Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 14, Number 1, pages 55–57, January 2025

Bee colony optimization for robust solutions of multi-objective knapsack problems

Kido Masaki Akihiro Fujiwara
Kyushu Institute of Technology
Iizuka, Fukuoka 820-8502, Japan
kido.masaki114@mail.kyutech.p, fujiwara@csn.kyutech.ac.jp

***Abstract*—In the multi-objective optimization problems, errors due to parameter variations may occur in the real world, and it is useful to find a solution that minimizes such errors. Solutions with high stability against such uncertainties are called robust solutions.**

In this study, we propose a bee colony optimization algorithm for finding robust Pareto optimal solutions to the multi-objective knapsack problem. We implement the proposed method in an experimental environment to verify the usefulness of the proposed method. In this experiment, we compare the proposed method with an existing algorithm and show that our proposed algorithm obtains a better robust Pareto optimal solution for the multi-objective knapsack problem.

I. Introduction

An optimization problem with one objective function is called a single-objective optimization problem, and there may exist a unique optimal solution for a single-objective optimization depending on the objective function. On the other hand, an optimization problem with multiple objective functions is called a multi-objective optimization problem. For multi-objective optimization problems, there is not a single optimal solution because there exists a trade-off between two or more conflicting objective functions.

For multi-objective optimization problems, the concept of Pareto-optimal solutions [4] was introduced to discuss optimal solutions. The set of Pareto-optimal solutions is the set of solutions that are not inferior to other solutions and is defined by the dominance relation of the solutions. Since the problem of computing the maximal Pareto-optimal solutions for a multi-objective optimization problem is generally computationally hard, approximation algorithms are needed, and a number of optimization algorithms that compute Pareto-optimal solutions have been proposed for the multi-objective 0-1 knapsack problem as a representative multi-objective optimization problem.

On the other hand, the solution obtained using a simple optimization is affected by changes in variables in the real world due to errors and other factors, and the predicted optimality of the solution may not be guaranteed. Therefore, the concept of a robust solution, which is a solution that is less affected by changes in variables, for multi-objective optimization problems was proposed in [2]. Specifically, two types of robust solutions were considered in [2]. One type of robust solutions optimizes an average of the values of the objective functions for neighboring solutions. The other type of robust solutions ensures robustness in the sense that the difference between an objective function and the average of the objective function over neighboring solutions is less than a given constant.

As an example of robust solutions to a multi-objective optimization problem, an optimization algorithm based on a population protocol is proposed in [5]. The population protocol is a computational model for a mobile sensor network with limited computing power, in which multiple agents communicate with each other and perform state transitions.

In the present paper, we propose a bee colony optimization algorithm for considering both types of robustness for the multi-objective 0-1 knapsack problem. We implement the proposed algorithm and a previous algorithm [5] in a simulation environment to evaluate the validity of the proposed algorithm. The experimental results show that the proposed algorithm obtains a better set of Pareto optimal solutions than the previous algorithm.

II. Preliminaries

A. Multi-objective optimization problem

We assume that an instance of the problem is m-dimensional decision variables (vector) $\boldsymbol{x}$. The multi-objective optimization problem consists of a set of n objective functions $\{f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), \cdots, f_{n-1}(\boldsymbol{x})\}$ and a set of k constraint functions $\{g_0(\boldsymbol{x}), g_1(\boldsymbol{x}), \cdots, g_{k-1}(\boldsymbol{x})\}$. Then, each objective function is defined as image from $\boldsymbol{x}$ to n objective function vector $\boldsymbol{y}$. The definition is mathematically formulated as follows.

$$\begin{aligned}
&\max/\min\ \boldsymbol{y} = \{f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), ..., f_{n-1}(\boldsymbol{x})\} \\
&\quad \text{such that } \boldsymbol{x} = (x_0, x_1, ..., x_{m-1}) \in X, \\
&\qquad\qquad X = \{\boldsymbol{x} \mid \forall i \in \{0, 1, ..., k-1\}, g_i(\boldsymbol{x}) \leq 0\}
\end{aligned}$$

In the above definition, X is called the set of feasible solutions for the problem.

Since no solution is optimal for all of the objective functions in the multi-objective optimization problem, only a solution that is not inferior to the other solutions is needed. Such a solution is called a *Pareto-optimal solution*, and we first define the dominance relationship of the solutions in order to define a Pareto-optimal solution.

We assume that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are two feasible solutions for the problem and that we want to maximize all the objective functions. Then, $\boldsymbol{x_2}$ dominates $\boldsymbol{x_1}$ if and only if the following two conditions hold.

$$\forall i \in \{0, 1, ..., n-1\}, f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$$
$$\exists j \in \{0, 1, ..., n-1\}, f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$$

In this paper, $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$ denotes that $\boldsymbol{x}_2$ dominates $\boldsymbol{x}_1$.

In addition, a feasible solution $\boldsymbol{x}$ is called *Pareto optimal* if and only if there is no feasible solution $\boldsymbol{x'} \in X$ such that $\boldsymbol{x} \prec \boldsymbol{x'}$. Since a Pareto-optimal solution is a solution that cannot be improved for any of the objective functions without degrading one of the other functions, the maximal set of Pareto-optimal solutions is considered the optimal solution for a multi-objective optimization problem.

There are various metrics for a set of Pareto-optimal solutions for a multi-objective optimization problem. In this paper, we evaluate a set of Pareto-optimal solutions using a *hypervolume indicator* [1]. Let h_x be the volume of the hypercube created by Pareto-optimal solution $\boldsymbol{x}$ and the reference point $\boldsymbol{r}$. The hypervolume H for a set of Pareto-optimal solutions is defined by $H = \bigcup_{\boldsymbol{x} \in X} h_x$.

B. Multi-objective 0-1 knapsack and robust solutions

The input of the multi-objective 0-1 knapsack problem is as follows.

- n knapsacks whose capacities are $c_0, c_1, \cdots c_{n-1}$.
- m items stored in the knapsacks. $p_{i,j}$ and $w_{i,j}$ denote value and weight of item j for knapsack i, respectively.

Let $\boldsymbol{x} = (x_0, x_1, \cdots, x_{m-1})$ be an m-dimensional Boolean vector. Then, the multi-objective 0-1 knapsack problem is formulated as follows.

$$\max \ \boldsymbol{y} = \{f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), ..., f_{n-1}(\boldsymbol{x})\}$$
$$f_i(\boldsymbol{x}) = \sum_{j=0}^{m-1} p_{i,j} x_j$$
$$\text{such that} \sum_{j=0}^{m-1} w_{i,j} x_j \leq c_i \quad (0 \leq i \leq n-1)$$

For solutions of multi-objective optimization problems, the previously mentioned two types of robustness defined in [2] are here called Type I and Type II, both of which guarantee that a solution is robust against variable perturbation. Using these definitions, we define robust solutions for the multi-objective 0-1 knapsack problem.

Type I robustness: A robust solution of Type I is obtained by optimizing the mean effective function, $f_i^{eff}(\boldsymbol{x})$, which is defined as follows.

$$f_i^{eff}(\boldsymbol{x}) = \frac{1}{N_s} \sum_{g=0}^{N_s-1} \sum_{j=0}^{m-1} p_{i,j} x_j^g$$

In the above definition, N_s is the number of neighborhood solutions $\boldsymbol{x}$: $\boldsymbol{x}^0, \boldsymbol{x}^1 ..., \boldsymbol{x}^{N_s-1}$.

Using the mean effective function, a robust solution of Type I for the multi-objective 0-1 knapsack problem is defined as follows.

$$\max \ \boldsymbol{y} = \{f_0^{eff}(\boldsymbol{x}), f_1^{eff}(\boldsymbol{x}), ..., f_{n-1}^{eff}(\boldsymbol{x})\}$$
$$\text{such that} \sum_{j=0}^{m-1} w_{i,j} x_j^g \leq c_i \quad (0 \leq i \leq n-1, 0 \leq g \leq N_s - 1)$$

Type II robustness: A robust solution of Type II is defined with the mean effective function and an additional constraint, a threshold η. Threshold η is used to guarantee that the difference between the objective function and the mean effective function at the solution is less than η.

A robust solution of Type II for the multi-objective 0-1 knapsack problem is defined as follows.

$$\max \ \boldsymbol{y} = \{f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), \cdots, f_{n-1}(\boldsymbol{x})\}$$
$$\text{such that} \sum_{j=0}^{m-1} w_{i,j} x_j^g \leq c_i \quad (0 \leq i \leq n-1, 0 \leq g \leq N_s - 1)$$
$$\frac{f_i^{eff}(\boldsymbol{x}) - f_i(\boldsymbol{x})}{f_i(\boldsymbol{x})} \leq \eta \quad (0 \leq i \leq n-1)$$

III. Bee colony optimization for robust solutions of the knapsack problem

Bee colony optimization [3] is an optimization technique based on the property of living bees in a group. We show an outline of our algorithm for robust solutions of the multi-objective 0-1 knapsack problem using the bee colony optimization in the followings.

Algorithm: a bee optimization for robust solutions of the multi-objective 0-1 knapsack problem

Step 1: Each bee i $(0 \leq i \leq n-1)$ generates an initial solution $x_i = (x_{i,0}, x_{i,1}, \cdots, x_{i,m-1})$. (Each $x_{i,j}$ is randomly set to 0 or 1.) Then, values of objective functions are computed according to the definition of Type I or Type II robustness. After the computation of the values of all bees, the Pareto-optimal solution is computed from the solutions of the bees, and stored in a set $\boldsymbol{P}$.

Step 2: The following operations (2-1) $\sim$ (2-3) are repeated up to a maximum number of generations.

(2-1): This step simulates a habit of the employed bee. Each employed bee searches a better solution by exchanging information with other bees. Based on the habit of the employed bee, a new solution $\widehat{x_i}$ for bee i is computed using the following formula.

$$\widehat{x_{i,j}} = \begin{cases} x_{i,j} & (\text{if } x_{i,j} = x_{k_1,j} = x_{k_2,j}) \\ \overline{x_{i,j}} & (\text{otherwize}) \end{cases} \quad (1)$$

In the above equation, k_1 and k_2 are randomly selected two numbers of other bees.

Next, the adjustment process, which adjusts items in the knapsack according to capacity limitation, is performed on the obtained solution $\widehat{x_i}$. Then, each bee i computes values of objective functions according to the definition of Type I or Type II robustness again, and the solution is added to the

Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 14, Number 1, pages 55–57, January 2025

TABLE I
HYPERVOLUMES FOR ROBUST SOLUTIONS

	Type I robustness	Type II robustness
Bee colony optimization	3.8992×10^8	3.9121×10^8
Population Protocol [5]	3.873×10^8	3.8862×10^8

set $\boldsymbol{P}$. Finally, Pareto optimal solution is computed for $\boldsymbol{P}$.

(2-2): This step simulates a habit of the onlooker bee. Each onlooker bee chooses a better solution near good solutions obtained by the other bees. Based on the habit of the onlooker bee, computation is executed as follows.

The onlooker bee chooses one of two solutions, x_{gb} and x_k. x_{gb} is a solution with the closest Hamming distance to the solution of the bee. On the other hand, x_k is a solution that is randomly chosen from the other bees.

Let $\widehat{x_i}$ be a new solution for the onlooker bee i. The first choice is executed according to the following formula. (r is a real random number between 0 and 1.)

$$\widehat{x_i} = \begin{cases} x_{gb} & (r < 0.9) \\ x_k & (\text{otherwise}) \end{cases}$$

Then, $\widehat{x_i}$ is modified as follows. First, two variables in $\widehat{x_i}$, $x_{i,u} = 1$ and $x_{i,v} = 0$, are randomly selected. (In the current solution, item u is in the knapsack, and item v is not in the knapsack.) Next, the values of the two variables are reversed as $x_{i,u} = 0$ and $x_{i,v} = 1$. (The item u is interchanged with the item v.)

After the modification, the adjustment process is performed on the obtained solution $\widehat{x_i}$, and each bee i computes values of objective functions again for the solution $\widehat{x_i}$ according to the definition of Type I or Type II robustness. The solution is added to $\boldsymbol{P}$, and Pareto optimal solution is computed for $\boldsymbol{P}$.

(2-3): This step simulates a habit of the scout bee. Each scout bee searches a new solution randomly.

In this step, 1% of bees are changed into a scout bee, and each scout bee i abandons its solution and initializes the solution $\widehat{x_i}$. The initialization procedure is the same as Step 1.

Step 3: Output the final Pareto optimal solution set $\boldsymbol{P}$.

IV. EXPERIMENTAL RESULTS

Our proposed algorithm and a previous algorithm [5] are implemented using Python 3, and we compare Pareto-optimal solutions and hypervolume indicators.

Table I shows hypervolumes for solutions of the proposed algorithm and the previous algorithm in the case that the execution time is about 1800 seconds. The table indicates that our proposed algorithm obtains a wider range of solutions than the previous algorithm.

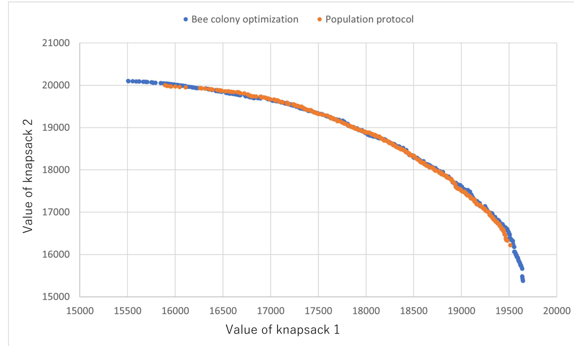

Fig. 1. Experimental results for Type I robustness

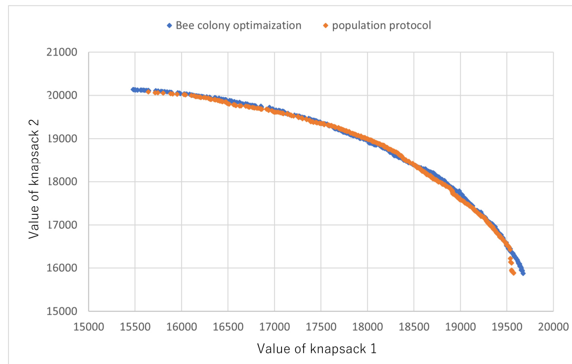

Fig. 2. Experimental results for Type II robustness

In addition, Fig. 1 and Fig. 2 show Pareto-optimal solutions for Type I and Type II robustness, respectively. These results indicate that the proposed algorithm is superior to the previous algorithm for computing solutions in case that one of the objective functions is highly weighted.

V. SUMMARY

In this study, we proposed an algorithm to find a robust Pareto-optimal solution set for the multi-objective knapsack problem using a bee colony optimization. Experimental results show that the proposed algorithm obtains a better set of robust Pareto-optimal solutions.

As our future research, we are considering cases such that there are three or more objective functions.

REFERENCES

[1] N. Beume, B. Naujoks, M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research, Volume 181, Issue 3, pp. 1653-1669, 2007.

[2] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. Evolutionary Computation, Vol. 14, No. 4 December 2006.

[3] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

[4] J. Knowles, D. Corne. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. IEEE Congress on Evolutionary Computation, 1999.

[5] T. Miyamoto, A. Fujiwara, Robust optimization algorithms for multi-objective knapsack problem, 13th International Workshop on Advances in Networking and Computing, pp. 430-432, 2022.