# CHAPTER 7

# CONCLUSIONS

In this chapter, we summarize the results obtained in the previous chapters. First we applied GAs to optimization problems where only a single objective is considered. In order to improve the performance of the GAs, hybridization of the GAs with other algorithms were attempted. Then we extended the GAs to multi-objective optimization problems. In the same manner as in the hybridization of the GAs with a single objective, we hybridized multi-objective genetic algorithms with some heuristics in order to improve their performance. We applied the GAs and the hybrid GAs with a single objective and multiple objectives to flowshop scheduling problems and linguistic rule selection problems.

In Chapter 2, we first described a genetic algorithm with a single objective. Next, we introduced genetic operators for multi-objective optimization in order to extend GAs to multi-objective optimization. Using a simple test problem, we compared 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. Using another test problem with a non-convex feasible region, we demonstrated that the MOGA could find non-dominated solutions of such problems.

In Chapter 3 and Chapter 4, we applied genetic algorithms to flowshop scheduling problems.

In Chapter 3, we first examined several crossover operators and mutation operators to construct a genetic algorithm for single-objective flowshop scheduling problems. By computer simulations, we pointed out that the combination of high performance crossover and mutation operators did not always lead to a high performance genetic algorithm. Next, we compared the genetic algorithm constructed for flowshop scheduling with other search algorithms such as local search, simulated annealing [90], and tabu search [108,119]. It was shown that the genetic algorithm was a bit inferior to the other search algorithms. Therefore, we introduced 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 introduced some modifications of search mechanisms in these hybrid genetic algorithms. While careful parameter specifications were required for constructing a high-performance genetic algorithm, it was shown that we could construct a high-performance genetic local search algorithm without careful parameter specifications.

Chapter 4 dealt with the application of GAs to multi-objective flowshop scheduling problems. We demonstrated the effectiveness of the MOGA on flowshop scheduling problems with two and 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 was shown by some computer simulations.

In Chapter 5 and Chapter 6, we applied genetic algorithms to a linguistic rule selection problem with the following two objectives: to maximize the number of correctly classified training patterns by selected rules, and to minimize the number of the selected rules.

In Chapter 5, we introduced a genetic algorithm with a single objective by combining these two objectives into a single scalar fitness function using constant weights. We combined a learning procedure with the genetic algorithm for the rule selection in order to improve performance of constructed fuzzy classification systems. Computer simulations showed the effectiveness of the GAs for the linguistic rule selection. Then we described another kind of genetic-algorithm-based method to the construction of fuzzy classification systems where both the number of fuzzy rules and the membership functions of antecedent fuzzy sets are determined simultaneously. We also hybridized the genetic algorithm with a learning procedure to improve performance of constructed fuzzy classification systems.

In Chapter 6, the MOGA was applied to multi-objective fuzzy rule selection problems. We compared the MOGA with some genetic algorithms with a single objective which find a set of non-dominated solutions. We also combined a learning procedure with the MOGA to get a better set of non-dominated solutions. Then we modified the genetic-algorithm-based multi-objective linguistic rule selection method for handling high-dimensional pattern classification problems with many continuous attributes. Simulation results showed the applicability of our modified method to high-dimensional pattern classification problems.