Fitness Switching Strategy for Developing Genetic Algorithm that Utilizes Infeasible Solutions

Kim Jun Woo
Department of Industrial and Management Systems Engineering
Dong-A University
Busan, South Korea
e-mail: kjunwoo@dau.ac.kr

***Abstract*—This paper introduces a general search strategy for genetic algorithm, which is called fitness switching. This strategy is developed to utilize the infeasible solutions during search procedure, and it provides two important benefits. First, it helps to find good solutions more effectively, since useful infeasible solutions can be exploited. Second, conventional feasibility handling strategies such as repair and penalization are not needed in fitness switching genetic algorithm, where fitness switching strategy is applied. Moreover, this strategy can be applied to a wide range of combinatorial optimization problems, while repair and penalization procedures are typically problem-specific.**

Keywords-genetic algorithm; fitness switching strategy; combinatorial optimization; meta heuristic; infeasible solution

I. INTRODUCTION

Genetic Algorithm (GA), proposed by Holland, is a well-known meta heuristic search method for solving combinatorial optimization problems [1]. Typically, meta heuristic search methods provide general search methodologies for exploring the search space of given problem effectively, and the search methodology of GA is usually defined by three genetic operators, selection, crossover, and mutation [2]. While the search methodology of GA is generally applicable to various problems, the genetic operators must be tailored to a specific problem, which is sometimes very difficult [3][4].

Feasibility is an important factor that can increase the complexities of genetic operators in that the infeasible solutions are not considered by conventional GAs. There are two approaches for handling the solution feasibility. One is to use carefully designed genetic operators which does not produce infeasible solutions at all, and the other is to apply additional procedures such as repair and penalization [5]. However, both approaches have two important limitations. First, they do not allow the infeasible solutions to be included within population, while such solutions can sometimes contain some features useful for finding better solutions. Second, both approaches are problem-specific, and complex genetic operators or additional procedures can be required.

Fitness switching can be used to address such problems, although it has been initially developed to solve specific combinatorial optimization problem with rare feasible solutions [5][6]. In this context, this paper introduces generalized form of fitness switching strategy and its application examples.

The remainder of this paper is organized as follows: In Section 2, the generalized structure of fitness switching strategy is introduced. The application examples of the strategy are illustrated in Section 3, and finally, the concluding remarks follow in Section 4.

II. FITNESS SWITCHING GENETIC ALGORITHM

Fitness switching strategy is characterized by three additional procedures, fitness switching, fitness leveling and simple local search, which are generally applicable to various combinatorial optimization problems [5][6][7].

A. *Fitness Switching*

Let us assume that desirability of a solution s can be measured by a function $X(s)$. For example, total value of a solution for knapsack problem and total length of a solution for traveling salesman problem can be used as $X(s)$. If a maximization problem is given, we have to increase the value of $X(s)$. However, too large $X(s)$ is typically obtained by infeasible solutions. Consequently, we have to decrease the value of $X(s)$ if s is infeasible, and fitness switching procedure suggests that

$$fitness^{+}(s) \propto X(s) \propto \frac{1}{fitness^{-}(s)}, \tag{1}$$

where fitness value of s, $fitness(s)$, is computed as follows:

$$fitness(s) = \begin{cases} fitness^{+}(s) & \text{, if } s \text{ is feasible} \\ fitness^{-}(s) & \text{, if } s \text{ is infeasible} \end{cases} \tag{2}$$

Note that (1) indicates that feasible solutions are enhanced when their fitness values increase, while infeasible ones are enhanced by decreasing their fitness values. Of course, fitness switching procedure can be written in additive form, for example, $fitness^{+}(s)$ = $X(s)$ and

 ISBN: 978-1-61208-655-2

$fitness^{-}(s) = T - X(s)$. However, we have to determine the value of additional parameter T in this case.

The fitness switching procedure proposed in this paper is applied to evaluation phase of standard GA (SGA) [8]. For details on the original version of FSWGA based on SGA, see Fig. 3 in [5].

B. Fitness Leveling

It is straightforward that the fitness of a feasible solution should be larger than the fitness of an infeasible one. This is satisfied if $fitness^{+}(s) \geq 0$. However, too large difference between $fitness^{+}(s)$ and $fitness^{-}(s)$ is not desirable in that it can cause too high selection pressure.

Fitness leveling procedure is used to maintain appropriate selection pressure by adjusting $fitness(s)$ as follows:

$$fitness'(s) = \begin{cases} fitness^{+}{}'(s) & \text{, if } s \text{ is feasible} \\ fitness^{-}{}'(s) & \text{, if } s \text{ is infeasible} \end{cases}, \quad (3)$$

where

$$fitness^{+}{}'(s) = 1 + L \times \frac{fitness(s) - \min_{x \in F} fitness(x)}{\max_{x \in F} fitness(x) - \min_{x \in F} fitness(x)} \quad (4)$$

and

$$fitness^{-}{}'(s) = (1-\alpha) \times \frac{fitness(s)}{\max_{x \in I} fitness(x)} \quad (5)$$

F and I denote the sets of feasible and infeasible solutions within population, respectively. Moreover, factor L (≥ 1) defines the relative desirability of feasible solutions, while factor α ($0 \leq \alpha < 1$) is used to guarantee that $fitness^{-}{}'(s) < fitness^{+}{}'(s)$. Consequently, $1 \leq fitness^{+}{}'(s) \leq L$ and $0 \leq fitness^{-}{}'(s) \leq 1$, if and only if $fitness^{+}(s) \geq 0$.

Note that fitness leveling procedure is not needed if current population consists of only feasible solutions or only infeasible ones, and this procedure can be incorporated into evaluation or selection phase of SGA.

C. Simple Local Search

Fitness Switching GA (FSWGA) allows infeasible solutions to be included within population. However, they are not suitable for solving given problems, inherently. In this context, the infeasible solutions can be slightly modified by applying simple local search procedure in hopes that they would be converted into better solutions, not necessarily feasible.

Unlike fitness switching and fitness leveling, this procedure is optional and problem-specific. This procedure is incorporated into evaluation phase of SGA.

III. APPLICATION EXAMPLES

FSWGA has been applied to Maze-type shortest path problem and 0-1 knapsack problem, and the details of fitness switching strategy for those problems are summarized in Table 1.

Table 1 indicates that $fitness^{-}(s)$ can be defined flexibly, as long as it is inversely proportional to $fitness^{+}(s)$. Moreover, no repair and penalization procedure are needed, and FSWGA has successfully solved given problem in both cases. For example, Fig. 2 shows the experiment result of FSWGA for maze-type shortest path problem with a maze-type network as shown in Fig. 1, where node 1 and node 27 are source node and destination node, respectively, and lengths of all edges are assumed to be 1 [5]. Then, it is straightforward that the optimal path from node 1 to node 27 is <1, 9, 11, 15, 26, 27> with length 5, while there are some competitive local optima such as <1, 9, 11, 15, 16, 26, 27> and <1, 2, 9, 11, 15, 26, 27> with length 6. Moreover, the network contains a number of dead-ends such as node 6, 8, and 10, etc. and we have many infeasible paths that fail to arrive at the destination node, such as <1, 2, 4, 6> and <1, 3, 5, 8>.

Nevertheless, FSWGA found the optimal solution successfully as shown in Fig. 2, where the search procedure of FSWGA for combinatorial optimization problems with rare feasible solutions consists of three periods. In initial period, there is no feasible solution in population, since it is not easy to find any feasible ones from scratch. During initial period, FSWGA aims to find longer paths in hopes that some feasible paths that arrives at the destination node would be found.

TABLE I. APPLICATION OF FITNESS SWITCHING STRATEGY.

Target problem	**Maze-type Shortest Path Problem [5][6]**	**0-1 Knapsack Problem [7]**
Problem type	Rare feasible solutions	Many feasible solutions
Objective	To find the shortest feasible path from source node to destination node	To find a set of items with maximum total value, satisfying pre-specified total weight limit
Feasible solution	A path from source node to destination node	A set of items with total weight does not exceed pre-specified upper limit
Infeasible solution	A path from source node to a non-destination node	A set of items that total weight exceeds pre-specified upper limit
$X(s)$	Length of path s	Total value of a set of some items s
$fitness^{+}(s)$	$\frac{\text{sum of all edges' lengths}}{\text{length of path}}$	Total value
$fitness^{-}(s)$	$\frac{\text{length of path}}{\text{sum of all edges' lengths}}$	(1) 1/ total value (2) 1/ total weight (3) 1/ total (value × weight)
Simple local search	Randomly modify the last move	Exclude a randomly chosen item

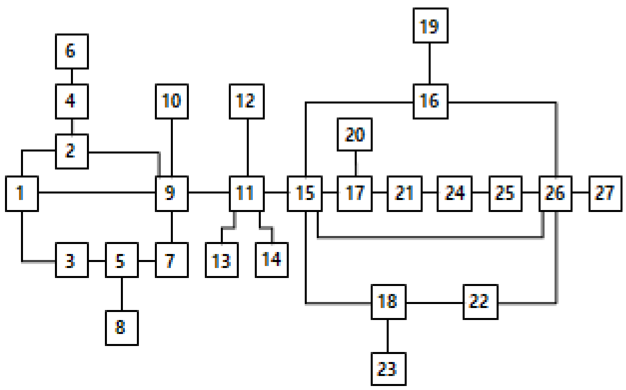

Figure 1. Example of a maze-type network [5].

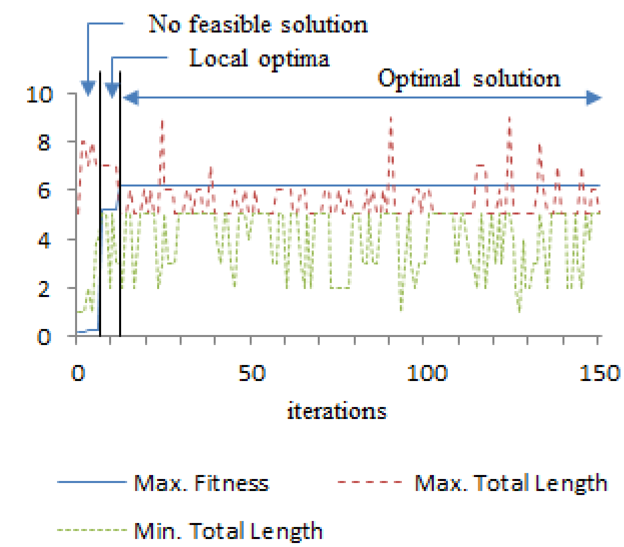

Figure 2. Experiment result of FSWGA for maze-type shortest path problem [5].

The second period begins when any feasible path is found, and FSWGA focuses on finding shorter feasible paths in this period. Finally, the optimal solution, the shortest feasible path is identified and maintained during the last period. Note that conventional GAs for classical shortest path problems have failed to find the optimal solution for the maze-type network shown in Fig. 1. On the contrary, Fig. 3 shows the experiment result of FSWGA for classical 0-1 knapsack problem with 50 items [7], which has many feasible solutions. In other words, it is easy to generate a number of feasible solutions that contain few items, and the graph in Fig. 3 represents the change in maximum total value, total value of the best feasible solution within current population. In this case, we can see that the initial population also has a number of feasible solutions and the maximum total value continuously increases until the optimal solution is found.

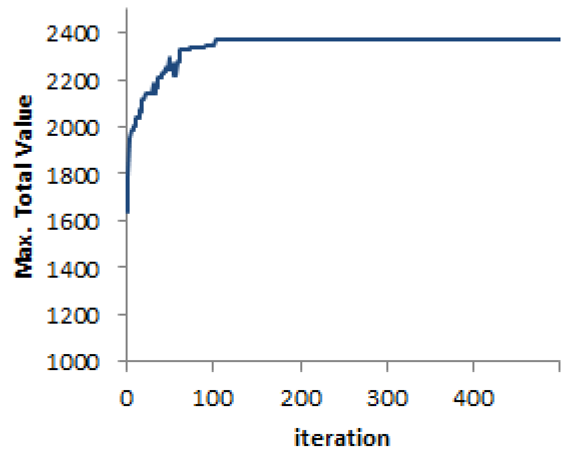

Figure 3. Experiment result of FSWGA for 0-1 knapsack problem [7].

The maze-type shortest path problem and 0-1 knapsack problem are quite different from each other for two reasons: (i) maze-type shortest path problem is inherently a sort of sequencing problem, but 0-1 knapsack problem is not. (ii) maze-type shortest path problem has rare feasible solutions, while 0-1 knapsack problem typically has many feasible solutions. Nevertheless, both problems have been successfully addressed by applying the fitness switching strategy, and we can conclude that the strategy can be widely applied to various combinatorial optimization problems.

IV. CONCLUSIONS

FSWGA utilizes infeasible solutions during search procedure, and it can be easily implemented. The fitness switching strategy is easy to implement and widely applicable in that it is applied to fitness values and solutions are not modified, and parameters are relatively intuitive. In this context, it will help to explore the search spaces of various combinatorial optimization problems efficiently.

Although the fitness switching strategy has been applied only to two types of combinatorial optimization problems, maze-type shortest path problem with rare feasible solutions and 0-1 knapsack problem with many feasible solutions, yet, the author plans to apply the strategy to various problems with complex constraints, in order to demonstrate its applicability.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(Ministry of Science, ICT & Future Planning) (NRF-2017R1C1B1008650).

REFERENCES

[1] J. H. Holland, Adaptations in Natural and Artificial Systems, University of Michigan Press, Michigan, 1975.

[2] J. W. Kim, "Candidate Order based Genetic Algorithm (COGA) for Constrained Sequencing Problems," Int. J. Ind. Eng-Appl. P. USA, vol. 23, pp. 1-12, 2016.

[3] K. Chakhlevitch and P. Cowling, “Hyperheuristics: Recent Developments,” In Adaptive and Multilevel Metaheuristics, Springer Berlin Heidelberg, pp. 3-29, 2008.

[4] E. Burke et al., “Hyper-heurisrics: A Survey of the State of the Art,” J. Oper. Res. Soc. Am. USA, vol. 64, pp. 1695-1724, 2013.

[5] J. W. Kim and S. K. Kim, “Fitness Switching Genetic Algorithm for Solving Combinatorial Optimization Problems with Rare Feasible Solutions,” J. Supercomput. Netherlands, vol. 72, pp. 3549-3571, 2016.

[6] J. W. Kim and S. K. Kim, “Genetic Algorithms for Solving Shortest Path Problem in Maze-Type Network with Precedence Consrtraints,” Wireless Pers. Commun. Netherlands, in press.

[7] J. W. Kim, “Application of Fitness Switching Genetic Algorithm for Solving 0-1 Knapsack Problem,” unpublished.

[8] Z. Michalewicz, Genetic Algorithm + Data Structure = Evolution Programs, 3rd ed., Springer-Verlag, New York, 1996.