Reference Point-based Nondominated Sorting Multi-objective Quantum-inspired Evolutionary Algorithm

Dick Sigmund
School of Electrical Engineering
KAIST
Daejeon, Republic of Korea
dick@rit.kaist.ac.kr

Jong-Hwan Kim, Fellow, IEEE
School of Electrical Engineering
KAIST
Daejeon, Republic of Korea
johkim@rit.kaist.ac.kr

***Abstract*—Various kinds of evolutionary algorithms have been developed to solve multi-objective optimization problems. One of them is multi-objective quantum-inspired evolutionary algorithm (MQEA) which utilizes quantum computing concepts to search the solution space effectively. MQEA used nondominated sorting and crowding distance calculation as the selection operator. This paper proposes MQEA with another kind of selection operator. The proposed RN-MQEA uses reference point-based nondominated sorting approach as the selection operator, which is adopted from NSGA-III. In the computer simulations, RN-MQEA is found to provide more diverse solutions compared to MQEA and NSGA-III in solving the DTLZ test problems.**

***Keywords*—Evolutionary algorithm, multi-objective optimization, multi-objective quantum-inspired evolutionary algorithm, reference points, nondominated sorting.**

I. Introduction

Evolutionary algorithm (EA) mechanism is inspired by biological evolution including reproduction, mutation, crossover, and selection based on Darwinian principle. Since the beginning of 1990s, a lot of research regarding EA has been conducted in order to solve multi-objective optimization problems [1]-[11]. Strength Pareto Evolutionary Algorithm (SPEA) [1] was introduced by E. Zitzler et al., which is achieved by using evolutionary process to provide Pareto optimal solutions. Later, E. Zitler et al. also improved SPEA to SPEA2 [2] which employs a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method.

On the other hand, Deb et al. also introduced (Nondominated Sorting Genetic Algorithm) NSGA [3] and later NSGA-II [4] which utilizes evolutionary process with nondominated sorting and crowding distance operator as its selection procedure in order to preserve the diversity of solutions. Recently, Deb and H. Jain also proposed NSGA-III [5] which uses all the procedure in NSGA-II except the selection procedure. The selection procedure is improved by predefined reference point to provide nondominated solutions. NSGA-III is proved to provide well-converged and well-diversified set of points.

There are also other kind of proposed EAs aside from those mentioned. Based on population-based incremental learning (PBIL) [6], multi-objective population-based incremental learning (MOPBIL) was presented [7]. MOPBIL was claimed to provide wider search space by randomly selecting nondominated solutions in archive when each element in probability vector gets updated [7]. On the other hand, based on particle swarm optimization (PSO) [8], dual multi-objective particle swarm optimization (DMOPSO) [9] was also introduced. DMOPSO utilizes dual stage objectives: primary objectives to find nondominated solution and secondary objectives to provide user-preferred solutions with better diversity from the nondominated solutions obtained by primary objectives. Also, multi-objective quantum-inspired evolutionary algorithm (MQEA) which is based on quantum-inspired evolutionary algorithm (QEA) [10] was proposed to provide solutions that is close to Pareto-optimal front with better spread of nondominated set [11]. MQEA utilizes nondominated sorting and crowding distance as its selection operator. The results showed that MQEA is capable to provide better diversity maintenance and proximity performance.

This paper proposes MQEA with reference point-based nondominated sorting, called RN-MQEA, to improve the spread of nondominated set. Basically, the procedure in MQEA proposed in [11] is still used; the improvement is in its crowding distance operator. This operator is substituted by other method which uses predefined reference point to select individuals in the population. The reference point-based nondominated sorting is based on NSGA-III framework. Thus, by combining MQEA with this selection operator in NSGA-III is expected to improve the performance of solving multi-objective optimization problems. Computer simulations verify that RN-MQEA provides more diverse solutions compared to MQEA and NSGA-III in solving the DTLZ test problems.

This paper is organized as follows. Section II presents an overview of MQEA. Section III describes the procedure of proposed RN-MQEA. Simulation results on DTLZ test problems that use RN-MQEA and its comparison with other algorithms are presented in Section IV. Finally, concluding remarks follow in Section V.

II. Multi-objective Quantum-inspired Evolutionary Algorithm (MQEA)

Similar to quantum-inspired evolutionary algorithm (QEA) [10], MQEA [11] adopts the concept and principles of quantum

```
Procedure MQEA
begin
          t ← 0
i)        initialize Q(t)
ii)       make P(t) by observing the states of Q(t)
iii)      evaluate P(t)
iv)       while (not termination condition) do
          begin
             t ← t + 1
v)           make P(t) by observing the states of Q(t)
vi)          evaluate P(t)
vii)         run the fast nondominated sort algorithm
             for P(t) U P(t-1)
viii)        calculate crowding distance and sort
ix)          P(t) is formed by the first N elements in the
             sorted population 2N
x)           Q(t) is classified into several groups
xi)          update Q(t) by using Q-gates refer to best
             group
          end
end
```

Fig. 1. Procedure of MQEA

computing such as uncertainty, superposition, and interference. In digital computer, data is encoded into binary digits which is consists of two definite states (0 or 1). However, quantum computation utilizes quantum bits (qubits) which are in the superposition of states as follows:

$$\alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β are the complex numbers that specify the probability amplitudes of the corresponding states; satisfying $|\alpha|^2 + |\beta|^2 = 1$.

A Q-bit, a smallest unit of information in MQEA, is defined with a pair of numbers as follows:

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \tag{2}$$

where $|\alpha|^2 + |\beta|^2 = 1$. A string of Q-bits as an individual **q** at t generation is represented as follows:

$$\mathbf{q}_j^t = \left[\begin{array}{c|c|c|c}\alpha_{j,m-1}^t & \alpha_{j,m-2}^t & \dots & \alpha_{j,0}^t\\ \beta_{j,m-1}^t & \beta_{j,m-2}^t & \dots & \beta_{j,0}^t\end{array}\right] \tag{3}$$

where m is the string length of the Q-bits individual **q** and j = 1, 2, …, n for the population size n. The population of Q-bit individuals, $Q(t)$, is represented as follows:

$$Q(t) = \{\mathbf{q}_1^t, \mathbf{q}_2^t, \dots, \mathbf{q}_n^t\}. \tag{4}$$

In general, the procedure of MQEA follows the procedure of quantum-inspired evolutionary algorithm (QEA) [10]. However, there are some procedure changes required in order to develop it [11]. The procedure of MQEA is summarized in Fig. 1.

i) Initiate linear superposition of all possible states in $Q(t)$ individual with $1/\sqrt{2}$.

ii) Form binary solutions $P(t)$ from making multiple observation of the states $Q(t\text{-}1)$ as in [10].

iii) Evaluate each binary solution for the fitness value.

iv) Do looping condition which will be the termination condition of MQEA.

v) Form binary solutions $P(t)$ from making multiple observation of the states $Q(t\text{-}1)$ as in [10].

vi) Evaluate each binary solution for the fitness value.

vii) Form population of *2N* elements with previous and current generation individuals ($P(t\text{-}1) \cup P(t)$) and sort them by nondominated sorting algorithm introduced in [4].

viii) Calculate the crowding distance of each individual from step vii) and sort them.

ix) Select the first N elements of the *2N* population and form $P(t)$. The Q-bit individual corresponded to $P(t)$ is also copied to $Q(t)$.

x) Group classification rule based on [11] is conducted in this step.

xi) Update Q-bit individual using Q-gates. Q-gates change the role of perturbation operation (crossover and mutation) in genetic algorithm. The rotation gate used is based on the one used in QEA [10].

III. Reference Point-based Nondominated Sorting Multi-objective Quantum-inspired Evolutionary Algorithm (RN-MQEA)

A. Main procedure

The procedure in RN-MQEA is basically the same as the one in MQEA. However, there are significant changes in the selection operator: step viii) and step ix) in MQEA procedure. The changes are shown from step ix) to step xix) in Fig. 2; and also there is reference points generation added in step ii). These changes are made based on NSGA-III [5] framework.

i) Initiate linear superposition of all possible states in $Q(t)$ individual with $1/\sqrt{2}$, same as MQEA.

ii) Predefine the reference points in a structured manner. More description will be given in Subsection B.

iii) ~ viii) These steps are the same as those steps in MQEA procedure.

ix) ~ x) Form a population $S(t)$ from each front (F_1, F_2, ...) arranged from nondominated sorting algorithm. Note that not all fronts will be included. If the population $S(t)$ is equal or larger than N elements, inclusion of fronts will be stopped.

xi) Record the last front to be included in population $S(t)$ of step viii) ~ ix) as a variable F_{last}.

Procedure RN-MQEA
begin
$t \leftarrow 0$
i) initialize $Q(t)$
ii) make $P(t)$ by observing the states of $Q(t)$
iii) evaluate $P(t)$
iv) **while** (**not** termination condition) **do**
begin
$t \leftarrow t + 1$
v) make $P(t)$ by observing the states of $Q(t)$
vi) evaluate $P(t)$
vii) run the fast nondominated sort algorithm for $P(t)$ U $P(t\text{-}1)$ that yields F_1, F_2, ...
viii) **while** ($|S(t)| < N$) **do**
begin
ix) $S(t) = S(t)$ U F_i
$i \leftarrow i + 1$
end
x) the last front to be included is $F_{last} = F_i$
xi) **if** ($|S(t)| = N$) **then**
xii) $P(t) = S(t)$
xiii) **else**
xiv) form $P(t)$ with F_1, F_2, ..., and $F_{last\text{-}1}$
xv) normalize objectives and create reference set
xvi) associate each member of $S(t)$ with a reference point
xvii) compute niche count of reference point
xviii) choose N - $|P(t)|$ members one at a time from F_{last} to form the rest of $P(t)$
end if
xix) $Q(t)$ is classified into several groups
xx) update $Q(t)$ by using Q-gates refer to best group
end
end

Fig. 2. Procedure of RN-MQEA

xii) ~ xiii) Check whether the number of population $S(t)$. If it is equal to N elements, then $S(t)$ will be passed to $P(t)$ and the procedure will directly continue to step xx).

xiv) This step is the condition where population $S(t)$ has more than N elements. This will lead the procedure to step xv).

xv) Form $P(t)$ from front F_1 to $F_{last\text{-}1}$. This will yield $P(t)$ with less than N elements.

xvi) Normalize the objective values of each individual in the population $S(t)$. More description will be given in Subsection C.

xvii) Associate each individual in the population $S(t)$ with a reference point defined in step ii). More description will be given in Subsection D.

xviii) ~ xix) Compute the number of individuals that are in association with a reference point; this is known as niche reference point. Then, decide which individuals in the F_{last} are needed to be included in population $P(t)$. More description will be given in Subsection E.

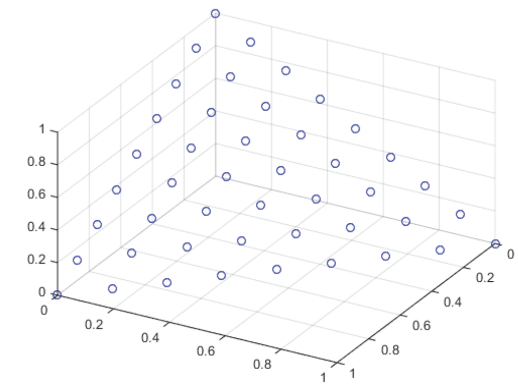

Fig. 3. Diverse set of reference points with $\#OBJ = 3$ and $p = 8$

xx) ~ xxi) These steps are the same as those steps in MQEA procedure.

B. Generation of Reference Points

The reference point generation used in NSGA-III [5] is based on Das and Dennis's method [12]. In RN-MQEA, the same method is also adopted.

The approach is done by putting a number of reference points on a normalized hyperplane. For example, a three-objective problem will yield a triangle-shaped hyperplane. If we set $p = 8$, where p is the number of divisions on each side of the triangle, we will get 45 reference points on the normalized hyperplane (shown in Fig. 3).

C. Normalization of Objective Values

Since we are using reference points to determine which members need to be added to the population $P(t)$, we normalize

Procedure Normalization
begin
i) **for** $i = 1$ **to** $\#OBJ$ **do**
ii) find the ideal points in front F_1 for the ith objective, denoted as $minOBJ_i$
iii) convert the objective value of each member in $S(t)$, denoted as $cOBJ_i = OBJ_i - minOBJ_i$
iv) compute the extreme points of $S(t)$ in the ith objective, denoted as exp_i
end
v) find the intercept points $a_{1..\#OBJ}$ between each objective value and linear hyperplane from $exp_{1..\#OBJ}$
vi) normalize all the objective values in $S(t)$ using Equation (5)
end

Fig. 4. Procedure of objective values normalization

all the objective values of population $S(t)$. The procedure for normalization is summarized Fig. 4 [5].

i) Do iteration for every objective value.

ii) Calculate the ideal point $minOBJ_i$ by finding the minimum corresponding objective value in front F_1.

iii) Convert to the new objective value $cOBJ_i$ by subtracting the objective value OBJ_i with the ideal point $minF_i$.

iv) Calculate the extreme points by identifying the solutions that make the achievement scalarization function minimum. Note that the achievement scalarization function here is formed with the objective value *OBJ* and the weight vector close to the corresponding objective value.

v) Calculate the intercept points $a_{1..\#OBJ}$. Note that these points are the interception between the corresponding objective value with its corresponding extreme point obtained from step iv).

vi) Normalize the objective values with the following equation:

$$normOBJ_i = \frac{cOBJ_i}{a_i}, \quad for\ i = 1, 2, \ldots, \#OBJ. \tag{5}$$

Note that once the normalization procedure is finished, the normalized objectives values at the intercept, $normOBJ_{i,}$ is 1. If we construct a hyperplane of these intercept points, it will yield $\sum_{i=1}^{\#OBJ} normOBJ[i] = 1$.

D. Association of Reference Points with Population Members

After normalizing all the objective values in $S(t)$, we can now associate them with the reference points. Fig. 5 summarizes the procedure for doing the association [5].

i) Do iteration for every reference points.

ii) Draw reference line $refL_i$ that pass through the origin and the corresponding reference point rps_i.

iii) Do iteration for each member in population $S(t)$.

iv) Do iteration for every reference points.

v) Calculate the perpendicular distance of member in $S(t)$ with every reference lines.

vi) Find the closest perpendicular distance, denoted as index *idxmin*, and check which front this member belongs to.

vii) ~ viii) If the member belongs to the last front F_{last}, then add this member as a potential member in $refL_{idxmin}.Potential$.

ix) ~ x) If the member does not belong to the last front F_{last}, then just add the number of members associated with the corresponding reference line, $refL_{idxmin}.sum + 1$.

Once the association procedure is done, every normalized objective values in population $S(t)$ belong to the associated member of one reference point. The associated members are the normalized objective values which have shortest perpendicular distance with the reference line stretching from the origin through the corresponding reference point. Note that one reference point might have more than one associated members of the normalized objective values from population $S(t)$.

```
Procedure Association
begin
i)      for i = 1 to #rps do
ii)         make reference line from origin to rps_i,
            denoted as refL_i
        end
iii)    for each member of S(t) do
iv)         for j = 1 to #rps do
v)              calculate the perpendicular
                distance of refL_j to cOBJ of
                each member of S(t)
            end
vi)         find the minimum perpendicular distance,
            denoted as idxmin
vii)        if member belong to F_last then
viii)           add corresponding member to
                refL_idxmin.Potential
ix)         else
x)              refL_idxmin.sum = refL_idxmin.sum + 1
            end if
        end
end
```

Fig. 5. Procedure of reference points with population members association

E. Operation of Niche-preservation

Now that all members in $S(t)$ are already associated to one reference point, we need to decide which members in front F_{last} should be included in population $P(t)$. Niche-preservation operation is used for this purpose. The procedure for doing the niche-preservation operation is summarized in Fig. 6 [5].

i) Do the loop while the number of members in population $P(t)$ is still less than N population size.

```
Procedure Niche Preservation
begin
i)      while (|P(t)| < N) do
ii)         look for the refL_1..#rps.sum with the
            minimum value, denoted as refLmin
iii)        find the member in refLmin which belong
            to F_last, return found or not found
iv)         if found then
v)              add member from refLmin.Potential
                to population P(t)
vi)             remove corresponding member from
                refLmin.Potential
vii)            refLmin.sum = refLmin.sum + 1
viii)       else
ix)             remove refLmin from refL
            end if
        end
end
```

Fig. 6. Procedure of niche-preservation operation

ii) Find the reference line, *refLmin,* with the minimum number of members, min($refL_{1..\#rps}.sum$).

iii) Search whether there is a member that belongs to the last Front F_{last} in *refLmin*.

iv) ~ vii) If there is a member that belongs to the last Front F_{last} in *refLmin*, then we will add this member to population $P(t)$. After adding this member, we need to remove it from *refLmin.Potential* so it will not be included at the next iteration. And also, the number of member *refLmin.sum* needs to be added since a new member is added to population $P(t)$.

viii) ~ ix) If there is no member that belongs to the last front F_{last} in *refLmin*, we will just remove this corresponding *refLmin* so it will not be used anymore in next iteration.

Niche-preservation operation plays a role in maintaining the diversity of the solutions by selecting which associated members of reference points should be included in population $P(t)$. As described above, the selection procedure is based on the number of members associated with the reference line *refLmin*. Notice that after niche-preservation operation is finished, the number of population $P(t)$ will be equal to N.

IV. Computer Simulations

The performance metrics used, simulation environment, and simulation results are presented in this section.

A. Performance Metrics

In order to compare the results obtained from different kinds of algorithms, two performance metrics were employed in the computer simulations: hypervolume and diversity metrics. Hypervolume is defined as the volume of the dominated set in the objective space. Large hypervolume value means that the quality of the solutions in the global population is high.

Diversity, the distribution and spread of the solutions, is defined [13] as follows:

$$\bar{D} = \frac{\sum_{k=1}^{n}(f_k^{(max)} - f_k^{(min)})}{1+\sqrt{\frac{1}{|N_O|}\sum_{i=1}^{|N_O|}(d_i-\bar{d})^2}}, \qquad (6)$$

where N_0 is the set of nondominated solutions, d_i is the minimal distance between the ith solution and the nearest neighbor, and $\bar{d}$ is the mean value of all d_i. $f_k^{(max)}$ ($f_k^{(min)}$) represents the maximum (minimum) objective values of the kth objective. Larger value of this metrics means a better diversity of the solutions.

TABLE I.

Parameter Settings of MQEA, RN-MQEA, and NSGA-III for DTLZ Test Problems

Algorithm	Parameter	Value
MQEA, DMQEA, and RN-MQEA	Number of generations	300
	Subpopulation size	4
	Number of subpopulations	25
	Number of multiple observation	10
	Rotation angle	0.20
NSGA-III	Number of generations	300
	Crossover rate	1.0
	Crossover distribution indexes	30
	Mutation distribution indexes	20

B. Simulation Environment

In the computer simulations, RN-MQEA was compared with MQEA and NSGA-III to solve three-objective to ten-objective Deb-Thiele-Laumanns-Zitzler (DTLZ) test problems [14]. Note that the NSGA-III code used is not an official released code by [5], instead it is obtained from [15].

Table I presents all of these algorithm parameters used in this paper. The number of variables is ($\#OBJ + K$ - 1), where $\#OBJ$ is number of objectives and K is different for each DTLZ problem. K was set to five for DTLZ1 problem, ten for DTLZ2, DTLZ3, DTLZ4, DTLZ5, and DTLZ6 problems, and twenty for DTLZ7 problem. All of the algorithms were run for ten times in each test problems simulation.

Beside the DTLZ test problems, RN-MQEA was also simulated with some selected reference points to provide preferred solutions in Subsection D.

C. Simulation Results

Table II shows the comparison of diversity and hypervolume metrics in solving three-objective DTLZ test problems for RN-MQEA, NSGA-III, and MQEA. Table III shows the same comparison as Table II; however, it is for five-objective test problems. Table IV is for eight-objective test problems; Table V is for ten-objective test problems.

Beside the hypervolume and diversity metrics, the *t*-test of both metrics with 0.05 significance level was also employed to statistically compare the performance metrics of the three algorithms, also shown in the tables as *p*-value. t-test is a statistical hypothesis test to determine test decision for the null hypothesis that the data come from the normal distribution. Large *p*-value (> 0.05) indicates the null hypothesis is supported, while small *p*-value (≤ 0.05) indicates the null hypothesis is rejected which means the alternative hypothesis is supported.

For three-objective problem, NSGA-III provides more diverse solution than RN-MQEA in DTLZ1, DTLZ4, DTLZ6, DTLZ7. However, as the number of objectives increased (five-objective, eight-objective, and ten-objective test problems), RN-MQEA outperformed NSGA-III for most cases in both diversity and hypervolume metrics. The *p*-value indicates that RN-MQEA and NSGA-III are statistically distinguishable. On the other hand, compared to MQEA, RN-MQEA appears to provide better solutions in most of the cases for three-objective to ten-objective DTLZ test problems. The *p*-value of RN-MQEA and MQEA t-test also indicates both of the performance metrics are statistically different.

D. Preference-based Selected Reference Points Simulation

In solving multi-objective optimization problems, decision making of selecting preferred solutions is often required. In other words, some users are interested in solutions that is nearby a preferred region. RN-MQEA utilizes reference points to provide nondominated solutions that spreads through all the region of objectives. Thus, in order to provide user-preferred solutions, the reference points generation should be modified.

Reference points in RN-MQEA are associated with the members in the population for the purpose of solutions selection. Consequently, by generating reference points that are located

TABLE II.

RESULTS COMPARISON OF RN-MQEA, NSGA-III, AND MQEA
WITH DIVERSITY METRICS, HYPERVOLUME METRICS, AND ITS T-TEST FOR THREE-OBJECTIVE DTLZ TEST PROBLEMS

DIVERSITY					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	97.415720	**138.76778**	62.982182	1.27012×10^{-1}	5.02866×10^{-2}
DTLZ2	**165.72496**	99.221702	123.97487	1.57760×10^{-15}	2.23550×10^{-8}
DTLZ3	**83.161023**	37.359410	78.820198	1.22331×10^{-2}	8.39942×10^{-1}
DTLZ4	83.161023	**120.70192**	117.83700	9.86967×10^{-2}	2.38095×10^{-1}
DTLZ5	**923.87501**	235.85852	577.56035	3.20640×10^{-13}	2.68140×10^{-6}
DTLZ6	68.759162	**77.274554**	71.260480	4.82682×10^{-1}	8.67727×10^{-1}
DTLZ7	83.556538	92.166600	**94.027717**	1.50415×10^{-1}	6.42773×10^{-1}
HYPERVOLUME					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	949.294	**996.568**	951.646	1.72631×10^{-7}	6.85647×10^{-1}
DTLZ2	**986.284**	981.626	942.011	3.21789×10^{-1}	1.45419×10^{-5}
DTLZ3	**976.284**	939.871	765.824	2.16858×10^{-2}	1.44276×10^{-7}
DTLZ4	920.059	**979.653**	899.649	3.98039×10^{-5}	6.56229×10^{-2}
DTLZ5	**987.777**	978.865	922.075	1.31750×10^{-2}	1.60294×10^{-2}
DTLZ6	957.873	912.741	**973.835**	7.82095×10^{-3}	5.98739×10^{-2}
DTLZ7	615.645	**641.320**	617.311	1.17333×10^{-1}	7.88551×10^{-2}

TABLE III.

RESULTS COMPARISON OF RN-MQEA, NSGA-III, AND MQEA
WITH DIVERSITY METRICS, HYPERVOLUME METRICS, AND ITS T-TEST FOR FIVE-OBJECTIVE DTLZ TEST PROBLEMS

DIVERSITY					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	89.44819	**136.5397**	62.14845	5.83279×10^{-3}	4.55586×10^{-3}
DTLZ2	**86.72549**	79.43354	81.50921	2.20550×10^{-2}	1.53269×10^{-1}
DTLZ3	**63.51847**	45.08173	55.58263	4.10101×10^{-2}	4.52467×10^{-1}
DTLZ4	**108.8338**	82.89593	103.0994	9.77926×10^{-2}	4.23915×10^{-1}
DTLZ5	**511.6641**	101.4586	186.1128	4.24341×10^{-4}	3.41916×10^{-3}
DTLZ6	53.80257	**73.86525**	56.33557	5.07816×10^{-6}	6.18789×10^{-1}
DTLZ7	70.23918	87.99216	**138.7897**	4.90114×10^{-4}	3.49516×10^{-6}
HYPERVOLUME					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	92592.8	**99962.4**	95869.7	2.98972×10^{-5}	3.72783×10^{-2}
DTLZ2	95757.7	**99477.4**	94397.5	1.88832×10^{-4}	2.79264×10^{-1}
DTLZ3	80090.7	**97830.5**	71261.6	6.45020×10^{-6}	4.06496×10^{-2}
DTLZ4	91546.8	**99731.2**	90014.7	8.73908×10^{-6}	1.88256×10^{-1}
DTLZ5	**94497.7**	93933.6	90070.0	6.94258×10^{-1}	9.37936×10^{-4}
DTLZ6	**83258.9**	78395.2	80191.4	1.33610×10^{-1}	5.48303×10^{-1}
DTLZ7	**47290.9**	29916.8	47140.3	9.29161×10^{-8}	2.21150×10^{-1}

nearby the preferred region, RN-MQEA will provide user preferred solutions. In this subsection, instead of using the reference points shown in Fig. 3, reference points in Fig. 7 (Experiment 1) and Fig. 9 (Experiment 2) were used as a preferred solution selection. Both of the simulations were conducted for three-objective DTLZ2 test problem.

In Experiment 1, the result obtained (Fig. 8) shows that the solutions provided are concentrated nearby the reference points defined in Fig 7. Same as Experiment 1, the solutions obtained in Experiment 2 (Fig. 10) are also concentrated nearby the reference points defined in Fig. 9. These implicate that by selecting reference points in preferred region, RN-MQEA has the potential to provide user preferred solutions in multi-objective optimization problems.

TABLE IV.

RESULTS COMPARISON OF RN-MQEA, NSGA-III, AND MQEA
WITH DIVERSITY METRICS, HYPERVOLUME METRICS, AND ITS T-TEST FOR EIGHT-OBJECTIVE DTLZ TEST PROBLEMS

DIVERSITY					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	**89.41656**	83.71142	64.61216	6.52109×10^{-1}	4.02289×10^{-2}
DTLZ2	80.12275	**96.10528**	77.90747	1.40792×10^{-2}	2.33727×10^{-1}
DTLZ3	**69.05601**	61.75149	55.54977	6.26954×10^{-1}	6.35449×10^{-2}
DTLZ4	**113.8526**	98.64624	88.40032	4.73171×10^{-1}	4.86295×10^{-1}
DTLZ5	**322.4386**	100.2390	153.3619	6.02130×10^{-3}	2.98535×10^{-2}
DTLZ6	**68.62539**	63.08344	68.58573	3.83624×10^{-1}	9.75315×10^{-1}
DTLZ7	83.80326	84.60009	**147.3618**	9.15724×10^{-1}	3.55174×10^{-7}
HYPERVOLUME					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	94276300	**99896900**	93731300	3.21699×10^{-5}	6.87948×10^{-1}
DTLZ2	97127900	**99696600**	94303700	2.10511×10^{-6}	9.69267×10^{-5}
DTLZ3	**69760600**	55698200	58134300	2.99645×10^{-1}	2.79665×10^{-1}
DTLZ4	91140200	**99939000**	90563400	3.57889×10^{-7}	6.32170×10^{-1}
DTLZ5	**93312800**	92640900	88943900	6.97546×10^{-1}	2.41624×10^{-3}
DTLZ6	**76335700**	60675500	71044100	1.31242×10^{-4}	2.18646×10^{-1}
DTLZ7	**31035100**	29516000	30313000	2.16293×10^{-13}	1.57033×10^{-3}

TABLE V.

RESULTS COMPARISON OF RN-MQEA, NSGA-III, AND MQEA
WITH DIVERSITY METRICS, HYPERVOLUME METRICS, AND ITS T-TEST FOR TEN-OBJECTIVE DTLZ TEST PROBLEMS

DIVERSITY					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	**102.0376**	95.17670	67.72907	4.92101×10^{-1}	1.74239×10^{-2}
DTLZ2	80.51177	**85.78128**	81.08340	3.82527×10^{-1}	8.39678×10^{-1}
DTLZ3	**67.97803**	54.09864	60.15323	1.45660×10^{-2}	2.33101×10^{-1}
DTLZ4	**132.5106**	103.1169	96.72596	1.45754×10^{-1}	9.24998×10^{-2}
DTLZ5	**186.8413**	128.8047	164.8984	4.16023×10^{-3}	2.69086×10^{-1}
DTLZ6	66.48750	**106.5612**	72.51591	1.12157×10^{-4}	2.68906×10^{-1}
DTLZ7	84.92734	128.3800	**136.1780**	3.85530×10^{-3}	2.57624×10^{-3}
HYPERVOLUME					
Test Problem	RN-MQEA	NSGA-III	MQEA	RN-MQEA vs NSGA-III (p-value)	RN-MQEA vs MQEA (p-value)
DTLZ1	9.54×10^{9}	$\mathbf{9.99 \times 10^{9}}$	9.79×10^{9}	8.71290×10^{-8}	4.67790×10^{-4}
DTLZ2	9.63×10^{9}	$\mathbf{9.95 \times 10^{9}}$	9.45×10^{9}	2.28057×10^{-4}	6.76206×10^{-2}
DTLZ3	$\mathbf{7.35 \times 10^{9}}$	4.15×10^{9}	4.79×10^{9}	1.43332×10^{-2}	1.13174×10^{-2}
DTLZ4	9.19×10^{9}	$\mathbf{9.99 \times 10^{9}}$	9.41×10^{9}	4.49849×10^{-8}	1.24100×10^{-1}
DTLZ5	$\mathbf{9.40 \times 10^{9}}$	8.86×10^{9}	8.91×10^{9}	1.21934×10^{-2}	2.09148×10^{-3}
DTLZ6	$\mathbf{7.96 \times 10^{9}}$	5.34×10^{9}	7.02×10^{9}	1.21512×10^{-7}	2.53805×10^{-2}
DTLZ7	$\mathbf{2.33 \times 10^{9}}$	2.29×10^{9}	2.31×10^{9}	6.78415×10^{-24}	2.95873×10^{-2}

V. CONCLUSION

In this paper, a novel variant of MQEA, called RN-MQEA, was developed with a reference-point based nondominated sorting selection operator. In computer simulations, RN-MQEA was compared with MQEA and NSGA-III for solving three-objective to ten-objective DTLZ test problems. The proposed RN-MQEA could provide wider spread of solutions in solving multi-objective problems compared to its predecessor, MQEA. Compared to NSGA-III, RN-MQEA showed improved performance in hypervolume and diversity metrics in most cases. Based on the selected reference points, RN-MQEA is also able to provide preferred nondominated solutions in the experiment conducted.

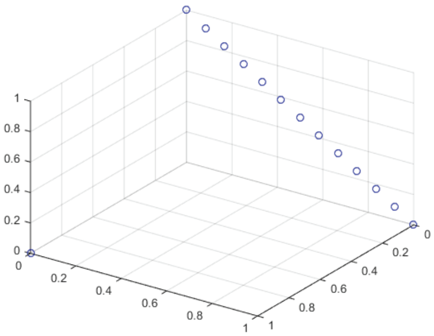

Fig. 7. Selected set of reference points for Experiment 1

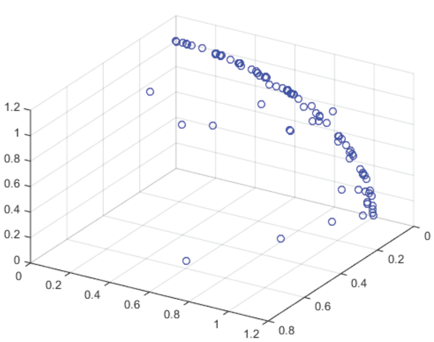

Fig. 8. Provided solutions for three-objective DTLZ2 Test Problem by RN-MQEA in Experiment 1

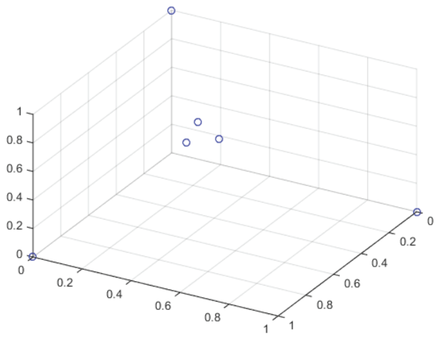

Fig. 9. Selected set of reference points for Experiment 2

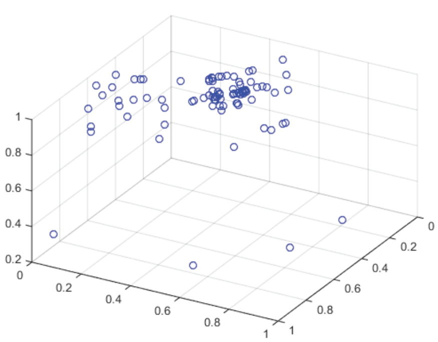

Fig. 10. Provided solutions for three-objective DTLZ2 Test Problem by RN-MQEA in Experiment 2

Acknowledgment

This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea MSIP (No. B0101-15-0551, Development of virtual creatures with digital DNA).

References

[1] E. Zitzler, and L. Thiele, “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.

[2] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm,” *Technical Report 103*, Computer Engineering and Communication Networks Lab, Swiss Federal Institute of Technology, Zurich (2001).

[3] N. Srinivas and K. Deb, “Multiobjective Function Optimization using Nondominated Sorting Genetic Algorithms,” *Evol. Comput*., vol. 2, no. 3, pp. 221-248, Fall 1995.

[4] 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.

[5] K. Deb and H. Jain "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints", *IEEE Trans. Evol. Comput*., vol. 18, no. 4, pp.577 -601, 2014.

[6] S. Baluja, “Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,” Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994.

[7] J.-H. Kim, Y.-H. Kim, S.-H. Choi, and I.-W. Park, “Evolutionary multiobjective optimization in robot soccer system for education*,” IEEE Computat. Intell. Mag.*, vol. 4, no. 1, pp. 31–41, Feb. 2009.

[8] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *in Proceedings of IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[9] K.-B. Lee and J.-H. Kim, "DMOPSO: Dual Multi-Objective Particle Swarm Optimization," *in Proc. 2014 IEEE World Congress on Computational Intelligence*, Beijing, China, Jun. 2014.

[10] K.-H. Han and J.-H Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” *IEEE Trans. Evolutionary Computation*, vol. 6, pp. 580-593, 2002.

[11] 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*, Vancouver, Canada, pp. 9151-9156, Jul. 2006.

[12] I. Das and J. Dennis, “Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems*,” SIAM J. Optimization,* vol. 8, no. 3, pp. 631–657, 1998.

[13] H. Li, Q. Zhang, E. Tsang, and J. A. Ford, “Hybrid estimation of distribution algorithm for multiobjective knapsack problem,” *in Proc. 4th EvoCOP, LNCS 3004.* 2004, pp. 145–154.

[14] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multi-objective optimization,” Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Tech. Rep., 2001.

[15] T.-C Chiang. (2016, Jan.). *nsga3cpp*: A C++ implementation of NSGA-III [Online]. Available: http://web.ntnu.edu.tw/~tcchiang/publications/nsga3cpp/nsga3cpp.htm