A Genetic Algorithm for Network Expanded Problem in Wireless ATM Network

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

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

Abstract

In this paper, we investigate the *network expanded problem* which optimally assigns new adding and splitting cells in *PCS* (Personal Communication Service) network to switches in an *ATM* (Asynchronous Transfer Mode) network. Moreover, the locations of all cells in PCS network are fixed and known, but new switches should be installed to ATM network and the topology of the backbone network may be changed. Given some potential sites of new switches, the problem is to determine how many switches will be added to the backbone network, the locations of new switches, the topology of the new backbone network, and the assignments of new adding and splitting cells in the PCS to switches on the new ATM network in an optimum manner. We would like to do the expansion in as attempt to minimize the total communication cost under budget and capacity constraints. This problem is modeled as a complex integer programming problem, and finding an optimal solution to this problem is *NP-hard*. A genetic algorithm is proposed to solve this problem. The genetic algorithm consists of three phases, *Switch Location Selection Phase*, *Switch Connection Decision Phase*, and *Cell Assignment Decision Phase*. First, in the Switch Location Selection Phase, the number of new switches and the locations of the new switches are determined. Then, Switch Connection Phase is used to construct the topology of the expanded backbone network. Final, Cell Assignment Phase is used to assign cells to switches on the expanded network. Experimental results indicate that the three-phase genetic algorithm has good performances.

keyword: Genetic algorithm, wireless ATM, network expanded problem, cell assignment problem.

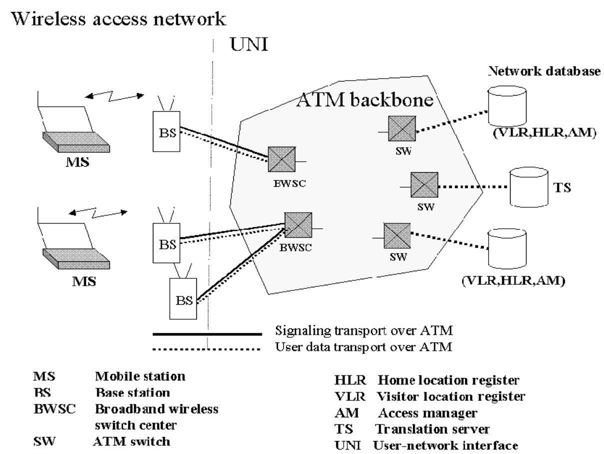

Figure 1: Architecture of wireless ATM PCS.

1 Introduction

The rapid worldwide growth of digital wireless communication services motivates a new generation of mobile switching networks to serve as infrastructure for such services. Mobile networks being deployed in the next few years should be capable of smooth migration to future broadband services based on high-speed wireless access technologies, such as *wireless asynchronous transfer* mode (*wireless ATM*)[1]. In the architecture presented in [1] (as shown in Fig. 1), the base station controllers (BSCs) in traditional PCS network are omitted, and the base stations (BSs or cells) are directly connected to the ATM switches. The mobility functions supported by the BSCs will be moved to the BSs and/or the ATM switches.

In the designing process of PCS network, first, the telephone company determined the global service area

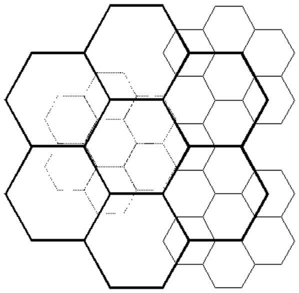

Figure 2: Cell splitting.

according to the usages of the mobile users, and divided the global service area into some smaller coverage areas which are covered by cells. Second, the cellular system and base stations are established and setup, BSs are connected to the switches on the ATM network to form the topology of wireless ATM. This topology may be out of date, since more and more users may use the PCS communication system. Some areas, which have not been covered in the original global service area, may now have mobile users to serve. The services requirement of some areas, which were originally covered by some BSs may be increased and exceeded the capacities provided by the original BSs and switches. Though, the wireless ATM system must be extended so that the system can provide higher quantity of services to mobile users. Two methods can be used to extend the capacities of system and provide higher quantity of services. The first one is: adding new cells to the wireless ATM network so that the non-covered areas can be covered by new cells. The other is: reducing the size of the cells so that the total number of channels available per unit cell and the capacity of a system can be increased. In practice, this can be achieved by using *cell splitting*[8] process. The cell splitting process establishes new BSs at specific points in the cellular pattern and reduces the cell size by a factor of 2 (or more) as shown in Fig. 2.

In this paper, we are given a two-level wireless ATM network as shown in Fig. 3. In the PCS network, cells are divided into two sets. One is the set of cells, which are built originally, each cell in this set has been assigned to a switch on the ATM network (*e.g.*, cells c_1, c_2 are assigned to switch s_1, cells c_3 and c_5 are assigned to switch s_4, and cell c_4 is assigned to switch s_4 in Fig. 3). The other is the set of cells which are newly added (*e.g.*, c_6, c_7, c_8) or established by performing the cell splitting process (*e.g.*, c_9, c_{10}, c_{11}, c_{12}, c_{13}, and c_{14}). Moreover, the locations of all cells in PCS network are fixed and known, but the number of switches in ATM network may be increased. Given some potential sites of new switches, the problem is to determine how many switches will be added to the backbone network, the locations of the adding switches, the connections between the adding switches and other switches, and the assignment of adding and splitting cells

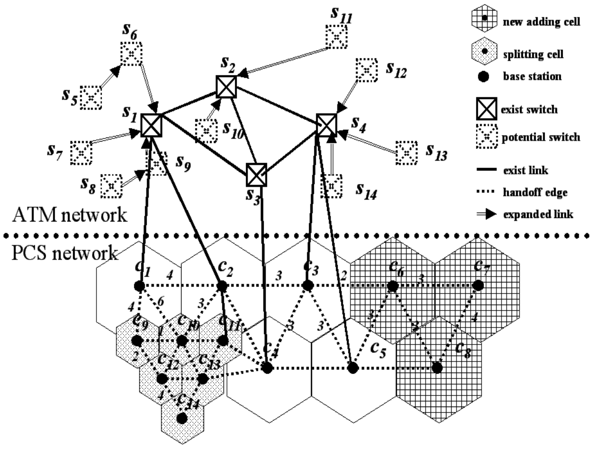

Figure 3: Example of the network expanded problem in the two layers wireless ATM network.

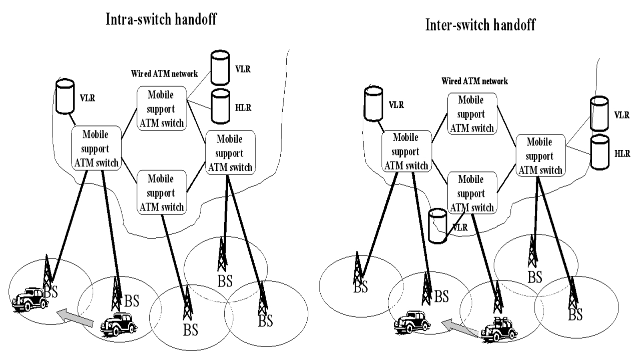

Figure 4: Two types of handoffs occurred in the wireless ATM network.

in the PCS to switches on the ATM network in an optimum manner. We would like to do the extension in as attempt to minimize the total communication cost under budget and capacity constraints.

The total communication cost has two components: one is the cost of handoffs that involve two switches, and the other is the cost of *cabling* (or *trucking*) [3][4][5][6][10]. During the wireless environment, two types of handoffs should be considered in the designing of the network, they are *intra-switch handoff* and *inter-switch handoff* as illustrated in Fig. 4. The intra-switch handoff involves only one switch and the inter-switch handoff involves two switches. The (inter-switch) handoffs that occur between two cells, which connected to different switches, consume much more network resources (therefore, are much more costly) than the intra-switch handoff that occurs between cells, which connected to the same switch[3][4][5][6][10]. Thus, we assume that the cost of (intra-switch) handoffs involving only one switch are negligible. Through this paper, we assume each cell to be connected to only one switch. The budget constraint used to constrain the sum of the switch setup cost, the link setup cost between two switches, and the link setup cost between cells and switches.

For the *cell assignment problem*, Merchant and Sengupta [10] considered the problem that assigns cells to switches in PCS network. They formulated the problem and proposed a heuristic algorithm to solve it so that the total cost can be minimized. The total cost consists of cabling cost and location update cost. The location update cost considered in [10], which depend only on the frequency of handoff between two switches, is not practical. Since switches of the ATM backbone are widely spread, the communication cost between two switches should be included in calculating the location update cost. In [3][4], this model was extended to solve the problem that grouped cells into clusters and assigned these clusters to switches on the ATM network in an optimum manner by including the communication cost between two switches. In [5] and [6], the *extended cell assignment problem* has been investigated and formulated which is assigning the new adding and the splitting cells to the switches on the ATM network so that the total cost can be minimized. In [5][6], the number of adding and splitting cells was not greater than the total remaining capacities provided by the switches of ATM network. That is, no new switch should be added into ATM network, and the topology of the backbone network was not changed in this problem. A simulated annealing and a genetic algorithms have been proposed in [5][6] to solve the extended cell assignment problem, respectively.

In this paper, a more complex problem is considered. Following the objective function formulated in [3][4][5][6], new cells and new switches should be introduced into the two-layer network. In this paper, the locations of new switches, the connections between switches, and the assignment of new and splitting cells should be determined so that the total communication cost can be minimized under budget and capacity constraints. This problem is denoted as *network expanded problem* in wireless ATM environment.

Since finding an optimal solution to this problem is NP-hard, in this paper, a three-phase genetic algorithm is designed to find an approximate solution. The organization of this paper is shown as follows. In Section 2, we formally define the problem. The backgrounds of genetic algorithms are described in Section 3. In Sections 4 and 5, we describe outline and details of the proposed genetic algorithm. The experimental results are presented in Section 5. Final, a conclusion is given in Section 6.

2 Problem Formulation

In this section, we give the formulation of network expanded problem in wireless ATM network. In what follows, we introduce a number of assumptions that are necessary for the proper modeling of the problem.

2.1 Backbone Network Assumption

- Each cell is connected to a switch through a link.
- The switches are interconnected with a specified topology through links.
- The number of cells that can handled by a new switch cannot exceed CAP.
- At most one switch may be installed at a given potential site.
- All links of the current backbone network are kept in place.
- A switch site in the current network is also a switch site in the expanded network.
- The backbone network topologies are preserved in the expanded backbone network.

2.2 Known Information

- The location of the new cells as well as the handoff frequency between cells.
- The potential switch sites.
- The setup cost of switch at a particular site.
- The link setup cost between cells and switches.
- The link setup cost between switches.

Our goal is to find the minimum-cost expanded network subject to all of the above assumption, facts and constraints (described later).

2.3 Mathematical Formulation

Let $CG(C, L)$ be the PCS network, where C is a finite set of cells with $|C|$ and L is the set of edges such that $L \subseteq C \times C$. We assume that $C^{new} \cup C^{old} = C$, $C^{new} \cap C^{old} = \emptyset$, C^{new} be the set of new and splitting cells where $|C^{new}| = n'$, cells in C^{new} have not yet been assigned to switches on the ATM, and C^{old} be the set of original cells where $|C^{old}| = n$. Without loss of generality, we assume that cells in C^{old} and C^{new} are indexed from 1 to n and $n+1$ to $n+n'$, respectively. If cells c_i and c_j in C are assigned to different switches, then an inter-switch handoff cost is incurred. Let f_{ij} be the frequency of handoff per unit time that occurs between cells c_i and c_j, $(i, j = 1, ..., n+n')$ and is fixed and known. We assume that all edges in C are undirected and weighted; and assume cells c_i and c_j in C are connected by an edge $(c_i, c_j) \in L$ with weight w_{ij}, where $w_{ij} = f_{ij} + f_{ji}$, $w_{ij} = w_{ji}$, and $w_{ii} = 0$[3][4][5][6]. Let $G^{old}(S^{old}, E^{old})$ be the currently exist ATM network, where S^{old} is the set of switches with $|S^{old}| = m$, $E^{old} \subseteq S^{old} \times S^{old}$ is the set of edges, s_k, s_l in S^{old}, (s_k, s_l) in E^{old}, and G^{old} is connected. We assume that the locations of cells in CG and switches in G^{old} are fixed and known. The topology of the ATM network $G^{old}(S^{old}, E^{old})$ is known and will be extended to $G(S,E)$. Let S^{new} is the set of potential sites of switches. Without loss of generality, we assume that switches in S^{old} and S^{new} are indexed from 1 to m and $m+1$ to $m+m'$, respectively. We assume that the expanded backbone network should be a connected network, *i.e.*, new switches can be connected to exist switches or another new switches. Let (X_{s_k}, Y_{s_k}) be the coordinate of switch s_k, $s_k \in S^{old} \cup S^{new}$, $k = 1, 2, ..., m+m'$, (X_{c_i}, Y_{c_i}) be the coordinate of cell c_i, $i = 1, .2, ..., n+n'$; and d_{kl} be the minimal communication cost between the switches s_k and $s_l, s_k, s_l \in S$; $k, l = 1, 2, ..., m+m'$.

The total communication cost has two components, the first is the cabling cost between cells and switches, the other is the handoff cost which occurred between two switches. To formulate the total communication cost, let us define the following variables: Let l_{ik} be the cabling cost per unit time between cell c_i switch s_k, $(i = 1, ..., n+n'; k = 1, ..., m+m')$ and assume l_{ik} is the function of Euclidean distance between cell c_i and switch s_k.

Assume the number of calls that can be handled by each cell per unit time is equal to 1 and CAP denotes the cell handling capacity of each new switch $s_k \in S^{new}$, $(k = m+1, m+2, ..., m+m')$. Let Cap_k be the number of remaining cells that can be used to assigned cells to switch $s_k \in S^{old}, (k = 1, 2, ..., m)$. Our goal is to determine the location of the new switches, construct the new topology of the expanded backbone network, and assign cells in C^{new} to switches on G so as to minimize the overall communication cost which is the sum of cabling communication cost and handoff costs per unit time under some constraints. Some variables are defined here and to be used to formulated this problem, let q_k =1, (k = 1, 2, ..., $m+m'$) if there is a switch installed on site s_k; q_k =0, otherwise (as we known, q_k =0, for k= 1, 2, ..., m). Let $setup_k$ be the setup cost of the switch at site $s_k \in S^{new}$, k= 1, 2, ..., $m+m'$ (as we known $setup_k = 0$, for k=1, 2, ..., m). Let $x_{ik} = 1$ if cell c_i is assigned to switch s$_k$; $x_{ik} = 0$, otherwise; where $c_i \in C$, i= 1, 2, ..., $n+n'$, $s_k \in S$, s=1, 2, ..., $m+m'$. Since each cell should be assigned to only one switch, we have the constraint $\sum_{k=1}^{m+m'} x_{ik} = 1$, for $i = 1, 2, .., n+n'$. Further, the constraints on the call handling capacity is as follows: For the new switch s_k,

$$\sum_{i=n+1}^{n+n'} x_{ik} \leq CAP, \ \ k = m+1, m+2, ..., m+m', \quad (1)$$

and for the existing switch s_k,

$$\sum_{i=n+1}^{n+n'} x_{ik} \leq Cap_k, k = 1, 2, ..., m. \quad (2)$$

If cells c_i and c_j are assigned to different switches, then an inter-switch handoff cost is incurred. To formulate handoff cost, variables $z_{ijk} = x_{ik}x_{jk}$, for $i, j, = 1, ..., n+n'$ and $k = 1, ..., m+m'$ are defined in [10]. Thus, z_{ijk} equals 1 if both cells c_i and c_j are connected to a common switch s_k; it is zero, otherwise. Further, let

$$y_{ij} = \sum_{k=1}^{m+m'} z_{ijk}, i, j = 1, 2, ..., n+n'. \quad (3)$$

Thus, y_{ij} takes a value of 1, if both cells c_i and c_j are connected to a common switch; $y_{ij} = 0$, otherwise. With this definition, it is easy to see that the cost of handoffs per unit time is given by

$$Handoff\ \ Cost =$$
$$\sum_{i=1}^{n+n'} \sum_{j=1}^{n+n'} \sum_{k=1}^{m+m'} \sum_{l=1}^{m+m'} w_{ij}(1-y_{ij})q_k q_l x_{ik} x_{jl} D_{kl}, \quad (4)$$

where D_{kl} is the minimal communication cost between switches s_k and s_l on $G(S,E)$.

The objective of the problem is to minimize the total communication cost subject to budget constraint. Thus, together with our earlier statement about the sum of cabling cost and handoff cost, the objective function is :

minimize :

$$\begin{aligned} & Total\ cost \\ = \ & Cabling\ Cost + Handoff\ \ Cost \\ = \ & \sum_{j=1}^{n+n'} \sum_{k=1}^{m+m'} l_{ik} x_{ik} + \alpha \times \\ & \sum_{i=1}^{n+n'} \sum_{j=1}^{n+n'} \sum_{k=1}^{m+m'} \sum_{l=1}^{m+m'} w_{ij}(1-y_{ij})q_k q_l x_{ik} x_{jl} D_{kl} \quad (5) \end{aligned}$$

where α is the ratio of the cost between cabling communication cost and inter-switch handoff cost.

Let e_{kl} be the variable that represents the link status between two switches s_k and s_l. If e_{kl}=1 then there is a link between two switches s_k and s_l $(s_k \in S^{new}, s_l \in S^{old} \cup S^{new})$; e_{kl}=0, otherwise. Let u_{ik} be link setup cost of constructing the connection between cell $c_i, (i = n+1, n+2, ..., n+n')$ and switch $s_k(k = 1, n+2, ..., m+m')$, and assume u_{ik} is the function of Euclidean distance between cell c_i and switch s_k. Let v_{kl} be link setup cost of constructing the connection between switch $s_k, (k = m+1, m+2, ..., m+m')$ and switch $s_l, (k = 1, 2, ..., m+m')$, and assume v_{ik} is the function of Euclidean distance between switch s$_k$ and switch s_l.

The following constraints must be satisfied:

$$EC = \sum_{k=m+1}^{m+m'} q_k \cdot setup_k + \sum_{i=n+1}^{n+n'} \sum_{k=1}^{m+m'} u_{ik} \cdot x_{ik} \cdot q_k$$

$$+\left(\sum_{k=m+1}^{m+m'}\sum_{l=1}^{m+m'} e_{kl}v_{kl}q_kq_l\right)/2 \le Budget \quad (6)$$

$$x_{ik} \le q_k, \ \ for \ \ k = 1, 2, ..., m. \quad (7)$$

$$w_{kl} \le q_k \ \ and \ \ w_{kl} \le q_l, \ \ for \ \ k = 1, 2, ..., m. \quad (8)$$

Example 1. Consider the graph shown in Fig. 3. There are 14 cells in CG which should be assigned to switches in ATM network. In CG, cells are divided into two sets, one is the set C^{old} of cells which are built originally, and cells in C^{old} have been assigned to switches in the ATM network (*e.g.*, $\{c_1, c_2, c_3, c_4, c_5\}$ in Fig. 3). The other is the set C^{new} of cells which are new adding cells (*e.g.*, $\{c_6, c_7, c_8\}$) or splitting cells (*e.g.*, $\{c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}\}$). The edge weight between two cells is the frequency of handoffs per unit time that occurs between them. Ten potential site of switches can be chosen to be the new switches. We assume that the capacity of each switch is 2.

3 Background of Genetic Algorithms

The *Genetic Algorithm (GA)* was developed by John Holland at the University of Michigan[7]. Genetic Algorithms are search techniques for global optimization in a complex search space. As the name suggests, GA employs the concepts of natural selection and genetic. Using past information, GA directs the search with expected improved performance. The concept of GA is based on the theory of adoption in natural and artificial systems[7]. In artificial adaptive systems, adaptation starts with an initial set of structures (possible solutions). These initial structures are modified according to the performance of their solution by using an adaptive plan to improve the performance of these structures. It has been proved by Holland that repeatedly applying this adaptive plan to input structures results in optimal or near optimal solutions [7]. The traditional methods of optimization and search do not fare well over a broad spectrum of problem domains[2]. Some are limited in scope because they employ local search techniques (e.g., calculus based methods). Others, such as enumerative schemes, are not efficient when the practical search space is too large.

3.1 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 0*s* and 1*s*. 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.

3.2 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[2]. Selection/reproduction uses the fitness values of the chromosome obtained after evaluating the objective function. It uses a biased roulette wheel[2] 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.

3.3 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.

3.4 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.

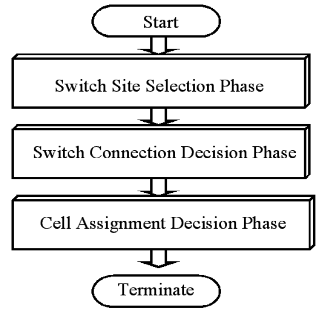

Figure 5: Outline of the genetic algorithm for solving the complex extended cell assignment problem.

4 Outline of Solution Algorithm

In general, the network expanded problem is a multi-constraints optimization problem. The designs are used to find optimal location of the switches, topological connections and cells assignment such that the total communication cost is minimized and yet satisfies the budget constraint and maximum capacity constraint. In fact, the problem is NP-hard and for the practical problem with a modest number of nodes, only approximate solutions can be obtained through heuristic algorithms. In this paper, considering that the problem is largely governed by the specification that is formulated in Section 2, we divide the problem into three subproblems. Thus, the whole complexity of the problem is broken down and driven only by the constraints and the requirements of cost. This can be simplified into three sets of design variables which corresponds to the number of optimization phases. In this way, each optimization level has the main core and a GA cycle, with similar architecture. This similarity can reduce the complexity of the system design.

The genetic algorithm proposed to solve the network expanded problem consists of three phases and the outline is shown in Fig. 5. They are *Switch Site Selection Phase*, *Switch Connection Decision Phase*, and *Cell Assignment Decision Phase*.

A detail description for each level of optimization is given in the following subsections.

4.1 Switch Site Selection Phase

In the Switch Site Selection Phase, we use genetic algorithm to determine the number of new switches to be setup and the locations of the new switches. During this phase, we assume that the connections between switches and the assignments of cells to switches are randomly generated. The subproblem can be formulated as follows:

Given CG, G^{old}, CAP, Cap_k, $Setup_k$, u_{ik}, v_{kl}, $Budget$, f_{ij}, and potential sites S^{new}
Minimize Total cost of the two-layer wireless ATM network
Subject to $EC \leq Budget$;
$\sum_{i=n+1}^{n+n'} x_{ik} \leq CAP$,
$k= m+1, m+2, ..., m+m'$;
$\sum_{i=n+1}^{n+n'} x_{ik} \leq Cap_k$,
k=1, 2,...,m;
$\sum_{k=1}^{m+m'} x_{ik} = 1$,
i=1, 2, ..., $n+n'$; and
$G(S,E)$ must correspond to a connected topology.
Determine q_k, k=m+1, m+2, ..., $m+m'$

4.2 Switch Connection Decision Phase

In the Switch Connection Decision Phase, we assume the number and the locations of the new switches are fixed and known, genetic algorithm is used to determine the connections between switches, and expanded the current exist network to a connected network. During this phase, we assume that the assignments of cells are randomly generated. The subproblem can be formulated as follows:

Given CG, $G^{old} \cup S^{new}$, CAP, Cap_k, u_{ik}, v_{kl}, $Budget$, f_{ij}, and location of new switches
Minimize Total cost of the two-layer wireless ATM network
Subject to $EC \leq Budget$
$\sum_{i=n+1}^{n+n'} x_{ik} \leq CAP$,
$k= m+1, m+2,..., m+m'$;
$\sum_{i=n+1}^{n+n'} x_{ik} \leq Cap_k$,
$k = 1, 2, ..., m$;
$\sum_{k=1}^{m+m'} x_{ik} = 1$,
i=1, 2, ..., $n+n'$;and
$G(S,E)$ must correspond to a connected topology.

Determine e_{kl}, k=m+1, m+2, ..., $m+m'$;
l=1,2, ...,$m+m'$.

4.3 Cell Assignment Decision Phase

In the Cell Assignment Decision Phase, we assume the topology of the expanded backbone network is fixed and known, genetic algorithm is used to determine the assignment of cells in C^{new} to switches in S.

Given $CG, G, CAP, Cap_k, u_{ik}, v_{kl}, Budget, f_{ij}$
Minimize *Total cost* of the two-layer wireless ATM network
Subject to $EC \leq Budget$
$\sum_{i=n+1}^{n+n'} x_{ik} \leq CAP$,
$k= m+1, m+2, ..., m+m'$
$\sum_{i=n+1}^{n+n'} x_{ik} \leq Cap_k$, $k = 1,2,...,m$
$\sum_{k=1}^{m+m'} x_{ik} = 1$, i=1, 2, ..., $n+n'$
$G(S,E)$ must correspond to a connected topology
Determine x_{ik}, i=n+1, n+2, ..., $n+n'$;
k=m+1, m+2, ..., $m+m'$

5 Genetic Algorithm for network expanded Problem

In this section, we discuss the details of GA developed to solve the network expanded problem. The development of GA requires: (1) a chromosomal coding scheme, (2) initial population generation, (3) a chromosome adjustment procedure, (4) a genetic crossover operator, (5) mutation operators, (6) a fitness function definition, (7) a replacement strategy, and (8) termination rules.

5.1 Chromosomal coding

To solve this problem, two two-level genes are introduced as illustrated in Figs. 6 and 7. In these encoding schemes, the activation of the low-level gene is governed by the value of the high-level gene. To indicate the activation of the high-level gene, an integer "1" is assigned for each high-level gene that is being ignited were "0" is for turning off. When "1" is signaled, the associated low-level gene due to that particular active high-level gene is activated in the lower level structure. It should be noticed that the inactive gene always exist within the chromosome even when "0" appears. This architecture implies that chromosome contains more information than that of the conventional GA structure. Hence, it is called *Hierarchical Genetic Algorithm (HGA)* [9].

To solve the network expanded problem in wireless ATM network, three types of genes, known as *switch-location gene*, *switch-connection gene*, and *cell-assignment gene*, are introduced. The switch-location gene in the form of bits decide the activation or deactivation of the corresponding new switches. The switch-connection gene defines the link connections between new switches and another (new or old) switches. The cell-assignment gene defines the assignment of cells to switches. For example, in Fig. 6, the switch-connection s_5 and s_6, with switch-location gene signified as "0" in the corresponding sites, are not being activated. Furthermore, the switch s_5 and s_6 will not appear in the content of the cell-assignment gene (as seen in Fig. 7).

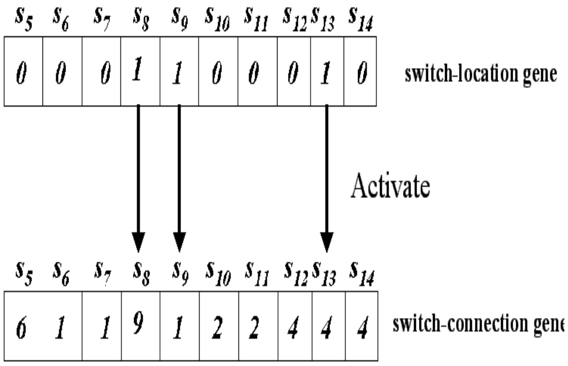

Figure 6: Two level genes used to represent the relation between the locations of switches and the connections between switches.

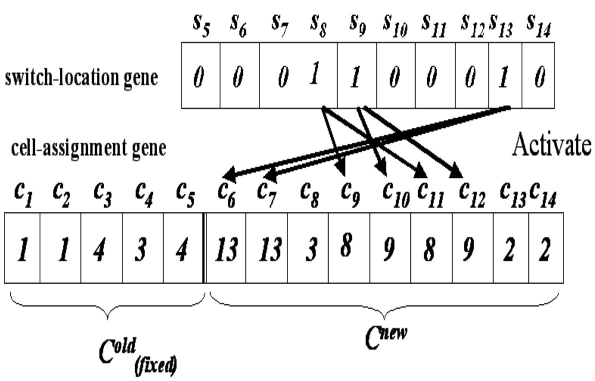

Figure 7: Two level genes used to represent the relation between the locations of switches and the assignments of cells.

The detail information of three types of genes are described as follows:

- *Switch-location gene (SL)*: Since there are m' potential sites for the choosing of news switches, a binary encoding method is used to represent whether the site is selected or not. A binary array $SL[m+1, ..., m+m']$ is used to represent the choose. If $SL[k] = 1$ $(m+1 \leq k \leq m+m')$ then a new switch is located at potential site s_k; $SL[k] = 0$, otherwise. For example, assume s_8, s_9, and s_{13} are selected be the new switches, the switch-location gene of the example shown in Fig. 3 is shown in Fig. 8(a).

- *Switch-connection genes (SC)*: A positive integral encoding method is used to describe the connections between switches. Since the existing ATM network is connected. Thus, the only thing that we have to do is to keep the information of how the new switch is connected to another switches. A integral array $SL[m+1, ..., m+m']$ with maximum positive integer number $m+m'$ is used to represent the connections of new switches to another switches. If $SC[k] = l (m+1 \leq k \leq m+m', 1 \leq l \leq m+m')$ then there is a link between switch s_k and s_l. For example, assume s_8, s_9, and s_{13} are selected to be the new switches, a possible switch-connection gene of the example shown in Fig. 3 is shown in Fig. 8(b).

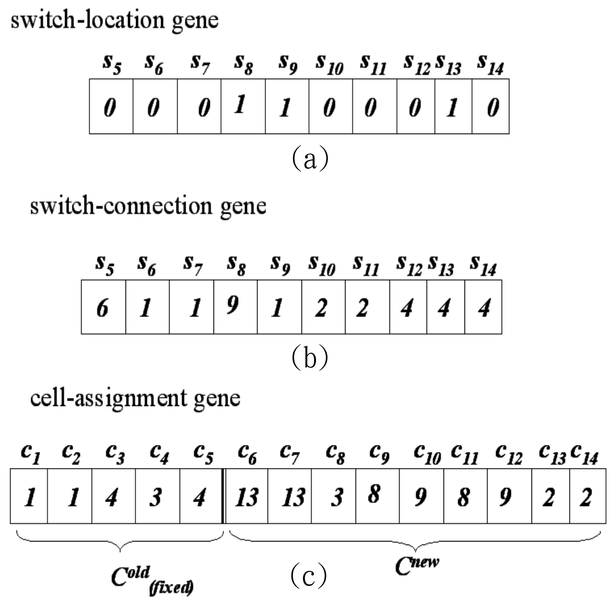

Figure 8: Three types of genes used to encode the example shown in Fig. 1, (a) switch-location genes, (b) switch-connection gene, and (c) cell-assignment genes.

- *Cell-assignment gene (CA)*: Since the cell assignment subproblem involves representing connections between cells and switches, we employ a coding scheme that use positive integer numbers. The cell–assignment gene is shown in Fig. 8(c), where the *ith* cell belongs to the $CA[i]$-th switch. For example, a possible cell-assignment gene of the example shown in Fig. 3 is shown in Fig. 8(c). It should be noticed that, the cell-assignment gene can be divided into two sets, the first set of cells which represents the assignment of cells in C^{old} is fixed in running of GA. Thus, the first set of cells can be ignored since it is unchanged during experiments.

  Furthermore, switch-location, switch-connection and cell-assignment genes are used in Switch Site Selection Phase; the value of each gene is randomly generated by a random number in the evolutionary process. In running Switch Connection Decision Phase, we assume that Switch Location genes are fixed, and the other two genes are variable. In running Cell Assignment Decision Phase, switch-location gene and switch-connection gene are fixed, and cell-assignment gene is variable.

5.2 Initial Chromosome Generating Procedure

As shown in Figs. 6 and 7, since switch-location gene (SL) be the high-level gene, the content of SL does effect the contents of the switch-connection gene (SC) and cell-assignment gene (CA). Once the location of new switches have been selected, the connections of switches should be updated according to the selection of new switches. To do this, let *switch-pool for connection (SPC)* be the set of numbers be the indices of the switch which will be used to determine SC. Thus $SPC = \{1, 2..., m\} \cup \{i \mid SL[i] = 1, m+1 \leq i \leq m+m'\}$. To generate the connection of new switches, the value of array SC are randomly selected from SPC. This process will guarantee that new switch will be connected to an exist switch in S. Similarly, the assignments of cells to switches should be updated according to the selection of new switches. Let *switch-pool for assignment (SPA)* be the set of numbers be the indexes of the switch can be used in choosing of CA. For switch s_k, $1 \leq k \leq m$, Cap_k "k" are inserted into set SPA. For switch c_k, if $m+1 \leq k \leq m+m'$ and $SL[k] = 1$, CAP "k" are inserted into set SPA. For example, assume s_8, s_9, and s_{13} are selected be the new switches, the SPC is $\{1, 2, 3, 4, 8, 9, 13\}$ (assume CAP=2) and the SPA is $\{2, 2, 3, 8, 8, 9, 9, 13, 13\}$. To assign cells to switches, the value of element in array CA is randomly selected a number from SPA and removed it from SPA.

5.3 Chromosome Adjustment Procedure

Since we assume that the expanded backbone network must be a connected network, but from the observation of the initial chromosome generating procedure described in previous subsection, there is no guarantee to generate a connected network. Moreover, the switch capacity constraint may be violated. These events are adjusted by means of the chromosome adjustment procedure described below.

- For the switch-connection gene: we have to test whether the expanded backbone generated by initial chromosome generating procedure is a connected network or not. If yes then there is no need to change; otherwise, a modified algorithm should be performed to change it to a connected network. Since the original existing network is connected, all we have to do is to test whether each new switch can reach to one of switches in S^{old} by repeatedly traversing the path to next switch as indicated in the switch-connection gene. If the expanded backbone network was not a connected network then there exist a cycle of new switches. To modified the backbone network to a connected one, for each cycle, arbitrarily select a new switch in this cycle and connect it to a randomly selected switch in S^{old}, then this will break the cycle and connect all switches in this cycle to a switch in the original existing backbone network. This test and modify process can be done in $O(m')$ time.
- For the cell-assignment gene: we have to test whether the cell assignment violates the capacity constraints or not. The Chromosome Adjustment Procedure[5] can be used to generate a constraint-satisfied assignment. Since the initial population of our solution method is randomly generated and the operator of GA sometimes generates a chromosome which does not represent a feasible assignment. This event is adjusted by means of the chromosome adjustment procedure described below: Let n_k be the number of cells assigned to switch s_k, k=1, ...,$m + m'$; three types of switches are defined:
(1) *over-switch: if* $n_k > Cap_k$;
(2) *saturated-switch: if* $n_k = Cap_k$;
(3) *poor–switch: if* $n_k < Cap_k$.

Switches are grouped into sets S_{over}, S_{sat}, and S_{poor} for over-switch, saturated-switch and poor-switch, respectively. To change infeasible chromosomes into feasible ones, chromosome adjustment procedure is repeatedly used to reassign the cells from over-switches to poor-switches until all over-switches become saturated-switches. We have following algorithm:

Algorithm: Chromosome Adjustment Procedure.

Step 1: Switches are grouped into sets S_{over}, S_{sat}, and S_{poor} according to the number of cells being assigned to it; without loss of generality, switches are renumbered such that $n_k \geq n_{k+1}$, $k = 1, ..., m + m' - 1$.

Step 2: Construct a set SP (switch pool) of number of switches by putting Cap_k-n_k "k" into SP, if $n_k < Cap_k$, for $k = 1, 2, ..., m + m'$.

Step 3: Randomly generate a number as the adjustment point z in $[n+1, n+n']$, while S_{over} is nonempty do Step 4.

Step 4: If $l = v_z \in S_{over}$, then randomly select and remove a number (say q) from SP; reassign cell c_z to switch s_q, i.e., set the value of v_z to q; decrease the n_l by 1; if $n_l = Cap_l$ then move switch s_l from S_{over} to S_{sat}. Otherwise, increase z by 1, if $z > n + n'$ then $z = n + 1$.

5.4 Genetic crossover operator

The traditional single point crossover was used in the genetic algorithm. The details are described in the follows:

- In the Switch Site Selection Phase, the single point crossover is randomly selecting two Switch-Location genes (say SL_1 and SL_2) for crossover from previous generations and then by using a random number generator, an integer value i is generated in the range $[m+1, m+m']$. This number is used as the crossover site. To create new offspring, first, all characters between i and $m+m'$ of two parents are swapped and children C_1 and C_2 are generated. Then, the following SL gene and CA gene are regenerated according to the contents of the C_1 and C_2.

- In the Switch Connection Decision Phase, switch-location gene is fixed. The single point crossover is randomly selecting two switch-connection genes (say SC_1 and SC_2) for crossover. After performing crossover operation, the resulting genes may represent a disconnected networks. Thus, the chromosome adjust procedure as illustrated in Section 4.3 must be applied to change the children SC genes into feasible genes. It should be notice that the cell-assignment gene does not change.

- In the Cell Assignment Decision Phase, switch-location gene and switch-connection gene are fixed. The single point crossover is randomly selecting two cell-assignment genes (say CA_1 and CA_2) for crossover. After performing crossover operation, the resulting genes may violate the capacity constraint, Thus, chromosome adjust procedure proposed in previous subsection should be applied to change the CA genes into an feasible one.

5.5 Mutation

The traditional single cell mutating operation, which mutates a cell in genes at a time, is used to develop of this algorithm.

- In the Switch Site Selection Phase, the mutating operator changes a randomly selected cell in switch-location gene from "0" to "1" or from "1" to "0". The value of the mutated cell in the SL is randomly assigned to a switch. If the resulting backbone network is not connected then the chromosome adjustment procedure should be applied. Final, the Cell-assignment gene is regenerated according to the content of the SL.

- In the Switch Connection Decision Phase, the mutating operator changes a randomly selected cell in switch-connection gene from current value to an integer which is randomly selected from the SPC. If the resulting backbone network is not a connected network then the chromosome adjustment procedure should be applied. It should be noticed that the switch-location gene and the cell-assignment gene need not change.

- In the Cell Assignment Decision Phase, switch-location gene and switch-connection genes are fixed. Four types of mutations can be applied to CA gene. It is worth noting that After mutation, the chromosome may became a infeasible one, thus, the Chromosome Adjustment Procedure must be applied to the chromosome.

  (1) *Traditional Mutation* (**TM**): randomly select a cell of vector v_i, where i in $[n+1, n+n']$ and transform to a random number between 1 to $m+m'$.

  (2) *Multiple Cells Mutation* (**MCM**): randomly select two random numbers k, l between 1 and $m+m'$, transform the value of cells in C^{new} which value is k to l and l to k.

  (3) *Heaviest Weight First Preference* (**HWFP**)[3]: Since the handoff cost involving only one switch is negligible, two cells can be assigned to the same switch so as to reduce the handoff cost between these cells. Two cells with higher weight w_{ij} should have a higher probability of being assigned to the same switch. Thus, if we consider two connected cells c_i and $c_j \in C$, then the probability of mutation from v_i of cell c_i to the value v_j of cell c_j is as follows:

$$P_{(i,j)} = \frac{w_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{degree(c_i)} w_{ij}},$$

  where $degree(c_i)$ is the number of cells connected to cell c_i in CG.

  (4) *Minimal Cabling Cost First Preference* (**MCCFP**)[3]: To reduce the cabling costs between cells and switches, we prefer to assign each cell to the nearer switch rather than the farther one. Cell c_i and switch s_k with lower cabling cost l_{ik} should result in higher probability that c_i will be assigned to s_k. Thus, if we consider the randomly selected cell c_i, then the probability of mutation from v_i of cell c_i to the value v_k is :

$$P_{(i,k)} = \frac{L_{max} - l_{ik}}{\sum_{l=1}^{m}(L_{max} - l_{il})},$$

  where $L_{max} = \max_{l=1}^{m}\{l_{il}\}$.

5.6 Fitness function definition

Generally, genetic algorithms use fitness functions to map objectives to costs to achieve the goal of an optimally designed two-level wireless ATM network. If cell c_i is assigned to switch s_k, then v_i in the CA genes is set to k. Let $d_{(v_i,v_j)}$ be the minimal communication cost between switches s_k and s_l in G. An objective function value is associated with each chromosome, which is the same as the fitness measure. We use the following objective function:

minimize

$$\sum_{j=1}^{n+n'} \sum_{k=1}^{m+m'} l_{iv_i} + \alpha \sum_{i=1}^{n+n'} \sum_{j=1}^{n+n'} \sum_{k=1}^{m+m'} \sum_{l=1}^{m+m'} w_{ij} \cdot (1 - y_{ij}) q_k q_l x_{ik} x_{jl} d_{(v_i, v_j)} + \Pi, \qquad (9)$$

where $\Pi = \beta \left(\sum_{k=1}^{m+m'} |n_k - Cap_k| \right) + \gamma \max\{(EC - Budget), 0\}$ is the penalty measure associated with a chromosome, and assume n_k be the number of cells in C^{new} be assigned to switch s_k, and β and γ are the penalty weights (see [7] for a further discussion of penalty measures).

Since the best-fit chromosomes should have a probability of being selected as parents 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:
maximize

$$\begin{aligned} C_{\max} \quad - \quad & [\sum_{i=1}^{n} l_{iv_i} + \alpha \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} d_{(v_i, v_j)} \\ + \quad & \beta \sum_{k=1}^{m+m\prime} (n_k - Cap_k) \\ + \quad & \gamma (\max\{EC - Budget, 0\})] \qquad (10) \end{aligned}$$

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, and C_{max} can be calculated by the following iterative equation:

$$C_{max} = max\{C_{max}, cost\}$$

where C_{max} is initialized to zero.

5.7 Replacement strategy

This subsection discusses a method used to create a new generation after crossover and mutation is carried out on the chromosomes of the previous generation. Each offspring generated after crossover is added to the new generation if it has a better objective function value than both of its parents. If the objective function value of an offspring is better than that of only one of the parents, then we select a chromosome randomly from the better parent and the offspring. If the offspring is worse than both parents then each of the parents is selected at random for the next generation. This ensures that the best chromosome is carried to the next generation, while the worst is not carried to the succeeding generations.

5.8 Termination rules

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

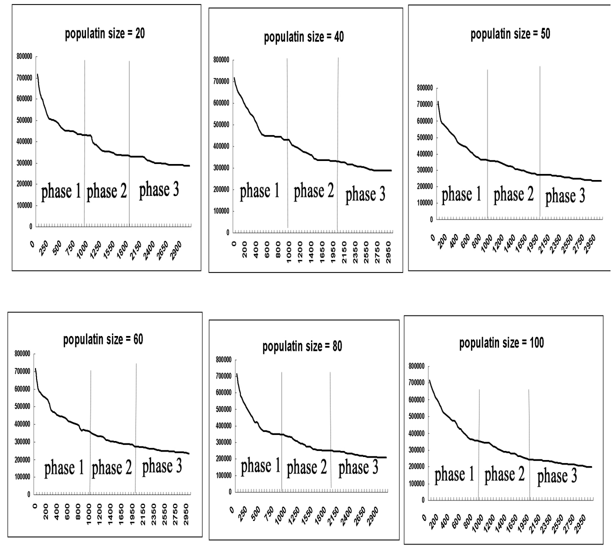

Figure 9: The effect of the GA with different population size.

6 Experimental Results

In order to evaluate its performance, 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 language was conducted in C, and all experiments were run on a Windows NT with a Pentium II 450MHZ CPU and 256MB RAM. We simulated a hexagonal system in which the cells were configured as an H-mesh. The handoff frequency f_{ij} for each border was generated from a normal random number with mean 100 and variance 20. To examine the effect of the different population size of genetic algorithms, we set $n = 400$, $n' = 200$, $m = 20$, $CAP = 25$, $\alpha = 1$, population size (popsize) is in set $\{20, 40, 50, 60, 80, 100\}$, crossover probability (P_c)=1.0, maximum number of generations of each phase is 1000, and mutation probability is 0.05. The coverage behaviors of the three-phase GA were shown in Fig. 9 and Fig. 10.

7 Conclusions

In this paper, we investigate the *network expanded problem* which optimum assignment new and splitting cells in *PCS* network to switches on an ATM network. This problem is currently faced by designers of mobile communication service and in the future, it is likely to be faced by designers of personal communication service (PCS).

Since finding an optimal solution of the network expended problem is NP-hard, a stochastic search method based on a genetic approach is proposed to solve it. Simulation results showed that genetic algorithm is robust

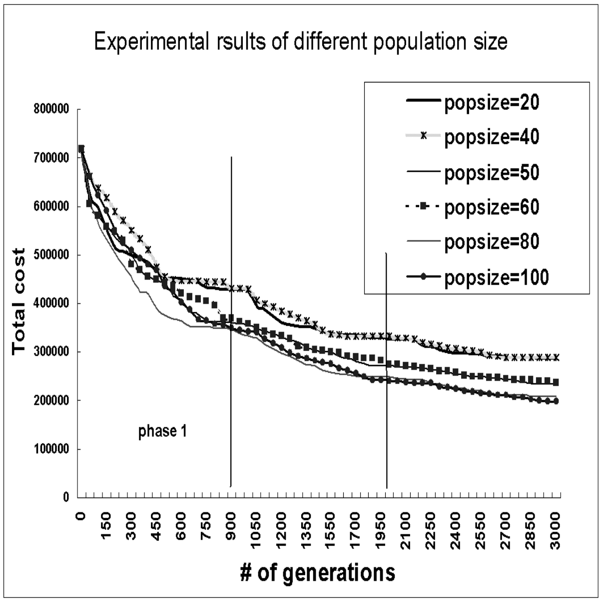

Figure 10: The effect of the GA with different population size.

for this problem. In our methods, three types of genes (switch-location, switch-connection, cell-assignment) are used to encode chromosome. Chromosome adjustment method is proposed to adjust chromosome to represent a feasible solution and find the fitness of chromosome. The traditional single point crossover and single cell mutation are employed in our method. Experimental results indicate that the algorithm run 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] M. Cheng, S. Rajagopalan, L. F. Chang, G. P. Pollini, and M. Barton, "PCS Mobility Support over Fixed ATM Networks," *IEEE Communication Magazine*, Nov. 1997, pp. 82–91.

[2] L. Davis (Eds.), *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York NY. U. S. A., 1991.

[3] D. R. Din and S. S. Tseng, "Genetic Algorithms for Optimal design of two–level wireless ATM network," *Proceeding of NSC*, Vol. 25, No. 3, pp. 151-162, 2001.

[4] D.R. Din and S. S. Tseng, "Heuristic Algorithm for Optimal design of two-level wireless ATM network, " *Journal of Information Science Engineering*. Vol. 17, pp. 674-665, 2001.

[5] D. R. Din and S. S. Tseng, "Genetic Algorithm for Extended Cell Assignment Problem in Wireless ATM Network," Jifeng He and Masahiko Sato (Eds.), ASIAN'00, Asian Computing Science Conference, Penang, Malaysia, November 25-27, 2000, Lecture Notes in Computer Science (LNCS) 1961, Springer-Verlag Berlin Heidelberg, pp. 69-87, 2000.

[6] D. R. Din and S. S. Tseng, "Simulated Annealing Algorithm for Extended Cell Assignment Problem of Wireless ATM Network, E. J. W. Boers et al. (Eds.) EvoWorkshop 2001, Lecture Notes in Computer Science (LNCS) 2037, Springer-Verlag Berlin Heidelberg, pp. 150-160, 2001.

[7] J. Holland, *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press (Ann Arbor), 1975.

[8] R. C. V. Macario, Cellular Radio. McGraw-Hill, New York, 1993.

[9] K. F. Man, K. S. Tang, S. Kwong and W. A. Halang, "Genetic Algorithms for control and signal processing." Springer Verlag, ISBN 3-540-76101-2, 1997.

[10] A. Merchant and B. Sengupta, "Assignment of Cells to Switches in PCS Networks", *IEEE/ACM Trans. on Networking*, Vol. 3, no. 5, 1995, pp. 521-526.