PARETO-BASED COST SIMULATED ANNEALING FOR MULTIOBJECTIVE OPTIMIZATION

Dongkyung Nam and Cheol Hoon Park

Division of Electrical Engineering
Department of Electrical Engineering & Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, 305-701, Republic of Korea
Email: ndk@eeinfo.kaist.ac.kr

ABSTRACT

In this paper, a multiobjective simulated annealing (MOSA) method is introduced and discussed with the multiobjective evolutionary algorithms (MOEAs). Though the simulated annealing is a very powerful search algorithm and has shown good results in various single-objective optimization fields, it has been seldom used for the multiobjective optimization because it conventionally uses only one search agent, which is inadequate in finding many solutions of the Pareto set. With the idea that the simulated annealing has a uniform state probability over global optima, a new multiobjective simulated annealing method is suggested. The experimental performance of the developed algorithm is compared with multiobjective evolutionary algorithms and shows that the proposed simulated annealing has good uniformity properties.

1. INTRODUCTION

This paper will address a simulated annealing method for solving multiobjective optimization problems and compare it with evolutionary methods. The multiobjective optimiza-tion problem has a bit different aspect to the scalar-objective one. Instead of finding one global optimum, which is a general aim in scalar-objective optimization, multiobjec-tive optimization must find a set of solutions, which is called *Pareto set*, or *Pareto optimal frontier*, as all the Pareto solutions are equivalently important and all of them are the global optimal solutions [1]. Many engineering and economical problems are often complex and have this characteristics of the multiple objectives, which must be optimized simultaneously. Typically multiobjective problems are solved with conventional optimization methods by using the penalty method or the weighted sum method [1]. However, the penalty method and the weighted sum method also have a difficulty in selecting proper penalty functions and weighting factors respectively. The other problem of using the weighted sum method is that it cannot find a solution of concave region [2]. To solve this problem, many researches for multiobjective optimizations have been suggested and new concepts are introduced [1]. One of these concepts is the *Pareto optimality* and it is widely used in the many multiobjective optimization algorithms including the evolutionary algorithms.

We suggest four important properties for the multiobjective optimization.

1) Searching precision. The algorithm must find the possible Pareto optimal solutions, which are global optima in multiobjective optimization.
2) Searching time. It must take less time to find the optimal set.
3) Uniform probability distribution over the optimal set. The solutions found must be widely spread, or uniformly distributed over the real Pareto optimal set instead of converging to one point.
4) Information about Pareto frontier. The algorithm must give the information about the Pareto frontier as much as possible.

The objective of this paper is to construct a simulated annealing method to find all the Pareto solutions, to verify the property of the suggested algorithm, and to compare the performance of it with the evolutionary algorithms. Simulated annealing method, which is suggested in this paper, uses the concept of Pareto optimality and domination, which is widely used in evolutionary approaches in multiobjective optimization, to have more searching power in many complex problems, which satisfies the *searching precision* property. Though it is reported that the main drawback of the simulated annealing is the *searching-time*, it is also reported that long searching-time is not always true [3]. The third property, *uniform probability distribution* property, is also very important in multiobjective optimization. The simulated annealing has an interesting advantage at this point as it is mathematically proved that it can find each of the global optima with the uniform probability [4, 5]. Considering that evolutionary algorithms generally use additional algorithms such as fitness sharing, niche induction for spreading the solutions, the suggested simulated annealing have more simple and compact structure.

2. MULTIOBJECTIVE OPTIMIZATION

For most multiobjective problems, there exists a set of non-dominated solutions that have a trade off relationship each other, and one of the multiple objectives of each solution cannot be improved without sacrificing any of others. This concept is known as the *Pareto optimality* [1].

Definition 1. *Pareto optimality*
Consider, without loss of generality, the minimization of the n components $f_k, k = 1, 2 ,n$, of a vector function $\boldsymbol{f}$ of a vector variable $\boldsymbol{x}$ in a universe A , where

$$\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), 2 , f_n(\boldsymbol{x})) .$$

Then a decision vector $\boldsymbol{x}_u \in$ A is said to be Pareto optimal if and only if there is no $\boldsymbol{x}_v$ for which $\boldsymbol{v} = \boldsymbol{f}(\boldsymbol{x}_v) = (v_1, 2 , v_n)$ dominates $\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}_u) = (u_1, 2 , u_n)$, that is, there is no $\boldsymbol{x}_v \in$ A such that

$$v_i \le u_i \quad \forall i \in \{1, 2 , n\} \text{ and } v_i < u_i \quad \exists i \in \{1, 2 , n\} \ \square$$

The set of all Pareto-optimal decision vectors is called the Pareto optimal set, efficient set, admissible set or the Pareto frontier of the problem. The corresponding set of objective vectors is called the non-dominated set. The notion of Pareto optimality is only a first step toward the practical solution of a multiobjective problem, which usually involves the choice of a single compromise solution from the non-dominated set according to some preference information.

The simultaneous optimization of multiple, possibly competing, objective functions deviate from scalar-objective optimization. Instead of finding one perfect solution, multiobjective optimization problem tend to be characterized by a family of alternatives that must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others. Therefore, the first objective in the multiobjective optimization is to find the Pareto set, and the next is to select a proper solution from the found Pareto solution set.

3. MULTIOBJECTIVE SIMULATED ANNEALING BY USING THE PARETO-BASED COST

One of the good properties of simulated annealing in single objective problem is that its properties are well proved by mathematical approaches. Geman and Geman [4] showed its convergence property to global optimum in finite state optimization using Markov chain analysis. Also Mitra, Romeo, and Sangiovanni-vincentelli [5] showed the same result in his paper with different approaches and also showed finite time analysis. One of the main results in their works is summarized in the next theorem [5].

Theorem 1. *Uniform searching probability on the optima.* The probability of the searching agent to be located on the global optima e' is uniform over the global optima. That is,

$$e' = \left[\begin{array}{ll} g(i)/g(*) & i \in S^* \\ 0 & i \notin S^* \end{array} \right. \qquad (1)$$

where S^* is the set of indices of global optimal-cost configuration, i.e.

$$S^* = \{\boldsymbol{x} \in S \mid \boldsymbol{c}(\boldsymbol{x}) \le \boldsymbol{c}(\boldsymbol{y}) \ \forall \boldsymbol{y} \in S\} \qquad (2)$$

with S is state configuration space, $c(\boldsymbol{x})$ is cost function of configuration $\boldsymbol{x}$ and,

$$g(*) = \underset{y \in S^*}{\text{—}} g(\boldsymbol{y})$$

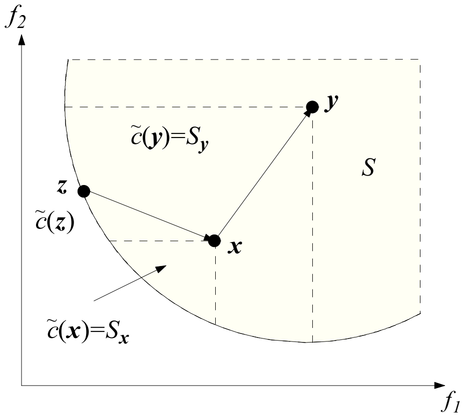
(3)

with $g(\boldsymbol{y})$ is normalizing factor of generation function which is usually considered to be one. $\square$

The proof of this theorem is shown in the [5]. This results show that the searching agent will be located to every global optimum with the uniform probability in the state space when the neighbor generating function is not biased. This indicates one important property for SA to be used in the multiobjective optimization - uniform distribution over all the global optima.

3.1 Pareto-based Cost

We suggest a new multiobjective optimization method that satisfies the *detailed balanced* condition of the SA. Instead using the cost functions directly, we used the *Pareto-based Cost Simulated Annealing* (PCSA). The following is the Pareto-based cost (or Pareto cost in brief) of the state $\boldsymbol{x}$:



f_2
$\boldsymbol{y}$
$\tilde{c}(\boldsymbol{y})=S_{\boldsymbol{y}}$
S
$\boldsymbol{z}$
$\tilde{c}(\boldsymbol{z})$
$\boldsymbol{x}$
$\tilde{c}(\boldsymbol{x})=S_{\boldsymbol{x}}$
f_1

Figure 1: The concept of the Pareto-based cost

$$\tilde{c}(\boldsymbol{x}) = \frac{\int_A dom(\boldsymbol{c}(\boldsymbol{y}), \boldsymbol{c}(\boldsymbol{x}))d\boldsymbol{y}}{\int_A d\boldsymbol{y}} \tag{4}$$

where A represents the whole state space and

$$dom(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \text{ dominates } \boldsymbol{y} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and the transition probability is

$$P_{\boldsymbol{xy}}(T) = \min[1, \exp(-\frac{\tilde{c}(\boldsymbol{x}) - c(\boldsymbol{y})}{T})] . \tag{6}$$

Then the stationary probability is

$$\pi(\boldsymbol{x}) = \exp[-\frac{\tilde{c}(\boldsymbol{x})}{T}] . \tag{7}$$

Figure 1 shows the concept of Pareto-based cost of two-objective optimization. The Pareto-based cost of state $\boldsymbol{x}$ is the area $S_{\boldsymbol{x}}$, in which all states dominate state $\boldsymbol{x}$, divided by whole state area S.

The costs of state $\boldsymbol{y}$ and $\boldsymbol{z}$ are also determined to be $S_{\boldsymbol{y}}/S$, 0 respectively as there are no state that dominates the state $\boldsymbol{z}$. We can also know that Pareto-based cost of state $\boldsymbol{x}$ is higher than Pareto-based cost of $\boldsymbol{z}$. As these two states are in the relation of *undominated*, it is clear that state $\boldsymbol{z}$ is closer to the Pareto optimal than $\boldsymbol{x}$.

3.2 Implementation

However, there are significant problems in using the Pareto-based cost algorithm practically. First, this algorithm is a little ridiculous because calculating one Pareto-based cost requires cost information of all the states including optimal states. It means that it is much worse than the full search algorithm. Second, when the states have real values, it is impossible to calculate exact Pareto-based cost with digital computer, as the Pareto-based cost requires integration on the state space. Therefore, we deal with these two significant problems by the sampling. After sampling N states, the Pareto-based cost is calculated as follows.

$$\tilde{c}^*(\boldsymbol{x}) = \frac{\sum_{k=1}^{N} dom(\boldsymbol{c}(\boldsymbol{s}_k), \boldsymbol{c}(\boldsymbol{x}))}{N} . \tag{8}$$

We suggest two methods for calculating the sampled Pareto-based cost.

3.2.1 Neighbor Sampling

Neighbor sampling takes N samples within the small boundary of current state $\boldsymbol{x}$, and reduces the boundary as time goes on. When a solution approaches the Pareto frontier, almost every sample is dominated by the solution. As only dominating samples have meanings in the Pareto-based cost algorithm, taking samples in a large area is wasteful. When the state transition is considered between states $\boldsymbol{x}$ and $\boldsymbol{y}$, neighbor sampling takes N samples within the boundary of hyper-sphere with radius of $\|\boldsymbol{x} - \boldsymbol{y}\|$ and center on the middle point of $\boldsymbol{x}$, $\boldsymbol{y}$. Generally the position of the next transition state $\boldsymbol{y}$ becomes nearer to the current state $\boldsymbol{x}$, the radius becomes smaller.

3.2.2 Population Sampling

If one wants to find many Pareto solutions at the same time, using population information is a reasonable choice. In this case, additional sampling is dispensable for calculating the Pareto-based cost as population itself gives enough information of the samples. Using population is very simple in the PCSA. PCSA does not need any information exchange like crossover of the genetic algorithms. If someone wants to find M Pareto solutions, running M independent PCSA is enough. Calculating Pareto-based cost from the population information is also simple: take $2M$ solutions (M current solutions and M next transition solutions which are generated by neighbor generation) as the samples for Pareto-based cost. Also additional sampling will be helpful when the population size is too small.

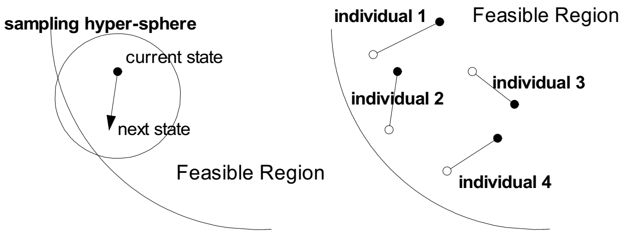


Figure 2: The concept of the neighbor sampling (left) and population sampling (right)

4. EXPERIMENTAL COMPARISON

Even though comparing algorithms by simulation is not always good approach because it has some problems, i.e. limitation of the simulation conditions, finite test-bed functions, dependency of the simulation environment, it must be helpful to understand the mechanism of the algorithm and to find proper algorithm for a specific problem. In this section, we compare the performance of the MOEAs and PCSA for the multiobjective optimization. The difficulties of comparisons lie mainly on the measures in the multiobjective optimization. In the experiment, we only test the uniformity performance between the well-known multiobjective genetic algorithms and the proposed simulated annealing method. Other measuring metrics for the multiobjective optimization, i.e., accuracy, are researching and remained as a future work. The comparison paradigms are as follows.

4.1 Experimental Setup

4.1.1 Objective of the comparison

First of all, this experiment is focused on the comparison between the PCSA and the MOEAs. We used three types of well-known MOEAs for comparison, which are FFGA [6], NSGA [7], and NPGA [8], but we did not consider the comparison of them because there have been already many research results about that. We considered them as variations of MOEAs, not specific algorithms even though their performances are different. Also we used only the proposed Pareto based PCSA for the comparison.

4.1.2 Test functions

The experiment is focused on the real-valued parameter optimization. The test problems are well known 18 optimization functions with and without the constraints [9, 10]. Almost problems have two parameters and two costs, which means there are two-dimensional parameter space and two-dimensional cost space. Also there are problems of four-dimensional parameter space and three-dimensional cost space. The range of the parameter varies much dependent on each problem.

4.1.3 Experiment condition

Basically it is not easy to set the experimental conditions of many algorithms equal because they have different parameters. For example, the initial temperature and cooling schedule are important for the PCSA, but selection scheme and mutation rate are important for the MOEAs. Additionally, in the multiobjective optimization, MOEAs have one more important parameter - niche size that mainly determines the uniformity and coverage characteristics. However, PCSA is not affected by the niche size because it does naturally maintain the uniformity and wide coverage characteristics. The parameters of both algorithms were chosen by the heuristic methods. Strictly speaking, by these reason, this experimental comparison may not be fare. However, the conditions are equal to both algorithms because there is no additional parameter tuning for the algorithms. Table 1 shows the parameters of both algorithms.

4.1.4 Uniformity measure

We propose uniformity metric for the two-dimensional cost function and suggest an algorithm for the higher dimensional case. In the two-dimensional problem, it is possible to use the distance sequence with the sorted index that is used in the coverage metric.

$$M_{uniformity} = \sqrt{\frac{1}{n-1}\underset{i\in I}{\overset{n-1}{—}}(\overline{?}? - {}_{i,i+1})^2} \qquad (9)$$

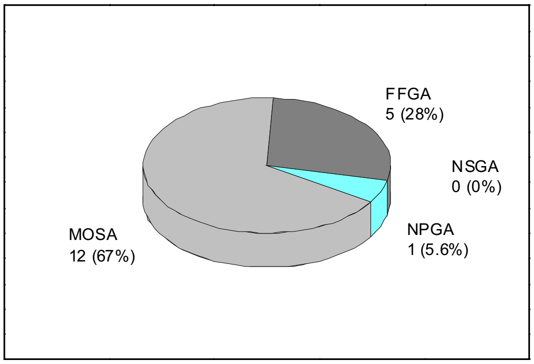
where the I is the sorted index of the found solutions as used in the coverage metric. There is no difference between sorting with respect to the first cost and the second cost. $?_{i,i+1}$ is the distance of the costs between the i-th solution and $(i+1)$-th solution and $\overline{?}$ is the average value of the distance sequence $?_{i,i+1}$. However, this method cannot be used in the higher dimensional costs because there is no easy sorting method for the problem with over three-dimensional cost. An easy method to solve this problem is the make a distance sequence with the following method. 1) Find a solution with maximum norm. 2) Calculate distance sequence of the solutions from that solution. 3) Sort the distance sequence and find the variance of them. It is not such a good metric because it does not measure uniformity exactly, but it can be a practically useful metric.

4.2 Experimental Result

We tested the 24 functions repeatedly by ten times and got averaged results. The graph in the Figure 3 shows how many times each algorithm outperforms others in the 24 test functions. This shows an unexpected result, not because the proposed algorithm outperforms well-known promising evolutionary algorithms, but because the ratio is only 67%. After simulations, we verified all the results one by one and found that, in all the cases, the proposed simulated annealing algorithm outperforms evolutionary algorithms in the view of uniformity. There is a main reason why the PCSA does not always outperform the evolutionary algorithms in the Figure 3. It is from the property of the disconnected Pareto frontier of the problem and developing good uniformity measure is remained as future work.

Table 1: Parameters for the experiment

Parameters of PCSA and Evolutionary Algorithms			
PCSA		Evolutionary Algorithms	
Pop size	100	Pop size	100
Initial temperature	100.0	Mutation rate	0.3
Neighbor generating	Fast SA	Crossover rate	1.0
Cooling method	Fast SA	Selection method	FFGA/NSGA/ NPGA
Acceptance method	Metropolis	Niche size	Problem dependent
Termination condition	10,000 iterations	Termination condition	10,000 iteration



Uniformity performance over all the test functions

Figure 3: Uniformity performance

5. CONCLUSION

We developed a new multiobjective optimization algorithm by using the simulated annealing method. To make the single objective algorithm to multiobjective one, we developed the Pareto-based cost and the test results showed that the proposed algorithm has good uniformity performance.

6. REFERENCES

[1] C. M. Fonseca and P. J. Fleming, "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, 3(1):1-16, 1995.

[2] P. J. Fleming, and A. P. Pashkevich, “Computer aided control system design using a multiobjective optimization approach,” *Proceedings of the IEE Control’85 Conference*, pp. 174-179, London, UK: IEE, 1985.

[3] L. J. Park, “Hybrid Evolutionary Algorithms with Heuristic Operators for Combinatorial Optimization Problems,” *Ph. D Thesis*, KAIST, 1997.

[4] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, pp. 721-741, November 1984.

[5] D. Mitra, F. Romeo, and A. Sangiovanni-Vincentelli, “Convergence and finite-time behavior of simulated annealing,” *Applied Probability Trust*, Vol. 18, pp. 747-771, 1986.

[6] C. M. Fonseca and P. J. Fleming, “Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms - Part II: Application Example,” *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, Vol. 28, No. 1, pp. 38-47, January 1998.

[7] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computations*, Vol. 2, No 3, pp.221-248, 1994.

[8] J. Horn and N. Nafpliotis, "Multiobjective optimization using the niched pareto genetic algorithm,” *IlliGAL Report No. 93005.* Urbana-Champaign, IL: Department of General Engineering, University of Illinois at Urbana-Champaign, 1993.

[9] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, PhD thesis, Department of Electrical and Com- puter Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[10] T. T. Binh. “A Multiobjective Evolutionary Algorithm: The Study Cases,” In Annie S. Wu, (Ed), *Proceedings of the 1999*

Genetic and Evolutionary Computation Conference. Workshop Program, pp. 127-128, Orlando, Florida, July 1999.