Multi-Objective Particle Swarm Optimization with Preference-based Sorting

Ki-Baek Lee and Jong-Hwan Kim
Department of Electrical Engineering
KAIST
291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Republic of Korea
email: {kblee, johkim}@rit.kaist.ac.kr

***Abstract*—Multi-objective particle swarm optimization (MOPSO) provides a set of nondominated solutions and the number of nondominated solutions increases exponentially when the number of objectives increases. To select a desired solution out of them, preference-based solution selection algorithm (PSSA) was proposed by incorporating user's preference into multi-objective evolutionary algorithms. In this paper, multi-objective particle swarm optimization with preference-based sorting (MOPSO-PS) is proposed, where a global best position is randomly selected from the archive of nondominated solutions sorted by global evaluation considering user's preferences for multiple objectives. The user's preference is represented as a degree of consideration for the objectives by the fuzzy measures. The global evaluation of the solutions is carried out by the fuzzy integral of partial evaluation with respect to the fuzzy measures, where the partial evaluation of each solution is obtained as a normalized objective function value. To demonstrate the effectiveness of the proposed MOPSO-PS, empirical comparisons to NSGA-II, MQEA, and MOPSO are carried out for the DTLZ functions. Experimental results show that the user's preference is properly reflected in the selected solutions without any loss of overall quality and diversity.**

***Index Terms*—Swarm Intelligence, Particle Swarm Optimization, Multi-Objective Evolutionary Algorithm, Fuzzy Integral, Fuzzy Measure, Preference-based Solution Selection Algorithm**

I. Introduction

Particle Swarm Optimization (PSO) is an algorithm to find an optimal solution in a complex search space inspired from the interaction of individuals of swarm, such as bird flocks, insects, etc. [1], [2]. In this algorithm, the individuals, called particles, each of which is defined as a vector of its position and velocity, search a space for a solution by modifying the trajectories of the individual vectors. The particles are attracted stochastically toward the better positions considering their personal best position and global best position. PSO is easy to implement and fast because of its simple and intuitive structure [3]–[7].

Over the past, there have been a number of approaches in the literature for extending PSO to multi-objective PSO (MOPSO) [8]–[12]. According to a recent survey [13], the research on MOPSO has mainly been studied through aggregating, lexicographic ordering, sub-population, and dominance-based approaches. However, since the number of nondominated solutions increases exponentially when the number of objectives increases, those approaches are less effective in multi-objective problems, in particular, with more than three objectives [14]. Most of all, decision making of selecting a preferred solution out of them is required in real applications. For this purpose, preference-based solution selection algorithm was proposed by incorporating user's preference into multi-objective evolutionary algorithms [15].

In this paper, multi-objective particle swarm optimization with preference-based sorting (MOPSO-PS) is proposed to select a global best position from the archive of nondominated solutions which are sorted by their global evaluation considering user's preferences to multiple objectives. It is based on both partial evaluation and global evaluation for each solution. The former is obtained by normalizing each objective function value of the solution to 1.0. For the latter, user's preference for the objectives or criteria is represented by a degree of consideration using the fuzzy measures. Global evaluation of each solution is calculated by the fuzzy integral of the partial evaluation with respect to the user's preference. At first, to maintain the diversity of the solutions, the solutions in the archive are sorted by crowding distance. After that, the upper half of the solutions, i.e. dispersed solutions, are sorted by the global evaluation value. Finally, global best position of each individual is selected randomly from the upper quarter of the solutions. Since the proposed algorithm can lay emphasis on specific objectives by the degree of consideration, it can select preferred solutions considering the preference for the specific ones in multi-objective optimization problems.

To demonstrate the effectiveness of the proposed MOPSO-PS, empirical comparisons to nondominated sorting genetic algorithm-II (NSGA-II) [16], multiobjective quantum-inspired evolutionary algorithm (MQEA) [17], [18], and MOPSO [10] are carried out for the DTLZ functions which are test problems for multi-objective optimization [19].

This paper is organized as follows. Section II describes the fuzzy measure and fuzzy integral and proposes preference-based sorting method. Section III briefly introduces PSO and proposes MOPSO-PS. In Section IV, experiment results are discussed. Finally, concluding remarks follow in Section V.

II. Preference-based Sorting Method

In the process of sorting the nondominated solutions by the user's preferences, it is required to have a global evaluation for each one considering both of partial evaluation over objectives and user's degree of consideration for objectives. The solutions

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

are sorted in descending order by their global evaluation value and then the ones with higher global evaluation values are regarded as the preferred solutions. In this paper, the fuzzy measures are employed to represent user's degree of consideration and a global evaluation is calculated by the fuzzy integral. The fuzzy measure and fuzzy integral are briefly described in the following and then detailed description of the proposed preference-based sorting method follows.

A. Fuzzy Measure and Fuzzy Integral

Fuzzy measure on the power set of X, denoted $P(X)$, in the finite space $X = \{x_1, \cdots, x_n\}$ is defined as follows.

Definition 1: A fuzzy measure g defined on $(X, P(X))$ is a set function $g : P(X) \to [0, 1]$ satisfying the following axioms:

(1) Boundary condition

$$g(\emptyset) = 0, \ \ g(X) = 1. \tag{1}$$

(2) Monotonicity

$$\forall A, B \subseteq P(X), \ \ if \ \ A \subseteq B, \ \ then \ \ g(A) \leq g(B). \tag{2}$$

Fuzzy measures are classified as belief measure, plausibility measure, probability measure, etc. Belief measure, Bel is a set function, $Bel : P(X) \to [0, 1]$, satisfying the following additional axiom:

$$Bel(A_1 \cup A_2 \cup \cdots \cup A_n) \geq \sum_i Bel(A_i) - \sum_{i>j} Bel(A_i \cap A_j) + \cdots + (-1)^{n+1} Bel(A_1 \cap A_2 \cap \cdots \cap A_n). \tag{3}$$

Since $Bel(A \cup \overline{A}) = 1$ and $Bel(A \cap \overline{A}) = 0$, $Bel(A) + Bel(\overline{A}) \leq 1$. In other words, the sum of all the belief measures is less than or equal to 1. Plausibility measure, Pl is a set function, $Pl : P(X) \to [0, 1]$, satisfying the following additional axiom:

$$Pl(A_1 \cap A_2 \cap \cdots \cap A_n) \leq \sum_i Pl(A_i) - \sum_{i>j} Pl(A_i \cup A_j) + \cdots + (-1)^{n+1} Pl(A_1 \cup A_2 \cup \cdots \cup A_n). \tag{4}$$

Since $Pl(A \cup \overline{A}) = 1$ and $Pl(A \cap \overline{A}) = 0$, $Pl(A) + Pl(\overline{A}) \geq 1$. It means that the sum of all the plausibility measures is greater than or equal to 1. Probability measure can also be defined as a special case of either belief measure or plausibility measure, which satisfies an additional axiom on additivity property.

Note that belief and plausibility measures are mutually dual and can be derived from one another, such as $Pl(A) = 1 - Bel(\overline{A})$. Belief measure indicates one's confidence of making a decision with certainty. Plausibility measure, on the other hand, represents one's confidence considering all the plausible cases in making a decision. Therefore, $Bel(A)$ is always less than or equal to $Pl(A)$.

As a general representation of fuzzy measure, λ-fuzzy measure, $g : P(X) \to [0, 1]$, is defined, which additionally satisfies the following axiom [20]:

$$\forall A_{i,j} \in P(X), \ i, j = 1, \cdots, n, \ A_i \cap A_j = \emptyset \ and \ -1 < \lambda,$$

$$g(A_i \cup A_j) = g(A_i) + g(A_j) + \lambda g(A_i) g(A_j) \tag{5}$$

where λ represents a degree of interaction between A_i and A_j. λ-fuzzy measure is considered as belief measure, plausibility measure or probability measure depending on the value of λ. If $\lambda > 0$, $\lambda < 0$ and $\lambda = 0$, they are considered respectively as belief measure, plausibility measure and probability measure.

Note that each kind of fuzzy measures indicates a different interaction between criteria [21]. Belief measure indicates a positive interaction, since $g(A_i \cup A_j) > g(A_i) + g(A_j)$. On the other hand, plausibility measure indicates a negative interaction, since $g(A_i \cup A_j) < g(A_i) + g(A_j)$. Probability measure does not represent any interactions among criteria, since it is the the same as conventional weighted sum which satisfies the additivity.

For global evaluation of each solution over criteria with respect to the degree of consideration for each of criteria, either Sugeno fuzzy integral or Choquet fuzzy integral [22] can be used, which are defined in the following.

Definition 2: Let $h : X \to [0, 1]$, where X can be any set. The Sugeno fuzzy integral of evaluated value, h over a subset of $X \in P(X)$ with respect to a fuzzy measure g is defined as

$$\int_X h \circ g = \max_i \ min[h(x_i), g(E_i)] \tag{6}$$

where $h(x_1) \leq h(x_2) \leq \cdots \leq h(x_n)$ and $E_i = \{x_i, x_{i+1}, \cdots, x_n\}$ for $x_i \in X$ and $i = 1, \cdots, n$.

Definition 3: Let $h : X \to [0, 1]$, where X can be any set. The Choquet fuzzy integral of evaluated value, h over a subset of $X \in P(X)$ with respect to a fuzzy measure g is defined as

$$\int_X h \circ g = \sum_{i=1}^{n} (h(x_i) - h(x_{i-1})) g(E_i) \tag{7}$$

where $h(x_1) \leq h(x_2) \leq \cdots \leq h(x_n)$, $E_i = \{x_i, x_{i+1}, \cdots, x_n\}$ and $h(x_0) = 0$, for $x_i \in X$ and $i = 1, \cdots, n$.

Note that x_i, $i = 1, \cdots, n$, denotes i-th criterion which corresponds to i-th objective in multi-objective problem and then $h(x_i)$ is the partial evaluation value over x_i. The fuzzy measure g represents the degree of consideration for each objective. Thus, fuzzy integral can be used for global evaluation of each solution.

B. Proposed Method Using Fuzzy Integral

Fuzzy integral requires neither criteria (or objectives in multi-objective problem) to be independent nor fuzzy measure to be additive for any subset in power set of criteria because it can effectively represent the interactions, i.e. positive interactions and negative interactions among criteria. Also, it is easy to set a preference for each criterion. It just needs to set the comparative preference between two criteria and to decide either plausibility measure or belief measure to be used. Thus, considering general multi-objective problems, it is a suitable method for the global evaluation compared to other existing methods. Overall sorting procedure using the λ-fuzzy measure and fuzzy integral is summarized in Algorithm 1 and each step is described in the following.

Algorithm 1 Preference-based Sorting

- $C = \{c_1, c_2, \cdots, c_n\}$: a set of criteria
- n: the number of criteria
- $P(C)$: a power set of C
- $h_k(c_i)$: partial evaluation value of k-th solution, $k = 1, \cdots, m$, over c_i
- m: the number of solutions
- e_k: global evaluation value of k-th solution

1) Define a set of objectives in multi-objective problem as C.
2) Calculate λ-fuzzy measures g's of $P(C)$.
    a) Make a pairwise comparison matrix, P.
    b) Calculate normalized weights of c_i, $\forall i$.
    c) Calculate λ-fuzzy measures of $P(C)$.
3) Normalize the solutions to get the partial evaluation value $h_k(c_i)$, $\forall i, k$.
4) Calculate the global evaluation value e_k, $\forall k$ using fuzzy integral for partial evaluation value and λ-fuzzy measures.
5) Sort and select the solutions with the high global evaluation value as the preferred solutions.

1) Define objectives in multi-objective problem as criteria

Multi-objective problems have predefined objectives to be optimized simultaneously. Partial evaluation over each objective is conducted for each candidate solution, which corresponds to calculating its normalized objective function value. The preferred solution is selected considering both the partial evaluation and user's degree of consideration for each objective. Thus, the objectives of multi-objective problems can be defined as the criteria of fuzzy integral for global evaluation.

2) Calculation of fuzzy measure

In this paper, λ-fuzzy measure is used to represent the degree of consideration for each criterion. According to (1) and (5), λ-fuzzy measure has to satisfy the following equation:

$$\begin{aligned} g(C) &= g(\{c_1, c_2, \cdots, c_n\}) \\ &= g(\{c_1, \cdots, c_{n-1}\}) + g_n + \lambda g(\{c_1, c_2, \cdots, c_{n-1}\}) g_n \\ &\qquad \vdots \\ &= (g_1 + g_2 + \cdots + g_n) + \lambda(g_1 g_2 + g_1 g_3 + \cdots + g_{n-1} g_n) \\ &\quad + \lambda^2 (g_1 g_2 g_3 + g_1 g_2 g_4 + \cdots + g_{n-2} g_{n-1} g_n) + \cdots + \\ &\quad \lambda^{n-1} (g_1 g_2 \cdots g_n) \\ &= 1 \end{aligned} \tag{8}$$

where C is a set of criteria, $\{c_1, c_2, \cdots, c_n\}$ and for notational simplicity $g_i = g(\{c_i\})$. Since (8) is an (n-1)-th order equation of λ, it is quite difficult to solve the equation for λ given g_i's, if the number of criteria is more than three. Thus, the following procedure is employed to calculate the fuzzy measures [23].

a) Make a pairwise comparison matrix

The pairwise comparison matrix of criteria, P, which represents preference degrees between criteria, is defined as follows [24]:

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \tag{9}$$

where p_{ij} represents the preference degree between i-th criterion, c_i and j-th criterion, c_j, p_{ii} is 1 and $p_{ji} = 1/p_{ij}$. For example, if p_{12} is 5, it means c_1 is five times more preferred to c_2.

b) Calculate normalized weight

The normalized weight, w_i of i-th criterion, c_i, $i, j = 1, \cdots, n$ is calculated as follows:

$$w_i = \frac{\sum_{j=1}^{n} p_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}}. \tag{10}$$

There are some other methods to derive priority vectors, like normalized weight, from pairwise comparison matrix [25]. Any one of them can be used in this step.

c) Calculate λ-fuzzy measures

$\phi_{\lambda+1}$ transformation is employed to calculate λ-fuzzy measures [23]. The transformation, $\phi_{\lambda+1} : [0,1] \times [0,1] \rightarrow [0,1]$ is defined as follows:

$$\phi_{\lambda+1}(\xi, w_i) = \begin{cases} 1 & if \;\; \xi = 1 \; and \; w_i > 0 \\ 0 & if \;\; \xi = 1 \; and \; w_i = 0 \\ 1 & if \;\; \xi = 0 \; and \; w_i = 1 \\ 0 & if \;\; \xi = 0 \; and \; w_i < 1 \\ w_i & if \;\; \xi = 0.5 \\ \frac{(\lambda+1)^{w_i} - 1}{\lambda} & other \;\; cases \end{cases} \tag{11}$$

where ξ is another interaction degree of which value lies in $[0, 1]$. Then, λ is determined by ξ, where $\lambda = \frac{(1-\xi)^2}{\xi^2} - 1$. It means $\xi \in (0, 1)$ has one to one correspondence with $\lambda \in (-1, \infty)$. Using (11), λ-fuzzy measure of each element of $P(C)$, $g(A)$ is calculated as follows:

$$g(A) = \phi_{\lambda+1}\left(\xi, \sum_{c_i \in A} w_i\right), \;\; \forall A \in P(C) \tag{12}$$

where A is an element of $P(C)$.

3) Partial evaluation of solutions

The function, h in (6) and (7) is a normalized objective function which represents partial evaluation of each solution over each criterion. The objective function values need to be normalized to 1, since h is defined from 0 to 1. This step calculates $h_k(c_i)$ of k-th solution over c_i.

4) Global evaluation of solutions

The global evaluation value of each candidate solution is calculated by the fuzzy integral using (6) or (7). g and h_k are λ-fuzzy measure and partial evaluation value obtained at Steps ii) and iii), respectively. It means the global evaluation value is calculated by considering both user's degree of consideration for each criterion and partial evaluation of the candidate solution.

5) Sort the preferred solutions

The preferred solutions are sorted and selected out of nondominated solutions based on their global evaluation value. The solutions are sorted in descending order by their global evaluation value and then the solutions with higher global evaluation values are regarded as the preferred solutions.

III. Multi-objective Particle Swarm Optimization with Preference-based Sorting

This section briefly summarizes particle swarm optimization (PSO) and proposes multi-objective particle swarm optimization with preference-based sorting (MOPSO-PS).

A. Particle Swarm Optimization Algorithm

Overall pseudo code of PSO is described in Algorithm 2 where $\mathbf{x}_i^p$ and $\mathbf{x}_i^g$ are the personal and global best positions of the i-th particle, respectively.

Algorithm 2 Particle Swarm Optimization

1) Initialize swarm.
2) Update swarm.
    for each particle **do**
        Evaluate the objective function.
    end for
    for each particle **do**
        Update $\mathbf{x}_i^p$.
        Update $\mathbf{x}_i^g$.
        Update velocity and position.
    end for
3) Repeat.

1) Initialize swarm

At first, the velocity and position of particles in a population are randomly initialized on D-dimensional space. A population is a set of N particles which have their own velocity and position. The velocity $\mathbf{v}_i$ and position $\mathbf{x}_i$ of the i-th particle p_i, $i = 1, 2, \ldots, N$, are the D-dimensional vectors as follows:

$$\mathbf{v}_i \in \mathbb{R}^D, \mathbf{x}_i \in \mathbb{R}^D.$$

The personal best position and the global best position is also initialized.

2) Update swarm

After initialization process, objective function value of each particle at generation t $f(\mathbf{x}_i^t)$, $i = 1, 2, \ldots, N$, is evaluated using the following function:

$$f(\mathbf{x}_i^t) : \mathbb{R}^D \to \mathbb{R}.$$

After fitness evaluation, the personal best position of each particle $\mathbf{x}_i^p$, $i = 1, 2, \ldots, N$, is updated. The personal best position of p_i is defined as the position where the fitness value is the largest all over the past positions of p_i. After that, the global best position of each particle $\mathbf{x}_i^g$, $i = 1, 2, \ldots, N$, is updated. The global best position of p_i is defined as the best position of the personal best positions of p_i's neighbors. Finally, the velocity and position of each particle p_i, $i = 1, 2, ..., N$, are updated as follows:

$$\begin{cases} \mathbf{v}_i^{t+1} = & w \cdot \mathbf{v}_i^t + c \cdot \{\phi_i^{1,t}(\mathbf{x}_i^p - \mathbf{x}_i^t) \\ & +\phi_i^{2,t}(\mathbf{x}_i^g - \mathbf{x}_i^t)\} \\ \mathbf{x}_i^{t+1} = & \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \end{cases} \quad (13)$$

where w and c are constants and $\phi_i^{1,t}$ and $\phi_i^{2,t}$ are random real values uniformly distributed in $[0, 1]$. $\mathbf{v}_i^t$ and $\mathbf{x}_i^t$ represent the velocity and position of p_i at generation t, respectively. New random values are generated for each particle and generation.

3) Repeat

Repeat 2) until a termination condition is met.

B. Overall Procedure of the proposed MOPSO-PS

This subsection describes overall structure and procedure of the proposed MOPSO-PS. The MOPSO-PS is designed by incorporating the proposed preference-based sorting method and PSO with the MOPSO framework [8]. Algorithm 3 shows the overall procedure of MOPSO-PS, where $\mathbf{A}_t$ is the external archive at generation t, $\mathbf{x}_i^p$ is the personal best position of the i-th particle, and $\mathbf{x}_i^g$ is the global best position of the i-th particle.

1) Initialize swarm and external archive

At first, the velocity and position of particles in a population are randomly initialized on D-dimensional space. A population is a set of N particles which have their own velocity and position. The velocity $\mathbf{v}_i$ and position $\mathbf{x}_i$ of the i-th particle p_i, $i = 1, 2, \ldots, N$, are the D-dimensional vectors as follows:

$$\mathbf{v}_i \in \mathbb{R}^D, \mathbf{x}_i \in \mathbb{R}^D.$$

The personal best position of each particle is initially set as itself. The external archive is also initialized with the nondominated solutions from the initial population.

2) Update Swarm

After initialization process, K-objective function of each particle at generation t, $f(\mathbf{x}_i^t)$, $i = 1, 2, \ldots, N$, is evaluated, which is defined as

$$f(\mathbf{x}_i^t) : \mathbb{R}^D \to \mathbb{R}^K.$$

The fuzzy integral, i.e. global evaluation value, of each particle is also evaluated by the method described in Section II. After

Algorithm 3 Multi-Objective Particle Swarm Optimization with Preference-based Sorting

1) Initialize population and external archive.
2) Update swarm.

for each particle **do**
  Evaluate the objective function. (Partial evaluation).
  Calculate the fuzzy integral (Global evaluation).
end for
for each particle **do**
  Update $\mathbf{x}_i^p$.
  Select $\mathbf{x}_i^g$ from the $\mathbf{A}_t$.
  Update velocity and position.
end for

3) Update $\mathbf{A}_t$.
   - Generate a new set of nondominated solutions from the union of $\mathbf{A}_t$ and new population.
   - Calculate the crowding distance of the solutions.
   - Sort the solutions based on the crowding distance.
   - Sort the upper half of the solutions based on the global evaluation value.
   - Fill $\mathbf{A}_{t+1}$ with the sorted nodiminated solutions.
4) Repeat.

fitness evaluation, the personal best position of each particle $\mathbf{x}_i^p$, $i = 1, 2, \ldots, N$, is updated. $\mathbf{x}_i^p$ is replaced by $\mathbf{x}_i^t$ if $\mathbf{x}_i^t$ weakly dominates $\mathbf{x}_i^p$ or $\mathbf{x}_i^t$ and $\mathbf{x}_i^p$ are mutually nondominating. After that, the global best position of each particle $\mathbf{x}_i^g$, $i = 1, 2, \ldots, N$, is updated. Since the solutions in $\mathbf{A}_t$ are mutually nondominating and no solutions in $\mathbf{A}_t$ is dominated by $\mathbf{x}_i^t$, all the solutions in $\mathbf{A}_t$ are candidates for the global best position. In the proposed algorithm, $\mathbf{x}_i^g$ is randomly selected from the upper quarter of $\mathbf{A}_t$. The reason of this selection strategy is explained in the next step. Finally, the velocity and position of each particle p_i, $i = 1, 2, ..., N$, are updated as follows:

$$\begin{cases} \mathbf{v}_i^{t+1} = & w \cdot \mathbf{v}_i^t + c \cdot \{\phi_i^{1,t}(\mathbf{x}_i^p - \mathbf{x}_i^t) \\ & +\phi_i^{2,t}(\mathbf{x}_i^g - \mathbf{x}_i^t)\} \\ \mathbf{x}_i^{t+1} = & \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \end{cases} \tag{14}$$

where w and c are constants and $\phi_i^{1,t}$ and $\phi_i^{2,t}$ are random real values uniformly distributed in $[0, 1]$. $\mathbf{v}_i^t$ and $\mathbf{x}_i^t$ represent the velocity and position of p_i at generation t, respectively. New random values are generated for each particle and generation.

3) Update external archive

After update process, a new solution set is generated by gathering nondominated solutions from the union of $\mathbf{A}_t$ and the new population. To maintain the diversity of the solutions in $\mathbf{A}_t$, the crowding distance of each solution is calculated and the solutions are sorted by their crowding distance [26]. The upper half of the sorted solutions, i.e. dispersed solutions, are sorted again based on the global evaluation value. Finally, $\mathbf{A}_{t+1}$ is filled with the sorted nondominated solutions. Since the upper half of the external archive is sorted by the global evaluation value, the upper quarter of the external archive consists of the solutions with higher global evaluation value. Therefore, by selecting $\mathbf{x}_i^g$ randomly from the upper quarter of the external archive, the particles are guided by the user's preferences.

4) Repeat

Repeat 2) and 3) until a termination condition is met.

Note that computational complexity of the proposed algorithm is governed by the sorting algorithms, i.e. the crowding distance-based sorting and the preference-based sorting. Since both sorting are done by the quick sort method, the proposed algorithm has the average computational complexity of O(nlog(n)).

IV. Experimental Results

To show the effectiveness of the proposed MOPSO-PS, the performance was compared with that of NSGA-II [16], MQEA [18], and MOPSO [10] for seven-objective DTLZ functions [19].

TABLE I: Parameter setting of NSGA-II, MQEA, MOPSO, and MOPSO-PS for 7-objective DTLZ functions.

Algorithms	Parameters	Values
NSGA-II	Population size (N)	100
	No. of generations	3000
	Mutation probability (p_m)	0.1
MQEA	Global population size ($n \cdot s$)	100
	No. of generations	3000
	Subpoppulation size (n)	25
	No. of subpopulations (s)	4
	No. of multiple observations	10
	The rotation angle ($\Delta\theta$)	0.23π
MOPSO/ MOPSO-PS	Population size (N)	100
	No. of generations	3000
	Max. archive size	500
	Inertia weight, w	$1/(2 \cdot \log 2)$
	Cognitive/Social parameter, c	$0.5 + \log 2$

A. Experiment Settings

Parameters used in experiments are given in Table I. The number of variables of each DTLZ function was set to 11 for DTLZ1, 16 for DTLZ2 - DTLZ6, and 26 for DTLZ7 function. Belief measure ($\xi = 0.25$) was employed in MOPSO-PS because preference-based sorting using belief measure selects solutions satisfying the preference with a given amount of certainty. If plausibility measure is used in preference-based sorting, it selects more biased solutions to the specific objectives which are emphasized by the user's high degree of consideration. If such solutions are selected in every generation, entire population converges toward local search area. Thus, it can give negative influence to the performance, i.e. proximity to Pareto-optimal solutions and solution diversity. Three out of the seven objectives in DTLZ problems were chosen as preferred objectives. The degree of consideration for them was set as $f_1 : f_2 : f_3 : f_4 : f_5 : f_6 : f_7 = 1 : 10 : 1 : 10 : 1 : 10 : 1$. The normalized weights according to pairwise comparison matrix were calculated as $(0.0295, 0.295, 0.0295, 0.295, 0.0295, 0.295, 0.0295)$.

B. Performance Metrics

In addition to the average objective function values, two performance metrics, the size of the dominated space and diversity, were employed to evaluate the results of NSGA-II, MQEA, MOPSO and MOPSO-PS. Brief explanations of the metrics are in the following [27]. The size of the dominated space, $\mathcal{S}$ is defined by the hypervolume of nondominated solutions. The quality of obtained solution set is high if this space is large. The diversity, $\mathcal{D}$ is to evaluate the spread of nondominated solutions, which is defined as follows [28]:

$$\mathcal{D} = \frac{\sum_{k=1}^{n}(f_k^{(max)} - f_k^{(min)})}{\sqrt{\frac{1}{|N_0|}\sum_{i=1}^{|N_0|}(d_i - \bar{d})^2}} \tag{15}$$

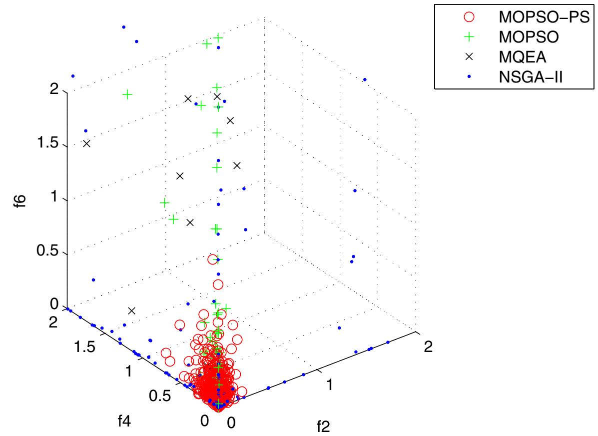
where N_0 is the set of nondominated solutions, d_i is the minimal distance between the i^{th} solution and the nearest neighbor, and $\bar{d}$ is the mean value of all d_i. $f_k^{(max)}$ and $f_k^{(min)}$ represent the maximum and minimum objective function value of the k^{th} objective, respectively. A larger value means a better diversity of the nondominated solutions.

C. Results

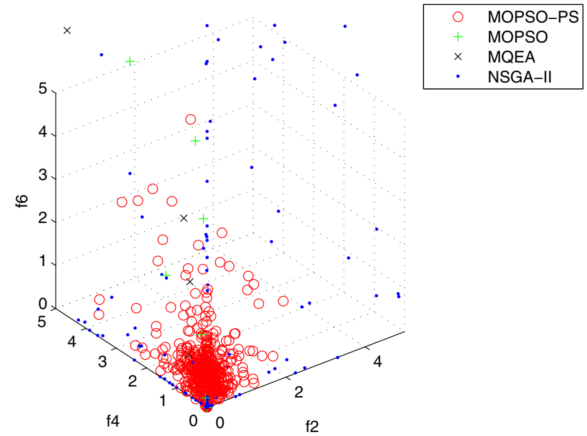
Since f_2, f_4, and f_6 were more considered in every generation of optimization process, MOPSO-PS could find the optimized solutions more focused on those preferred objectives. Fig.1 and Table II show the nondominiated solution distribution of the median result among 100 runs and the average function values, respectively. As the figure shows, the solutions of MOPSO-PS, marked as red circle, are distributed toward smaller value of f_2, f_4, and f_6 compared to those of the other algorithms. The table also shows that the average values of f_2, f_4, and f_6 of MOPSO-PS are the smallest among all the algorithms except for DTLZ2 function.

The size of the dominated space $\mathcal{S}$ and the diversity $\mathcal{D}$ of NSGA-II, MQEA, MOPSO and MOPSO-PS are shown in Table III, where $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ and $\mathcal{S}_4$ represent $\mathcal{S}$ of NSGA-II, MQEA, MOPSO and MOPSO-PS, respectively, and $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$ represent $\mathcal{D}$ of NSGA-II, MQEA, MOPSO and MOPSO-PS, respectively. The values were averaged over 100 runs and Welch's t-test values [29] were also calculated. A t-test value $t_{X_1-X_2}$ represent the statistical difference between the two samples, X_1 and X_2. As the table shows, both $\mathcal{S}$ and $\mathcal{D}$ of MOPSO-PS were not larger than those of the other algorithms for all DTLZ functions. However, it can be a distinctive advantage of MOPSO-PS that its $\mathcal{S}$ and $\mathcal{D}$ were competitive with those of the other algorithms though the preferred objectives were more considered.

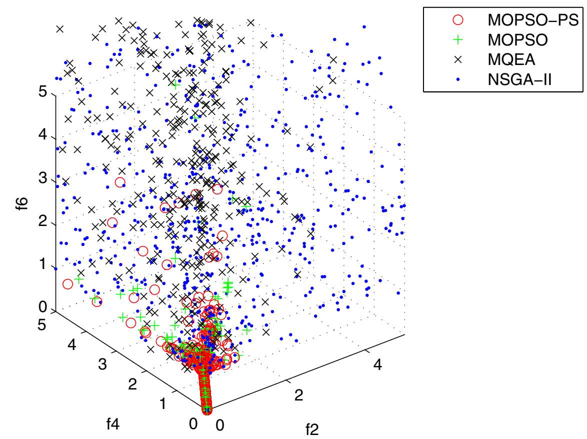
When the conventional utility function method like weighted sum method is used in selection process, the weights need to be set very carefully to select the solutions not only optimized mostly for preferred objectives but also optimized



(a) DTLZ1



(b) DTLZ3



(c) DTLZ5

Fig. 1: Nondominated solution distribution of DTLZ function 1, 3, and 5 with respect to the preferred objectives f_2, f_4, and f_6

TABLE II: Average of the preferred objective function values of NSGA-II, MQEA, MOPSO, and MOPSO-PS.

(a) f_2

Problem	NSGA-II	MQEA	MOPSO	MOPSO-PS
DTLZ1	12.58	3.86	2.19	**0.04**
DTLZ2	0.29	**0.12**	0.43	0.18
DTLZ3	22.24	38.28	64.09	**0.33**
DTLZ4	0.28	0.23	0.20	**0.15**
DTLZ5	1.86	0.79	0.23	**0.13**
DTLZ6	0.33	0.21	0.1 6	**0.13**
DTLZ7	0.51	0.51	0.74	**0.35**

(b) f_4

Problem	NSGA-II	MQEA	MOPSO	MOPSO-PS
DTLZ1	15.88	16.88	11.18	**0.09**
DTLZ2	0.33	0.32	0.37	**0.30**
DTLZ3	18.40	144.08	107.88	**0.44**
DTLZ4	0.26	0.31	0.24	**0.12**
DTLZ5	2.49	1.99	0.69	**0.25**
DTLZ6	0.94	0.44	0.31	**0.28**
DTLZ7	0.53	0.44	0.72	**0.39**

(c) f_6

Problem	NSGA-II	MQEA	MOPSO	MOPSO-PS
DTLZ1	6.86	45.68	16.91	**0.13**
DTLZ2	**0.31**	0.73	0.44	0.41
DTLZ3	32.39	321.32	112.00	**0.54**
DTLZ4	0.37	0.19	0.18	**0.18**
DTLZ5	2.73	4.99	0.83	**0.51**
DTLZ6	1.34	1.02	0.70	**0.67**
DTLZ7	0.47	0.51	0.66	**0.45**

to a certain level for the other objectives. On the other hand, MOPSO-PS can solve this problem by employing the fuzzy measure.

V. Conclusions

In this paper, multi-objective particle swarm optimization with preference-based sorting (MOPSO-PS) was proposed. It could find preferred nondominated solutions according to the user's degree of consideration for the objectives. The effectiveness of the proposed MOPSO-PS was demonstrated by comparing its performance with that of NSGA-II, MQEA, and MOPSO for the DTLZ functions. It could successfully find more optimized solutions considering specific preferred objectives, which maintain the overall quality and the diversity of the solutions. Since the preference-based sorting algorithm is a general one for multiple criteria decision making, it can be used for any one of multi-objective optimization algorithms to select preferred solutions or to sort the obtained solutions based on the user's preference during optimization process.

VI. Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0000321).

TABLE III: Comparison results of NSGA-II, MQEA, MOPSO, and MOPSO-PS with the size of the dominated space and diversity metric for DTLZ functions over 100 runs.

(a) Average size of the dominated space

Problem	(NSGA-II)	(MQEA)	(MOPSO)	(MOPSO-PS)
DTLZ1	5.06	9.97	10.00	10.00
DTLZ2	10.00	9.90	9.89	9.98
DTLZ3	8.88	9.46	99.85	9.94
DTLZ4	10.00	9.95	9.90	9.99
DTLZ5	9.88	9.85	9.75	9.85
DTLZ6	7.56	99.75	9.86	9.81
DTLZ7	4.26	2.52	2.34	1.90

(b) Welch's t-test value of size of the dominated space

Problem	$t_{\mathcal{S}_4-\mathcal{S}_1}$	$t_{\mathcal{S}_4-\mathcal{S}_2}$	$t_{\mathcal{S}_4-\mathcal{S}_3}$
DTLZ1	14.28	4.62	-7.04
DTLZ2	-23.02	2.73	0.88
DTLZ3	67.21	116.20	0.96
DTLZ4	-7.58	1.31	0.93
DTLZ5	-20.08	-0.85	11.36
DTLZ6	77.70	13.97	-15.09
DTLZ7	-68.59	-21.28	-10.24

(c) Average diversity

Problem	(NSGA-II)	(MQEA)	(MOPSO)	(MOPSO-PS)
DTLZ1	138.15	69.48	79.19	84.70
DTLZ2	93.78	58.18	105.99	103.71
DTLZ3	106.99	53.22	70.09	80.21
DTLZ4	97.64	89.43	124.62	122.67
DTLZ5	110.65	127.52	81.48	79.31
DTLZ6	90.71	79.52	73.56	78.38
DTLZ7	135.73	131.22	70.25	58.88

(d) Welch's t-test value of diversity

Problem	$t_{\mathcal{D}_4-\mathcal{D}_1}$	$t_{\mathcal{D}_4-\mathcal{D}_2}$	$t_{\mathcal{D}_4-\mathcal{D}_3}$
DTLZ1	-31.18	9.25	3.44
DTLZ2	16.25	73.91	-3.06
DTLZ3	-22.73	32.72	11.85
DTLZ4	31.39	40.04	-2.18
DTLZ5	-59.36	-89.75	-4.66
DTLZ6	-21.14	-1.52	6.85
DTLZ7	-73.22	-64.25	-11.93

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[2] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proc. IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 1999.

[3] R. Eberhart, Y. Shi, and J. Kennedy, *Swarm intelligence*. Elsevier, 2001.

[4] A. Engelbrecht, *Fundamentals of computational swarm intelligence*. Wiley New York, 2005, vol. 1.

[5] K.-B. Lee and J.-H. Kim, "Mass-spring-damper motion dynamics-based particle swarm optimization," in *Proc. IEEE Congress on Evolutionary Computation*. IEEE, pp. 2348–2353.

[6] H.-M. Park and J.-H. Kim, "Potential and dynamics based particle swarm algorithm," in *Proc. IEEE Congress on Evolutionary Computation*. IEEE, pp. 2354–2359.

[7] K.-B. Lee and J.-H. Kim, "Particle swarm optimization driven by evolving elite group," in *Proc. IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 2114–2119.

[8] C. Coello and M. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," *Proc. IEEE Congress on Evolutionary Computation*, pp. 1051–1056, 2002.
[9] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," *Proc. IEEE Congress on Evolutionary Computation*, pp. 1677–1681, 2002.
[10] C. Coello, G. Pulido, and M. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.
[11] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proc. IEEE Swarm Intelligence Symposium*. IEEE, 2003, pp. 26–33.
[12] J. Alvarez-Benitez, R. Everson, and J. Fieldsend, "A MOPSO algorithm based exclusively on pareto dominance concepts," in *Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 459–473.
[13] M. Reyes-Sierra and C. Coello, "Multi-objective particle swarm optimizers: A survey of the state-of-the-art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287–308, 2006.
[14] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 3983–3990.
[15] J.-H. Kim, J.-H. Han, Y.-H. Kim, S.-H. Choi, and E.-S. Kim, "Preference-Based Solution Selection Algorithm for Evolutionary Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–15, doi: 10.1109/TEVC.2010.2 098 412, 2011.
[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
[17] Y.-H. Kim, J.-H. Kim, and K.-H. Han, "Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problems," in *Proc. IEEE Congress on Evolutionary Computation*. IEEE, 2006, pp. 2601–2606.
[18] Y.-H. Kim and J.-H. Kim, "Multiobjective quantum-inspired evolutionary algorithm for fuzzy path planning of mobile robot," in *Proc. IEEE Congress on Evolutionary Computation*. IEEE, 2009, pp. 1185–1192.
[19] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 2002, pp. 825–830.
[20] M. Sugeno, *Theory of fuzzy integrals and its applications*. Tokyo Institute of Technology Tokyo, Japan, 1974.
[21] J. Marichal, "An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 6, pp. 800–807, 2002.
[22] M. Sugeno, "Fuzzy measures and fuzzy integrals: a survey," *Fuzzy automata and decision processes*, vol. 78, no. 33, pp. 89–102, 1977.
[23] E. Takahagi, "On Identification Methods of λ-Fuzzy Measures using Weights and λ." *Journal of Japan Society for Fuzzy Theory and Systems*, vol. 12, no. 5, pp. 665–676, 2000.
[24] T. L. Saaty, "Decision making with the analytic hierarchy process," *International Journal of Services Sciences*, vol. 1, no. 1, pp. 83–98, 2008.
[25] G. Bajwa, E. Choo, and W. Wedley, "Effectiveness analysis of deriving priority vectors from reciprocal pairwise comparison matrices," *Asia-Pacific Journal of Operational Research*, vol. 25, no. 3, pp. 279–299, 2008.
[26] C. Raquel and P. Naval Jr, "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proceedings of the 2005 conference on Genetic and evolutionary computation*. ACM, 2005, pp. 257–264.
[27] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," *Berichte aus der Informatik, Shaker Verlag, Aachen-Maastricht*, 1999.
[28] H. Li, Q. Zhang, E. Tsang, and J. Ford, "Hybrid estimation of distribution algorithm for multiobjective knapsack problem," *Evolutionary Computation in Combinatorial Optimization*, pp. 145–154, 2004.
[29] B. Welch, "The generalization of student's problem when several different population variances are involved," *Biometrika*, vol. 34, no. 1/2, pp. 28–35, 1947.