Multiobjective Particle Swarm Optimization with Preference-based Sort and its Application to Path Following Footstep Optimization for Humanoid Robots

Ki-Baek Lee and Jong-Hwan Kim, *Fellow, IEEE*

***Abstract*—This paper proposes Multiobjective Particle Swarm Optimization with Preference-based Sort (MOPSO-PS) in which the user's preference is incorporated into the PSO update process to determine the relative merits of nondominated solutions while handling the mutual dependencies and priorities of objectives. In MOPSO-PS, the user's preference is represented as the degree of consideration for each objective using the fuzzy measure. The global evaluation of a particle, which represents the quality of the particle according to the user's preference, is carried out by the fuzzy integral, which integrates the partial evaluation value of each objective with respect to the degree of consideration. Since the global best attractor of each particle in the population is randomly chosen among the nondominated particles having a relatively higher global evaluation value in each PSO update iteration, the optimization is gradually guided by the user's preference. After the optimization, the most preferable particle can be chosen for practical use by selecting the particle with the highest global evaluation value. The effectiveness of the proposed MOPSO-PS is demonstrated by the application of path following footstep optimization for humanoid robots in addition to empirical comparison with the other algorithms. The footsteps optimized by the MOPSO-PS were verified by simulation. The results indicate that the user's preference is properly reflected in optimized solutions without any loss of overall solution quality and diversity.**

***Index Terms*—Particle Swarm Optimization, Multiobjective Evolutionary Optimization, Fuzzy Integral, Preference-based sort, Humanoid Robot, Footstep Optimization.**

I. Introduction

Solving Multiobjective Optimization Problems (MOPs) has become important in engineering recently. For example, there are a lot of MOPs in robotics, such as footstep planning for humanoid robots, autonomous control for Unmanned Aerial Vehicles (UAVs), and path planning for UAVs [1]–[3]. In order to solve MOPs, various Multiobjective Evolutionary Algorithms (MOEAs) have been developed and have shown outstanding results through solving complex multiobjective benchmark functions. The Pareto Archived Evolutionary Strategy (PAES) was developed by using the adaptive grid [4]. The Strength Pareto Evolutionary Algorithm (SPEA) was created, which used a mixture of established and new techniques in order to approximate the Pareto-optimal set [5]. SPEA2, the improved version of SPEA, was developed by employing a refined fitness assignment and an enhanced archive truncation technique [6]. The Nondominated Sorting Genetic Algorithm (NSGA) was developed by the classification of nondominated fronts and the sharing operation [7]. The improved version of NSGA, NSGA-II, was created, which is a strong elitist method with a mechanism to maintain diversity efficiently by using a fast nondominated sort and crowding distance assignment [8]. The Multiobjective Quantum-inspired Evolutionary Algorithm (MQEA) was proposed to improve proximity to the Praeto-optimal set while preserving diversity [9], [10]. The Multiobjective Particle Swarm Optimization (MOPSO) was developed by extending the Particle Swarm Optimization (PSO), which is a population-based stochastic algorithm inspired by the interaction among the individuals of a swarm, such as a flock of birds and insects [11]–[20].

However, for real applications, which are different from the benchmark functions, two other issues need to be considered. The first one is how to handle the mutual dependencies and priorities of objectives. For example, in the problem of the humanoid robot path following footstep optimization, the remaining distance to the goal, the mean distance from the path's center line to the footsteps, the mean lateral movement, and the mean rotational movement can be employed as objectives. They are not always independent from each other and their priorities may vary with the preference of the user. The second one is how to determine the relative merits of nondominated solutions. A dominance-based approach is not effective in many-objective problems since the number of nondominated solutions increases exponentially with the number of objectives. Therefore, the relative merits of the nondominated solutions should be determined to update the population and archive at each generation. Moreover, for real applications, such as operating humanoid robots, it is necessary to choose the most preferable solution among the finally obtained solutions.

To solve the issues mentioned above, MOPSO with preference-based sort (MOPSO-PS) is proposed in this paper. According to the results from [18], MOPSO showed a highly competitive performance and was able to cover the full Pareto front of all the benchmark functions used. The relatively low computational burden of MOPSO is also an advantage for real applications. The preference-based sorting concept

Ki-Baek Lee and Jong-Hwan Kim are with the Department of Electrical Engineering, KAIST, Republic of Korea. (phone: +82-42-350-8048; fax: +82-42-350-8877; email: {kblee, johkim}@rit.kaist.ac.kr).

is based on a recent algorithm, Preference-based Solution Selection Algorithm (PSSA), in which the user's preference to each objective is represented by the degree of consideration using the fuzzy measure [21]. This successfully helps MOPSO determine the relative merits of the nondominated solutions in the consideration of the user's preference. In MOPSO-PS, every particle, i.e. every solution, is updated by following the personal best and the global best attractors. The global best attractor of each particle in the population is randomly chosen among the relatively more preferred particles in the archive at each generation. Therefore, as a result, the optimization is gradually guided by the user's preference.

To demonstrate the effectiveness of MOPSO-PS, empirical comparisons with NSGA-II [8], MQEA [9], [10], and MOPSO [18] were carried out through DTLZ functions, which are well known benchmark functions for multiobjective optimization algorithms [22]. The experiments of the path following footstep optimization for humanoid robots were also performed. For each of the three kinds of short paths, footsteps were optimized 50 times to check the robustness of the proposed algorithm. After that, footsteps were optimized for a long complex path that considers the real environment. The footsteps optimized to the path were verified by simulation.

This paper is organized as follows. Section II proposes MOPSO-PS and briefly introduces the fuzzy measure and fuzzy integral. In Section III, the comparison among MOPSO-PS and the other algorithms are discussed. Section IV presents the application to path following footstep optimization for humanoid robots. Finally, concluding remarks are in Section V.

II. Multiobjective Particle Swarm Optimization with Preference-based Sort

The key point of the Multiobjective Particle Swarm Optimization with Preference-based Sort (MOPSO-PS), compared to the original MOPSO, is that user's preference is taken into account. In MOPSO-PS, preference-based sort is additionally employed in order to incorporate the user's preference into the PSO update process. This is because the dominance-based approach is not effective in many-objective problems since the number of nondominated solutions increases exponentially with the number of objectives.

The most important thing for implementing the preference-based sort is that the Global Evaluation value (GEval) of every particle should be carried out, which represents the quality of the particle according to the user's preference. Therefore, in this paper, the user's preference is represented as the degree of consideration for each objective using the fuzzy measure and the GEval of a particle is calculated by the fuzzy integral, which integrates the Partial Evaluation value (PEval) of each objective with respect to the degree of consideration. The PEval of each objective can be obtained by normalizing the objective function value. For real applications, the most preferable solution can also be selected by the GEvals of the solution after finishing the optimization. Note that the fuzzy integral requires neither objectives to be independent nor the fuzzy measure to be additive for any subset in the power set of objectives because it can effectively represent the interactions, i.e. positive interactions and negative interactions between objectives. Thus the fuzzy integral is more suitable for determining the relative merits of the nondominated particles than the other existing methods, including multicriteria decision making and aggregation methods [23]–[26].

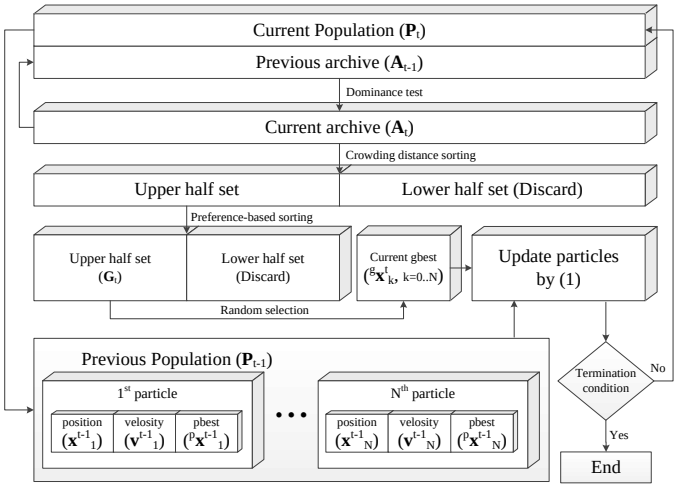


Fig. 1: The flow diagram of MOPSO-PS.

In this subsection, MOPSO-PS is explained step by step and the procedure of global evaluation is described in detail.

A. Proposed MOPSO-PS

In MOPSO-PS, the optimization is gradually guided by the user's preference. Each particle in the population chooses its Global Best attractor (GBest) randomly from the GBest candidate pool, which consists of relatively more preferable particles. Fig. 1 shows the flow diagram of MOPSO-PS, where $\mathbf{A}_t$ is the external archive, $\mathbf{P}_t$ is the population, $\mathbf{G}_t$ is the GBest candidate pool, ${}^p\mathbf{x}_t^k$ is the personal best attractor (PBest) position of the k-th particle, and ${}^g\mathbf{x}_t^k$ is the GBest position of the k-th particle at generation t. As shown in Fig .1, the GBest candidate pool is extracted from the archive by the following three steps. First, the archive is updated and every particle dominated by the others is discarded. Second, the particles in the archive are sorted by Crowding Distance (CD) and the lower half set of the archive is discarded. This is for preserving diversity and avoiding premature convergence. Third, the remaining particles of the archive, which have a relatively higher crowding distance, are sorted by the GEval of each particle and the lower half set of the archive is discarded again. As a result, dispersed and relatively more preferable particles are gathered in the GBest candidate pool.

The overall procedure of MOPSO-PS is summarized as Algorithm 1 and each step of the algorithm is described in detail in below:

1) Initialize $\mathbf{P}_0$ and $\mathbf{A}_0$

A population is a set of N particles, which have their own position and velocity. The position $\mathbf{x}^k$ and velocity $\mathbf{v}^k$ of the k-th particle p^k, $k = 1, 2, \ldots, N$, are the D-dimensional vectors

Algorithm 1 Multiobjective Particle Swarm Optimization with Preference-based Sort

- $\mathbf{P}_t$: The population at iteration t
- $\mathbf{A}_t$: The external archive at iteration t
- $\mathbf{G}_t$: GBest candidate pool at iteration t
- N_P: The number of particles in $\mathbf{P}_t$
- N_A: The number of particles in $\mathbf{A}_t$
- N_G: The number of particles in $\mathbf{G}_t$
- D: The dimension of the search space
- $rand(L, U)$: Random integer value between L and U
- $\mathbf{x}_t^k$: The position of the k-th particle at iteration t
- $\mathbf{v}_t^k$: The velocity of the k-th particle at iteration t
- $f(\mathbf{x}_t^k)$: The objective function value of $\mathbf{x}_t^k$
- ${}^g\mathbf{x}_t^k$: GBest position of the k-th particle at iteration t
- ${}^p\mathbf{x}_t^k$: PBest position of the k-th particle at iteration t

1) Initialize $\mathbf{P}_0$ and $\mathbf{A}_0$
   $t = 0$
   for $k = 1, 2, \ldots, N_P$ **do**
   $\mathbf{x}_t^k$ = random vector $\in \mathbb{R}^D$
   $\mathbf{v}_t^k = \mathbf{0}$
   Evaluate $f(\mathbf{x}_t^k)$
   ${}^p\mathbf{x}_t^k = \mathbf{x}_t^k$
   end for
   $\mathbf{A}_t$ = Null set
2) Update $\mathbf{A}_t$
   $t = t + 1$
   $\mathbf{A}_t = \mathbf{A}_{t-1} \cup \mathbf{P}_{t-1}$
   Discard dominated particles out of $\mathbf{A}_t$
   for $k = 1, 2, \ldots, N_A$ **do**
   Evaluate the PEval of each particle in $\mathbf{A}_t$
   Evaluate the GEval of each particle in $\mathbf{A}_t$
   Calculate the CD of each particle in $\mathbf{A}_t$
   end for
3) Extract $\mathbf{G}_t$ from $\mathbf{A}_t$
   Sort the particles in $\mathbf{A}_t$ based on their CDs
   $\mathbf{A}'_t$ = the upper half of $\mathbf{A}_t$
   Sort the particles in $\mathbf{A}'_t$ based on their GEvals
   $\mathbf{G}_t$ = the upper half of $\mathbf{A}'_t$
4) Update particles
   for $k = 1, 2, \ldots, N_P$ **do**
   $r = rand(0, N_G)$
   ${}^g\mathbf{x}_t^k$ = the position of the r-th particle in $\mathbf{G}_t$
   Update $\mathbf{v}_t^k$ and $\mathbf{x}_t^k$ by (1)
   Evaluate $f(\mathbf{x}_t^k)$
   Update ${}^p\mathbf{x}_t^k$
   end for
5) Go to 2) and Repeat

as follows:

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

For every particle in the population, the position is randomly initialized in a D-dimensional space and the velocity is initially set to $\mathbf{0}$. The M-objective function of each particle $f(\mathbf{x}_t^k)$ is evaluated, which is defined as

$$f(\mathbf{x}_t^k) : \mathbb{R}^D \to \mathbb{R}^M$$

and the PBest position of each particle is set to be the position of itself. The external archive is also initialized as a null set.

2) Update $\mathbf{A}_t$

The current archive $\mathbf{A}_t$ is updated as the union of the previous archive $\mathbf{A}_{t-1}$ and the previous population $\mathbf{P}_{t-1}$. After that, through a dominance test, the dominated particles are discarded out of $\mathbf{A}_t$. For each particle in $\mathbf{A}_t$, the PEval and GEval are evaluated, which are described in Section II.B, and CD is also calculated [27].

3) Extract $\mathbf{G}_t$ from $\mathbf{A}_t$

At first, the particles in $\mathbf{A}_t$ are sorted by their CDs and the upper half of $\mathbf{A}_t$, which is a relatively more dispersed set, is stored into $\mathbf{A}'_t$. Then, the particles in $\mathbf{A}'_t$ are sorted again based on their GEvals, and the upper half of $\mathbf{A}'_t$, which is a relatively more preferable set, is stored into $\mathbf{G}_t$. Since the particles in $\mathbf{G}_t$ are mutually nondominating and no particles in $\mathbf{G}_t$ are dominated by $\mathbf{x}_{t-1}^k$, every particle in $\mathbf{G}_t$ can be a candidate for the GBest of each particle in $\mathbf{P}_{t-1}$. Moreover, by choosing ${}^g\mathbf{x}_t^k$ from $\mathbf{G}_t$, the particles are guided by the user's preference while maintaining diversity.

4) Update particles

For each particle, ${}^g\mathbf{x}_t^k$ is randomly chosen from $\mathbf{G}_t$. The velocity and position of each particle are updated as follows:

$$\begin{cases} \mathbf{v}_t^k = & w \cdot \mathbf{v}_{t-1}^k + c \cdot \{\phi_{1,t}^k({}^p\mathbf{x}_{t-1}^k - \mathbf{x}_{t-1}^k) \\ & +\phi_{2,t}^k({}^g\mathbf{x}_t^k - \mathbf{x}_{t-1}^k)\} \\ \mathbf{x}_t^k = & \mathbf{x}_{t-1}^k + \mathbf{v}_t^k \end{cases} \quad (1)$$

where w and c are constants and $\phi_{1,t}^k$ and $\phi_{2,t}^k$ are random real values uniformly distributed in $[0, 1]$. New random values are generated for each particle at each and every generation. After that, $f(\mathbf{x}_t^k)$ for every particle in the updated $\mathbf{P}_{t-1}$ is evaluated. Finally, the personal best position of each particle ${}^p\mathbf{x}_t^k$, $k = 1, 2, \ldots, N$, is updated. ${}^p\mathbf{x}_t^k = \mathbf{x}_t^k$ if $\mathbf{x}_t^k$ weakly dominates ${}^p\mathbf{x}_{t-1}^k$ or they are mutually non-dominating. Otherwise, ${}^p\mathbf{x}_t^k = {}^p\mathbf{x}_{t-1}^k$.

5) Go to 2) and Repeat

Go to 2) and repeat until a termination condition is met.

Note that the computational complexity of the proposed algorithm is governed by the sorts, i.e. the CD-based sort and the GEval-based sort. Since both sorts are done by the quick sort, the proposed algorithm has an average computational complexity of O($n$log(n)).

B. The Procedure of Global Evaluation

To carry out the GEval of each particle, the user's preference is represented as the degree of consideration to each objective using the fuzzy measure and with which the PEval of each obective is integrated by the fuzzy integral. In this study, a method of diamond pairwise comparisons and transformation

Algorithm 2 The Procedure of Global Evaluation

- M: The number of objectives
- $O = \{o_1, o_2, \cdots, o_M\}$: A set of objectives
- $P(O)$: A power set of O
- N: The number of particles
- $f_i(\mathbf{x}_k)$: Objective function value of the k-th particle over o_i
- $h_i(\mathbf{x}_k)$: Partial evaluation value of the k-th particle over o_i
- $e(\mathbf{x}_k)$: Global evaluation value of the k-th particle

1) Fuzzy measure identification
  Make a pairwise comparison matrix, P
  for $i = 1, 2, \ldots, M$ **do**
    Calculate normalized weight w_i with P by (6)
  end for
  Generate interaction diagram
  Generate hierarchy diagram
  for each $A \in P(O)$ **do**
    Calculate fuzzy measure $g(A)$ of $P(O)$ by (9)-(12)
  end for
2) Global evaluation of particles
  for $i = 1, 2, \ldots, M$ **do**
    for $k = 1, 2, \ldots, N$ **do**
      Find the maximum value f_i^{MAX} and the minimum value f_i^{MIN}
    end for
  end for
  for $k = 1, 2, \ldots, N$ **do**
    for $i = 1, 2, \ldots, M$ **do**
      $h_i(\mathbf{x}_k) = \frac{f_i(\mathbf{x}_k) - f_i^{MIN}}{f_i^{MAX} - f_i^{MIN}}$
    end for
  end for
  for $k = 1, 2, \ldots, N$ **do**
    Calculate $e(\mathbf{x}_k) = \int_O h \circ g$ by (13)
  end for

was employed for fuzzy measure identification [28], [29] and Choquet fuzzy integral was employed as a fuzzy integral [30].

The overall procedure of global evaluation using the fuzzy measure and fuzzy integral is summarized in Algorithm 2 and each step is described in the following:

1) Fuzzy measure identification

In this study, λ-fuzzy measure was used to represent the degree of consideration for each objective. The fuzzy measure of the power set of X, denoted as $P(X)$ in the finite space $X = \{x_1, \cdots, x_M\}$ is defined as follows:

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

(1) Boundary condition

$$g(\emptyset) = 0, \;\; g(X) = 1. \tag{2}$$

(2) Monotonicity

$$\forall A, B \subseteq P(X), \;\; if \;\; A \subseteq B, \;\; then \;\; g(A) \leq g(B). \tag{3}$$

As a general representation of fuzzy measure, λ-fuzzy measure, $g : P(X) \rightarrow [0, 1]$, is defined, which additionally satisfies the following axiom [31]:

$$\forall A_{i,j} \in P(X), \; i, j = 1, \cdots, M, \; A_i \cap A_j = \emptyset \; and \; -1 < \lambda,$$
$$g(A_i \cup A_j) = g(A_i) + g(A_j) + \lambda g(A_i) g(A_j) \tag{4}$$

where λ represents an interaction degree between A_i and A_j. λ-fuzzy measure is considered as a 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 a belief measure, plausibility measure and probability measure [32]. The following procedure was employed to calculate the fuzzy measures [28], [29]:

a) Make a pairwise comparison matrix

The pairwise comparison matrix of objectives, P, which represents preference degrees between objectives, is defined as follows [26]:

$$\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1M} \\ p_{21} & p_{22} & \cdots & p_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MM} \end{bmatrix} \tag{5}$$

where p_{ij} represents the preference degree between the i-th objective o_i and the j-th objective o_j, p_{ii} is 1 and $p_{ij} = 1/p_{ji}$. The preference degrees of the pairwise matrix are determined by the relative importance between each pair of objectives from the user's perspective, e.g., if p_{12} is 10, it means o_1 is ten times more preferable to o_2.

b) Calculate normalized weight

The normalized weight, w_i of the i-th objective, o_i, $i, j = 1, \cdots, M$ is calculated as follows:

$$w_i = \frac{\sum_{j=1}^{M} p_{ij}}{\sum_{i=1}^{M} \sum_{j=1}^{M} p_{ij}}. \tag{6}$$

c) Generate interaction diagram

The interaction diagram of objectives (Fig. 3 in Section IV) shows the interaction degrees between two different objectives. If the two objectives have a negative correlation, the interaction degree between them has a value between 0.0 and 0.5. In contrast, if the two objectives have a positive correlation, the interaction degree between them has a value between 0.5 and 1.0. If the two objectives are independent, the interaction degree between them has a value of 0.5. Therefore, the interaction degree between the i-th and j-th objectives $\xi_{i,j}$ lies in $[0.0, 1.0]$.

d) Generate hierarchy diagram

As shown in the interaction diagram, since the interaction degrees are different from each other, it is hard to directly identify the fuzzy measures. In this study, a hierarchy diagram was generated to get a merged interaction degree. The hierarchy diagram of objectives (Fig. 3 in Section IV) represents hierarchical interaction relations among the objectives by clustering two closely-related objectives. To estimate how much

the two objectives are related, a dissimilarity measure between them was employed. The dissimilarity $D_{\{G_p,G_q\}}$ between two clusters G_p and G_q is defined as an average distance to other objectives, as follows:

$$D_{\{G_p,G_q\}} = \frac{\sum_{G_r\in\Phi, G_r\neq G_p, G_r\neq G_q}\left[\xi_{\{G_p,G_r\}} - \xi_{\{G_q,G_r\}}\right]^2}{|\Phi| - 2} \tag{7}$$

where Φ is a set of all clusters that consist of one or more objectives, $G_p, G_q, G_r \in \Phi$ are clusters, $\xi_{\{G_i,G_j\}}$ is the interaction degree between objectives G_i and G_j, and $|\Phi|$ is the number of clusters in Φ. Two objective clusters that have the smallest dissimilarity are merged into one and the interaction degrees among will clusters are recalculated. The interaction degree $\xi_{\{G_p,G_q\}}$ between two clusters G_p and G_q is calculated as

$$\xi_{\{G_p,G_q\}} = \frac{\sum_{(i,j)\in(E(G_p)\times E(G_q))}\xi_{ij}}{|E(G_p)\times E(G_q)|} \tag{8}$$

where $A\times B$ is the direct sum and $E(G_p)$ is the function that picks up all objectives in the set G_p. The merging procedure is done until all objectives are merged.

e) Calculate fuzzy measures

After getting the hierarchy diagram, a fuzzy measure $g(A)$, where $A \in P(O)$, was identified by employing ϕ_s transformation, as follows:

$$g(A) = \phi_s(\xi_R, \sum_{P\subset R} u_P^R), \tag{9}$$

where R is the root level set in the hierarchy diagram and ξ_R is the interaction degree of R. The transformation $\phi_s : [0,1]\times[0,1]\to[0,1]$ is defined as follows:

$$\phi_s(\xi,u) = \begin{cases} 1, & \text{if } \xi = 1 \text{ and } u > 0 \\ 0, & \text{if } \xi = 1 \text{ and } u = 0 \\ 1, & \text{if } \xi = 0 \text{ and } u = 1 \\ 0, & \text{if } \xi = 0 \text{ and } u < 1 \\ \frac{s^u-1}{s-1}, & \text{other cases} \end{cases} \tag{10}$$

where $s = (1-\xi)^2/\xi^2$ and u_Q^P is defined as follows:

$$u_Q^P = \begin{cases} w_i, \text{where } o_i \in Q, & \text{if } |Q| = 1 \text{ and } o_i \in A \\ 0, & \text{if } |Q| = 1 \text{ and } o_i \notin A \\ \phi_s^{-1}(\xi_P, \phi_s(\xi_Q, \sum_{V\subset Q} u_V^Q)\times T_Q^P), & \text{other cases} \end{cases} \tag{11}$$

where the value of $\phi_s^{-1}(\xi,x)$ is u, which satisfies $\phi_s(\xi,u) = x$. To calculate the value of $\phi_s^{-1}(\xi,x)$, the recursive least squares method was employed. The conversion ratio T_Q^P from Q to P is computed as

$$T_Q^P = \frac{\phi_s(\xi_P, \sum_{o_i\in P} w_i)}{\phi_s(\xi_Q, \sum_{o_i\in Q} w_i)}. \tag{12}$$

2) Global evaluation of particles

For the global evaluation of each particle over objectives with respect to the degree of consideration for each of the objectives, the following Choquet fuzzy integral [30] can be used.

***Definition** 2:* Let $h : O \to [0,1]$, where O can be any set. The Choquet fuzzy integral of partial evaluation, h over a subset of $O \in P(O)$, with respect to the fuzzy measure g, is defined as

$$\int_O h\circ g = \sum_{i=1}^{M}(h_i - h_{i-1})g(E_i) \tag{13}$$

where $h_1 \le h_2 \le \cdots \le h_M$, $E_i = \{o_i, o_{i+1}, \cdots, o_M\}$ and $h_0 = 0$, for $o_i \in O$ and $i = 1, \cdots, M$.

The value h in (13) is a normalized objective function value that represents the partial evaluation value of each particle over each objective. The objective function values need to be normalized to 1.0 since h is defined from 0.0 to 1.0. In this way, the $h_i(\mathbf{x}_k)$ of the k-th particle over o_i is calculated. g is the λ-fuzzy measure obtained from the fuzzy measure identification. It means the global evaluation value is calculated by considering both the user's degree of consideration for each objective and the partial evaluation of the particle.

III. Experiments for Comparison

A. Configuration for the Comparison

The parameters used in the comparison 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 the DTLZ7 function. For MOPSO-PS, three out of the seven objectives in DTLZ functions were chosen as preferred objectives. Thus, the preference degree for this was set to $f_1 : f_2 : f_3 : f_4 : f_5 : f_6 : f_7 = 1 : 10 : 1 : 10 : 1 : 10 : 1$ from which a pairwise comparison matrix (5) was provided. The normalized weights according to the pairwise comparison matrix were calculated as $(0.0295, 0.295, 0.0295, 0.295, 0.0295, 0.295, 0.0295)$. Since it is hard to figure out the exact correlation degrees between the objectives in DTLZ functions, it was assumed that they have negative correlations with each other and every interaction degree between them was set to 0.25. Therefore, the interaction diagram and hierarchy diagram generation process could be skipped and the fuzzy measure $g(A)$ was identified as follows:

$$g(A) = \phi_s(\xi, \sum_{o_i\in A} w_i) \tag{14}$$

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

In addition to average objective function values, two performance metrics, the size of dominated space and diversity, were employed to evaluate the performance of NSGA-II, MQEA, MOPSO and MOPSO-PS. The size of dominated space, $\mathcal{S}$ is defined by the hypervolume of nondominated solutions [5]. The reference point to calculate $\mathcal{S}$ was set to (10, 10, 10, 10, 10, 10, 10). The quality of the obtained solution set is high if this space is large. Diversity, $\mathcal{D}$ is for evaluating the spread of nondominated solutions, which is defined as follows [33]:

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

where N_0 is the set of nondominated solutions, d_i is the minimal distance between the i-th solution and the nearest

TABLE I: The parameter settings of the algorithms

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	$0.5 + \log 2$

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 values of the k-th objective, respectively. A larger value means a better diversity of the nondominated solutions.

B. Comparison Results

Since f_2, f_4, and f_6 were more considered in every generation of the evolutionary process, MOPSO-PS could obtain the optimized solutions that were more focused on those preferred objectives. Table II shows the average objective function values over 100 runs. As the table shows, the average values of f_2, f_4, and f_6 of MOPSO-PS are the smallest among all the algorithms, except for the DTLZ2 function. To see how the preference degree contributes to the final solutions, the MOPSO-PS with different preference degree settings was also studied for the DTLZ functions. Table III shows the average of the preferred objective function values of the three DTLZ functions, DTLZ2, DTLZ4 and DTLZ6, over 100 runs for each setting. As shown in the table, the preference degree had a pronounced effect on the final solutions. For the rest of the DTLZ functions, the result was similar. However, the effect was not highly sensitive to the magnitude of the preference degree.

The size of dominated space $\mathcal{S}$ and diversity $\mathcal{D}$ of NSGA-II, MQEA, MOPSO and MOPSO-PS are shown respectively in Table IV and Table V, where $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ and $\mathcal{S}_4$ represent the $\mathcal{S}$ of NSGA-II, MQEA, MOPSO and MOPSO-PS, and $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$ represent the $\mathcal{D}$ of NSGA-II, MQEA, MOPSO and MOPSO-PS, respectively. The values were averaged over 100 runs and Welch's t-test values [34] 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 tables show, both the $\mathcal{S}$ 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, even though the preferred objectives were considered more. When the conventional utility function method, like the weighted sum method, is used in selection process, the weights need to be set very carefully in order to obtain the solutions optimized, not only for preferred objectives, but also for the other objectives to a certain level. On the other hand, MOPSO-PS can solve this problem by employing the fuzzy measure representing the

TABLE II: Average of the preferred objective function values obtained by 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	0.16	**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**

TABLE III: Average of the preferred objective function values of the three DTLZ functions, DTLZ2, DTLZ4 and DTLZ6, obtained by MOPSO-PS with different preference degree settings

(a) DTLZ2

$f_1 : f_2 : f_3 : f_4 : f_5 : f_6 : f_7$	f_2	f_4	f_6
$1:1:1:1:1:1:1$	0.1069	0.2455	0.5735
$1:5:1:5:1:5:1$	0.0921	0.1577	0.1947
$1:10:1:10:1:10:1$	0.0959	0.1722	0.1785
$1:20:1:20:1:20:1$	0.0766	0.1273	0.1654
$1:2:1:5:1:10:1$	0.1047	0.1371	0.1419

(b) DTLZ4

$f_1 : f_2 : f_3 : f_4 : f_5 : f_6 : f_7$	f_2	f_4	f_6
$1:1:1:1:1:1:1$	0.0850	0.1164	0.1920
$1:5:1:5:1:5:1$	0.0058	0.0070	0.0023
$1:10:1:10:1:10:1$	0.0020	0.0009	0.0005
$1:20:1:20:1:20:1$	0.0001	0.0001	0.0003
$1:2:1:5:1:10:1$	0.0140	0.0023	0.0020

(c) DTLZ6

$f_1 : f_2 : f_3 : f_4 : f_5 : f_6 : f_7$	f_2	f_4	f_6
$1:1:1:1:1:1:1$	0.2036	0.4808	0.9023
$1:5:1:5:1:5:1$	0.0630	0.1356	0.2732
$1:10:1:10:1:10:1$	0.0411	0.0886	0.1751
$1:20:1:20:1:20:1$	0.0383	0.0767	0.1533
$1:2:1:5:1:10:1$	0.0500	0.1084	0.2187

TABLE IV: The average and Welch's t-test value of the size of the dominated space of NSGA-II, MQEA, MOPSO, and MOPSO-PS through DTLZ functions over 100 runs

(a) Average size of the dominated space

Problem	(NSGA-II)	(MQEA)	(MOPSO)	(MOPSO-PS)
DTLZ1	5061372	9974885	9998271	9996018
DTLZ2	9999093	9901558	9891993	9979433
DTLZ3	8884523	9458221	9845668	9941199
DTLZ4	9999878	9948522	9897862	9990820
DTLZ5	9876226	9851258	9750803	9848487
DTLZ6	7564857	9752218	9863796	9808835
DTLZ7	4234421	2522185	2341579	1903191

(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

TABLE V: The average and Welch's t-test value of the diversity of NSGA-II, MQEA, MOPSO, and MOPSO-PS through DTLZ functions over 100 runs

(a) 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

(b) 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

interactions between the objectives and the user's preference to them.

IV. Application to Path Following Footstep Optimization

A. Configuration for the Experiments

The main goal of this application was to make a humanoid robot follow a predefined path with the footsteps obtained by the MOPSO-PS by considering the objectives and the user's preference for them. The problem was formulated as a Multiobjective Optimization Problem (MOP) by defining the objective functions and setting the footsteps as variable vectors. To solve this kind of problem, the MOPSO-PS is more suitable than the other algorithms because the objectives are not always independent from each other and the user's preference for the objectives should be considered. Moreover, at the end of the evolutionary process, MOPSO-PS provides the humanoid robot with one preferred solution among the nondominated solutions, which is to be used for its footsteps.

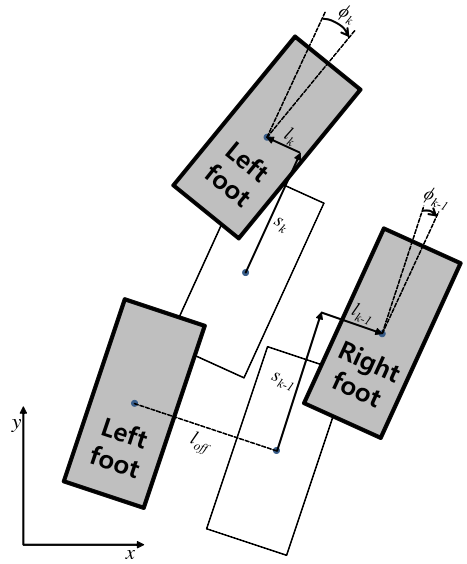


Fig. 2: An example of footsteps and related notations.

It was assumed that a humanoid robot always started walking from the right footstep. It was also set that K footsteps ($\frac{K}{2}$ left footsteps and $\frac{K}{2}$ right footsteps) was optimized at a time. The simulation model was based on a small-sized humanoid robot, HanSaRam-IX (HSR-IX), which subsequently verified the optimized footsteps by simulation. The HSR-IX was developed by the Robot Intelligent Technology (RIT) laboratory at the Korea Advanced Institute of Science and Technology (KAIST) [35]. Its height and weight are 52.8 cm and 5.5 kg, respectively. It has 26 degrees of freedom, which consists of 12 dc motors with harmonic drives in the lower body and 16 RC servomotors in the upper body. The on-board Pentium-III compatible PC, running RT-Linux, calculates the proposed algorithm every 5 ms in real time. To measure the ground reaction forces on the feet and the real ZMP trajectory while walking, four force sensing resisters were equipped on the sole of each foot. It can stride a maximum 90.0 mm in a sagittal direction and 20.0 mm in a lateral direction at once. The possible rotation angle of the leg is at maximum ± 0.15 rad. The offset length between the centers of both feet, l_{off}, is 78.0 mm.

Each footstep was represented by a corresponding vector defined as follows:

$$(l_k, s_k, \phi_k) \tag{16}$$

where $k = 1, \ldots, K$ and l_k, s_k, and ϕ_k are the lateral, sagittal, and rotational movement of the k-th footstep from the $(k-1)$-th footstep, respectively. Fig. 2 illustrates an example of footsteps.

The position and orientation vector of the k-th footstep were

calculated as follows:

$$\begin{cases} {}^{f}x_k = {}^{f}x_{k-1} + (l_{\text{off}} + l_k) \cdot cos({}^{f}\theta_{k-1} + (-1)^k \cdot \frac{\pi}{2}) \\ \quad\quad +s_k \cdot sin({}^{f}\theta_{k-1} + (-1)^k \cdot \frac{\pi}{2}) \\ {}^{f}y_k = {}^{f}y_{k-1} + (l_{\text{off}} + l_k) \cdot cos({}^{f}\theta_{k-1}) \\ \quad\quad +s_k \cdot sin({}^{f}\theta_{k-1}) \\ {}^{f}\theta_k = {}^{f}\theta_{k-1} + \phi_k \end{cases} \tag{17}$$

where ${}^{f}x_k$, ${}^{f}y_k$, and ${}^{f}\theta_k$ are the positions along the x-axis and y-axis and the orientation of the k-th footstep, respectively. ${}^{f}x_0$, ${}^{f}y_0$, and ${}^{f}\theta_0$ were set to be 0.0, 0.0, and $\frac{\pi}{2}$, respectively. Since a footstep was represented by a three dimensional vector, it was necessary to optimize $3 \times K$ variables for K footsteps.

A path can be represented by a lane which consists of vectors, each of which, $\mathbf{p}_n, n = 1, \ldots, N$, can be defined as follows:

$$\mathbf{p}_n = ({}^{p}x_n, {}^{p}y_n) \tag{18}$$

where ${}^{p}x_n$ and ${}^{p}y_n$ are the positions of the n-th point of the lane along the x-axis and y-axis, respectively. In experiments, N was set as the value which makes a distance of 1.0 mm between the two adjacent points. The k-th footstep was regarded as being inside of the path if $min_n[\sqrt{({}^{f}x_k - {}^{p}x_n)^2 + ({}^{f}y_k - {}^{p}y_n)^2}] < 60.0$ mm. Otherwise, it was regarded as being outside of the path.

Four objectives (minimum remaining distance to goal (o_1), minimum mean distance from the path's center line (o_2), minimum mean lateral movement (o_3), and minimum mean rotational movement (o_4)) were employed in experiments and their corresponding objective functions to be minimized were defined as follows:

$$f_1 = \sum_{n=i}^{N-1} |\mathbf{p}_{n+1} - \mathbf{p}_n| \tag{19}$$

$$f_2 = \frac{1}{K} \sum_{k=1}^{K} \min_n [\sqrt{({}^{f}x_k - {}^{p}x_n)^2 + ({}^{f}y_k - {}^{p}y_n)^2}] \tag{20}$$

$$f_3 = \frac{1}{K} \sum_{k=1}^{K} |l_k| \tag{21}$$

$$f_4 = \frac{1}{K} \sum_{k=1}^{K} |\phi_k| \tag{22}$$

where i is the index of the nearest point on the path from the last footstep. Note that f_1, f_2, f_3, and f_4 are related with travelling distance per step, safety, stability of walking, and energy consumption, respectively.

In experiments, the same parameters of the MOPSO-PS used in the performance comparison were used. K was set to 10 and the number of variables of this problem was 30 (3×10 footsteps). For each of the three different kinds of short paths, footsteps were optimized 50 times to check the robustness of the proposed algorithm. After that, footsteps were optimized for a long complex path that assumes a real environment. Three cases of preference degrees were considered for each path to generate footsteps.

The preference degree, the pairwise comparison matrix, and the normalized weights corresponding to each case is shown in Table VI and the interaction diagram of objectives is shown in Fig. 3. The number on the line connecting the two objectives is the interaction degree (ξ) between them. Every correlation between the objectives can be determined intuitively. For example, the closer a robot follows the path's center line, the longer the distance to the goal remains. Since every correlation was assumed to be negative or independent, every interaction degree was set in $[0.0\ 0.5]$.

TABLE VI: Preference degree, pairwise comparison matrix, and normalized weights

Case	Preference degree ($o_1 : o_2 : o_3 : o_4$)	Pairwise comparison matrix	Normalized weights
C_1	$10 : 1 : 1 : 5$	$\begin{bmatrix} 1 & 10 & 10 & 2 \\ 0.1 & 1 & 1 & 0.2 \\ 0.1 & 1 & 1 & 0.2 \\ 0.5 & 5 & 5 & 1 \end{bmatrix}$	$w_1 = 0.588$ $w_2 = 0.059$ $w_3 = 0.059$ $w_4 = 0.294$
C_2	$5 : 10 : 1 : 1$	$\begin{bmatrix} 1 & 0.5 & 5 & 5 \\ 2 & 1 & 10 & 10 \\ 0.2 & 0.1 & 1 & 1 \\ 0.2 & 0.1 & 1 & 1 \end{bmatrix}$	$w_1 = 0.294$ $w_2 = 0.588$ $w_3 = 0.059$ $w_4 = 0.059$
C_3	$5 : 1 : 10 : 1$	$\begin{bmatrix} 1 & 5 & 0.5 & 5 \\ 0.2 & 1 & 0.1 & 1 \\ 2 & 10 & 1 & 10 \\ 0.2 & 1 & 0.1 & 1 \end{bmatrix}$	$w_1 = 0.294$ $w_2 = 0.059$ $w_3 = 0.588$ $w_4 = 0.059$

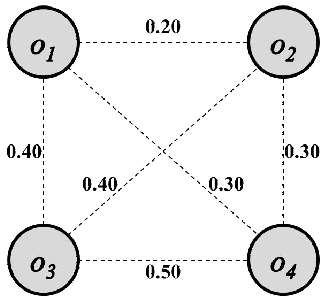


Fig. 3: Interaction diagram of objectives.

To merge two highly related objectives into one, the dissimilarity between them was accounted. For example, the dissimilarity between o_1 and o_2 was calculated as

$$D_{\{o_1,o_2\}} = \frac{(\xi_{23} - \xi_{13})^2 + (\xi_{24} - \xi_{14})^2}{|\Phi| - 2} = \frac{0.00}{2} = 0.00.$$

After calculating the dissimilarity, o_1 and o_2 were merged into one since they had the smallest dissimilarity. Another pair of objectives, o_3 and o_4, were also merged, and the interaction degree between the two pairs, $\{o_1, o_2\}$ and $\{o_3, o_4\}$, was computed as

$$\xi_{\{\{o_1,o_2\},\{o_3,o_4\}\}} = \frac{\xi_{13} + \xi_{14} + \xi_{23} + \xi_{24}}{4} = 0.35.$$

Fig. 4 shows the simplified hierarchy diagram.

The fuzzy measures of the objective sets were computed by using (9), (10), (11), (12), and the simplified hierarchy diagram. As an example, the fuzzy measure $g(A)$ of $A = \{o_1\}$

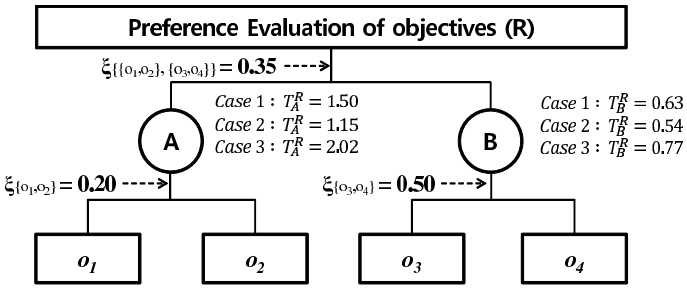


Fig. 4: Simplified hierarchy diagram of objectives.

TABLE VII: Identified fuzzy measures

A	g(A) Case 1	Case 2	Case 3
{1}	0.411	0.097	0.170
{2}	0.018	0.315	0.024
{3}	0.037	0.032	0.456
{4}	0.186	0.032	0.046
{1,2}	0.501	0.809	0.224
{1,3}	0.486	0.136	0.815
{1,4}	0.785	0.136	0.234
{2,3}	0.057	0.372	0.506
{2,4}	0.212	0.372	0.072
{3,4}	0.224	0.064	0.501
{1,2,3}	0.584	0.904	0.929
{1,2,4}	0.917	0.904	0.294
{1,3,4}	0.860	0.176	0.880
{2,3,4}	0.251	0.429	0.555
{1,2,3,4}	1.000	1.000	1.000

for Case 1 was calculated as

$$\begin{aligned} g(A) &= \phi_s(\xi_R, u_A^R + u_B^R) \\ &= \phi_s(0.35, u_A^R) \\ &= \phi_s(0.35, \phi_s^{-1}(0.35, \phi_s(0.2, 0.588) \times T_A^R)) \\ &= \phi_s(0.35, \phi_s^{-1}(0.35, 0.274 \times 1.50)) \\ &= \phi_s(0.35, 0.563) \\ &= 0.411. \end{aligned}$$

Table VII shows the finally identified fuzzy measures. In the table, objectives are abbreviated to their indices for the simplicity of notation. These values were used for the global evaluation.

The optimized footsteps were verified by simulation. The HSR-IX can modify its walking period, step length, and walking direction independently in realtime by the Modifiable Walking Pattern Generator (MWPG) [36]. Therefore, it can walk with the optimized footsteps by making a command state list of MWPG that corresponds to the footsteps. The simulation was carried out using 3-D robotics simulation software Webots [37] on which the model of the HSR-IX was programmed.

B. Experimental Results

Table VIII presents the average and Standard Deviation (STDEV) of the partial evaluation values for each case, where the partial evaluation values are the normalized objective function values between 0.0 and 1.0. Since o_1 for Case 1, o_2 for Case 2, and o_3 for Case 3 were set to have a relatively high preference degree, the average of f_1 for Case 1, f_2 for Case 2, and f_3 for Case 3 were the highest for all given paths. Extremely small STDEV values indicate that the MOPSO-PS robustly optimized the footsteps, regardless of path and case. In fact, there was no failure at all.

TABLE VIII: Average and standard deviation (STDEV) of partial evaluation values

			f_1	f_2	f_3	f_4
Path 1	Case 1	Average	**0.938**	0.793	0.627	0.673
		STDEV	0.003	0.053	0.050	0.041
	Case 2	Average	0.880	**0.900**	0.711	0.649
		STDEV	0.087	0.018	0.049	0.049
	Case 3	Average	0.920	0.615	**0.988**	0.558
		STDEV	0.024	0.035	0.007	0.030
Path 2	Case 1	Average	**0.977**	0.433	0.557	0.500
		STDEV	0.005	0.042	0.057	0.021
	Case 2	Average	0.835	**0.903**	0.786	0.402
		STDEV	0.020	0.011	0.042	0.026
	Case 3	Average	0.912	0.440	**0.987**	0.337
		STDEV	0.016	0.042	0.004	0.044
Path 3	Case 1	Average	**0.972**	0.456	0.642	0.531
		STDEV	0.007	0.073	0.124	0.060
	Case 2	Average	0.844	**0.924**	0.647	0.378
		STDEV	0.029	0.020	0.061	0.066
	Case 3	Average	0.952	0.332	**0.992**	0.282
		STDEV	0.014	0.036	0.003	0.054

Figs. 5, 6, and 7 show the optimized footsteps of the median performance for each of the three short paths and three cases. The footsteps optimized for Case 1 traveled the furthest since f_1 was considered more than the other objectives in Case 1. Similarly, the footsteps optimized for Case 2 were the closest to the path's center line because f_2 was considered more than the other objectives. Also, the footsteps optimized for Case 3 showed a rotational movement instead of lateral movement when the robot needed to move in a lateral direction because f_3 was considered more than the other objectives. Note that the there were no swaying movements in every case since f_4 was also sufficiently considered.

Fig. 8 shows the footsteps optimized for a long path. Similarly to the short paths, the consideration in each case was successfully reflected to the footsteps. The robot in Fig. 8(a) traveled further with the same number of footsteps than the others. The robot in Fig. 8(b) followed the path's center line as close as possible. The robot in Fig. 8(c) reached the goal with minimum lateral movement. Video (1) demonstrates the applicability of the proposed MOPSO-PS by Webots simulation. In the simulation environment, HSR-IX successfully walked along the path according to the preference degree in each case.

V. Conclusion

In this paper, Multiobjective Particle Swarm Optimization with Preference-based Sort (MOPSO-PS) was proposed. It could solve the Multiobjective Optimization Problems (MOPs) in consideration of the user's preference by applying preference-based sort, for which the fuzzy measure and fuzzy integral were employed. The effectiveness of this algorithm was demonstrated by comparison with NSGA-II, MQEA, and MOPSO through the DTLZ functions. The comparison results

indicated that the user's preference was properly reflected to optimized solutions by MOPSO-PS without any loss of overall solution quality and diversity. Moreover, MOPSO-PS avoided premature convergence even though the preferred objectives were considered more. The MOPSO-PS was also applied to the path following footstep optimization for humanoid robots and the footsteps optimized for predefined paths were successfully obtained. The obtained footsteps were verified by the simulation based on the model of humanoid robot HanSaRam-IX (HSR-IX). Through these results, it is certain that the MOPSO-PS can be applied to various kinds of real world applications.

ACKNOWLEDGEMENT

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0000150).

REFERENCES

[1] Y.-D. Hong, Y.-H. Kim, J.-H. Han, J.-K. Yoo, and J.-H. Kim, "Evolutionary multiobjective footstep planning for humanoid robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, no. 99, pp. 1–13, 2011.

[2] G. Barlow, C. Oh, and E. Grant, "Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming," in *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems*, vol. 2, 2004, pp. 689–694.

[3] S. Mittal and K. Deb, "Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2007, pp. 3195–3202.

[4] J. Knowles and D. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.

[5] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," *Doctoral dissertation ETH 13398, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland*, 1999.

[6] E. Zitzler, M. Laumanns, L. Thiele, *et al.*, "SPEA2: Improving the strength pareto evolutionary algorithm," in *EUROGEN*, vol. 3242, no. 103, 2001, pp. 1–21.

[7] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.

[8] 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.

[9] Y.-H. Kim, J.-H. Kim, and K.-H. Han, "Quantum-inspired multiobjective evolutionary algorithm for multiobjective 0/1 knapsack problems," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2006, pp. 2601–2606.

[10] Y.-H. Kim and J.-H. Kim, "Multiobjective quantum-inspired evolutionary algorithm for fuzzy path planning of mobile robot," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2009, pp. 1185–1192.

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

[12] K.-B. Lee and J.-H. Kim, "Mass-spring-damper motion dynamics-based particle swarm optimization," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2008, pp. 2348–2353.

[13] H.-M. Park and J.-H. Kim, "Potential and dynamics based particle swarm algorithm," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2008, pp. 2354–2359.

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

[15] C. Coello and M. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1051–1056, 2002.

[16] X. Hu and R. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 1677–1681, 2002.

[17] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proceedings of IEEE Swarm Intelligence Symposium*, 2003, pp. 26–33.

[18] 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.

[19] J. Alvarez-Benitez, R. Everson, and J. Fieldsend, "A MOPSO algorithm based exclusively on pareto dominance concepts," in *Evolutionary Multi-Criterion Optimization*, 2005, pp. 459–473.

[20] 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.

[21] 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.

[22] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1, 2002, pp. 825–830.

[23] R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[24] K. Miettinen, *Nonlinear multiobjective optimization*. Springer, 1999, vol. 12.

[25] M. Emmerich, N. Beume, and B. Naujoks, "An emo algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization*, 2005, pp. 62–76.

[26] T. L. Saaty, "Decision making with the analytic hierarchy process," *International Journal of Services Sciences*, vol. 1, no. 1, pp. 83–98, 2008.

[27] 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*, 2005, pp. 257–264.

[28] 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.

[29] ——, "A fuzzy measure identification method by diamond pairwise comparisons and transformation," *Fuzzy Optimization and Decision Making*, vol. 7, no. 3, pp. 219–232, 2008.

[30] M. Sugeno, "Fuzzy measures and fuzzy integrals: a survey," *Fuzzy automata and decision processes*, vol. 78, no. 33, pp. 89–102, 1977.

[31] ——, "Theory of fuzzy integrals and its applications," *Doctoral Thesis, Tokyo Institute of Technology Tokyo, Japan*, 1974.

[32] 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.

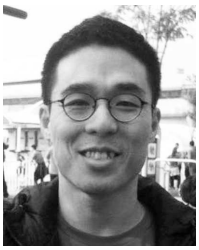
[33] H. Li, Q. Zhang, E. Tsang, and J. Ford, "Hybrid estimation of distribution algorithm for multiobjective knapsack problem," *Evolutionary Computation in Combinatorial Optimization*, pp. 145–154, 2004.

[34] 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.

[35] J.-K. Yoo, B.-J. Lee, and J.-H. Kim, "Recent progress and development of the humanoid robot HanSaRam," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 973–981, 2009.

[36] B.-J. Lee, D. Stonier, Y.-D. Kim, and J.-H. Kim, "Modifiable walking pattern of a humanoid robot by using allowable ZMP variation," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 917–925, 2008.

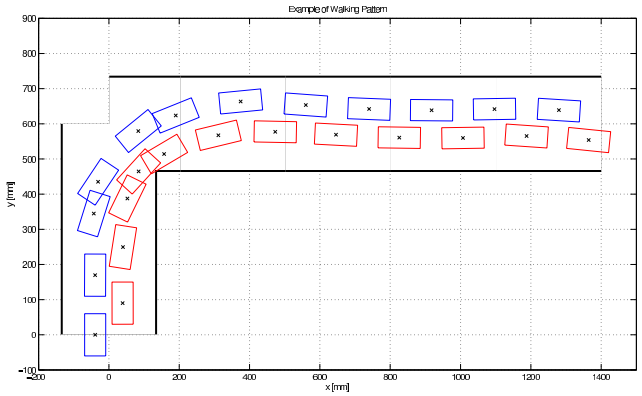
[37] O. Michel, "Cyberbotics Ltd. WebotsTM: Professional mobile robot simulation," *Int. J. Advanced Robot. Syst.*, vol. 1, no. 1, pp. 39–42, 2004.



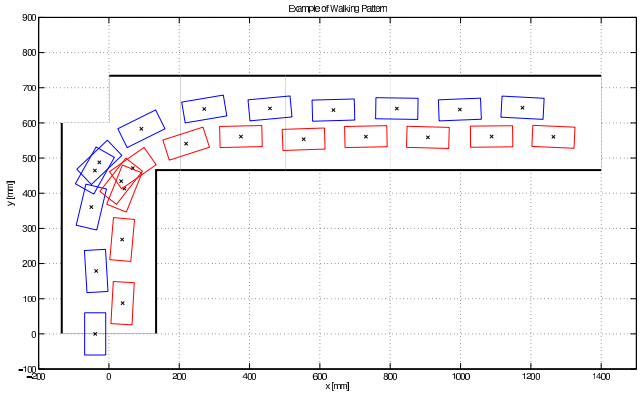
Ki-Baek Lee received the B.S. degree in Electrical Engineering, in 2005, from KAIST, Daejeon, Korea, where he is currently working toward the M.S.-Ph.D. joint degree.

He has researched computational intelligence, in particular, in the area of swarm intelligence, multiobjective particle swarm optimization and multiobjective quantum-inspired evolutionary algorithm (MQEA). His research interests also include the real world applications of MOEA such as on-line MOEA through distributed computing and on-line multiobjective evolutionary navigation for humanoid robots.

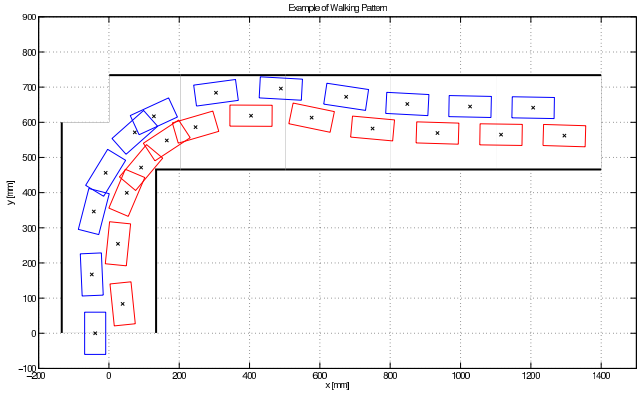
Jong-Hwan Kim (F'09) received the B.S., M.S., and Ph.D. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981, 1983, and 1987, respectively. Since 1988, he has been with the Department of Electrical Engineering, KAIST, Daejeon, Korea, where he is currently KT Chair Professor and Director for the National Robotics Research Center for Robot Intelligence Technology. His research interests include intelligence technology and ubiquitous and genetic robotics. He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE. His name was included in the Barons 500 Leaders for the New Century in 2000 as the Father of Robot Football. He is the Founder of the Federation of International Robosoccer Association (FIRA, www.FIRA.net) and the International Robot Olympiad Committee (IROC, www.IROC.org), and is currently the President of both.



(a)

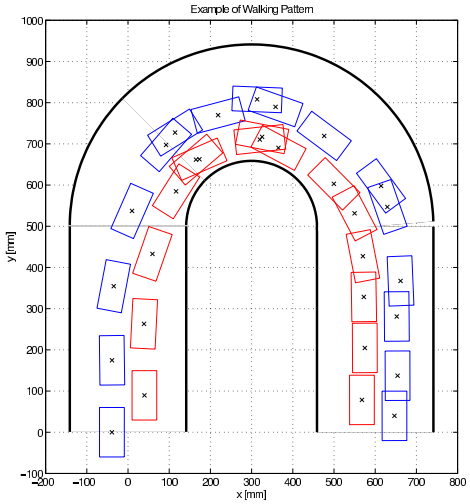


(b)

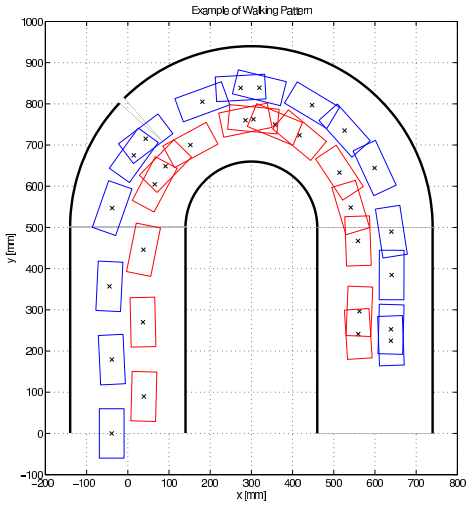


(c)

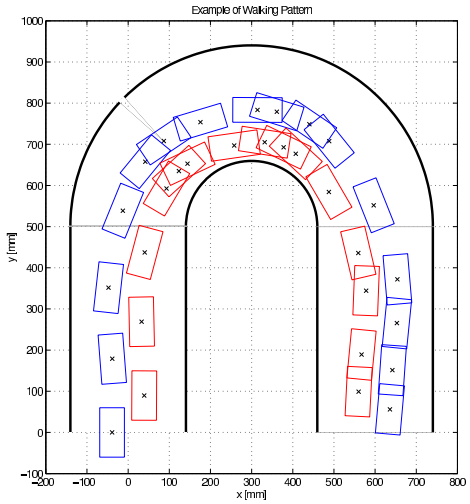
Fig. 5: 24 footsteps optimized for Path 1 ($\times$: foot position). (a) Case 1. (b) Case 2. (c) Case 3.



(a)

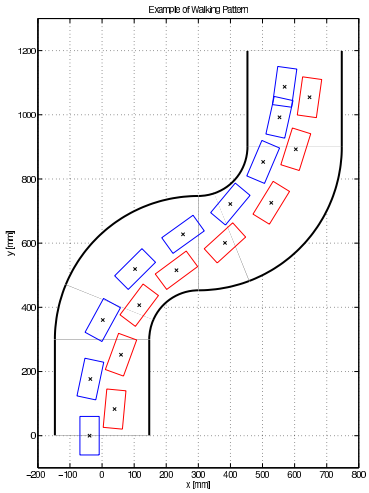


(b)

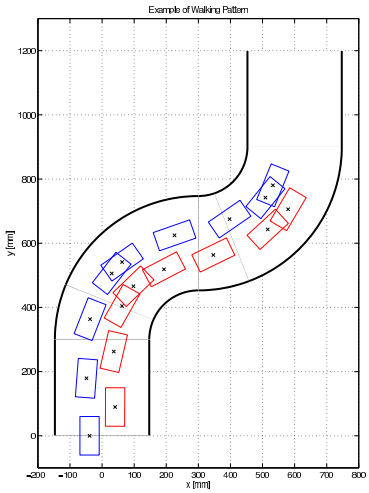


(c)

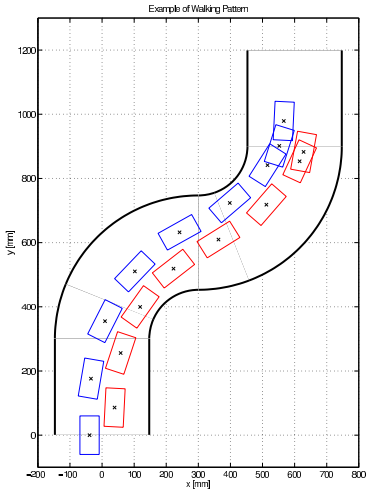
Fig. 6: 31 footsteps optimized for Path 2 (×: foot position). (a) Case 1. (b) Case 2. (c) Case 3.



(a)

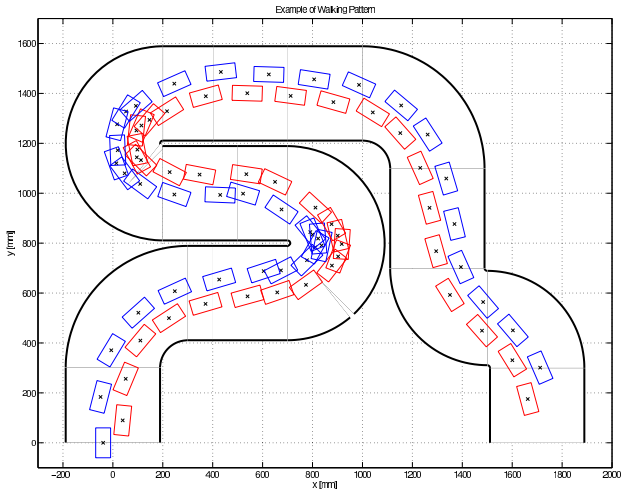


(b)

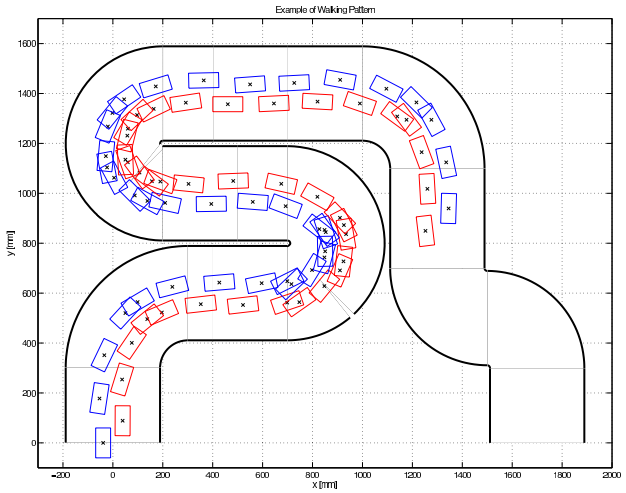


(c)

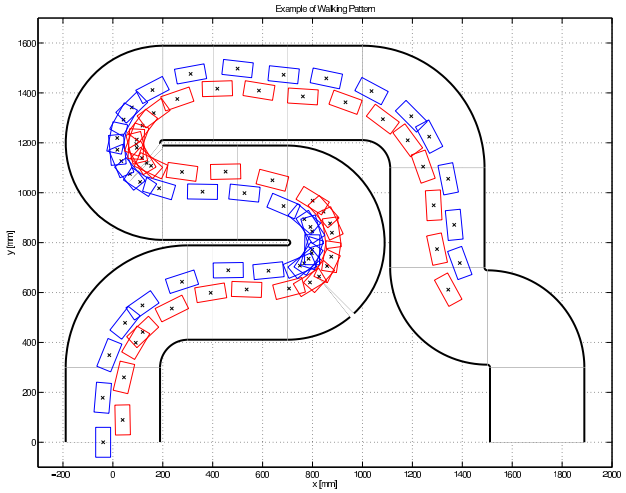
Fig. 7: 17 footsteps optimized for Path 3 (×: foot position). (a) Case 1. (b) Case 2. (c) Case 3.



(a)



(b)



(c)

Fig. 8: 76 footsteps optimized for a long path (×: foot position). (a) Case 1. (b) Case 2. (c) Case 3.