Figure 5: Relationships between wavelength and connection chromosomes.

4.2 Fitness function definition

Generally, genetic algorithms use fitness functions to map objectives to costs to achieve the goal of finding optimally assignments. Assume connection i is assigned to wavelength l ($l=1, 2, \dots, W$), if the route is in clockwise direction, then c_i in the connection chromosome is set to l ; otherwise, c_i is set to $-l$. According to the content of the connection chromosome, the value of wavelength chromosome can be determined. The goal is to minimize the number of required wavelengths. Thus the fitness function is $\sum_{j=1}^W y_j$. A fitness function value is associated with each wavelength chromosome, which is the same as the measure mentioned above. We use the following objective function:

$$\text{minimize } \sum_{w=1}^W y_w. \quad (6)$$

Note that if constraints (2) and (3) in Section 2 are considered with this objective function, we have a complete problem formulation. While breeding chromosomes, the genetic algorithm does not require the chromosome to reflect a feasible solution. Thus, we need to attach a penalty to the fitness function in the event the solution is not constraint-satisfied. For a pair of connections (c_i, c_j) in connection chromosome C , let $\text{conflict}(c_i, c_j) = 1$, if c_i and c_j are assigned to the same wavelength and the routing paths of two connections are overlapped. Then, we can rewrite the formulation above in an unconstrained form:

$$\begin{aligned} \text{minimize } & : \\ \text{cost} & = \sum_{w=1}^W y_w + \alpha \sum_{i=1}^M \sum_{j=1}^M \text{conflict}(c_i, c_j) \end{aligned} \quad (7)$$

where α is the penalty weight. It is worth noting that the value of α should be greater than W . Since the best-fit chromosomes should have a probability of being selected as parents that is proportional to their fitness, they need to be expressed in a maximization form. This is done by subtracting the objective from a large number C_{max} . Hence, the fitness function becomes:

$$\text{maximum } : \quad C_{max} - \left(\sum_{w=1}^W y_w + \alpha \sum_{i=1}^M \sum_{j=1}^M \text{conflict}(c_i, c_j) \right) \quad (8)$$

where C_{max} denotes the maximum value observed, so far, of the cost function in the population. Let cost be the value of the cost function for the chromosome; C_{max} can be calculated by the following iterative equation:

$$C_{max} = \max\{C_{max}, \text{cost}\}, \quad (9)$$

where C_{max} is initialized to zero.

4.3 Conflict-detect algorithm

Consider the following example, assume two connections $c_1 = (1, 2)$ and $c_2 = (1, 4)$ are assigned to wavelength 1 with clockwise direction, then conflict occurred. For each pair of connections, if conflict occurred then a penalty should be attached into the fitness function. Now, the problem is: How to detect the number of conflicts in a connection chromosome? To do this, a *conflict-detection algorithm* should be developed. It is worth noting that there are M^2 pairs of connections in a connection chromosome; every time a new chromosome is generated, the conflict-detection algorithm should be performed to detect the conflict. If the time spent by the conflict-detection algorithm is high, the whole time spent by genetic algorithm will be high. Thus, our goal is to develop an algorithm such that the conflict of each pair of connections can be detected in constant time. To do this, four *bipartite graphs* (termed *conflict graph*) AA , AB , BA , and BB are constructed. The terms A and B meaning that the routing paths of the connections are in clockwise and counter-clockwise directions, respectively. The positions of A and B are the corresponding connections. Therefore, AB meanings that the first connection is in clockwise and the second connection is in counter-clockwise directions. BA meanings that the first connection is in counter-clockwise and the second connection is in clockwise directions. In these conflict graphs, for each connection c_i , construct nodes c_i in the vertex set U and V . Edges of graphs are constructed as follow:

(1) $AA(U, V, E^{AA})$: Assume two connections c_i and c_j use the same wavelength and route in clockwise direction. If two routing paths are overlapped then an edge which connects c_i and c_j is added in E^{AA} .

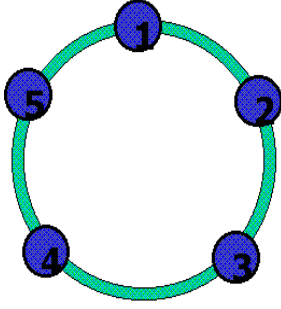
(2) $AB(U, V, E^{AB})$: Assume two connections c_i and c_j use the same wavelength, and c_i is routing in clockwise direction and c_j is routing in counter-clockwise direction. If two routing paths are overlapped then an edge which connects c_i and c_j is added in E^{AB} .

(3) $BA(U, V, E^{BA})$: Assume two connections c_i and c_j use the same wavelength, and c_i is routing in counter-clockwise direction and c_j is routing in clockwise direction. If two routing paths are overlapped then an edge which connects c_i and c_j is added in E^{BA} .

(4) $BB(U, V, E^{BB})$: Assume two connections c_i and c_j use the same wavelength, and route in the counter-clockwise direction. If two routing paths are overlapped then an edge which connects c_i and c_j is added in E^{BB} .

Example 2. Assume four connections $c_1 = (1, 4)$, $c_2 = (2, 4)$, $c_3 = (1, 2)$, and $c_4 = (5, 2)$ should be assigned to a five nodes ring network as shown in Fig. 6. Four conflict graphs can be constructed and shown in Fig. 7.

Each graph is represented by a 0-1 $M \times M$ adjacency matrix and the element of matrix AA is denoted by



$$\begin{aligned} c_1 &= (1, 4), \\ c_2 &= (2, 4), \\ c_3 &= (1, 2), \\ c_4 &= (5, 2) \end{aligned}$$

Figure 6: Ring network and connections of Example 2.

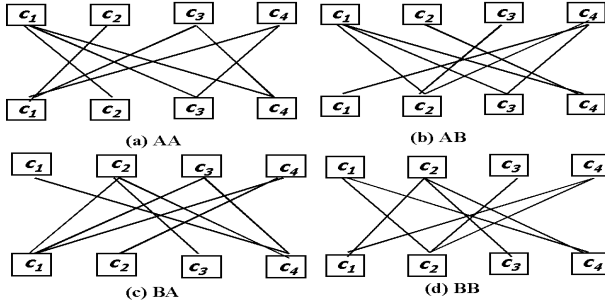


Figure 7: Conflict graphs of Example 2 (a) AA, (b) AB, (c)BA, and (d)BB.

$AA(c_i, c_j)$, and so on. It is easy to find that four conflict graphs can be constructed in $O(M^2)$ time. Once conflict graphs are constructed, the test of whether two connections are conflict or not can be achieved in constant time.

$$\text{Let } test(c_i, c_j) = \begin{cases} AA(c_i, c_j) & \text{if } c_i > 0 \text{ and } c_j > 0 \\ AB(c_i, c_j) & \text{if } c_i > 0 \text{ and } c_j < 0 \\ BA(c_i, c_j) & \text{if } c_i < 0 \text{ and } c_j > 0 \\ BB(c_i, c_j) & \text{if } c_i < 0 \text{ and } c_j < 0 \end{cases}$$

and

$$\text{conflict}(c_i, c_j) = \begin{cases} 1 & \text{if } |c_i| = |c_j| \text{ and } test(c_i, c_j) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Thus, if AA, AB, BA, and BB are given, the conflict of two connections can be known in $O(1)$ time.

4.4 Genetic crossover operator

Five types of crossover operators were used to develop this algorithm: (1) single point crossover (SPC); (2) multiple points crossover (MPC); (3) globally reverse routing operator (GRRO) (4) partially reverse routing operator (PRRO) (5) wavelength exchange operator (WEO)

These operators only operated on the connection chromosome, and the single and multiple point crossovers are the traditional ones described in Section 3.

In *globally reverse routing operator (GRRO)*, randomly selected one chromosome for operation from previous generations and then each value of c_i in C is changed to $c_i * (-1)$, for $i=1, 2, \dots, M$. Assume

parent P_1 is

$$\begin{bmatrix} 5 & -3 & 4 & 3 & 2 & 4 & -1 & -3 & 2 & 4 & 2 & -1 \end{bmatrix}.$$

After operating, parent P_1 is changed to

$$\begin{bmatrix} -5 & 3 & -4 & -3 & -2 & -4 & 1 & 3 & -2 & -4 & -2 & 1 \end{bmatrix}.$$

In *partially reverse routing operator (PRRO)*, first, randomly selected one chromosome for operation from previous generations, then randomly selected z ($1 \leq z \leq M$) connections, each value of c_i in these connections is changed to $c_i * (-1)$. Assume

parent P_1 is

$$\begin{bmatrix} 5 & -3 & 4 & 3 & 2 & 4 & -1 & -3 & 2 & 4 & 2 & -1 \end{bmatrix},$$

$z=4$, and connections c_2, c_5, c_8 , and c_{12} are selected. After operating, parent P_1 is changed to

$$\begin{bmatrix} 5 & 3 & 4 & 3 & -2 & 4 & -1 & 3 & 2 & 4 & 2 & 1 \end{bmatrix}.$$

In *wavelength exchange operator (WEO)*, randomly selected one chromosome for operation from previous generations. Then, randomly selected 2 connections c_i and c_j , and the assigned wavelengths of these two connections are exchanged. Assume

parent P_1 is

$$\begin{bmatrix} 5 & -3 & 4 & 3 & 2 & 4 & -1 & -3 & 2 & 4 & 2 & -1 \end{bmatrix},$$

and connections c_2 and c_{12} are selected.

After operating,

parent P_1 is changed to

$$\begin{bmatrix} 5 & -1 & 4 & 3 & 2 & 4 & -1 & -3 & 2 & 4 & 2 & -3 \end{bmatrix}.$$

4.5 Mutation

Four types of mutations are used to develop the genetic algorithm:

(1) *Single connection mutation (SCM)*: randomly selected a connection, the single connection mutation (SCM) changes the value of the connection to a random integral number l , which is in $[1, W] \cup [-1, -W]$.

(2) *Multiple connection mutation (MCM)*: randomly selected z ($1 \leq z \leq M$) connections, the MCM changes the values of these connections to random integral integers, which are in $[1, W] \cup [-1, -W]$.

(3) *Conflict-free mutation (CFM)*: It is worth noting that if two connections c_i and c_j ($i \neq j$) are assigned to the same wavelength and the routing paths of these connections are not overlapped, then these connections can be assigned to the same wavelength such that the number of required wavelengths can be reduced. This pair of connections are denoted as a *conflict-free pair*. Let CF be the set of pairs (i, j) , $i \in [1, W] \cup [-1, -W]$ and $(c_{|i|}, c_{|j|})$ is conflict-free pair in AA, AB, BA, or BB. Moreover, CF_{AA} , CF_{AB} , CF_{BA} , and CF_{BB} , be

the set of conflict-free pair in AA , AB , BA , BB , respectively. Then $CF = CF_{AA} \cup CF_{AB} \cup CF_{BA} \cup CF_{BB}$ can be constructed by following rules:

Initially, $CF_{AA} = CF_{AB} = CF_{BA} = CF_{BB} = 0$.

- If (c_i, c_j) ($i \neq j$) is a conflict-free pair in AA then $CF_{AA} = CF_{AA} \cup (i, j)$,
- If (c_i, c_j) ($i \neq j$) is a conflict-free pair in AB then $CF_{AB} = CF_{AB} \cup (i, -j)$,
- If (c_i, c_j) ($i \neq j$) is a conflict-free pair in BA then $CF_{BA} = CF_{BA} \cup (-i, j)$,
- If (c_i, c_j) ($i \neq j$) is a conflict-free pair in BB then $CF_{BB} = CF_{BB} \cup (-i, -j)$.

Consider the conflict graphs shown in Fig. 7, we can easily find that $CF_{AA} = \{(2,3), (2,4), (3,2), (4,2)\}$, $CF_{AB} = \{(2,-1), (2,-3), (3,-1), (3,-4)\}$, $CF_{BA} = \{(-1,2), (-1,3), (-3,2), (-4,3)\}$, and $CF_{BB} = \{(2,3), (2,4), (3,2), (4,2)\}$. Thus, $CF = \{(2,3), (2,4), (3,2), (4,2), (2,-1), (2,-3), (3,-1), (3,-4), (-1,2), (-1,3), (-3,2), (-4,3), (2,3), (2,4), (3,2), (4,2)\}$.

To perform the conflict-free mutation, several steps should be performed. First, each conflict-free pair in CF is assigned a distinct integer number from 1 to $|CF|$. Second, randomly selected a number between 1 to $|CF|$ and let (i, j) be the corresponding pair of connections. Four cases of mutation are as follows:

- If $i > 0$ and $j > 0$ then mutate $(c_i$ to $|c_j|$ and c_j to $|c_j|)$ or $(c_j$ to $|c_i|$ and c_i to $|c_i|)$.
- If $i > 0$ and $j < 0$ then mutate $(c_i$ to $|c_i|$ and c_j to $-|c_i|)$ or $(c_i$ to $|c_j|$ and c_j to $-|c_j|)$.
- If $i < 0$ and $j > 0$ then mutate $(c_i$ to $-|c_i|$ and c_j to $|c_i|)$ or $(c_i$ to $-|c_j|$ and c_j to $|c_j|)$.
- If $i < 0$ and $j < 0$ then mutate $(c_i$ to $-|c_i|$ and c_j to $-|c_i|)$ or $(c_i$ to $-|c_j|$ and c_j to $-|c_j|)$.

For the simple example with four connection described in Example 2, a possible connection chromosome is

$$\begin{bmatrix} 5 & -3 & 4 & 3 \end{bmatrix}.$$

Consider the set CF of conflict-free pairs, if the conflict-free pair $(2,3)$ is selected then there are two possible mutations: The one is that the values of c_2 and c_3 and mutated to 4. The other is that the values of c_2 and c_3 are mutated to $|-3| = 3$. Since c_2 and c_3 are conflict-free in the conflict graph AA , the mutation described above may reduce the number of required wavelength. Similarly, the example of the other cases of mutations is shown on Fig. 8.

(4) *Maximal Path Overlapped Mutation (MPOM)*: It is worth noting that if two connections use the same wavelength and the routing paths are overlapped then

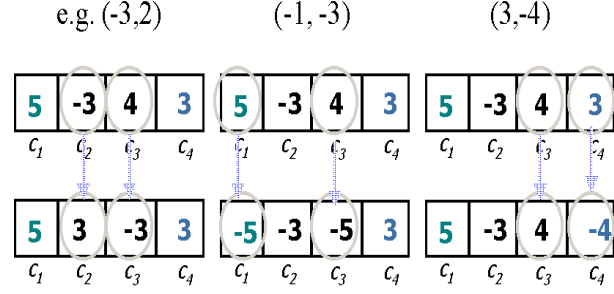


Figure 8: Examples of the conflict-free mutation.

conflict occurred. To avoid conflict, for the pair of connections with maximal path overlap, the assigned wavelength of one of these two connections should be reassigned (or mutated) to another wavelength. This can be achieved by tacking little modification on the conflict graphs AA , AB , BA , and BB . A weight is associated with the edge in two nodes in the conflict graphs AA , AB , BA , and BB . The weight w_{ij} is defined by the length of the overlapped path. For example, if $c_1 = (1,3)$ and $c_2 = (1,4)$ are assigned to the same wavelength and in the same clockwise direction in five nodes ring network, then the length of overlapped path is 2. Two connections with higher weight w_{ij} should have a higher probability of being assigned to the different wavelength.

4.6 Replacement strategy

Initially, assume N_{parent} connection chromosomes are randomly constructed and $N_{population}$ be the number chromosomes to be generated. In the process of selection, $N_{population}/2$ pairs of connections are randomly selected for crossover to generate the new generation of chromosomes. After crossover, chromosomes are sorted according to the fitness function in increasing order, N_{parent} chromosomes with smaller fitness is selected to construct the new generation. The values of $N_{population}$ and N_{parent} will be determined through experiment.

4.7 Termination rules

Execution of GA can be terminated when the number of generations exceeds an upper bound (n_g) specified by the user.

5 Genetic Algorithm for RWA Problem with $r_k \geq 1$

In this section, we extend the genetic algorithm for RWA problem with $r_k = 1$ presented in previous section to solve the generalized one with $r_k \geq 1$. The connection chromosome must be extended to tackle multi-

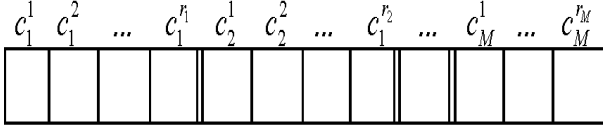


Figure 9: Connection chromosome for multiple requests $r_k \geq 1$.

ple connections requirements. It is easy to find that the wavelength chromosome need not be changed, only the connection chromosome should be changed. The connection chromosome for RWA with multiple connection requests is shown in Fig. 9. The z^{th} ($z \leq r_k$) request of connection i is mapped to connection $\sum_{j=1}^{i-1} r_j + z$ connection of new connection chromosome C^{new} .

The fitness function can be extended to

$$\text{cost} = \sum_{w=1}^W y_w + \alpha \sum_{i=1}^{M'} \sum_{j=1}^{M'} \text{conflict}(c_{l(i)}, c_{l(j)}), \quad (10)$$

minimize :

where $M' = \sum_{k=1}^M r_k$ and $l(i)$ is a function that maps new index in C^{new} to the original index in C .

Four conflict graphs (AA , AB , BA , BB) should be modified by adding links between nodes $c_i \in U$ and $c_i \in V$; since in the case of RWA problem with $r_k \geq 1$, conflict may be occurred in the same connections (c_i, c_i) . If two requests of same connection are assigned to the same wavelength in the same directions. It is worth noting that the crossover and mutation operation in previous section can be used in the case with $r_k \geq 1$.

6 Experimental Results

In order to evaluate the performance of the proposed genetic algorithm, we have implemented the algorithm and applied it to solve problems that were randomly generated. The results of these experiments are reported below. In all the experiments, the implementation was conducted in C, and all the experiments were run on a personal computer (PC) with a Pentium III 1GHZ CPU and 256MB RAM.

We tested our algorithm on two ring networks which have different numbers of nodes (10, 30). For the test, we generated connection requirements between all pairs of two nodes in the networks. The number of required connections for each pair of nodes is one, two, or three with the same probability.

In the following, we will discuss the effects of the various parameters of the genetic algorithm. Let $n=10$, $M=58$, crossover probability=1.0, mutation probability=0.3, population size $N_{population} = 1200$, parent size $N_{parent} = 200$, $W=1000$, $\alpha = W$, and maximum number of generations = 1000.

In Fig. 10, we present the effect of the different types of crossovers, five crossover operators and a randomly selected scheme which is randomly selected these operators, are used in this experiment. We observe that the randomly selected scheme is the best one, and the GPPRO and PRRO is the worst.

In Fig. 11, we present the effect of the different types of mutations, four mutations and a randomly selected scheme which is randomly selected from these operators, are used in this experiment. We observe that the CFM and the randomly selected scheme is the best one, and the MPOM is the worst one.

In Fig. 12, we present the effect of the different number of chromosome in populations ($N_{population}$), where $N_{population}$ is in $\{200, 400, 600, 800, 1000, 1200\}$. Crossover and mutation operators are the randomly selected schemes. We found that $N_{population} = 1000$ and 1200 get the best results, and $N_{population} = 200$ ($=N_{parent}$) is the worst. The CPU time in seconds of different values of $N_{population}$ is shown in Fig.13 and is proportional to the $N_{population}$.

In Fig. 14, we present the effect of the different number of chromosomes in parent (N_{parent}), where N_{parent} is in $\{200, 400, 600, 800, 1000, 1200\}$. Crossover and mutation operators are the randomly selected schemes. We found that $N_{parent} = 200$ and 400 get the best results, and $N_{parent} = 1200$ is the worst. The CPU time in seconds of different values of N_{parent} is shown in Fig.15 and is proportional to the $1/N_{parent}$.

In Fig. 16, we present the effect of the mutation probability on the evolution of the required wavelengths in GA. In this experiment, let $n=30$, $M=938$, crossover probability=1.0, population size $N_{population} = 1200$, parent size $N_{parent} = 200$, $W=1000$, $\alpha = W$, and maximum number of generations = 1000.

We found that when the mutation probability is large (0.60, 0.55), the algorithm leads to both fast convergence and to the global optimal. The CPU time in seconds of different values of mutation probability is shown in Fig.17.

To evaluate the effect of the heuristic mutations and the crossover operators described in Section 4, we simple constructed a simple genetic algorithm called *SGA*. The mutation operation of SGA is the single connection mutation (SCM) and the crossover operator is the single point crossover (SPC). Fig. 18 shows the result of the algorithms (SGA and our EGA). let $n=30$, $M=938$, crossover probability=1.0, population size $N_{population} = 1200$, parent size $N_{parent} = 200$, $W=1000$, $\alpha = W$, and maximum number of generations = 2000. Observe the results shown in Fig. 18, the required wavelengths of EGA is smaller than that of SGA. Moreover, the time spent by our GA (EGA) is less that of SGA. That is, using these crossover and mutation operators in EGA can rapidly decrease the number of required wavelengths and kept it from getting trapped in a local minimum.

7 Conclusions

In this paper, we consider the RWA problem on ring networks without wavelength conversion and with static traffic. Since the problem is formulated as an integer-programming proposed by [9], and the optimal solution of this problem has been found to be NP-hard. A genetic algorithm is proposed to solve this problem and simulation results show that the proposed genetic algorithm is robust for this problem.

In our method, connection and wavelength chromosomes are used to represent the assignment of connection and the usage of wavelength. In the GA method, three general genetic operators - selection, crossover, and mutation - are employed. Five types of operators (SPC, MPC, GRRO, PRRO, and WEO) and four types of mutations (SCM, MCM, CFM, and MPOM) are employed in our genetic algorithm. Experimental results indicate that GA runs efficiently.

8 Acknowledgment

This work was supported in part by the MOE Program of Excellence Research under Grant 90-E-FA04-1-4.

References

- [1] D. Banerjee and B. Mukherjee, "Practical approaches for routing and wavelength assignment in large all optical wavelength routed networks," in Proc. IEEE Infocom'94, June 1994, pp. 110-119.
- [2] C. Chen and S. Banerjee, "A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks," in Proc. IEEE Infocom'96, Apr. 1996, pp. 164-171.
- [3] I. Chlamtac, A. Ganz, and G. Karmi, "Light path communications: An approach to high bandwidth optical WAN's," IEEE Trans. Commun., vol. 40, 1992, pp. 1171-1182.
- [4] T. Erlebach and K. Jansen, "Scheduling of virtual connections in fast networks," in Proc. 4th Parallel Syst. Algorithms Workshop, 1996, pp.11-32.
- [5] O. Gerstel and S. Kutten, "Dynamic wavelength allocation in all-optical ring networks," in Proc. IEEE Int. Conf. Commun., June 1997, pp.432-436.
- [6] J. Holland, *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press (Ann Arbor), 1975.
- [7] D. K. Hunter and D. Marcenac, "Optimal mesh routing in four fiber WDM rings," Electron. Lett., vol. 34, 1998, pp. 796-797.
- [8] K. C. Lee and V. O. K. Li, "A wavelength routing algorithm in wide area all-optical networks," J. Lightwave Technol., 1996, pp. 1218-1229.
- [9] T. Lee, K. Lee, and S. Park, "Optimal routing and wavelength assignment in WDM ring networks," IEEE Journal on Select. Areas in Communication, vol. 18, no. 10, Oct., 2000, pp. 2146-2154.
- [10] N. Nagatsu, S. Okamoto, and K. Sato, "Optical path cross-connect system scale evaluation using path accommodation design for restricted wavelength multiplexing," IEEE J. Select. Areas Commun., vol. 14, 1996, pp. 893-902.
- [11] C. Qiao, X. Zhang, and L. Zhou, "Scheduling all-to-all connections in WDM rings," in Proc. Int. Soc. Opt. Eng., vol. 2219, 1994, pp. 970-979.
- [12] P. Raghavan and E. Upfal, "Efficient routing in all optical networks," in Proc. ACM STOC'94, 1994, pp. 134-143.
- [13] R. Ramaswami and K. N. Sivarajan, "Optimal routing and wavelength assignment in all-optical networks," in Proc. IEEE Infocom'94, June 1994, pp. 970-979.
- [14] R. Ramaswami and K. N. Sivarajan, *Optical Networks-A Practical Perspective*. San Francisco: Morgan Kaufmann, 1998.
- [15] S. Subramaniam and R. A. Barry, "Wavelength assignment in fixed routing WDM networks," in Proc. IEEE Int. Conf. Commun., June, 1997, pp. 406-410.
- [16] N. Wauters and P. Demeester, "Design of the optical layer in multi wavelength cross-connected networks," IEEE J. Select. Areas Commun., vol.14, 1996, pp. 881-892.
- [17] L. Wuttisittikulij and M. J. O'Mahony, "Design of efficient and practical algorithm for wavelength assignment in multi wavelength ring transport networks," in Proc. IEEE GLOBECOM'97, Nov. 1997, pp. 571-575.

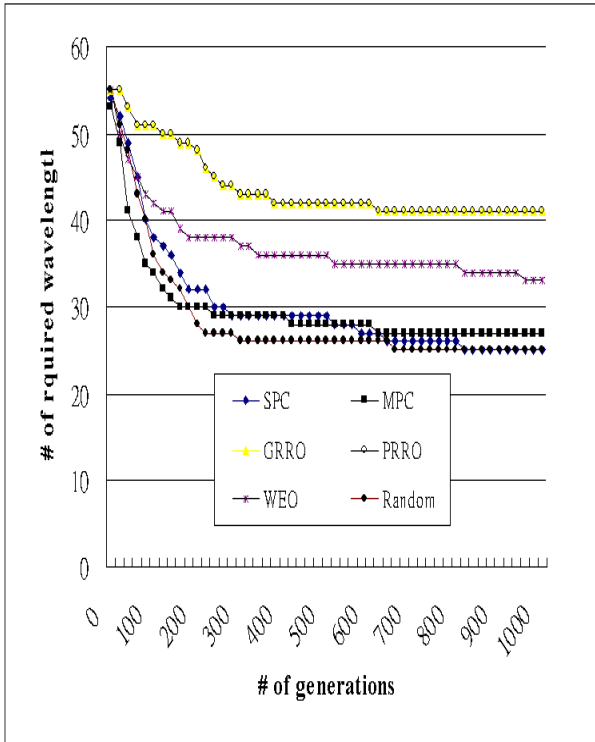


Figure 10: Effect of the different types of crossover operators.

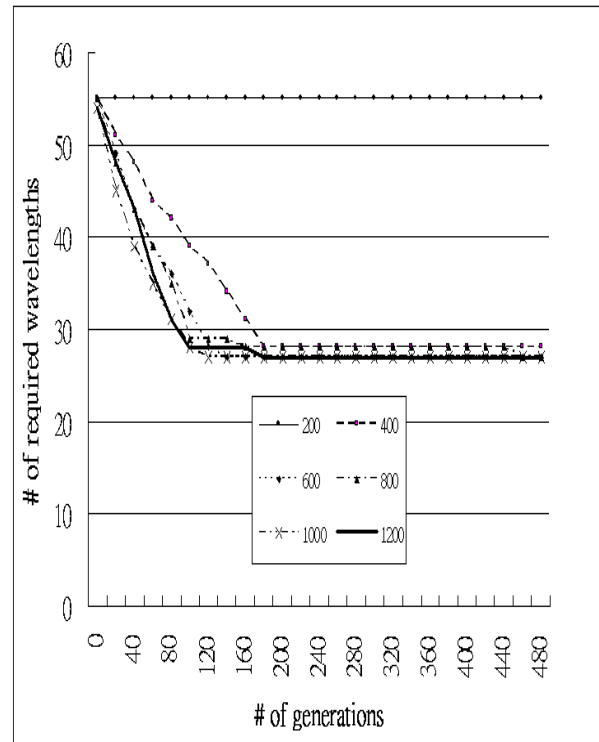


Figure 12: Effect of the different number of chromosomes in population.

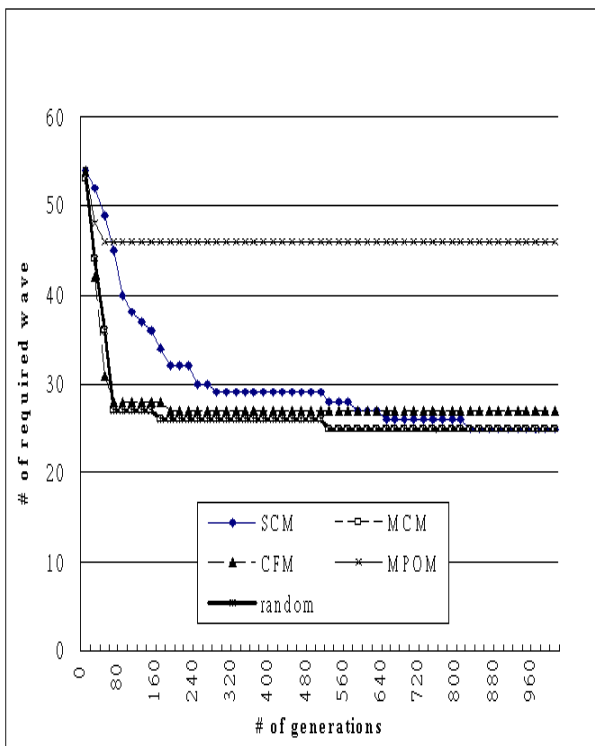


Figure 11: Effect of the different types of mutation operators.

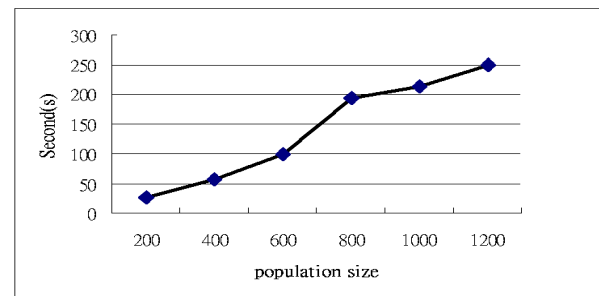


Figure 13: The time spent by the different number of chromosomes in population.

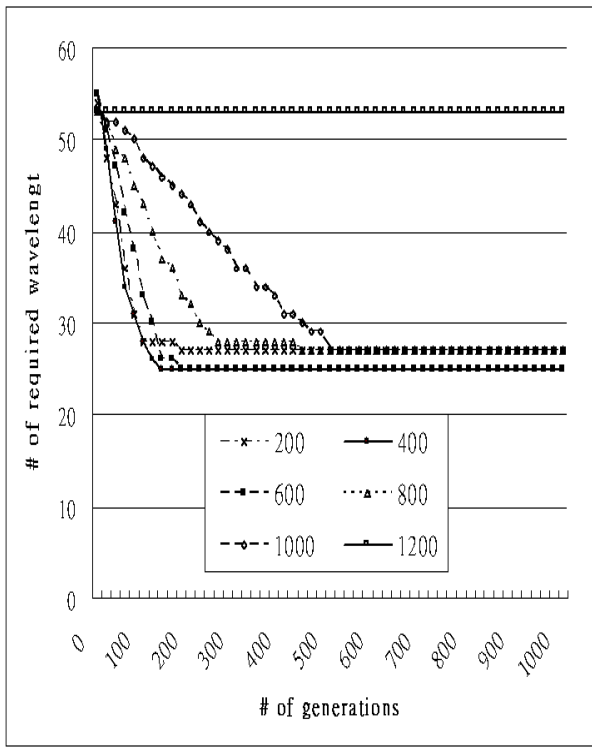


Figure 14: Effect of the different number of chromosomes in parent generations.

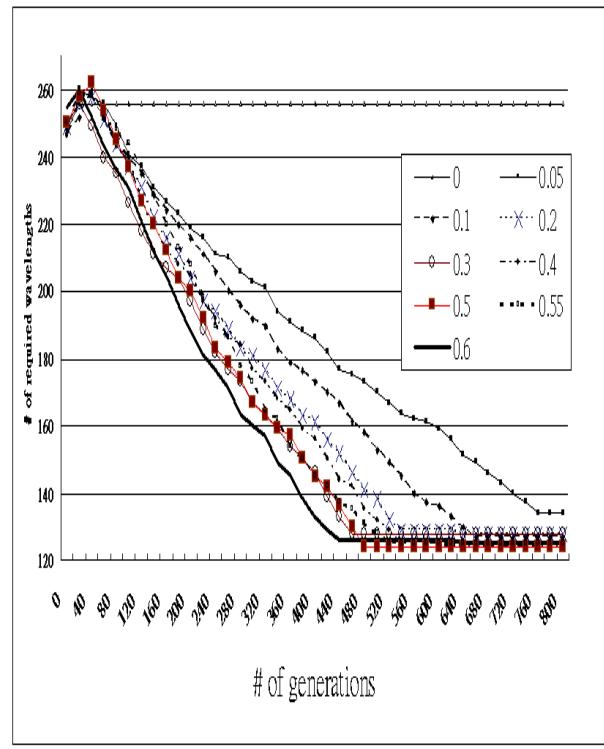


Figure 16: Effect of the different values of mutation probabilities.

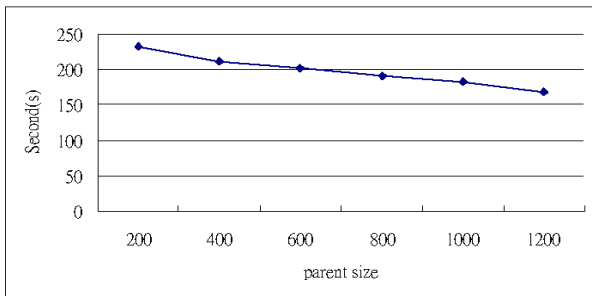


Figure 15: The time spent by the different number of chromosomes in parent generations.

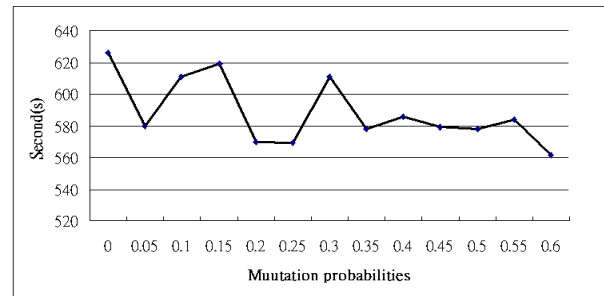


Figure 17: The time spent by the different values of mutation probabilities.

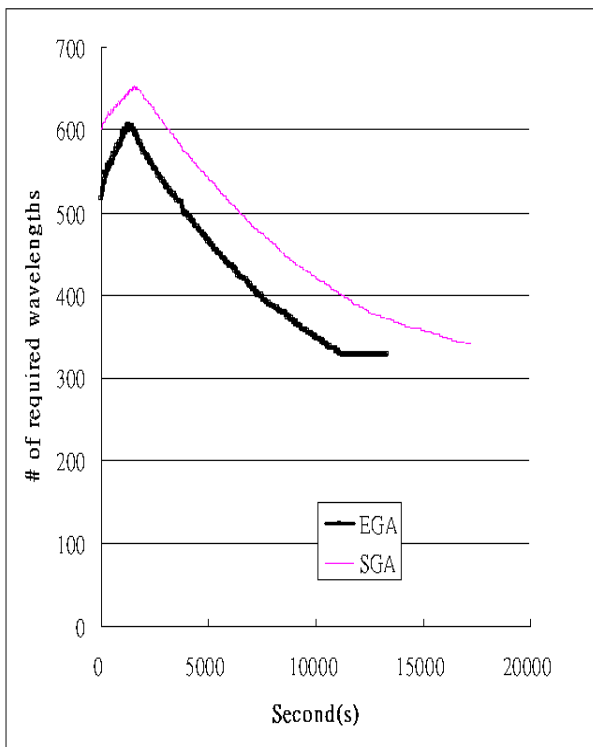


Figure 18: Comparison of SGA and EGA.