Improved Version of a Multiobjective Quantum-inspired Evolutionary Algorithm with Preference-based Selection

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

Abstract—Multiobjective quantum-inspired evolutionary algorithm (MQEA) employs Q-bit individuals, which are updated using rotation gate by referring to nondominated solutions in an archive. In this way, a population can quickly converge to the Pareto optimal solution set. To obtain the specific solutions based on user's preference in the population, MQEA with preference-based selection (MQEA-PS) is developed. In this paper, an improved version of MQEA-PS, MQEA-PS2, is proposed, where global population is sorted and divided into groups, upper half of individuals in each group are selected by global evaluation, and selected solutions are globally migrated. The global evaluation of nondominated solutions is performed by the fuzzy integral of partial evaluation with respect to the fuzzy measures, where the partial evaluation value is obtained from a normalized objective function value. To demonstrate the effectiveness of the proposed MQEA-PS2, comparisons with MQEA and MQEA-PS are carried out for DTLZ functions.

I. INTRODUCTION

Quantum-inspired evolutionary algorithm (QEA) is an evolutionary algorithm, which employs the probabilistic mechanism inspired by the concept and principles of quantum computing, such as a quantum bit and superposition of states [1], [2]. QEA starts with a global search scheme and changes automatically into a local search scheme as generation advances because of its inherent probabilistic mechanism. Thus, QEA leads to a good balance between exploration and exploitation [3].

To solve multiobjective optimization problems, multiobjective quantum-inspired evolutionary algorithm (MQEA) has been developed [4]. Since the probabilistic individuals are updated by referring to nondominated solutions in an archive, the population converges to the Pareto-optimal solution set in MQEA. In other words, MQEA provides high quality solutions for multiobjective problems. However, dominance-based MOEAs are less effective in multiobjective problems because the number of nondominated solutions increases exponentially as the number of objectives increases.

Instead of dominance-based sorting, preference-based solution selection algorithm (PSSA) is proposed by considering user’s preference. Then, MQEA with preference-based selection (MQEA-PS), which employs PSSA in MQEA in each and every generation of evolutionary process, has been developed [5]. In MQEA-PS, the global population is sorted by preference-based sorting instead of dominance-based sorting, whereas the subpopulation is sorted by fast nondominated sorting. In this way, solutions that consider user’s preference are obtained. However, only specific objectives emphasized by user’s high degree of consideration converge closely to the Pareto-optimal set and the other objectives are far away from the Pareto-optimal set. Considering all objectives, the quality of solutions obtained by MQEA-PS falls off compared to MQEA.

In this paper, MQEA-PS2 is proposed to improve the performance, especially the hypervolume of MQEA-PS. The main differences of MQEA-PS2 compared to MQEA-PS are as follows: in MQEA-PS2, global population is sorted and divided into groups. After this process, upper half of individuals in each group are selected by the global evaluation and randomly migrated. By doing this procedure, globally migrated solutions include not only the most preferred solution, but also less preferred solutions. This causes an improvement in the quality of solutions over generation for multiobjective optimization problems. To demonstrate the effectiveness of the proposed MQEA-PS2, experiments are carried out for seven DTLZ functions. In addition, its nondominated solutions are compared with those of existing algorithms including MQEA [4] and MQEA-PS [5].

The rest of this paper is organized as follows: preference-based sorting algorithm is described in Section II. Section III proposes an improved version of MQEA-PS, MQEA-PS2. The experimental results are discussed in Section VI and concluding remarks follow in Section V.

II. PREFERENCE-BASED SORTING ALGORITHM

In the process of sorting nondominated solutions according to 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 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 users degree of consideration and the global evaluation is calculated by the fuzzy integral. The fuzzy measure and fuzzy integral are briefly described in the following and then detailed descriptions of the preference-based sorting method follow [5].
A. Fuzzy Measure and Fuzzy Integral

Fuzzy measure on the power set of X, denoted \( P(X) \), in the finite space \( X = \{x_1, \ldots, x_n\} \) 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 two axioms:

1. **Boundary condition:**
   \[
g(\emptyset) = 0, \quad g(X) = 1. \tag{1}
   \]
2. **Monotonicity:**
   \[
   \forall A, B \subseteq P(X), \quad \text{if} \ A \subseteq B \quad \text{then} \quad g(A) \leq g(B). \tag{2}
   \]

Fuzzy measures are classified as belief measure, plausibility measure, probability measure, etc. The 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 the other words, the sum of all belief measures is less than or equal to 1. The 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 plausibility measures is greater than or equal to 1. Lastly, the probability measure can be defined as a special case of either belief measure or plausibility measure, which satisfies an additional axiom on additivity property.

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

As a general representation of fuzzy measure, \( \lambda \)-fuzzy measure, \( g : P(X) \to [0, 1] \), is defined, which additionally satisfies the following axiom [6]:

\[
\forall A_i, j \in P(X), i, j = 1, \ldots, n, A_i \cap A_j = \emptyset \quad \text{and} \quad 1 < \lambda
\quad g(A_i \cup A_j) = g(A_i) + g(A_j) + \lambda g(A_i)g(A_j) \tag{5}
\]

where \( \lambda \) represents the degree of interaction between \( A_i \) and \( A_j \). \( \lambda \)-fuzzy measure is considered as belief measure, plausibility measure or probability measure depending on the value of \( \lambda \). 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 [7]. The belief measure indicates a positive interaction due to \( g(A_i \cup A_j) > g(A_i) + g(A_j) \). On the other hand, the plausibility measure indicates a negative interaction due to \( g(A_i \cup A_j) < g(A_i) + g(A_j) \). Lastly, the probability measure does not represent any interactions among criteria because it is identical as a conventional weighted sum, which satisfies the additivity.

For global evaluation of each solution over criteria with respect to the degree of consideration for each criteria, either Sugeno fuzzy integral or Choquet fuzzy integral [8] 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

\[
\int_X h \circ g = \max_i \min_i [h(x_i)g(E_i)] \tag{6}
\]
where \( h(x_1) \leq h(x_2) \leq \cdots \leq h(x_n) \) and \( E_i = \{x_i, x_{i+1}, \ldots, x_n\} \) 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 \( h(x_1) \leq h(x_2) \leq \cdots \leq h(x_n) \), \( E_i = \{x_i, x_{i+1}, \ldots, x_n\} \) and \( h(x_n) = 0 \), for \( x_i \in X \) and \( i = 1, \cdots, n \).

Note that \( x_i, i = 1, \ldots, n \), 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, the fuzzy integral can be used for the global evaluation of each solution.

B. Preference-based Solution Selection Algorithm (PSSA)

PSSA needs to define the comparative preference between two criteria and to decide either belief measure or plausibility measure. Thus, considering general multi-objective problems, it is a suitable method for the global evaluation compared to other existing methods [9]–[13]. Overall sorting procedure using \( \lambda \)-fuzzy measure and fuzzy integral is summarized in Algorithm 1, where each step is described in the following.

1) Define objectives in multi-objective problem as criteria

Multi-objective problems have predefined objectives, which have 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 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) Calculate 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:

\[ g(C) = g(\{c_1, c_2, \ldots, c_n\}) = g(\{c_1, \ldots, c_{n-1}\}) + g_n + \lambda g(\{c_1, c_2, \ldots, c_{n-1}\})g_n \]

\[ = (g_1 + g_2 + \cdots + g_n) + \lambda(g_1g_2 + g_1g_3 + \cdots + g_{n-1}g_n) \]

\[ + \lambda^2(g_1g_2g_3 + g_1g_2g_4 + \cdots + g_{n-2}g_{n-1}g_n) + \cdots + \lambda^{n-1}(g_1g_2 \cdots g_n) \]

\[ = 1 \] (8)

where \( C \) is the set of criteria, \( \{c_1, c_2, \ldots, c_n\} \) and \( g_i = g(\{c_i\}) \) for notational simplicity. Since (8) is \((n-1)\)-th order equation of \( \lambda \), it is quite difficult to solve the equation for \( \lambda \) given \( g_i \)'s if the number of criteria is more than three. Thus, the following procedure is employed to calculate the fuzzy measures [14].

a) Make pairwise comparison matrix

The pairwise comparison matrix of criteria, \( P \), which represents preference degrees between criteria, is defined as follows [15]:

\[
\begin{bmatrix}
p_{11} & p_{12} & \cdots & p_{1n} \\
p_{21} & p_{22} & \cdots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n1} & p_{n2} & \cdots & p_{nn}
\end{bmatrix}
\] (9)

where \( p_{ij} \) represents the preference degree between \( i \)-th criterion, \( c_i \), and \( j \)-th criterion, \( c_j \), \( p_{ii} = 1 \) and \( p_{ji} = 1/p_{ij} \).

b) Calculate normalized weight

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

\[ w_i = \frac{\sum_{j=1}^{n} p_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij}}. \] (10)

There are other methods to derive the priority vectors, like normalized weight, from pairwise comparison matrix [16]. 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 [14]. The transformation, \( \phi_{\lambda+1} : [0, 1] \times [0, 1] \rightarrow [0, 1] \) is defined as follows:

\[
\phi_{\lambda+1}(\xi, w_i) = \begin{cases} 
1 & \text{if } \xi = 1 \text{ and } w_i > 0 \\
0 & \text{if } \xi = 1 \text{ and } w_i = 0 \\
1 & \text{if } \xi = 0 \text{ and } w_i = 1 \\
0 & \text{if } \xi = 0 \text{ and } w_i < 1 \\
\frac{w_i}{(\lambda+1)^{w_i-1}} & \text{other cases}
\end{cases}
\] (11)

where \( \xi \) is another interaction degree of which value lies in \([0, 1]\). Then, \( \lambda \) is determined by \( \xi \), where \( \lambda = (1 - \xi)^2/\xi^2 - 1 \). 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:

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

where \( A \) is the 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. Note that the objective function values need to be normalized to 1.0 because \( h \) 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 the λ-fuzzy measure and the partial evaluation value obtained from step 2) and step 3), 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.

III. MQEA-PS2

A. Quantum-inspired Evolutionary Algorithm (QEA)

Building block of classical digital computer is represented by two binary states, ‘0’ or ‘1’, which is a finite set of discrete and stable state. In contrast, QEA utilizes a novel
representation, called a Q-bit representation [1], for the probabilistic representation that is based on the concept of qubits in quantum computing [17]. Quantum system enables the superposition of such state as follows:

$$\alpha|0\rangle + \beta|1\rangle$$

(13)

where $\alpha$ and $\beta$ are the complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$.

Qubit is shown in Fig. 1, which can be illustrated as a unit vector on the two dimensional space as follows:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

(14)

where $|\alpha|^2 + |\beta|^2 = 1$. Q-bit individual is defined as a string of Q-bits as follows:

$$\mathbf{q}_j^t = \left[ \begin{array}{cccc} \alpha_{11}^t & \alpha_{12}^t & \cdots & \alpha_{1m}^t \\ \beta_{11}^t & \beta_{12}^t & \cdots & \beta_{1m}^t \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{jm}^t & \alpha_{jm}^t & \cdots & \alpha_{jm}^t \\ \beta_{jm}^t & \beta_{jm}^t & \cdots & \beta_{jm}^t \end{array} \right]$$

(15)

where $m$ is the string length of Q-bit individual, and $j = 1, 2, \ldots, n$ for the population size $n$. The population of Q-bit individuals at generation $t$ is represented as $Q(t) = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n\}$.

Since Q-bit individual represents the linear superposition of all possible states probabilistically, diverse individuals are generated during the evolutionary process. The procedure of QEA and the overall structure for single-objective optimization problems are described in [1].

B. MQEA-PS2

MQEA-PS2 employs preference-based sorting and crowding distance sorting in the step of archive generation. Preference-based sorting gives the solutions whose specific objectives are more considered, and crowding distance sorting enables an archive to include spread solutions. The overall steps of MQEA-PS2 are summarized in Algorithm 2, and the procedure of MQEA-PS2 is depicted in Fig. 2. Each step is described in the following.

1), 2) In this step, $Q_k(0)$ containing $\mathbf{q}_j^0$, which consists of $\alpha_{ij}^0$ and $\beta_{ij}^0$, is initialized with $1/\sqrt{2}$, where $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, and $k = 1, 2, \ldots, s$. Note that $m$ is the string length of Q-bit individual, $n$ is the subpopulation size, and $s$ is the number of subpopulations. It means that one Q-bit individual, $\mathbf{q}_j^0$, represents the linear superposition of all possible states with same probability.

3) Binary solutions in $P_k(0)$ are produced by observing the states of $Q_k(0)$, where $P_k(0) = \{x_{1j}^0, x_{2j}^0, \ldots, x_{nj}^0\}$. One binary solution, $x_{ij}^0$, has a value either 0 or 1 according to the probability either $|\alpha_{ij}^0|$ or $|\beta_{ij}^0|$, $i = 1, 2, \ldots, m$, as follows:

$$x_{ij}^0 = \begin{cases} 0 & \text{if rand}[0,1] \geq |\beta_{ij}^0|^2 \\ 1 & \text{if rand}[0,1] < |\beta_{ij}^0|^2 \end{cases}$$

(16)

4) Evaluation is performed in each binary solution, $x_{ij}^0$, in $P_k(0)$.

5) Nondominated solutions in $P(0)$ are copied to the archive $A(0)$, where $A(0) = \{a_1^0, a_2^0, \ldots, a_l^0\}$ and $l (l \leq N)$ is the size of present archive.

6) The process runs until the termination condition is fulfilled. Termination condition is satisfied when the number of generation reaches the maximum number.

7), 9) In the while loop, binary solutions in $P_k(t)$ are produced through the multiple observing the states of $Q_k(t-1)$ and fitness values are calculated for each binary solution. Then, based on the dominance check, $x_{ij}^t$ is substituted by the best $x_{ij}^o$, where $o$ is the observation index.

10) Individuals in the population of size $2n$ ($P_k(t-1) \cup P_k(t)$) are sorted by the fast nondominated sort and the crowding distance method to select $n$ individuals [18]. The crowding distance method estimates the density of each individual.

Algorithm 2: Procedure of MQEA-PS2

1: $t \leftarrow 0$
2: Initialize $Q_k(t)$
3: Make $P_k(t)$ by observing the states of $Q_k(t)$
4: Evaluate $P_k(t)$
5: Store all solutions in $P_k(t)$ into $P(t)$ and nondominated solutions in $P(t)$ to $A(t)$
6: while (not termination condition) do
7: $t \leftarrow t + 1$
8: Make $P_k(t)$ by observing the states of $Q_k(t-1)$
9: Evaluate $P_k(t)$
10: Run the fast nondominated sort and crowding distance sort assignment $P_k(t) \cup P_k(t-1)$
11: Form $P_k(t)$ by the first $n$ individuals in the sorted population of size $2n$
12: Store all solutions in every $P_k(t)$ into $P(t)$
13: Sort the solutions in $A(t-1) \cup P(t)$ based on users’ preference
14: Divide the sorted solutions into $M$ groups
15: Run crowding distance sort for all groups
16: Form $A(t)$ by upper half solutions in each group
17: Migrate randomly selected solutions in $A(t)$ to every $R_k(t)$
18: Update $Q_k(t)$ using Q-gates referring to the solutions in $R_k(t)$
19: end while
Instead of crossover and mutation, the rotation gate $U$ compared to decide the update direction of Q-bit individuals. Every generation.

Note that the solutions in $R_k(t)$ are randomly replaced by the selected solutions, where $R_k(t) = \{r_1, r_2, \ldots, r_n\}$. Note that the solutions in $R_k(t)$ are employed as references to update Q-bit individuals, which are correspondent to the best solutions. Global random migration procedure occurs at every generation.

17) Solutions in $A(t)$ are randomly selected and solutions in every reference population, $R_k(t)$, are randomly replaced by the selected solutions, where $R_k(t) = \{r_1, r_2, \ldots, r_n\}$. Note that the solutions in $R_k(t)$ are employed as references to update Q-bit individuals, which are correspondent to the best solutions. Global random migration procedure occurs at every generation.

18) Fitness values of $r_i^j$ and $x_i^j$ in each subpopulation are compared to decide the update direction of Q-bit individuals. Instead of crossover and mutation, the rotation gate $U(\Delta \theta)$ is employed as an update operator for Q-bit individuals, which is defined as follows:

$$q_i^t = U(\Delta \theta) \cdot q_{i-1}^t$$

with

$$U(\Delta \theta) = \begin{bmatrix} \cos(\Delta \theta) & -\sin(\Delta \theta) \\ \sin(\Delta \theta) & \cos(\Delta \theta) \end{bmatrix}$$

where $\Delta \theta$ is the rotation angle of each Q-bit as shown in Fig. 1.

IV. EXPERIMENTAL RESULTS

A. Experimental Settings

Parameters setting for experiment is given in Table I. The number of variables for each DTLZ function was set to 11 for DTLZ1, 16 for DTLZ2 to DTLZ6, and 26 for DTLZ7 function. Belief measure ($\xi = 0.25$) was used for MQEA-PS and MQEA-PS2. As the preferred objectives, three objectives among seven objectives in DTLZ problems were selected. The degree of consideration for seven objectives 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, 0.0295, 0.295).

B. Performance Metrics

Two performance metrics, the size of dominated space and the diversity, were employed to evaluate the results of MQEA, MQEA-PS, and MQEA-PS2 [19]. The size of dominated space, $S$, was defined by the hypervolume of nondominated solutions. The quality of obtained solution set was high if this
TABLE II: Comparisons of preferred objective values between MQEA, MQEA-PS, and MQEA-PS2 for seven DTLZ functions

<table>
<thead>
<tr>
<th>Problem</th>
<th>MQEA</th>
<th>MQEA-PS</th>
<th>MQEA-PS2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>0.0609</td>
<td>0.0020</td>
<td>0.0009</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>0.0527</td>
<td>0.0230</td>
<td>0.0440</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>0.1301</td>
<td>0.1982</td>
<td>0.0801</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>0.0685</td>
<td>0.1338</td>
<td>0.0169</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>0.1313</td>
<td>0.0716</td>
<td>0.1228</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>0.3726</td>
<td>0.0975</td>
<td>0.6502</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>0.8331</td>
<td>0.0161</td>
<td>0.0181</td>
</tr>
</tbody>
</table>

(a) $f_2$

(b) $f_4$

(c) $f_6$

The proposed MQEA-PS2 was able to find the optimized solutions concentrated on the selected preferred objectives: $f_2$, $f_4$, and $f_6$. Table II indicates the average of preferred objective values over 10 runs, and Fig. 4(a), Fig. 4(b), and 4(c) show the corresponding distribution of nondominated solutions for DTLZ1, DTLZ2, and DTLZ7, respectively. The average values of $f_2$, $f_4$, and $f_6$ of MQEA-PS2 in Table II were the smallest for one third of DTLZ functions among the whole algorithms, while those of MQEA-PS were the smallest for two thirds of DTLZ functions. As shown in Fig. 4, the solutions of MQEA-PS2 indicated with square were distributed toward smaller value of $f_2$, $f_4$, and $f_6$ compared to those of the other algorithms for DTLZ1, DTLZ2, and DTLZ7. Thus, MQEA-PS2 had better performance on these three DTLZ functions.

The diversity and hypervolume of MQEA, MQEA-PS, and MQEA-PS2 are summarized in Table III(a) and Table III(b), respectively. The performance of the previous MQEA-PS was better than the proposed MQEA-PS2 in terms of preference-based selection. However, the proposed MQEA-PS2 had better performance than MQEA-PS in terms of multiobjective optimization because the hypervolume of MQEA-PS2 was larger than that of MQEA-PS in the most of DTLZ functions as shown in Table III(b).

V. CONCLUSION

This paper proposed an improved version of MQEA-PS, MQEA-PS2, which sorted and divided the global population into groups by using global evaluation. The upper half of individuals in each group formed a current archive and these individuals were randomly migrated. The global evaluation was performed by the fuzzy integral of partial evaluation with respect to the fuzzy measures, where the partial evaluation values were directly obtained from the normalized objective function values. The comparisons of nondominated solutions, hypervolume, and diversity between MQEA, MQEA-PS, and MQEA-PS2 for seven DTLZ functions confirmed that the proposed MQEA-PS2 was able to generate a higher value of preferred objectives and a larger size of dominates solutions compared to the existing two algorithms.
Fig. 4: Distribution of non-dominated solutions according to the preferred objective functions for DTLZ1, DTLZ2, and DTLZ7.

ACKNOWLEDGMENT

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 (2012-0000150).

REFERENCES


