CHAPTER 1

INTRODUCTION

There are myriad search and optimization techniques for optimization problems in the world. Researchers in economics, political science, psychology, linguistics, immunology, biology, and computer science need an efficient tool to tackle their optimization problems. It is difficult, however, to model realistic systems because the behavior of the systems is complex. In general, an optimization problem to be addressed has several objectives to be optimized. Thus, the complexity of the problem increases as the number of objectives increases because the objectives considered are often contradictory to one another. Such complex optimization problems have a lot of feasible solutions. However, only a few solutions among them are desirable.

In order to use an optimization technique for such complex optimization problems without difficulties, the technique should be robust. Goldberg defined robustness in his book [23] as “the balance between efficiency and efficacy necessary for survival in many different environment.” Then we can define two purposes in constructing an optimization technique as its efficacy and efficiency. Efficacy means whether the optimization technique can reach the optimum or not. The common purpose in constructing optimization techniques is this efficacy, that is, their convergence to the optimum of the problem. The other purpose, efficiency, means whether the technique can find a better solution under the constraints the problem has. The technique may not find the optimal solution of the problem due to the constraints, but it is important that better solutions are searched by the algorithm within the constraints. From this point of view, all search techniques are not robust because some search technique tends to find only the local optimum due to its local scope, depends on existence of derivatives, or requires enormous computation time. Therefore Goldberg concluded that “the most important goal of optimization is improvement. ... Attainment of the optimum is much less important for complex systems.” As for complex systems, Zadeh also said in [121], “most realistic problems tend to be complex, and many complex problems are either algorithmically unsolvable or, if solvable in principle, are computationally infeasible." Thus, robust algorithms which can find better solutions under a lot of constraints are required for optimizing complex systems.

The central theme of research on genetic algorithms (GAs) [9,23,27] has been robustness. Genetic algorithms, first specified by John Holland in the early 1970's [27], are becoming an important tool for combinatorial optimization, function optimization, and machine learning. GAs are a kind of (i) stochastic search, (ii) multi-point search, (iii) direct search, and (iv) parallel search. These characteristic features of GAs contribute robustness of the algorithms. While it is easy to apply GAs to optimization problems, several researchers [22,49,78,82] pointed out that the performance of GAs on some combinatorial optimization problems was a bit inferior to that of neighborhood search algorithms (*e.g.*, local search, simulated annealing [90], and tabu search [108,119]). Therefore hybridization of GAs with other heuristic methods is required for improving the performance of GAs.

Genetic algorithms have been mainly applied to single-objective optimization problems. In order to handle multi-objective optimization problems, the objective functions should be combined into a scalar fitness function. But the characteristic features of GAs can be utilized for the search in the feasible region of multi-objective optimization problems. Since Schaffer's work [98], extensions of GAs to multi-objective optimization problems were proposed in several manners (*e.g.*, see Fonseca & Fleming [14,15], Horn *et al.*[30], Kita *et al.*[60], Kursawe [63], Murata & Ishibuchi [77], and Tamaki *et al.* [111,112]). In their papers, the ability of GAs to address multi-objective optimization problems is well described.

In this dissertation, genetic algorithms for optimization problems are considered. First we apply GAs to optimization problems where only a single objective is considered. In order to improve the performance of the GAs, hybrid algorithms of GAs with other search algorithms are attempted. Then we extend GAs to multi-objective optimization problems. In the same manner as the hybridization of GAs for single-objective problems, we hybridize multi-objective genetic algorithms with some heuristics in order to improve their performance. We apply GAs to flowshop scheduling and fuzzy rule selection. The former is a kind of permutation problem. For many problems like scheduling problems and traveling salesman problems, a permutation of a set of numbers is encoded as a string which is governed by genetic operators. The latter is a kind of knapsack problem where elements are selected in order to heighten the total value of elements in a knapsack. For this kind of problem, a binary string is often used as an individual which codes a solution of the problem. Both problems are combinatorial optimization problems, but the coding method of the solution is usually different from each other. In this dissertation, we apply genetic algorithms to each of these two problems with a single objective and with multiple objectives.

The remainder of this thesis are organized as follows.

In Chapter 2, we first explain the basic scheme of GAs. A simple genetic algorithm with a single objective is described. Next, genetic operators for multi-objective optimization are introduced in order to design a multi-objective genetic algorithm. Using a simple test problem, we compare our multi-objective genetic algorithm (MOGA) with several genetic algorithms for multi-objective optimization. In general, when an algorithm is applied to multi-objective optimization problems, it is important whether the algorithm works well for problems with non-convex feasible regions in objective spaces or not. By using another test problem with a non-convex feasible region, we demonstrate that the MOGA consisting of the modified genetic operators can find non-dominated solutions of such problem.

In Chapter 3, we apply GAs to single-objective flowshop scheduling problems. We first examine several crossover operators and mutation operators to construct genetic algorithms for flowshop scheduling. By computer simulations, we point out that the combination of high performance crossover and mutation operators does not always lead to a high performance genetic algorithm. Next, we compare the genetic algorithm constructed for flowshop scheduling with other search algorithms such as local search, simulated annealing [90], and tabu search [108,119]. It is shown that the genetic algorithm is a bit inferior to the other search algorithms. Then, we examine two hybrid genetic algorithms for improving the performance of the genetic algorithm. One is a genetic local search algorithm and the other is a genetic simulated annealing algorithm. We also introduce some modifications of search mechanisms in these hybrid genetic algorithms. While careful parameter specifications are required for constructing GAs with high performance, it is shown that we can construct the genetic local search algorithm without careful parameter specifications.

Chapter 4 deals with the application of GAs to multi-objective flowshop scheduling problems. We demonstrate the effectiveness of the MOGA on a flowshop scheduling problem with two objectives and a problem with three objectives. We also hybridized our MOGA with a local search algorithm in the same manner as in Chapter 3. The effectiveness of the hybrid algorithm is shown by some computer simulations.

In Chapter 5, we consider GAs for designing fuzzy classification systems with two objectives: one is to maximize the number of correctly classified training patterns by selected rules and the other is to minimize the number of the selected rules. By combining these two objectives into a single scalar fitness function using constant weights, a single-objective genetic algorithm can be applied to the problem. We combine a kind of learning procedure with the genetic algorithm for rule selection in order to improve performance of the constructed classification system. Computer simulations show the effectiveness of the GAs for rule selection. Next we describe another kind of genetic-algorithm-based approach to the construction of fuzzy classification systems where both the number of fuzzy rules and the membership function of each antecedent fuzzy set are determined simultaneously. We also hybridize the genetic algorithm with a learning procedure to improve performance of the constructed classification system.

In Chapter 6, the MOGA is applied to multi-objective fuzzy rule selection problems. We compare the MOGA with some single-objective genetic algorithms which are implemented to find non-dominated solutions of this problem. We combine a learning procedure with the MOGA to get a better set of non-dominated solutions. Then we modify the genetic-algorithm-based multi-objective fuzzy rule selection method for handling high-dimensional pattern classification problems with many continuous attributes. Simulation results show the applicability of our modified method to high-dimensional pattern classification problems.

Last, we summarize the results of this dissertation in Chapter 7.