2003 International Symposium on Advanced Intelligent Systems
September 25-28, 2003, Jeju, Korea

Evolutionary Network Optimization:
Hybrid Genetic Algorithms Approach

Mitsuo Gen
Graduate School of Information, Production & Systems
Waseda University
Kitakyushu 808-0135, Japan
E-mail: gen@waseda.jp

Abstract - Network optimization is being increasingly important and fundamental issue in the fields such as engineering, computer science, operations research, transportation, telecommunication, decision support systems, manufacturing, and airline scheduling. Networks provide a useful way to modeling real world problems and are extensively used in practice.

Many real world applications impose on more complex issues, such as, complex structure, complex constraints, and multiple objects to be handled simultaneously and make the problem intractable to the traditional approaches. Recent advances in evolutionary computation have made it possible to solve such practical network optimization problems.

The invited talk introduces a thorough treatment of evolutionary approaches, *i.e.*, hybrid genetic algorithms approach to network optimization problems, such as, fixed charge transportation problem, minimum cost and maximum flow problem, minimum spanning tree problem, multiple project scheduling problems, scheduling problem in FMS.

Keywords: Genetic algorithms, multistage process planning, minimum spanning tree, fixed charge transportation problem, minimum cost & maximum flow problem, resource constrained project scheduling problem, and scheduling problem in FMS.

I. INTRODUCTION

Network optimization is being increasingly important and fundamental issue in the fields such as engineering, computer science, operations research, transportation, telecommunication, decision support systems, manufacturing, and airline scheduling. Networks provide a useful way to modeling real world problems and are extensively used in practice (Bertsekas and Gallager 1992). Many real world applications impose on more complex issues, such as, complex structure, complex constraints, and multiple objects to be handled simultaneously and make the problem intractable to the traditional approaches. Recent advances in evolutionary computation have made it possible to solve such practical network optimization problems (Gen and Kim 1998; Gen, Cheng & Oren, 2001).

Genetic algorithms are one of the most powerful and broadly applicable stochastic search and optimization techniques based on principles from evolution theory (Holland, 1975; Goldberg, 1989). In the past few years, the genetic algorithms community has turned much of its attention toward the optimization of network design problems (Gen and Cheng, 1997; Gen and Cheng, 2000). This paper is intended to introduce the applications of GAs to some difficult-to-solve network design problems (Gen, Zhou and Kim, 1999; Gen, Cheng and Oren, 2001).

The invited talk introduces a thorough treatment of evolutionary approaches, *i.e.*, adaptation of genetic algorithms approach to network optimization problems, such as, fixed charge transportation problem, minimum spanning tree problem, minimum cost and maximum flow problem, multiple project scheduling problems, scheduling problem in FMS.

II. ADAPTATION OF GENETIC ALGORITHMS

Genetic algorithms were first created as a kind of generic and weak method featuring binary encoding and binary genetic operators. This approach requires a modification of an original problem into an appropriate form suitable for the genetic algorithms, as shown in Figure 2.1. The approach includes a mapping between potential solutions and binary representation, taking care of decoders or repair procedures, *etc.* For complex problems, such an approach usually fails to provide successful applications.

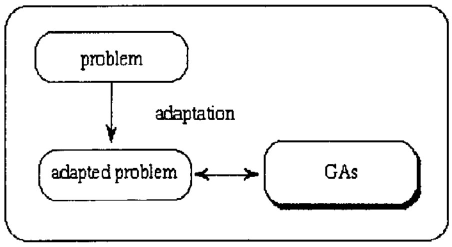

Figure 2.1 Adapting a problem to GA

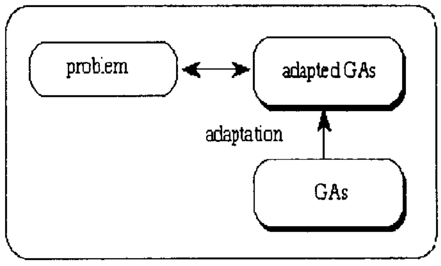

Figure 2.2 Adapting GA to a problem

To overcome such problems, various non-standard implementations of the genetic algorithms have been created for particular problems. As shown in Figure 2.2, this approach leaves the problem unchanged and adapts the genetic algorithms by modifying a chromosome representation of a potential solution and applying appropriate genetic operators. But in general, it is not a good choice to use the whole original solution of a given problem as the chromosome because many real problems are too complex to have a suitable implementation of genetic algorithms with the whole solution representation. Generally, the encoding methods can be either direct or indirect. In the direct encoding method, the whole solution for a given problem is used as a chromosome. For a complex problem, however, such a method will make almost all of the conventional genetic operators unusable because a vast number of offspring will be infeasible or illegal. On the contrary, in the indirect encoding method, just the necessary part of a solution is used as a chromosome. A decoder then produces the solution. A decoder is a problem-specific and determining procedure to generate a solution according to the permutation and/or the combination of the items produced by genetic algorithms. With this method, the genetic algorithms will focus their search solely on the interesting part of solution space.

A third approach is to adapt both the genetic algorithms and the given problem, as shown in Figure 2.3. A common feature of combinatorial optimization problems is to find a permutation and/or a combination of some items associated with side constraints. If the permutation and/or combination can be determined, a solution then can be derived with a problem-specific procedure. With this third approach, genetic algorithms are used to evolve an appropriate permutation and/or combination of some items under consideration, and a heuristic method is subsequently used to construct a solution according to the permutation and combination.

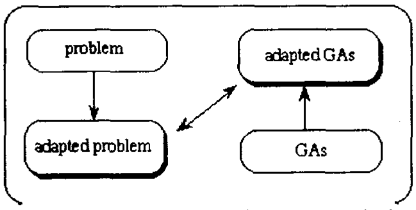

Figure 2.3 The third approach: adapt both the genetic algorithms and the problem

This approach has been successfully applied in the area of industrial engineering and has recently become the main approach for the practical use of genetic algorithms in the network design and optimization.

III. FIXED CHARGE TRANSPORTATION PROBLEM

The fixed charge transportation problem (fc-TP) is an extension of the transportation problem (TP) and many practical transportation and distribution problems can be formulated as this problem. For instance, in a transportation problem, a fixed cost may be incurred for each shipment between a given plant and a given consumer and a facility of a plant or warehouse may result in a fixed amount on investment. The fc-TP problem is much more difficult to solve due to the presence of fixed costs, which cause discontinuities in the objective function. Given m plants and n consumers, the problem can be formulated as follows:

fc-TP:

$$\min \ f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n}(f_{ij}(x) + d_{ij}g(x_{ij})) \tag{3.1}$$

$$\text{s. t.} \quad \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1, 2, \quad m \tag{3.2}$$

$$\sum_{i=1}^{m} x_{ij} \ge b_j, \quad j = 1, 2, \quad n \tag{3.3}$$

$$x_{ij} \ge 0, \quad \forall\, i,\ j \tag{3.4}$$

where $x = [x_{ij}]$ is the unknown quantity to be transported from plant i to consumer j, $f_{ij}(x)$ is the objective function of shipping, and

$$g(x_{ij}) = \begin{cases} 1, & \text{if } x_{ij} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

where d_{ij} is the fixed cost. Many solution procedures have been proposed for the fixed charge transportation problem range from exact solution algorithms to heuristic methods. Recently, Gottlieb and Paulmann (1998) proposed a genetic algorithm based on permutation representation for this problem. Sun, Aronson, Mckeown, and Drinka (1998) proposed a tabu search method. Since the solution of the problem has a network structure characterized as spanning tree, Gen, Li and Ida proposed a spanning tree-based genetic algorithms (Gen and Li, 1997, 1998; Gen, Li, and Ida, 1999, 2000; Gen, Choi & Ida, 2000). Figure 3.1 shows a simple example of transportation alternatives, expressed as a spanning tree. A transportation alternative can be encoded by a Pr er number as shown in Figure 3.2. The detail of the decoding procedure from a Pr er number to a transportation tree was given in the book (Gen and Cheng, 2000).

Because a transportation tree is a special type of spanning tree, the Pr er number encoding may correspond to an infeasible solution. Gen and Li designed a criterion for checking the feasibility of chromosomes. One-point crossover and inversion mutation were used to explore new solutions.

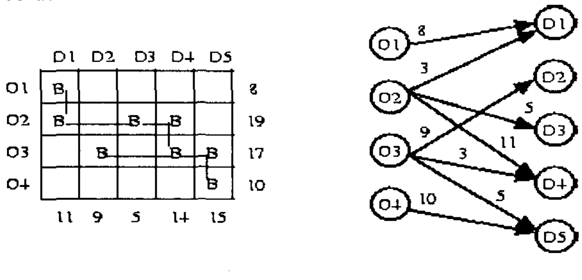

Figure 3.1 A transportation a lternatives with a spanning tree structure

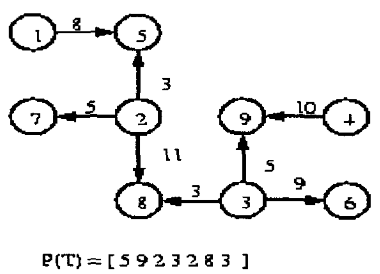

Figure 3.2 A transportation alternatives with a spanning tree structure

To demonstrate the effectiveness and efficiency of the spanning tree based genetic algorithm for the problem, Gen and Li carried out nume rical experiments and compared with the matrix-based genetic algorithm (Vignaux and Michalewicz, 1991). For the small-scale problem, there was no obvious difference on results. For the larger scale problems, the spanning tree-based genetic algorithm can get the optimal or near optimal solutions wit (Li & Gen, 1997; Li, 1999; Kim, Gen & Ida, 1999; Kim, 2000).

IV. MINMUM SPANNING TREE PROBLEM

A spanning tree structure is the best topology for telecommunication network designs, which usually consists of finding the best way to link n nodes at different locations. They may be the host, concentrators, multiplexors, and terminals. In a real-life network optimization situation, a spanning tree is often required to satisfy some additional constraints, such as the edge or capacity on a node. A tree structure network with the constraint on the edges is denoted as the degree-constrained minimum spanning tree problem (dc-MST) (Narula and Ho, 1980; Hall, 1996).

Considering an undirected graph $G = (V, E)$, let $V = \{1, 2, \quad n\}$ be the set of nodes and $E = \{(i,j) \mid i, j \in V\}$ be the set of edges. For a subset of nodes S ($\subseteq$ V), define $E(S) = \{(i,j) \mid i, j \in S\}$ be the edges whose end points are in S. Define the following binary decision variables for all edges $(i,j) \in E$.

$$x_{ij} = \begin{cases} 1, & \text{if edge}(i,j) \text{ is selectedin a spanningtree} \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

Let w_{ij} be the fixed cost related to edge (i, j), the problem can be formulated as follows:

dc-MST:

$$\min \quad z(x) = \sum_{i=1}^{n-1} \sum_{j=2}^{n} w_{ij} x_{ij} \tag{4.2}$$

$$\text{s. t.} \quad \sum_{i=1}^{n-1} \sum_{j=2}^{n} x_{ij} = n ? 1 \tag{4.3}$$

$$\sum_{i \in S} \sum_{j \in S, j>1} x_{ij} \leqq |S| - 1, \; S \subseteq V \setminus \{1\}, \quad |S| \geqq 2 \tag{4.4}$$

$$\sum_{j=1}^{n} x_{ij} \leqq b_i, \; i = 1, 2, \quad n \tag{4.5}$$

$$x_{ij} = 0 \text{ or } 1, \; i = 1, 2, \quad n - 1, \; j = 2, 3, \quad n \tag{4.6}$$

where b_i is the constrained degree value for node i. Inequality (5.3) is the constrained degree on each node. Equality (5.4) is true of all spanning trees. The dc-MST problem is NP-hard and there are no effective algorithms to deal with it. Zhou and Gen proposed a genetic algorithms approach to solve this problem (Zhou and Gen, 1997, 1999; Gen, Zhou and Takayama 2000). There are two factors which should be taken into consideration if we want to keep the tree topology in genetic representation: one is the connectivity among nodes; the other is the degree value of each node. Therefore, a two-dimension structure was used as the genetic representation. One dimension encodes a spanning tree; another dimension encodes degree value. For a undirected tree, we can take any node as the root node of it and all other nodes are regarded as being connected to it hierarchically. Figure 4.1 illustrates an example of this degree-based permutation.

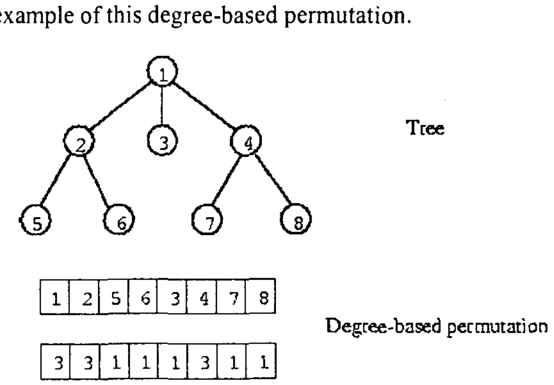

Figure 4.1 A rooted tree and its degree-based permutation encoding

In order to keep the degree constraint and connectivity between nodes, the genes in the degree dimension need to satisfy the following conditions: For an n-node tree, the total degree value for all nodes is $2(n-1)$. Suppose that d_{used} is the total degree value of the nodes whose degree value in degree dimension have been assigned and d_{rest} is the total lower bound of the degree values for all those nodes whose degree value in degree dimension have not been assigned. Then the degree value for the current node in degree dimension should hold: no less than 1. The degree value for the current node together with the number of the rest nodes should hold: no less than d_{rest} and no greater than $2(n-1) - d_{used}$.

Because this encoding is essentially a permutation one, uniform crossover and insertion mutations were adopted. Especially the insertion mutation plays a very important role for the dc-MST problem as it always keep the individuals as tree structure and evolves them to the fitter tree structures. This operator selects a string of genes (branch of a tree) at random and inserts it in a random gene (node). The operation is illustrated in Figure 4.2.

As a related research on a spanning tree problem Malik and Yu proposed a branch and bound method for the Capacitated MST Problem (Malik and Yu, 1993)

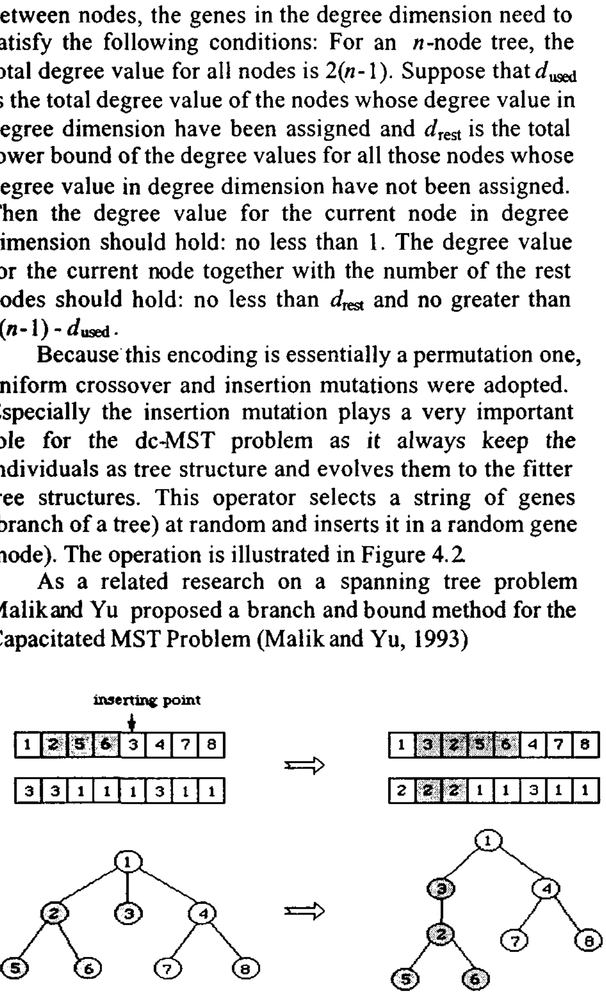

Figure 4.2 Illustration of insertion mutation

V. SHORTEST PATH PROBLEM

One of the most common problems encountered in the analysis of networks is the shortest path problem (Jensen and Barnes, 1980). The problem is to find a path between two designated nodes with the minimum total length or cost. It is a fundamental problem that appears in many applications involving transportation, routing and communication. In many applications, however, there are several criteria associated with traversing each edge of a network. For example, cost and time measures are both important in transportation networks, economic and ecological factors for highway construction. As a result, there has been recent interest in solving bicriteria shortest path problem. It is to find paths that are efficient with respect to both criteria. There is usually no single path that gives the shortest path with respect to both criteria. Instead, a set of Pareto optimal paths is preferred. Cheng and Gen proposed a compromise approach-based genetic algorithm to solve the bicriteria shortest path problem (Cheng and Gen 1994; Gen, Cheng and Wang 1997). The compromise approach, contrary to generating approach, identifies solutions, which are closest to the ideal solution as determined by some measure of distance.

How to encode a path for a graph is a critical step. Special difficulty arises because (1) a path contains variable number of nodes, and (2) a random sequence of edges usually does not correspond to a path. To overcome such difficulties, Cheng and Gen adopted an indirect approach: encode some guiding information for constructing a path in a chromosome, but not a path itself (Cheng and Gen, 1994). A new encoding method, called *proposed priority-based encoding,* was introduced. In this method, the position of a gene was used to represent a node and the value of the gene was used to represent the priority of the node for constructing a path among candidates. The path corresponding to a given chromosome is generated by sequential node appending procedure with beginning from the specified node 1 and terminating at the specified node n. At each step, there are usually several nodes available for consideration, only the node with the highest priority is added into path. Consider the undirected graph shown in Figure 5.1 and a priority-based encoding shown in Figure 5.2. Suppose we want to find a path from node 1 to node 10.

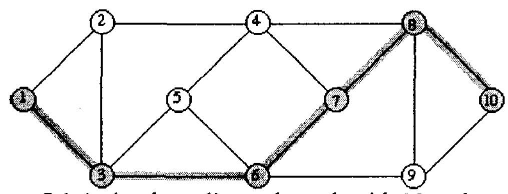

Figure 5.1 A simple undirected graph with 10 nodes and 16 edges

position: node ID	1	2	3	4	5	6	7	8	9	10
value: priority	7	3	4	6	2	5	8	10	1	9

Figure 5.2 An example of priority-based encoding

At the beginning, we try to find a node for the position next to node 1. Nodes 2 and 3 are eligible for the position, which can be easily fixed according to adjacent relation among nodes. The priorities of them are 3 and 4, respectively. The node 3 has the highest priority and is put into the path. The possible nodes next to node 3 are nodes 2, 5 and 6. Because node 6 has the largest priority value, it is put into the path. Then we form the set of nodes available for next position and select the one with the highest priority among them. Repeat these steps until we obtain a complete path (1, 3, 6, 7, 8, 10).

The compromise approach can be regarded as a kind of mathematical formulation of goal seeking behavior in terms of a distance function. It identifies solutions that are closet to the ideal solution as determined by following weighted L_p-norm:

$$r(z;p,w) = \left[\sum_{k=1}^{2} w_k^p \left| z_k - z_k^* \right|^p \right]^{1/p} \qquad (8.1)$$

where $z^*=(z_1^*, z_1^*)$ is the ideal solution to the problem. The parameter p is used to reflect the emphasis of decision-makers. For the bicriteria shortest path problem, the ideal point can be easily obtained by solving two single criterion problems. For many complex problems, to obtain an ideal point is also a difficult task. To overcome the difficulty, a concept of proxy ideal point was suggested to replace the ideal point. The proxy ideal point is the ideal point corresponding to current generation but not to a given problem. In the other words, it is calculated in the explored partial solution space but not the whole solution space. The proxy ideal point is easy to obtain at each generation. Along with evolutionary process, the proxy ideal point will gradually approximate to the real ideal point. Recently, Ahn and Ramakrishna proposed a new method for solving the shortest path routing problem (Ahn and Ramakrishna, 2002).

VI. MINIMUM COST/ MAXIMUM FLOW PROBLEM

The flow problem is structured on a network; where each arc is imposed to some attribute and the problem is to find a flow possible from some given source node to a given sink node subject to conservation of flow constraints at each node to optimize some criteria. In the maximum flow problem, each arc is imposed a upper bound, and the attempt is to find a maximal flow subject to flow bounds on each arc. In the minimum flow problem, each arc is associated with a cost, and the attempt is to find a flow with minimal cost over all arcs. The minimum cost maximum flow problem is a combination of Maximum Flow Problem (MXF) and the Minimum Cost Flow (MCF) problem. It asks for a maximum flow from a source node to a target node such that the total cost of the flow is minimal. Gen, Lin and Cheng have proposed a genetic algorithm to solve this problem (Gen, Lin, and Cheng, 2003).

Consider a given graph $G=(V,A)$ with m nodes $i \in V$ and n arcs $(i,j) \in A$, c_{ij} and u_{ij} denote the cost and upper bound for acr (i,j), the bicriteria optimal flow problem can be formulated as follows:

bound for arc (i,j), the bicriteria optimal flow problem can be formulated as follows:

$$\max \ z_1 = v \qquad (6.1)$$

$$\min \ z_2 = \sum_{i=1}^{m} \sum_{j=1}^{m} c_{ij} x_{ij} \qquad (6.2)$$

$$\text{s.t.} \ \sum_{i=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = \begin{cases} v & (i=1) \\ 0 & (i=2,3,\cdots,m-1) \\ -v & (i=m) \end{cases} \qquad (6.3)$$

$$0 \le x_{ij} \le u_{ij} \quad , \forall (i,j) \in A$$

$$v \ge 0$$

In Gen, Lin and Cheng implementation, a priority-based representation was adopted to encode a path for the problem. The encoding method is capable to represent all feasible paths for a given graph. A special decoding method is designed to generate a minimum cost and maximum flow from a given chromosome. Partially match crossover and swap mutation were used to generate new solutions. Adaptive weight approach was implemented to evaluate current population in order to give a search pressure towards to the positive ideal point (Gen and Cheng, 2000; Gen, Ida and Kim, 1998). Figure 6.1 shows an example of the bicriteria optimal flow problem with 25 nodes and 49 arcs. Figure 62 shows the set of Pareto solutions found by the proposed genetic algorithm.

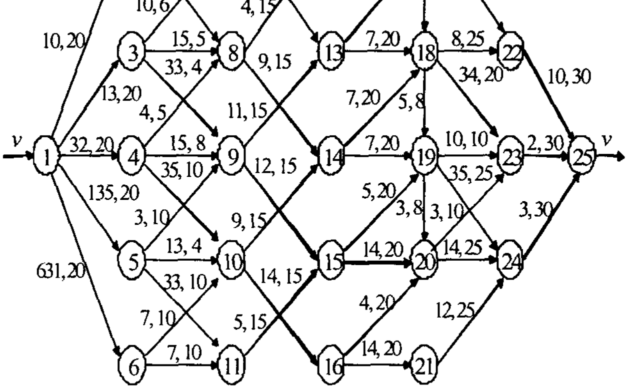

Figure 6.1 An example of bicriteria optimal flow problem

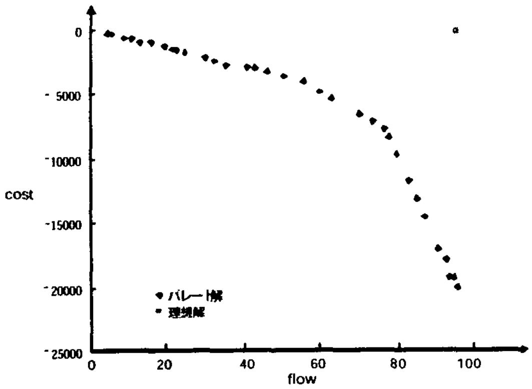

Figure 6.2 The set of Pareto solutions found by GA

VII. SCHEDULING PROBLEMS IN FMS ENVIRONMENTS

Kim, Gen and Yamazaki have investigated a special case of scheduling problems in flexible manufacturing systems (FMS), transformed the problem into a Multistage Process Planning Problem (Chang and Wysk 1985) as described by Zhou and Gen (Zhou and Gen, 1997), and then solved with a hybrid genetic algorithm (Yang, 2001; Gen, Kim and Yamazaki, 2003; Kim, Gen, Moon and Yamazaki, 2003).

The flexible manufacturing system is an enhancement of the cellular manufacturing paradigm. Typical flexible manufacturing system is composed of multiple workstations (or machine centers), a material handling system, and a loading-unloading station (Bedworth and Bailey, 1987). In FMS environment, the objectives of the scheduling problem are to minimize the makespan t_M, total flow time t_F, and total tardiness penalty p_T of the project.

$$\min\ t_M = t_{mn} \tag{7.1}$$

$$\min\ t_F = \sum_{j=1}^{J} t_{mn}(J_j) \tag{7.2}$$

$$\min\ p_T = \sum_{j=1}^{J} \max\{\ t_{mn}(J_j) - t_{DD}(J_j),\ 0\} \times c_{TP}(J_j) \tag{7.3}$$

$$\text{s. t.}\quad t_{ik} - t_{(i-1)k} \geq p_{(i-1)k},\ \forall_i \in S_i \tag{7.4}$$

$$t_{ik} \geq 0,\ i = 1,2,\cdots,m,\cdots,I,\ k = 1,2,\cdots,n,\cdots K \tag{7.5}$$

$$J_j \geq 0,\ j = 1,2,\ldots J \tag{7.6}$$

where t_{ik} denotes the finish time of operation o_i on workstation W_k, p_{ik} the processing time of operation o_i on workstation W_k, A_{ik} the set of operations at the time t_{ik}, $t_{mn}(J_j)$ the finish time of the last operation o_i on workstation W_k about job J_j, $t_{DD}(J_j)$ the due date of the job J_j, $c_{TP}(J_j)$ the total penalty cost of the job J_j. Equation (7.1) is to minimize the makespan of whole system. Equation (7.2) is to minimize the total flow time. Equation (7.3) is to minimize total penalty. Constraint (7.4) ensures that none of the precedence constraints are violated.

Consider a simple FMS scheduling case with three workstations (W_1, W_2, W_3), three jobs (J_1, J_2, J_3) run in each workstation, and each job requires two kinds of operations among possible four operations (o_1, o_2, o_3, o_4). Suppose the processing time of each operation is given in Table 7.1. Table 7.2 shows relevant data pertaining to each job that must be scheduled in the system.

Table 7.1 Workstation and processing time data of each operation

Workstation	o_1	o_2	o_3	o_4
W_1	20	20	8	39
W_2	26	8	52	41
W_3	8	30	38	8
Average	23	25	45	40

Table 7.2 Job-related data for example of FMS problem

Job no.	*Required operations*	t_{ATP}	t_{DD}	c_{TP}
J_1	$o_1 \rightarrow o_3$	68	94	1
J_2	$o_2 \rightarrow o_3$	70	100	1
J_3	$o_4 \rightarrow o_3$	85	101	1

t_{ATP}: The average total processing time,

t_{DD}: The due date, c_{TP}: The total penalty cost

Then the problem can be represented as a network flow shown in Figure 7.1 by the chromosome in Figure 7.2.

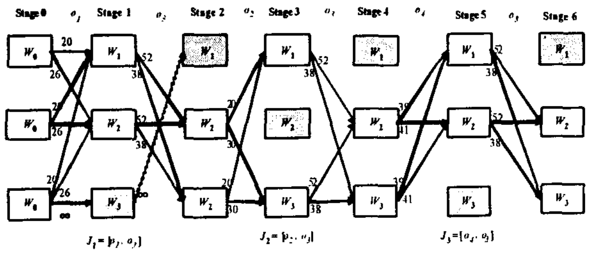

Figure 7.1 The flow network representation for a simple FMS problem.

Kim *et al* used the state permutation encoding method in their genetic algorithm implementation. A chromosome for the example given in above figure is shown as follows. A SPT (shortest processing time) first heuristic rule is used to decode the chromosome into a feasible schedule.

1	3	1	2	2	3

Figure 7.2 State permutation encoding

A neighborhood search technique based mutation operation was used in order to have more chance to find out improved solutions. A fuzzy logic controller was used to adjust crossover and mutation parameters during evolutionary process.

VIII. RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM

The problem of scheduling activities under resource and precedence restrictions with the objective of minimizing the project duration is referred to as the resource constrained project scheduling problem in literature (Baker, 1974). The basic problem can be stated as follows. A project consists of a number of interrelated activities. Each activity is characterized by a known duration and given resource requirements. Resources are available in limited quantities but renewable from period to period. There is no substitution between resources and activities cannot be interrupted. A solution is to determine the start times of activities with respect to the precedence and resource constraints so as to optimize the objective. Following picture shows a well-known benchmark problem given by Daivs in 1975, presented as a *directed acyclic graph* (Davis, 1975).

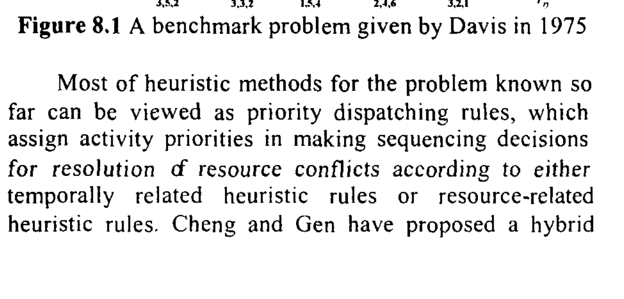

Figure 8.1 A benchmark problem given by Davis in 1975

Most of heuristic methods for the problem known so far can be viewed as priority dispatching rules, which assign activity priorities in making sequencing decisions for resolution of resource conflicts according to either temporally related heuristic rules or resource-related heuristic rules. Cheng and Gen have proposed a hybrid genetic algorithm to the resource-constrained project scheduling problem (Cheng and Gen, 1994). In essentials, the problem consists of two basic issues: (a) how to determine the processing order of activities without violating the precedence constraints and (b) how to subsequently determine the start time for each activity without violating the resource constraint, resource constraint. How to determine the order of activities is critical to the problem because that if the order of activities is determined, a schedule then can be easily constructed with some determining procedures according to the order. A priority-based encoding method is proposed to handle this difficulty, based on the concepts of the *topological sort* of a directed acyclic graph (Cheng and Gen, 1998). A *local search-based mutation* method was proposed to hunt for an improved solution other than just random search as the usual mutation does.

Kim, Gen, and Yamazaki proposed a hybrid genetic algorithm for solving the basic type of resource constrained project scheduling problem (Kim, Gen, and Yamazaki, 2003a). A fuzzy logic controller (FLC) is used to adjust crossover and mutation ratio (Lee, 1990). The basic idea is that to increase the ratio when the change of average fitness value among population is insignificant and to decrease the ratio when the change is significant. The architecture of hybrid genetic algorithm by the combination of FLC and GA is shown in Figure 8.2.

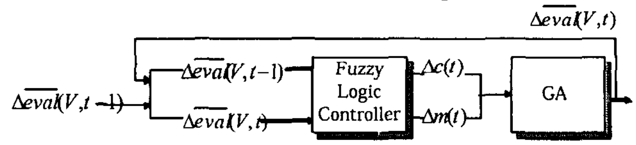

Figure 8.2 The architecture of hybrid GA with FLC.

The fuzzy logic controller is a kind of rule-based system, which is based on fuzzy logic and fuzzy set theory. Generally, the behavior of genetic algorithms depends on many uncertain factors, and only incomplete knowledge and imprecise information are available for identification of the relationship between the strategy parameters and the behavior of genetic algorithms. Therefore, it is suitable for the fuzzy logic controllers to dynamically adjust these parameters. When utilizing a fuzzy logic controller to adjust the strategy parameters of genetic algorithms, diversity measure, fitness values, and current parameters are taken as inputs of if-then rules. Outputs indicate values of strategy parameters, that is, the crossover and mutation ratio. They adopted the implementation of the fuzzy logic controller suggested by Wang et al (Wang, Wang, and Hu, 1997). The basic structure of the method consists of two fuzzy logic controllers: one adjusts the crossover ratio and one adjusts the mutation ratio. The heuristic method for updating the crossover ratio is to consider the changes in the average fitness of the population.

Kim *et al* further extended their work into multiple projects case (Kim, Yun, Yoon, Gen, and Yamazaki, 2003). The problem consists of multiple projects, and precedence constraints among projects are given. In each project, there are a number of activities with known processing time and multiple resources consumption. Start time of each activity is dependent upon the completion of some other activities (precedence constraints of activities). The multiple resources are available in limited quantities but renewable from period to period. Activities can not be interrupted; there is only one execution mode for each activity. The managerial objective is to minimize the total project time and the total tardiness penalty for all projects. The following figure shows an illustrative example of the problem.

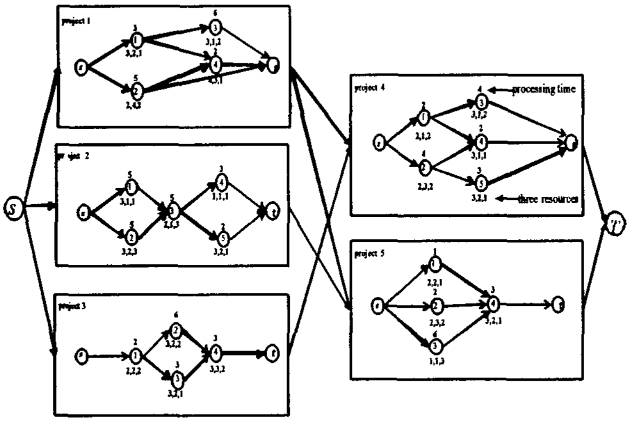

Figure 8.3 An illustrative example of resource constrained multiple projects scheduling problem

Let i denote the project index, j the activity index in each project, r the nonrenewable resource index, p_{ij} the processing time of activity j in project i, t_{ij}^F the finish times of activity j in project i, t_{iJ}^F the finish times of last activity J in project i, t_{ij}^S the start times of activity j in project i, l_{ijr} the scheduling activity j in project i consumes resource units per period from resource r, l_r^M the maximum-limited resource r only available with the constant period availability, A_t the set of activities being in progress in period t, t_i^D the due date of the project i (the promised delivery time of project), c_i^{TP} the total penalty cost of the project i, then the problem can be formulated as follows:

$$\min\ t_F = \sum_{i=1}^{I} t_{iJ}^F \tag{8.1}$$

$$\min\ p_T = \sum_{i=1}^{I} c_i^{TP} \times (t_{iJ}^F - t_i^D) \tag{8.2}$$

$$\text{s.t.}\quad t_{(i-1)J}^S + p_{(i-1)J} \le t_{iJ}^S \quad \forall i \tag{8.3}$$

$$t_{i(j-1)}^S + p_{i(j-1)} \le t_{ij}^S \quad \forall i,j \tag{8.4}$$

$$\sum_{i\in A_t}\sum_{j\in A_t} l_{ijr} \le l_r^M$$

$$i = 0,\dots,I+1,\ j = 0,\dots,J+1,\ r \in R \tag{8.5}$$

where

$$(t_{iJ}^F - t_i^D)_+ = \begin{cases} t_{iJ}^F - t_i^D; & t_{iJ}^F > t_i^D \\ 0 & ;\ t_{iJ}^F \le t_i^D \end{cases} \tag{8.6}$$

The objective function (8.1) minimizes the total project time. Equation (8.2) defines the penalty costs for all projects. Constraint (8.3) states precedence relations among projects. Constraint (8.4) indicates precedence relations among activities. Constraint (8.5) corresponds to resource constrains.

A hybrid genetic algorithm with fuzzy logic controller was adopted to solve the resource-constrained multiple project scheduling problem. The priority-based encoding was extended to cope with the multiple projects situation and swap mutation and local search-based mutation were adapted for this encoding. The fuzzy logic controller was used to adjust parameters of genetic algorithm during evolutionary process.

Lastly as a related area with network optimization problems there are several areas such as the location-allocation problem with obstacle and the capacitated location-allocation problem by hybrid genetic algorithms (Gong, Gen, Yamazaki and Xu, 1995; 1997), Walters and Smith proposed evolutionary algorithm for optimal layout of tree networks (Walters and Smith, 1995) and recently Zhou and Gen reported a genetic algorithm approach on tree-like telecommunication network design problem (Zhou and Gen, 2003).

IX. CONCLUSION

With the development of modern society, the data communication has become very important part in human being life. Actually, this trend will continue to the next century or even further future. Simultaneously, it also brings about many problems related with varieties of network designs to us. In this paper, we gave out a brief review about our recent research works in this field. Different from many other conventional techniques, we developed the genetic algorithms to deal with all these network design problems. Our limited computational experience showed that the genetic algorithms approach is very effective to solve such kinds of networks design problems. Especially, with the increase of problem scale, and some complicated constraints, the genetic algorithms showed their even great potential power to cope with all these network design problems. The key issue for solve those problems successfully is if we can invent an encoding method that well matches the instinct nature of a given problem. From this point of view, the genetic algorithms are not only means of algorithms or techniques, but also a kind of art in the sense that the problems were solved in coding space instead of the solutions space themselves. The paper therefore focused on such kind of state-of-the-art in the genetic algorithms approach on these network design problems.

ACKNOWLEDGEMENT

This research work was partly supported by the Grant-in-Aid for Scientific Research (No. 10044173) by the Ministry of Education, Science and Culture, the Japanese Government.

REFERENCES

1. Ahn, C. W. and R. S. Ramakrishna (2002) A genetic algorithms for shortest path routing problem and the size of populations, *IEEE Trans. On Evolutionary Computation*, Vol. 6, No. 6, pp.566–579.
2. Bedworth, D., and J. Bailey (1987) *Integrated Production Control System*. John Wiley & Sons, New York.
3. Bertsekas, D. and R. Gallager (1992) *Data Networks*, 2nd ed., Prentice-Hall, New Jersey.
4. Chang, T. C. and Wysk, R. A. (1985) *An Introduction to Automated Process Planning Systems*, Prentice-Hall.
5. Cheng, R. and M. Gen (1994), Evolution program for resource constrained project scheduling problem, In Fogel, D., editor, *Proceedings of the First IEEE Conference on Evolutionary Computation*, pp.736-741, IEEE Press, Florida.
6. Cheng, R. and M. Gen, An evolution program for resource constrained project scheduling problem, *Computer Integrated Manufacturing*, Vol.11, no.3, pp.274-287, 1998.
7. Davis, E.W. and J.H. Patterson, A comparison of heuristic and optimum solutions in resource-constrained project scheduling, Management Sciences, Vol.21, pp. 944-955, 1975.
8. Gen, M. and R. Cheng (1997) *Genetic Algorithms and Engineering Design*, John & Wiley Sons, New York.
9. Gen, M., R. Cheng, and D. Wang (1997) Genetic Algorithms for solving Shortest Path Problems, *Proceedings of IEEE International Conference on Evolutionary Computation*, Indiana, pp.401-406.
10. Gen, M., K. Ida and J. R. Kim (1998) A Spanning Tree-based Genetic Algorithm for Bicriteria Topological Network Design, *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, USA, pp.15-20.
11. Gen, M. and J. R. Kim (1998) GA-based Optimal Network Design: A state-of-the-art Survey, *Proceedings of the Artificial Neural Networks in Engineering Conference*, St. Louis, USA, pp.247-252.
12. Gen, M. and Y. Z. Li (1998) Solving Multiobjective Transportation Problem by Spanning Tree-based Genetic Algorithm, in I. Parmee ed., *Adaptive Computing in Design and Manufacture*, Springer-Verlag, pp.98-108.
13. Gen, M., G. Zhou and J. R. Kim (1999) Genetic algorithms for solving network design problem: state-of-the-art-survey, *Evolutionary Optimization*, Vol.1, No.2, pp. 121-141.
14. Gen, M., Y. Z. Li, and K. Ida (1999) Solving Multi-objective Transportation Problem by Spanning Tree-based Genetic Algorithm, *IEICE Trans. on Fundamentals*, Vol.E82-A, No.12, pp.2802-2810
15. Gen, M. and Cheng, R. (2000) *Genetic Algorithms and Engineering Optimisation*, John & Wiley Sons, NY.
16. Gen, M., Y. Z. Li, and K. Ida (2000) Spanning Tree-based Genetic Algorithm for Bicriteria Fixed Charge Transportation Problem, *J. of Japan Soc. for Fuzzy Theory & Systems*, Vol.12, No.2, pp.295-303.
17. Gen, M., J. Choi and K. Ida (2000) Improved Genetic Algorithm for General Transportation Problem, *Artificial Life and Robotics*, Vol. 4, pp.96-102.
18. Gen, M., G. Zhou and M. Takayama (2000) Matrix-based Genetic Algorithm Approach on Bicriteria Minimum Spanning Tree Problem with Interval Coefficients, *J. of Japan Society for Fuzzy Theory and Systems*, Vol. 10, No. 6, pp.643-656.
19. Gen, M., R. Cheng and S. Oren (2001) Network Design Techniques using Adapted Genetic Algorithms, *Advances in Engineering Software*, Vol. 32, pp. 731-744.
20. Gen, M., L. Lin, and R. Cheng. (2003) Priority-based Genetic Algorithm for Bicriteria Network Optimization Problem, *Proc. of The 4th International Symposium on Advanced Intelligent Systems*, Jeju, Korea.
21. Gen, M., Kim, K. and G. Yamazaki (2003) Multistage-based Hybrid Genetic Algorithm for Scheduling in FMS Environments, *Proceedings. of International Conf. on Production Research*, Blacksburg, USA.
22. Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading.
23. Gong, D., Gen, M., Yamazaki, G., and Xu, W. (1995) Hybrid evolutionary method for obstacle location-allocation problem, *Computers and Industrial*

Engineering, Vol. 29, No.1-4, pp.525-530.

24. Gong, D., Gen, M., Yamazaki, G., and Xu, W. (1997) Hybrid evolutionary method for capacitated location-allocation, *Engineering Design & Automation*, Vol. 3, No. 2, pp.166-173.
25. Gottlieb, J. and L. Paulmann, (1998) Genetic Algorithms for the Fixed Charge Transportation Problem, *Proceedings of IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, 330-335.
26. Hall, L. (1996) Experience with a Cutting Plane Algorithm for the Capacitated Minimal Spanning Tree Problem, *INFORMS Journal on Computing*, Vol. 8, No. 3, pp.219-234.
27. Holland, J. H. (1975) *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.
28. Jensen, A. P. and Barnes, J. W. (1980) *Network Flow Programming*, John Wiley, New York.
29. Kim, J. R., Gen, M. and Ida, K. (1999) Bicriteria Network Design using Spanning Tree-based Genetic Algorithm, *Artificial Life and Robotics*, Vol. 3, pp.65-72.
30. Kim, J. R. (2000) *Study on Advanced Genetic Algorithms for Reliable Network Design*, PhD dissertation, Ashikaga Institute of Technology.
31. Kim, K., Gen, M. and G, Yamazaki (2003) Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling, *Applied Soft Computing*, Vol.2, No.3, pp. 174-188.
32. Kim, K., Gen M., C. Moon and G, Yamazaki (2003) Constraints-Based Scheduling in MTO Manufacturing using Hybrid GA, *Proc. of International Conf. on Production Research*, Blacksburg, USA.
33. Kim, K., Y. Yun, J. Yoon, M. Gen, and G. Yamazaki (2003) Hybrid Genetic Algorithm with Adaptive Abilities for Resource-constrained Multiple Project Scheduling, submitted.
34. Kusiak, A. and Finke, G. (1988) Selection of process plans in automated manufacturing systems, *IEEE Journal of Robotics and Automation*, Vol. 4, No. 4, pp.397-402.
35. Li, Y. Z. and M. Gen (1997) Spanning Tree-based Genetic Algorithm for Bicriteria Transportation Problem with Fuzzy Coefficients, *Australian J. of Intelligent Inform. Processing Systems*, Vol. 4 No. 3/4, pp.220-229.
36. Li, Y. Z. (1999) *Study on Hybridized Genetic Algorithm for Production Distribution Planning Problems*, PhD dissertation, Ashikaga Institute of Technology.
37. Malik, K. and Yu, G. (1993) A Branch and Bound Algorithm for the Capacitated Minimum Spanning tree Problem, *Networks*, vol. 23, pp. 525-532.
38. Narula, S. C. and Ho, C. A. (1980) Degree-constrained minimum spanning tree. *Computers & OR*, vol. 7, pp.239-249.
39. Sun, M., Aronson, J. E. Mckeown, P. G. and Drinka, D. (1998) A Tabu Search Heuristic Procedure for the Fixed Charge Transportation Problem, *European J. of Operational Research*, Vol. 106, pp.441-456.
40. Vignaux, G. A. and Michalewicz, Z. (1991) A genetic algorithm for the linear transportation problem, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, pp.445-452.
41. Walters, G. A. and Smith, D. K. (1995) Evolutionary design algorithm for optimal layout of tree networks, *Engineering Optimization*, Vol. 24, pp.261-281.
42. Wang, P.T., G.S. Wang, and Z.G. Hu, Speeding Up the Search Process of Genetic Algorithm by Fuzzy Logic, *Proceedings of the 5th European Congress on Intelligent Techniques and Soft Computing*, pp. 665-671, 1997.
43. Whatley, J. K. (1985) *SAS/OR User's Guide: Version 5. Netflow Procedure*, SAS Institute Inc., Cary, NC, pp.211-223.
44. Yang, J. B. (2001) GA-Based Discrete Dynamic Programming Approach for Scheduling in FMS Environment, *IEEE Trans. on Systems, Man, and Cybernetics-Pt B: Cybernetics*, Vol. 31, No. 5, pp. 824-835.
45. Zhou, G. and M. Gen (1997) A note on genetic algorithm approach to the degree-constrained spanning tree problems, *Networks*, Vol. 30, pp.105-109.
46. Zhou, G. and Gen, M. (1997) An effective genetic algorithm approach to the quadratic minimum spanning tree problem, *Computers & Operations Research*, Vol. 25, No. 3, pp.229-237.
47. Zhou, G. and Gen, M. (1997) Evolutionary computation on multicriteria production process planning problem," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation*, edited by T. B k, Z. Michalewicz and X. Yao, IEEE Press, Indianapolis, pp.419-424.
48. Zhou, G. and Gen, M. (1999) Genetic Algorithm Approach on Multicriteria Minimum Spanning Tree Problem, *European J. of Operational Research*, Vol. 114, No. 1, pp.141-152.
49. Zhou, G. (1999) *Study on Constrained Spanning Tree Problems with Genetic Algorithms*, PhD dissertation, Ashikaga Institute of Technology.
50. Zhou, G. and Gen, M. (2003), A Genetic Algorithm Approach on Tree-like Telecommunication Network Design Problem, *J. of Operational Research Society*, Vol. 54, No. 3, pp.248-254.