Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 9, Number 1, pages 27–30, January 2020

Flower pollination optimization for the multi-objective knapsack problem

Yuta Hadachi Akihiro Fujiwara
Graduate School of Computer Science and Systems Engineering
Kyushu Institute of Technology
Iizuka, Fukuoka, 820-8502, Japan

***Abstract*—A multi-objective optimization problem involves a number of objective functions to be maximized or minimized, and no single solution exists for the problem because there is no solution that simultaneously satisfies all of the objective functions. Therefore, a set of Pareto-optimal solutions, which are non-dominated solutions for the problem, is defined for the multi-objective optimization problem.**

In the present paper, we propose an optimization algorithm based on flower pollination algorithm (FPA) for the multi-objective knapsack problem. The experimental results show that the proposed algorithm obtains a better set of Pareto solutions than the existing algorithm.

***Index Terms*—multi-objective optimization problem, multi-objective knapsack problem, flower pollination algorithm**

I. Introduction

For a multi-objective optimization problem that involves a number of objective functions to be maximized or minimized, there is no single solution because no solution satisfies all of objectives simultaneously. Since there exists a trade-off between two or more conflicting objective functions, the non-dominated solutions, which is called Pareto optimal solution [2] are needed for the multi-objective optimization problem. The problem for computing the maximal Pareto optimal solutions for the multi-objective optimization problem is generally computationally hard, and a number of approximation algorithms have been proposed for the problem.

As an example of the multi-objective optimization problems, a number of approximation algorithms have proposed for a multi-objective 0-1 knapsack problem. There exists a number of knapsacks in the multi-objective 0-1 knapsack problem, and values and weights of items are defined for each knapsack. A number of algorithms have been proposed [1], [3] for the multi-objective knapsack problem.

In the present paper, we propose an approximation algorithm for the multi-objective 0-1 knapsack problem using flower pollination algorithm (FPA) [4]. The FPA is an optimization method based on the process of the flower pollination.

We implement our proposed algorithm and an existing algorithm [1] in experimental environment, and evaluate validity of the proposed algorithms. The experimental results show that our proposed algorithms obtain better sets of Pareto solutions than the existing algorithm.

II. Preliminaries

A. Multi-objective optimization problem

We assume that an instance of the problem is m-dimensional decision 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.

$$\begin{aligned}\max/\min\ \boldsymbol{y} &= \{y_0, y_1, ..., y_{n-1}\} = \{f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), ..., f_{n-1}(\boldsymbol{x})\}\\ \text{such that } \boldsymbol{x} &= (x_0, x_1, ..., x_{m-1}) \in X,\\ X &= \{\boldsymbol{x} \mid \forall i \in \{0, 1, ..., k-1\}, g_i(\boldsymbol{x}) \le 0\}\end{aligned}$$

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

B. Pareto-optimal solution

Since no solution satisfies all of the objective functions in the multi-objective optimization problem, a solution that is not inferior to the other solutions is needed for the multi-objective problem. The solution is called *Pareto optimal solution*, and we first define the dominance relationship of the solutions for defining the Pareto-optimal solution. We assume that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are two feasible solutions for the problem and all objective functions are maximized. Then, $\boldsymbol{x_2}$ dominates $\boldsymbol{x_1}$ if and only if the following two conditions holds.

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

In this paper, $\boldsymbol{x}_1 \prec \boldsymbol{x}_2$ denotes $\boldsymbol{x}_2$ 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 the Pareto-optimal solution is a solution that cannot be improved in any of the objective function without degrading one of the other functions, a maximal set of the Pareto optimal solution is considered as an optimal solution for the multi-objective optimization problem.

C. The hypervolume indicator

There are various metrics for a set of Pareto-optimal solutions of the multi-objective optimization problem. In the paper,

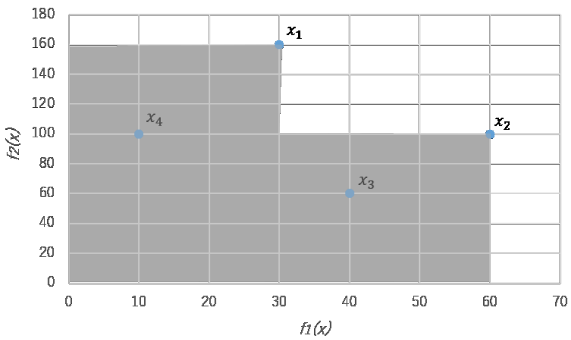

Fig. 1. An example of Hypervolume indicator

we evaluate a set of the Pareto-optimal solutions using *hypervolume indicator* [2]. Let v_x be the volume of the hypercube created by Pareto optimal solution $\boldsymbol{x}$ and the reference point $\boldsymbol{r}$. The hypervolume V for a set of Pareto optimal solutions is defined as follows.

$$V = \bigcup_{\boldsymbol{x} \in X} v_x$$

Fig. 1 shows an example of the hypervolume indicator in case that the problem is bi-objective. Let $X = \{\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}\}$ be a set of Pareto optimal solutions. The hypervolume for X is defined as an union, which is an gray-shaded area in the figure, of four rectangular regions whose opposite vertices are a reference point $\boldsymbol{r}$ and $(f_1(\boldsymbol{x_i}), f_2(\boldsymbol{x_i}))$.

III. The optimization algorithm for the multi-objective knapsack problem

A. The multi-objective 0-1 knapsack problem

An input of the multi-objective 0-1 knapsack problem is given 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 $_x = (x_0, x_1, \cdots, x_{m-1})$ are m-dimensional Boolean vector. Then, the multi-objective 0-1 knapsack problem is formulated as follows.

$$\max \ \boldsymbol{y} = \{f_0(\boldsymbol{x}), f_1(\boldsymbol{x}), ..., f_{n-1}(\boldsymbol{x})\}$$

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

$$\text{s.t. } g_i(\boldsymbol{x}) = \left(\sum_{j=0}^{m-1} w_{i,j} x_j \right) - c_i \leq 0$$

In the above expression, $x_j = 1$ denotes that item j is stored in the knapsacks.

B. The common procedures for optimization algorithms

We explain two common procedures used in the proposed optimization algorithm. The first is a procedure that updates best Pareto optimal solutions, and the second is a procedure that repair infeasible solutions, which do not satisfy constraint functions, to feasible solutions.

1) A procedure for updating best solutions: In the proposed algorithm, Pareto optimal solutions are created at every iteration. Then, Pareto ranking sort is used as a procedure for obtaining a set of l best optimal solutions and a set of Pareto optimal solutions. Let P be a temporal Pareto optimal solutions, and F is a set of feasible solutions obtained in the algorithm. Then, a procedure, Pareto ranking sort, is summarized as follows.

A procedure of Pareto ranking sort:

Step 1: Set $P' = P \cup F$ and $r = 1$. Then, compute Pareto rank of each solution in P' by repeating the following sub-steps until $P' = \phi$.

(1-1) Select Pareto optimal solutions in P', and set Pareto ranks of the selected solutions to r.

(1-2) Remove the selected solutions from P', and set $r = r + 1$.

Step 2: Sort all solutions in $P \cup F$ according to the computed Pareto ranks. Then, select l best solutions $\{t_0, t_1, \cdots, t_{l-1}\}$ for the next iteration. In addition, all solutions whose rank is 1 is selected as a set of Pareto optimal solutions P.

2) A procedure for repairing infeasible solutions: We next show a procedure that repair infeasible solutions, which do not satisfy constraint functions, to feasible solutions. In the procedure, a number of items are removed from a tentative solution using a greedy method.

Let $x_i = (x_{i,0}, x_{i,1}, \cdots, x_{i,m-1})$ be an infeasible solution obtained the in the proposed algorithm. Then, the following procedure is executed and the solution is repaired so that all constraints are satisfied.

A procedure for repairing an infeasible solution:

Step 1: Sort all items in descending order according to the following value s_j $(0 \leq j \leq m-1)$.

$$s_j = \sum_{i=0}^{n-1} \frac{p_{k,j}}{w_{k,j}}$$

Step 2: For each item j $(0 \leq j \leq m-1)$ in solution x_i, the following sub-steps are repeated until x_i is feasible.

(2-1) Select $x_{i,j}$ that satisfies the following condition.

$$s_j = \min\{s_k \mid x_{i,k} = 1, 0 \leq k \leq m-1\}$$

(2-2) Set $x_{i,j} = 0$.

(2-3) Select $x_{i,j}$ that satisfies the following condition.

$$s_j = \max\{s_k \mid x_{i,k} = 0, 0 \leq k \leq m-1\}$$

(2-4) Set $x_{i,j} = 1$ and evaluate the solution. In case that the solution is infeasible, set $x_{i,j} = 0$ again.

C. An optimization algorithm based on flower pollination

The flower pollination algorithm (FPA) [4] is an optimization method inspired by the nature from the pollination process of flowers. Flower pollination is a transfer of pollen from a stamen to the stigma of a flower.

The pollination is classified into four kinds of pollination. Pollination that the pollen is transferred by pollinators such as insects or animals is called biotic pollination. In this case, animals and insects are attracted by flowers that have bright colors and strong smell. On the one hand, biotic pollination transfer pollen through the wind or diffusion in the water. In addition, pollination can be classified into self-pollination and cross-pollination. A transfer of pollen from one flower to the same flower is called as self-pollination, and a transfer of pollen from one flower to a different flower is called cross-pollination.

The pollination increases the quality and strength of flowers. To simulate the pollination process, the following four rules are used.

- In the global pollination, biotic and cross-pollination are considered. The pollinators move in a way which obeys a Levy flight distribution.
- In the Local pollination, biotic and self-pollination are considered.
- Flower constancy is considered as the reproduction probability, which is proportional to the similarity of two flowers involved.
- A choice between global pollination or local pollination is controlled by a switch probability $p \in [0, 1]$.

We now formulate the flower pollination as an optimization technique. In global pollination, flower pollen are carried over a long distance by the pollinators. To simulate the above first and third rules, the following equation is used.

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(x_i^t - b^*) \tag{1}$$

where x_i^t is the solution vector x_i at iteration t, and b^* is the current best solution. γ is a scaling factor to control the step size. The parameter $L(\lambda)$ control the strength of the pollination. In [4], the parameters are suggested to set $\gamma = 0.1$ and $\lambda = 1.5$. To simulate move of pollinators over a long distance, the Levy flight distribution, which is given below, is used.

$$L(\lambda) \sim \frac{\lambda\Gamma(\lambda)\sin(\frac{\pi\lambda}{2})}{\pi} \frac{1}{S^{1+\lambda}} (S >> S_0 > 0) \tag{2}$$

$\Gamma(\lambda)$ is the standard gamma function. The step size S is generated by using Gaussian distributions through generating two random numbers U and V as follows.

$$S = \frac{U}{|V|^{\frac{1}{\lambda}}}, U \sim N(0, \sigma^2), V \sim N(0, 1) \tag{3}$$

$$\sigma^2 = \left[\frac{\Gamma(1+\lambda)}{\lambda\Gamma(\frac{1+\lambda}{2})} \frac{\sin(\frac{\pi\lambda}{2})}{2^{\frac{\lambda-1}{2}}} \right]^{\frac{1}{\lambda}} \tag{4}$$

To simulate the above second rule as local pollination, the following equation is defined.

$$x_i^{t+1} = x_i^t + U(x_j^t - x_k^t) \tag{5}$$

where x_j^t, x_k^t are pollen from different flowers of same plant and U is a uniform distribution in $[0, 1]$.

We now propose an optimization algorithm for the multi-objective knapsack problem using FPA.

FPA for the multi-objective knapsack problem

Step 1: Create random initial l solutions as m-dimensional Boolean vector $x_i^0 = (x_0, x_1, \cdots, x_{m-1})$ $(0 \leq i \leq l-1)$. Each x_k of Boolean vector is 0 or 1 with provability of $\frac{1}{2}$. Then, store the vector as a set of solution X. If x_j is an infeasible solution, execute a procedure for repairing for the infeasible solution. In addition, find a set of Pareto optimal solutions P in the initial solutions, and choose a best solution b^* from P.

Step 2: Repeat the following sub-steps T times. (T is a parameter that denotes the number of generations.) We assume that t denotes the number of executed generations in the following.

(2-1) Generate solutions x_i^{t+1} $(0 \leq i \leq l-1)$ as follows.

(2-1-1) Generate a random value r from a uniform distribution in $[0, 1]$. In case that $p \leq r$, execute global pollination according to (1). Otherwise, execute local pollination according to (5).

(2-1-2) Convert real values obtained in (2-1-1) into binary values using the sigmoid function given below. (r is a random value in $[0, 1]$.)

$$x_{sig} = \frac{1}{1 + \mathrm{e}^{-x_i}} \tag{6}$$

$$x_{bin} = \begin{cases} 1 & r \leq x_{sig} \\ 0 & \text{(otherwise)} \end{cases} \tag{7}$$

(2-1-3) In case that the obtained solution is infeasible, execute a procedure for repairing the infeasible solution. Then, store the solution in a set of solutions F.

(2-3) Execute a procedure of Pareto ranking sort for P and F, and obtain a set of Pareto optimal solutions P.

Step 3: Output a final P as a set of Pareto optimal solutions.

IV. Experimental results

Our proposed algorithm and an existing algorithm [1] are implemented using C++, and we compare Pareto optimal solutions and hypervolume indicators. Table I shows the simulation environment.

TABLE I
SIMULATION ENVIRONMENT

CPU	AMD Ryzen 7 1800X
RAM	64GB
SSD	512GB
OS	CentOS 7.5

First, we describe parameters of the multi-objective 0-1 knapsack problem. The values of the variables used in the simulation are as follows.

- The number of knapsacks n: 2
- The number of items m: 500
- A value of item $p_{i,j}$ $(1 \leq i \leq n, 1 \leq j \leq m)$: [10, 100]

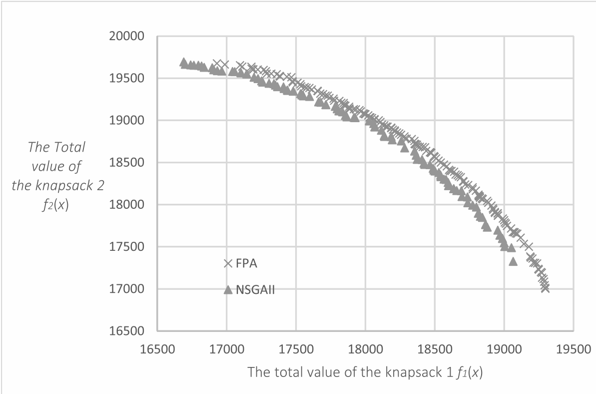

Fig. 2. Pareto optimal solutions of the algorithms

TABLE II
HYPERVOLUME INDICATORS OF THE ALGORITHMS

FPA	NSGA II [1]
3.78×10^8	3.74×10^8

- A weight of item $w_{i,j}$ $(1 \le i \le n, 1 \le j \le m)$: [10,100]
- A capacity of a knapsack c_i $(1 \le i \le n)$: $c_i = \frac{1}{2}\sum_{j=1}^{m} w_{i,j}$

Fig. 2 shows a part of our experimental results. Pareto optimal solutions obtained by the proposed algorithm are distributed in wider range than the solutions obtained by the existing algorithm. In addition, Table II shows hypervolume indicators of the algorithms. The hypervolumes of the proposed algorithm are better than the existing algorithm.

V. CONCLUSIONS

In this paper, we proposed an approximation algorithm for the multi-objective 0-1 knapsack problem using FPA. As our future research, we are considering improvement of our proposed algorithm for diversity of the Pareto optimal solutions, and also considering reduction of the execution time of the proposed algorithms.

REFERENCES

[1] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-Ⅱ. *IEEE Transactions on Evolutionary Computation (IEEE-TEC)*, 2002.
[2] J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. *TIK-Report No.214, Computer Enfineering and Networks Laboratory*, 2005.
[3] D. Simian R. Jovanovic, M. Tuba. Ant colony optimazation applied to minimum weight dominating set problem. In *Proc. 12th WSEAS International Conference on AUTOMATIC CONTROL*, 2010.
[4] X.-S. Yang. Flower pollination algorithm global optimization. In *Proc. Unconventional Computation and Natural Computation*, 2012.