

Non-dominated sorting Harris's hawk multi-objective optimizer based on reference point approach

Shaymah Akram Yasear, Ku Ruhana Ku-Mahamud

School of Computing, Universiti Utara Malaysia, Malaysia

Article Info

Article history:

Received Oct 1, 2018

Revised Dec 10, 2018

Accepted Jan 25, 2019

Keywords:

Global optimization

Metaheuristic

Optimization algorithm

Swarm-intelligence

ABSTRACT

A non-dominated sorting Harris's hawk multi-objective optimizer (NDSHHMO) algorithm is presented in this paper. The algorithm is able to improve the population diversity, convergence of non-dominated solutions toward the Pareto front, and prevent the population from trapping into local optimal. This was achieved by integrating fast non-dominated sorting with the original Harris's hawk multi-objective optimizer (HHMO). Non-dominated sorting divides the objective space into levels based on fitness values and then selects non-dominated solutions to produce the next generation of hawks. A set of well-known multi-objective optimization problems has been used to evaluate the performance of the proposed NDSHHMO algorithm. The results of the NDSHHMO algorithm were verified against the results of an HHMO algorithm. Experimental results demonstrate the efficiency of the proposed NDSHHMO algorithm in terms of enhancing the ability of convergence toward the Pareto front and significantly improve the search ability of the HHMO.

*Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Shaymah Akram Yasear,

School of Computing,

Universiti Utara Malaysia.

Email: shayma.akram.yasear@gmail.com

1. INTRODUCTION

Several real-world optimization problems take into consideration multiple conflicting goals (objectives) simultaneously. This type of problem is known as a multiobjective optimization problem (MOP). Multi-objective optimization has been applied in several areas, such as, manufacturing process optimization, engineering design, chemical engineering [1, 2] and cloud computing [3]. In general, solving an MOP is much more difficult than a single objective optimization problem (SOP) because an optimal solution of the SOP can, generally, be clearly defined. On the other hand, in the MOP, there is usually no single solution that is optimal for all objectives. In other words, improving the value on one objective can make the value on other objectives worse because, there is a trade-off between different objectives. Therefore, it is not possible to simply define the optimal solution for an MOP. Instead, there is a set of solutions, not dominated by any other feasible solution, with different trade-offs, known as a Pareto optimal solution set or Pareto boundary (Pareto frontier) of a MOP [4].

MOPs using conventional single-objective optimization methods is not efficient. Therefore, the search for new optimization methods that can overcome the great challenge of this type of problem has become necessary. One way to deal with this challenge is to convert MOPs into SOPs using an aggregated objective function, which requires finding solutions with priorities, or weights, associated with the objectives and utilizing a single objective optimization algorithm to solve a problem. However, for a complex MOP with a large number of objectives and high-dimension, it is difficult to find an optimal solution using traditional methods. Therefore, the interest moves toward using metaheuristics. With the development of swarm intelligence (SI) theory [5], several multi-objective SI-based metaheuristics have been proposed. Most

of these algorithms were deduced by various existing single-objective algorithms such as multi-objective ant colony optimization (MOACO) [6], multi-objective particle swarm optimization (MOPSO) [7], multi-objective firefly algorithm (MOFA) [8], multi-objective artificial bee colony (MOABC) [9], multi-objective grey wolf optimizer (MOGWO) [10].

One of the most successful SI-based algorithms is the grey wolf optimizer (GWO) algorithm [11], which has been widely used in solving complex optimization problems [12-15] because of its simple concept, few parameters to be adjusted, easy implementation and strong global search ability. In [10], the MOGWO algorithm shows a competitive performance in solving different MOPs as compared to other well-known algorithms such as MOPSO [7] and multi-objective evolutionary algorithms based on decomposition (MOEA/D) [16]. The MOGWO algorithm, similar to most SI-based multi-objective optimization algorithms, tries to approximate the whole Pareto front and returns a set of non-dominated solutions which are evenly distributed across the whole Pareto front as shown in Figure 1(a). However, in real situations, different regions of the Pareto front could be more preferred than others and some regions could be not at all interesting. Therefore, the main drawbacks of this approach are time is wasted in exploring undesired solutions and difficulties for the decision-maker (DM) in determining the most preferred solution among a large number of solutions.

In general, the ultimate goal of multi-objective optimization algorithms is to help the DM to find the most satisfactory solution rather than all optimal Pareto solutions [17, 18]. This can be achieved by combining the preference of DM with the optimization process. In the preference approach, preference information is added to the search process to guide the search to the region of greatest interest to the DM. This helps to improve optimization efficiency and reduce computational cost [17, 18]. According to different ways of interacting with users, multi-objective optimization algorithms can be divided into three categories, namely, a priori, posterior and interactive [19, 20]. Interactive algorithms, as preference algorithms, have become a new trend, in which the user dynamically guides the search process in an interactive manner until an output that is satisfactory to the user is obtained.

DeBruyne and Kaur [21] proposed the Harris's hawk multi-objective optimizer (HHMO) which is developed based on the GWO [11]; this can be considered as an improved version of the MOGWO [10]. Instead of spending time in searching for non-dominated solutions in undesired regions, as in the case of the MOGWO, the HHMO algorithm focuses on a particular region of the Pareto front approximated based on preference points (reference points) determined by the DM and the non-dominated solutions are clustered near a reference point as illustrated in Figure 1(b). Thus, it inherits the advantages of the preference interactive approach in terms of effectively finding the most satisfactory solutions. This can help in reducing the computational cost [22, 23].

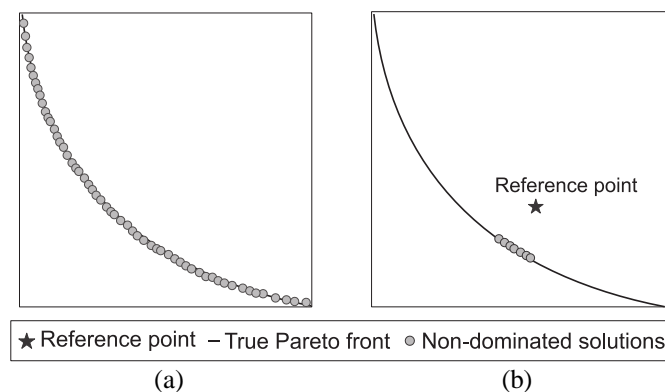


Figure 1. Non-dominated objective vectors: (a) evenly distributed across the whole Pareto front; (b) clustered near a reference point

Although, DeBruyne and Kaur [21] claimed that the HHMO algorithm is able to solve three or more objectives, they do not provide any test that shows the performance of HHMO in solving three or more objectives with high-dimensional MOPs. This algorithm is relatively new and, according to the No-free-lunch (NFL) theorem [24], there is no an algorithm that can be efficiently used to solve all optimization problems. If the algorithm provides an efficient performance in solving a particular problem, this does not mean it will be able to provide the same performance in solving other problems. This has encouraged

researchers to propose new algorithms or improve existing algorithms. Therefore, this study focuses on performing more tests to validate the performance of the algorithms using different benchmark functions. Additionally, to the best of the authors' knowledge, there is no improved version of HHMO that has been proposed in the literature. This paper aims to propose an improved Harris's hawk multi-objective optimizer algorithm, called the non-dominated sorting HHMO (NDSHHMO) by integrating the HHMO algorithm with non-dominated sorting (NDS). The NDS is considered as one of the most common vector sorting schemes in solving MOPs. Several studies have proved the efficiency and effectiveness of NDS, with most common multi-objective evolutionary algorithms (MOEAs) [25-30]. NDS helps in improving the convergence of the algorithm towards the true Pareto front (PF_{true}), especially for dealing with complex MOPs with a large number of local Pareto fronts [31]. The performance of proposed algorithm is evaluated using a set of well-known MOPs and its results are compared with the original HHMO using several performance metrics.

In multi-objective optimization, the following important basic concepts are usually used. In general, a MOP can be defined as a problem whose search for the set of decision variables, $X = (x_1, x_2, \dots, x_d)$, satisfying certain constraints and simultaneously optimizing a set of criteria of dimensions greater than or equal to two, $f(X) = (f_1(X), f_2(X), \dots, f_M(X))$. A MOP can be defined as follows:

$$\begin{aligned} \min(\max) F(X) &= (f_1(X), f_2(X), \dots, f_M(X)) \\ \text{s. t. } x &\in S \subset R^n \end{aligned} \quad (1)$$

where d is the dimension of decision vector; S is a feasible domain of x ; $f_m(X)$, $m=1, \dots, M$; M is the number of objectives. In multi-objective optimization, Pareto dominance relations are used to measure the quality of the solutions in the objective space and defined as: for any two solution vectors $S1$ and $S2$, we denote that $S1$ forms a Pareto domination for the solution $S2$ if and only if: $S1$ is superior to $S2$ for all objectives, then we say $S1$ dominates $S2$. $S1$ is called a non-dominated solution, also known as a Pareto solution, if it is not dominated by other solutions. For the sake of simplicity, we can also say the solution $S1$ is dominating solution $S2$, denoted as $S1 < S2$, and the solution $S1$, by dominating solution $S2$, represents that solution $S1$ is better than solution $S2$. If the two do not dominate each other, then $S1$ and $S2$ are equivalent. Based on this definition, the essence of solving the MOP is to find all solutions that are not dominated by any other solutions.

This paper is organized as follows. Section 2 presents the concept of fast non-dominating sorting in multi-objective optimization. Description of the original HHMO is presented in Section 3 while Section 4 describes the proposed NDSHHMO algorithm. The experimental setting is presented in Section 5 while Section 6 presents the experimental results and analysis of the MOPs. Finally, the conclusion and future work are presented in Section 7.

2. FAST NON-DOMINATED SORTING

The concept of NDS was first proposed by [32]. However, its main drawback is the high computational cost which has been resolved in [27], by proposing fast non-dominated sorting (FNDS). In FNDS, fitness calculation based on the concept of Pareto optimum is used to move a population up to the Pareto front in a MOP. The fundamental idea is to select the non-dominated solutions (individuals) with respect to the current population to calculate the hierarchy and the higher aptitude. Individuals that have the best quality in the population are considered as a first level of frontier and assigned the first rank. Subsequently, these individuals are temporarily eliminated from the competition. The non-dominated individuals in the remaining population are selected to construct the second level of frontier and assigned the second rank. These processes are repeated until there is no individual left. Thus, solutions that are not dominated by other solutions are assigned a rank equal to 1, if they are dominated by only one solution they are assigned a rank equal to 2 and solutions dominated by only two solutions are assigned a rank equal to 3, and so on. In this way, the population is divided into multiple non-dominated frontiers, each defining a specific quality level. Figure 2 illustrates the principle of non-dominated sorting.

The FNDS divides the population of N individuals into four non-dominated frontiers and it stratifies all individuals in a population P . The first layer, F_1 , is composed of non-dominated solutions in the initial population. Then the individuals that have been assigned to F_1 are removed from P . The remaining individuals are a composition of the set $P - F_1$ and the second layer, F_2 is composed of the non-dominated solutions in $P - F_1$. The third layer, F_3 is composed of the non-dominated solutions in $P - F_1 - F_2$. The subsequent layers are analogous. Details about FNDS can be obtained from [27]. Figure 3 shows the main steps of FNDS [27].

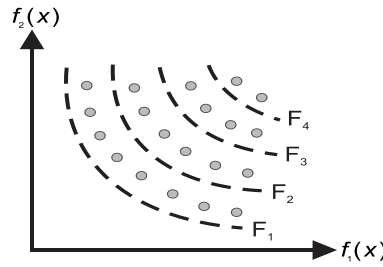


Figure 2. Dividing a population into four levels of front (F₁, F₂, F₃, F₄) by fast non-dominated sorting

Algorithm 1: Fast non-dominated sorting of P		
1	For each $p \in P$	
2	$S_p = \emptyset$	
3	$n_p = 0$	
4	For each $q \in P$	
6	If $p \prec q$ then	If p dominates q
5	$S_p = S_p \cup \{q\}$	Add q to the set of solutions dominated by p
11	Else If $q \prec p$ then	
12	$n_p = n_p + 1$	Increment the domination counter of p belongs to the first front
13	If $n_p = 0$ then	
14	$p_{rank} = 1$	
15	$F_1 = F_1 \cup \{p\}$	
16	$i = i + 1$	Initialize the front counter
17	While $F_i \neq \emptyset$	
	$Q = \emptyset$	Used to store the members of the next front
	For each $q \in S_p$	
	$n_q = n_q - 1$	
	If $n_q = 0$ then	q belongs to the next front
	$q_{rank} = i + 1$	
	$Q = Q \cup \{q\}$	
	$i = i + 1$	
	$F_i = Q$	

Figure 3. Pseudo code of fast non-dominated sorting

In general, NDS-based algorithms, when making survival choices, it is possible to discard F₃ and F₄ directly but discarding F₂ will lead to losing too many individuals. In this case, the number of selected individuals will not be enough to produce the next generation. If the total number of F₁ and F₂ has exceeded the required population size of the next generation, all individuals in F₁ will survive to the next generation and the rest will be selected from the next front, F₂, based on another quality criterion (such as diversity) [31]. In this paper, ϵ -clearing strategy, proposed by [33], has been used as a second quality criterion to select between individuals that belong to the same front (have the same rank). This strategy divides the objective space into grids of size, ϵ . Then, individuals with the smallest Euclidean distance to a reference point are selected to be added to the next generation, P . This helps in preserving diversity among solutions of the same front.

3. HARRIS'S HAWK MULTI-OBJECTIVE OPTIMIZER ALGORITHM

The HHMO algorithm [21] is a kind of SI-based optimization algorithm proposed by mimicking the social hierarchy and hunting behavior of the Harris's hawk predator in nature [34]. In the social hierarchy of the Harris's hawk, there are four social ranks, from high to low, alpha (α), beta (β), delta (δ) and gamma (ω) hawks. The hunting process is divided into two main stages, namely, encircling and attacking, which closely resemble the encircling and attacking behavior of grey wolves [21]. The mathematical model of the HHMO is developed based on the GWO algorithm. Therefore, it inherits the characteristics and advantages of the GWO algorithm [11]. In the HHMO algorithm, the hawks represent candidate solutions in the decision space. The population of hawks is divided into groups according to the number of reference points. In each group, the leaders α , β and δ are the hawks that have the three shortest distances to a reference point and they

represent the first best three non-dominated solutions, respectively. The remaining hawks are represented by ω . To simulate the collective hunting behavior of the Harris's hawks, it is assumed that, α , β and δ hawks have a better understanding of the potential position of the prey. Therefore, during each iteration, their positions are saved and used to comprehensively determine the direction of the ω hawks and their positions are updated to move toward the prey. The behavior of the group approaching and surrounding the prey is formulated as follows:

$$\begin{aligned} \vec{X}(t + 1) &= \vec{X}_p(t) - \vec{A} * \vec{D} \\ \vec{D} &= C * \vec{X}_p(t) - \vec{X}(t) \end{aligned} \tag{2}$$

D is the distance between the hawk and the prey, t is the number of current iterations, and $X_p = (x_{p1}, x_{p2}, \dots, x_{pd})$ is the position vector of the prey, while $X = (x_1, x_2, \dots, x_d)$ represents the position vector of the hawks in d dimension. A and C are parameter vectors, formulated as follows:

$$\vec{A} = 2\vec{a} * r_1 - \vec{a} \tag{3}$$

$$\vec{C} = 2 * r_2 \tag{4}$$

where r_1 and r_2 are random vectors in interval $[0,1]$. $a = 2 - t(2/MaxIteration)$, is the control parameter, in the range $[0,2]$ and decreases linearly during the optimization process, with the number of iterations t . The average position of α , β and δ hawks is used to calculate a new position of hawks, as shown in Equation (5).

$$\begin{aligned} \vec{X}(t + 1) &= \frac{(\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t))}{3} \\ \vec{X}_1(t) &= \vec{X}_\alpha(t) - \vec{A}_\alpha * \vec{D}_\alpha \\ \vec{X}_2(t) &= \vec{X}_\beta(t) - \vec{A}_\beta * \vec{D}_\beta \\ \vec{X}_3(t) &= \vec{X}_\delta(t) - \vec{A}_\delta * \vec{D}_\delta \end{aligned} \tag{5}$$

4. NON-DOMINATED SORTING HHMO BASED ON REFERENCE POINT

In general, an efficient, simple and fast method is required to reach an optimum with acceptable accuracy within a reasonable time. One of the challenges of metaheuristics is therefore to facilitate the choice of a method and simplify its adjustment to best adapt to a problem. In [21], to enhance the performance of the HHMO algorithm, the authors introduce two evolutionary strategies (ES), namely, direct replacement (μ, λ)-ES and selection of the best ($\mu + \lambda$)-ES [35, 36], with HHMO. These strategies were used to generate the next generation from the current and old generations. In the former strategy, the next generation is produced by replacing the entire parent population, μ , with the offspring, λ . In later strategy, the parent and offspring populations are combined, then the best individuals are selected to produce the next generation. According to the authors [21], the ($\mu + \lambda$)-ES shows superior performance compared to (μ, λ)-ES. However, in HHMO, the non-dominated solutions are selected based on the closest distance to a reference point, v_i . In this case, the solutions in the objective space are tends to move toward a reference point instead of Pareto front. To overcome this limitation, this study integrates the FNDS, as another criterion to select non-dominated solutions and improve the convergence toward the true Pareto front. This aims to improve the stability and convergence of the algorithm. The pseudo code of NDSHHMO algorithm is shown in Figure 4.

In NDSHHMO, the optimization process starts by randomly initializing the population of hawks. Then, each hawk, X_i , is evaluated by calculating the fitness value for all objectives, $f_m(X_i)$. The distance between the reference points, V and all hawks in the objective space, $f_m(X_i)$, is calculated using Euclidean distance formula. The population of hawks is then divided into groups based on the number of reference points. At each iteration, the position of hawks is updated according to Equation (5). Then, the binomial recombination procedure [37] is performed to produce a new generation of hawks (offspring). The new positions are evaluated and the new leaders are selected based on the shortest Euclidean distance to a reference point. In the proposed NDSHHMO, in addition to the main procedures of HHMO with ($\mu + \lambda$)-ES [21], the FNDS procedure is incorporated to divide the objective space into front levels. Then, non-dominated solutions are selected from the front to be used in the next generation. These processes are repeated until the loop termination condition is met and, finally, output the non-dominated solutions set. In NDSHHMO algorithm, the mutation operation used in the enhanced HHMO (HHMO with ($\mu + \lambda$)-ES), has been eliminated to reduce the complexity of the algorithm and the number of parameters represented by a mutation factor. Basic procedures of the NDSHHMO algorithm are:

- a) Initialize a population of hawks.
- b) Evaluate each hawk in the population.
- c) Select the leaders, α , β and γ from the initial population, based on the smallest Euclidean distance to a reference point.
- d) Update the position of hawks with respect to the positions of leaders.
- e) Recombination.
- f) Evaluate each hawk.
- g) Perform FNDS and select non-dominated solutions to produce the next generation of hawks.

Algorithm 2: NDSHHMO	
1	Given the set of objective functions $F(X) = (f_1(X), f_2(X), \dots, f_M(X))$
2	Initialize the population of hawk, X within the boundaries of the decision space d
3	Evaluate each hawk in the population according to objective functions
4	Divide the search agents into groups based on based on number of reference points
5	while ($t \leq$ maximum number of iterations or termination condition)
6	For each reference point group
7	Select leaders (α , β and γ):
8	i. Calculate the Euclidean distances to a reference point, v_i for each hawk, X_i , on all objectives $F(X)$.
9	ii. Select first three hawks that have a shortest distance to a reference point, V_z , to be α , β and γ .
10	For each search agent in a group
11	Update position of a hawk based on Equations set (5)
12	end for
13	Perform recombination procedure
14	For each search agent in a group
15	Calculate fitness values for all hawks in a group
16	end for
17	Perform FNDS and select non-dominated solutions to produce the next generation of hawks.
18	end for
19	$t = t + 1$
20	end while
21	Return the best non-dominated solutions

Figure 4. Pseudo code of non-dominated sorting Harris's hawk multi-objective optimizer

5. RESEARCH METHOD

A set of eight (8) test functions (Deb & Sundar, 2006) has been used to evaluate and compare the performance of the proposed, NDSHHMO and the original HHMO algorithms with respect to convexity, non-convexity multimodality and non-uniformity. These functions are ZDT1, ZDT2, ZDT3 and ZDT4, with two objectives. The test function DTLZ2 with three, five and 10 objectives has also been used [38]. These test functions are commonly used to test the stability and efficiency of an algorithm. For each algorithm, the maximum number of iteration has been set at 300, and the population size of 100 individuals for each problem. Table 1 shows the reference point used with each problem.

Problem	Reference point
ZDT1,4,3	(0.1,0.6) (0.5,0.2)
ZDT2	(0.8,0.2) (0.15,0.8)
ZDT6	(0.90,0.3) (0.5,0.7)
DTLZ2 with 3 objectives	(0.2,0.2,0.6) (0.8,0.6,1.0)
DTLZ2 with 5 objectives	(0.5, 0.5, 0.5, 0.5, 0.5) (0.2 0.2 0.2 0.2 0.8)
DTLZ2 with 10 objectives	(0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25,0.25)

To compare the results obtained by the multi-objective optimization algorithms, three (3) commonly used performance metrics have been utilized, namely, inverted generation distance (IGD) [39] and R-metrics [40], which includes R-IGD and R-HV. For each metric, the statistical measures have been used to compare the performance of algorithms. They seek to capture the characteristics that make an approximation of the Pareto front better than another in some criterion. In general, the values of these metrics reflect a certain quality aspect to a particular approximation set, such as, diversity of solutions and convergence. The inverted

generation distance measures the average distance between all individuals in the PF_{true} to the nearest individual in the solution set obtained by the algorithm. The final solution represents information about the convergence and the diversity of the non-dominated solutions. The IGD value is calculated as shown in Equation (6):

$$IGD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \tag{6}$$

where $p = 2$ and d_i^2 is the minimum Euclidean distance between a point, i in the approximate Pareto front and a nearest point in the PF_{true} . n is the number of solutions belonging to the PF. IGD values close to zero indicate that non-dominated solutions are very close to true Pareto optimal set.

6. RESULTS AND ANALYSIS

Each algorithm was executed 10 times, independently with each MOP, to obtain the statistical significance of the results and for fair comparison. The comparison between NDSHHMO and HHMO was carried out based on the mean, standard deviation (SD), best and worst values of quality metrics. Table 2 shows the results of the IGD metrics.

Table 2. Results of Mean, Standard Deviation, Best and Worst IGD Values Over 10 Independent Runs for Both HHMO and NDSHHMO Algorithms

Algorithm		HHMO		NDSHHMO	
Problem	m	Mean (SD)	Best (Worst)	Mean (SD)	Best (Worst)
ZD1	2	3.015766e-05	2.289158e-05	4.097678e-05	3.465444e-05
		3.456729e-06	3.395342e-05	(4.313639e-06)	(4.928885e-05)
ZDT2	2	3.462937e-05	2.454948e-05	4.137926e-05	3.217633e-05
		5.876528e-06	4.608892e-05	(8.304714e-06)	(5.937574e-05)
ZDT3	2	3.054700e-04	2.507444e-04	6.919176e-05	6.022995e-05
		3.639905e-05	3.652381e-04	(7.736077e-06)	(8.499650e-05)
ZDT4	2	1.209638e-04	2.581917e-05	5.070034e-05	2.390682e-05
		2.012809e-04	6.664835e-04	(1.821928e-05)	(8.380816e-05)
ZDT6	2	3.174891e-05	1.531363e-05	2.161897e-05	1.409194e-05
		(1.213382e-05)	(5.010032e-05)	(6.071884e-06)	(3.070295e-05)
DTLZ2	3	9.162699e-04	8.520694e-04	4.271674e-04	3.849329e-04
		4.509744e-05	9.986688e-04	(3.098091e-05)	(4.825089e-04)
DTLZ2	5	1.441510e-03	1.368030e-03	1.421138e-03	1.380836e-03
		2.746606e-05	1.465777e-03	(3.995864e-05)	(1.518529e-03)
DTLZ2	10	2.384243e-03	2.333402e-03	2.369540e-03	2.330149e-03
		(3.145295e-05)	(2.418701e-03)	(2.957656e-05)	(2.406577e-03)

Both HHMO and NDSHHMO perform almost equal in solving DTLZ2 with five and ten objectives problems. For solving the ZDT6 problem, the performance of the NDSHHMO is slightly better than the HHMO. For other problems, namely, ZDT3, ZTD4 problems and the DTLZ2 problem with three objectives, NDSHHMO performed significantly better than the HHMO. However, in solving the ZDT1 and ZDT2 problems, based on the average IGD value, HHMO shows slightly better performance as compared to NDSHHMO.

Li, et al. [40] proposed R-metrics based on the MCDM approach, which include R-IGD and hypervolume with R-Metric (R-HV). These metrics adapt the existing metrics, namely, IGD [39] and hypervolume (HV) [41] to evaluate the quality of a set of non-dominated solutions obtained by a reference point-based multi-objective optimization algorithm. The HV metric is used when the optimal Pareto solutions are unknown, where the larger value of HV indicates a better result. Mathematically, the HV is described by Equation (7).

$$HV(A) = \lambda(\cup_{a \in A} [f_1(a), r_1] \times [f_2(a), r_2] \times \dots \times [f_k(a), r_k]) \tag{7}$$

HV is denoted as the hyper-volume of a space that is dominated by a set of solution A and is bounded by a reference point, $r = (r_1, r_2, r_3, \dots, r_k) \in R^k$. $\lambda(S)$ is the Lebesgue measure of a set S [42]. The R-metric first filters out the non-dominated solutions, then determines the pivot point and the preferred region and eliminates the solution outside the preferred region. Using the achievement scalarization function, the compromise solution is transformed according to the degree of satisfaction of the preference

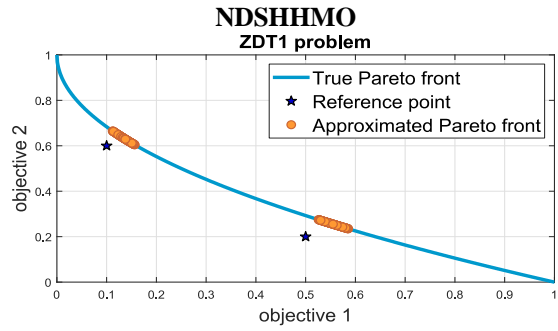
information. The R-metrics are calculated using the sets approximated Pareto fronts ($PF_{\text{approx.}}$) generated by each algorithm [40]. Table 3 shows the mean, SD, best and worst R-metrics values obtained by the HHMO and NDSHHMO algorithms.

Table 3. Results of Mean, Standard Deviation, Best and Worst R-IGD and R-HV Values Over 10 Independent Runs for Both HHMO and NDSHHMO Algorithms

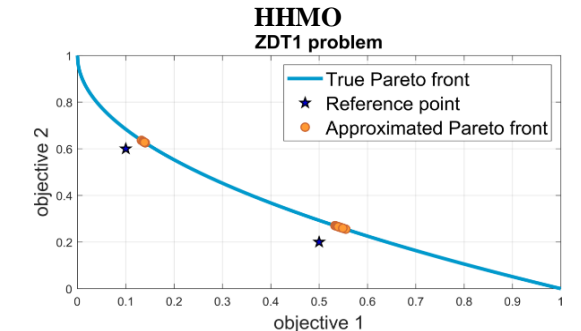
Algorithm			HHMO		NDSHHMO	
R-metrics	Problem	m	Mean (SD)	Best (Worst)	Mean (SD)	Best (Worst)
R-IGD	ZD1	2	5.750499e-02	5.370099e-02	3.086646e-02	2.916731e-02
			1.629131e-03	5.943255e-02	(8.668946e-04)	(3.180344e-02)
	ZDT2	2	3.582150e-02	2.162508e-02	2.526445e-02	2.429212e-02
			1.223893e-02	6.550052e-02	(5.869044e-04)	(2.629707e-02)
	ZDT3	2	1.564317e-01	1.133154e-01	8.126304e-02	7.990480e-02
			1.232284e-01	4.850325e-01	(8.941625e-04)	(8.219142e-02)
	ZDT4	2	1.595845e-01	5.389915e-02	3.200888e-02	3.178943e-02
			2.151564e-01	5.684010e-01	(2.148493e-04)	(3.245914e-02)
	ZDT6	2	5.203782e-01	3.375339e-02	2.653745e-02	2.536410e-02
			(2.565047e-01)	(6.513960e-01)	(6.092902e-04)	(2.727200e-02)
R-HV	DTLZ2	3	1.493500e-01	1.251975e-01	1.258225e-01	1.132261e-01
			1.384537e-02	1.645151e-01	(6.984063e-03)	(1.348985e-01)
	DTLZ2	5	2.241562e-01	2.208843e-01	1.977805e-01	1.918947e-01
			2.027524e-03	2.269642e-01	(4.233870e-03)	(2.041274e-01)
	DTLZ2	10	6.902470e-01	6.850544e-01	6.933113e-01	6.890396e-01
			1.938679e-03	6.920374e-01	(2.822722e-03)	(6.966718e-01)
	ZD1	2	3.876157e+00	3.890523e+00	3.964742e+00	3.961318e+00
			7.576768e-03	3.865809e+00	(3.223440e-03)	(3.971095e+00)
	ZDT2	2	3.856566e+00	3.884824e+00	3.864925e+00	3.861443e+00
			5.710995e-02	3.690229e+00	(2.247994e-03)	(3.868524e+00)
ZDT3	2	3.607186e+00	3.729507e+00	3.797716e+00	3.790436e+00	
		3.538229e-01	2.663683e+00	(6.756720e-03)	(3.809657e+00)	
ZDT4	2	3.595390e+00	3.882197e+00	3.960615e+00	3.958948e+00	
		5.860350e-01	2.470387e+00	(7.989916e-04)	(3.961462e+00)	
ZDT6	2	2.887313e+00	2.515327e+00	4.265944e+00	4.262028e+00	
		(7.063620e-01)	(4.227197e+00)	(2.698068e-03)	(4.270751e+00)	
DTLZ2	3	5.181027e+00	5.420532e+00	5.287903e+00	5.224974e+00	
		1.258370e-01	5.062406e+00	(4.667065e-02)	(5.378672e+00)	
DTLZ2	5	3.258571e+01	3.276106e+01	3.438223e+01	3.399767e+01	
		9.250290e-02	3.246725e+01	(2.987349e-01)	(3.490270e+01)	
DTLZ2	10	7.486208e+02	7.539277e+02	7.365107e+02	7.311533e+02	
		3.149022e+00	7.442124e+02	(3.038939e+00)	(7.419556e+02)	

The results in Table 3 imply that the NDSHHMO has superior performance in solving MOPs under test. According to mean R-IGD values, NDSHHMO outperforms HHMO in solving the ZDT1, ZDT2, ZDT3 and ZTD4 problems. In solving DTLZ2 with 3, 5 and 10 objectives, NDSHHMO shows competitive performance compared with HHMO, except for DTLZ2 with 10 objectives. The results of HHMO are slightly better than NDSHHMO. Figure 5 shows the final solution set obtained by HHMO and NDSHHMO algorithms.

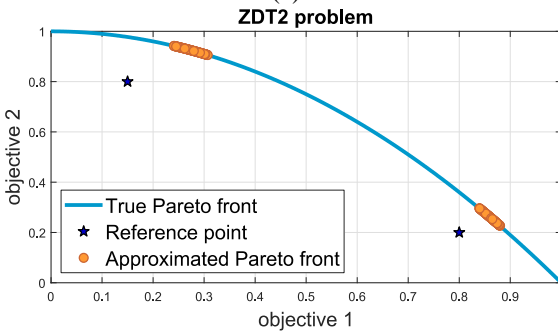
The distribution of the solution obtained by both the HHMO and NDSHHMO algorithms (refer to Figure 5(a-p)) are clustered near the reference points. However, in the ZDT6 problem, HHMO moved towards the reference point instead of PF_{true} . The same occurred in the ZDT3 problem, the distribution and the convergence of solutions obtained by NDSHHMO is much better than HHMO. For the ZDT2 and ZDT4 problems, the solution obtained by NDSHHMO has better spread than the HHMO. For ZDT1 the convergence of the solution obtained by NDSHHMO is slightly less, as the mean IGD value indicates, than that obtained by HHMO. However, according to the R-metrics, the convergence of the solution obtained by HHMO is significantly less than that obtained by NDSHHMO. In solving DTLZ2 with 3, 5 and 10 objectives, both algorithms showed good distribution and convergence. However, based on the R-IGD and R-HV values, NDSHHMO showed better convergence and distribution compared to HHMO. From the experimental results on eight benchmark functions, it can be found that NDSHHMO has advantages in terms of speed, convergence and distribution compared with the HHMO algorithm. According to the R-metrics, the solutions obtained by NDSHHMO are closer to the PF_{true} compared to those of HHMO.



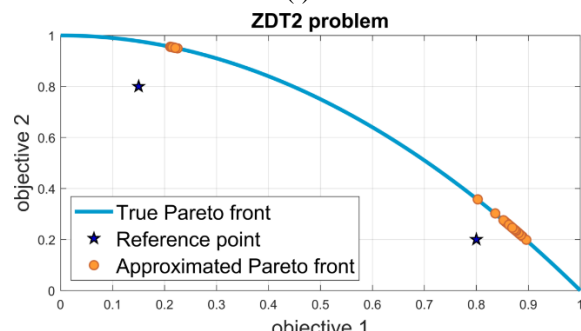
(a)



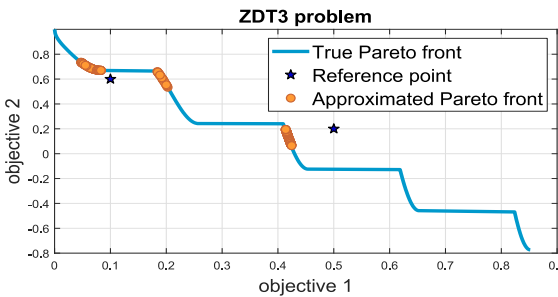
(i)



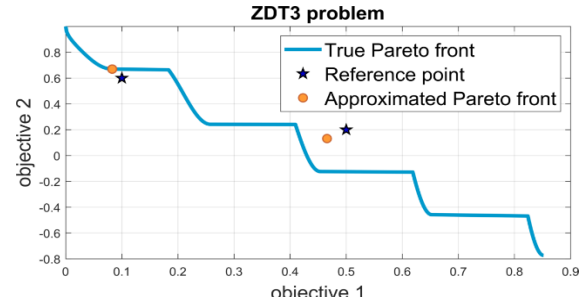
(b)



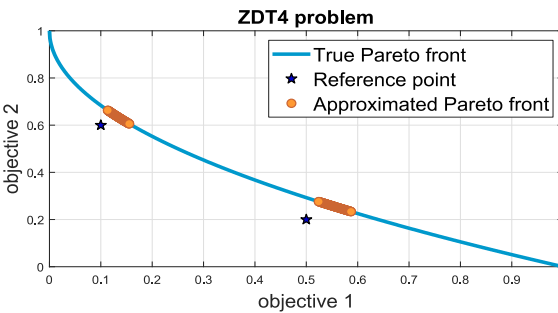
(j)



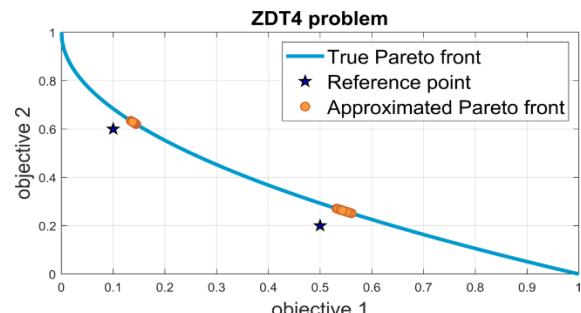
(c)



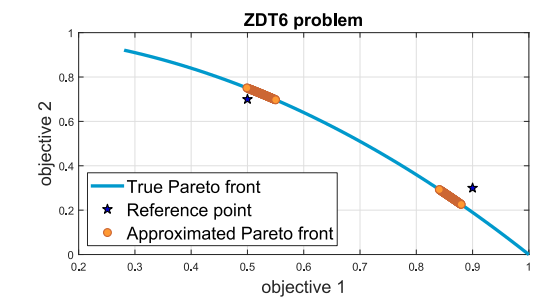
(k)



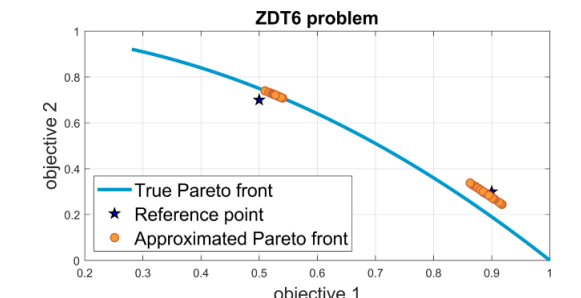
(d)



(l)



(e)



(m)

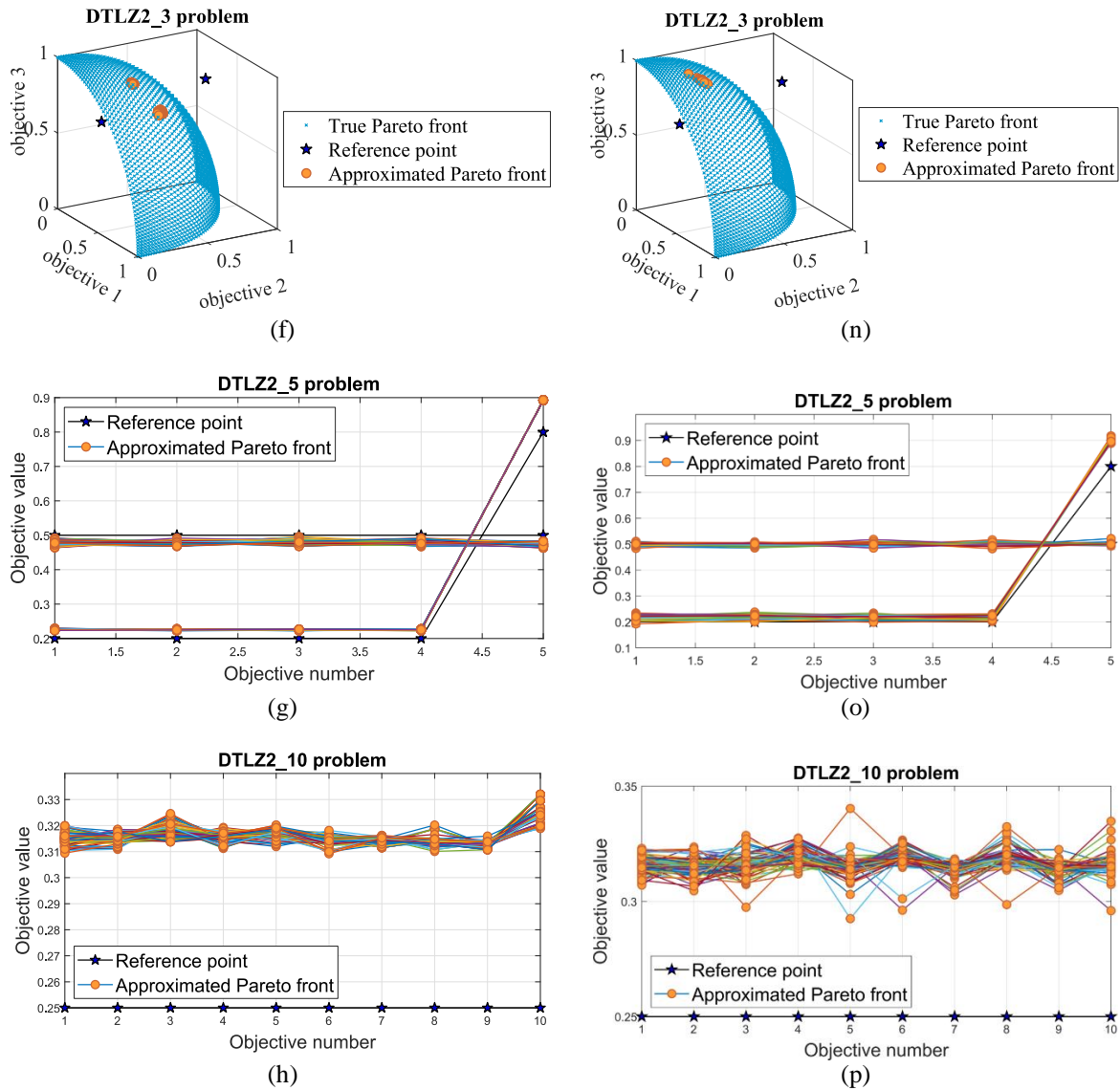


Figure 5. Convergence of non-dominated solutions obtained by (a-h) NDSHHMO and (i-p) HHMO algorithms

7. CONCLUSION

The NDSHHMO algorithm is proposed to overcome the limitations of HHMO. The fast non-dominated sorting has been integrated with the original HHMO to improve the convergence toward the true Pareto front. This has helped in maintaining the population diversity and enhancing the search ability of the algorithm. The performance of the proposed algorithm is evaluated using different MOPs. The proposed algorithm is not only suitable for high-dimensional functions, but can also effectively deal with MOPs with more than three objectives. The proposed algorithm is expected to be used for other optimization problems, such as, structural design engineering optimization problems. Future work can focus on the use of NDS to improve the performance of other SI-based multi-objectives optimization algorithms.

ACKNOWLEDGEMENT

In this The Higher Education Ministry of Malaysia has funded this work under the Fundamental Research Grant Scheme, S/O code 13794.

REFERENCES

- [1] J. Andersson, "A survey of multiobjective optimization in engineering design," Department of Mechanical Engineering, Linköping University, Sweden, 2000.
- [2] G. P. Rangaiah and A. B. Petriciolet, "Multi-objective optimization in chemical engineering: developments and applications," *John Wiley & Sons*, 2013.
- [3] D. Gabi, et al., "Hybrid cat swarm optimization and simulated annealing for dynamic task scheduling on cloud computing environment," *Journal of ICT*, vol. 17, pp. 435-467, 2018.
- [4] J. Branke, et al., "Multiobjective optimization: Interactive and evolutionary approaches," *Springer Science & Business Media*, 2008.
- [5] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in P. Dario, et al., "Robots and Biological Systems: Towards a New Bionics?" Berlin, Heidelberg, Springer, vol. 102, pp. 703-712, 1993.
- [6] D. Angus and C. Woodward, "Multiple objective ant colony optimisation," *Swarm intelligence*, vol. 3, pp. 69-85, 2009.
- [7] C. A. C. Coello, et al., "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on evolutionary computation*, vol. 8, pp. 256-279, 2004.
- [8] X. S. Yang, "Multiobjective firefly algorithm for continuous optimization," *Engineering with Computers*, vol. 29, pp. 175-184, 2013.
- [9] R. Hedayatzaeh, et al., "A multi-objective Artificial Bee Colony for optimizing multi-objective problems," in 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), vol. 5, pp. V5-277-V5-281, 2010.
- [10] S. Mirjalili, et al., "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization," *Expert Systems with Applications*, vol. 47, pp. 106-119, 2016.
- [11] S. Mirjalili, et al., "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014.
- [12] L. Dilip, et al., "Optimal Power Flow Problem Solution Using Multi-objective Grey Wolf Optimizer Algorithm," in *Intelligent Communication and Computational Technologies*, Springer, pp. 191-201, 2018.
- [13] H. Faris, et al., "Grey wolf optimizer: a review of recent variants and applications," *Neural Computing and Applications*, pp. 1-23, 2017.
- [14] R. E. Precup, et al., "An Easily Understandable Grey Wolf Optimizer and Its Application to Fuzzy Controller Tuning," *Algorithms*, vol. 10, pp. 68, 2017.
- [15] M. Siavash, et al., "An application of grey wolf optimizer for optimal power flow of wind integrated power systems," in Electric Power Engineering (EPE), 2017 18th International Scientific Conference on, pp. 1-6, 2017.
- [16] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, pp. 712-731, 2007.
- [17] L. Thiele, et al., "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary computation*, vol. 17, pp. 411-436, 2009.
- [18] K. Yang, et al., "Preference-based multiobjective optimization using truncated expected hypervolume improvement," in 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), pp. 276-281, 2016.
- [19] T. Bäck, et al., "Handbook of evolutionary computation," *CRC Press*, 1997.
- [20] J. Horn, "Multicriteria decision making and evolutionary computation," 1996.
- [21] A. S. DeBruyne and B. D. Kaur, "Harris's Hawk Multi-Objective Optimizer for Reference Point Problems," in Proceedings on the International Conference on Artificial Intelligence (ICAI), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p. 287, 2016.
- [22] R. Liu, et al., "Interactive reference region based multi-objective evolutionary algorithm through decomposition," *IEEE Access*, vol. 4, pp. 7331-7346, 2016.
- [23] A. Mohammadi, et al., "Reference point based multi-objective optimization through decomposition," in Evolutionary Computation (CEC), 2012 IEEE Congress on, pp. 1-8, 2012.
- [24] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, pp. 67-82, 1997.
- [25] X. Cai, et al., "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, pp. 508-523, 2015.
- [26] D. W. Corne, et al., "PESA-II: Region-based selection in evolutionary multiobjective optimization," in Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc, pp. 283-290, 2001.
- [27] K. Deb, et al., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, pp. 182-197, 2002.
- [28] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1, pp. 443-450, 2005.
- [29] A. J. Nebro, et al., "Smpso: A new pso-based metaheuristic for multi-objective optimization," in *Computational intelligence in multi-criteria decision-making, 2009. mcdm'09. IEEE symposium on*, pp. 66-73, 2009.
- [30] Y. Tian, et al., "A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric," in *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pp. 5222-5229, 2016.
- [31] Y. Tian, et al., "Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization," *Complex & Intelligent Systems*, vol. 3, pp. 247-263, 2017.
- [32] D. E. Goldberg, "Genetic algorithms in search, optimization, and machine learning/David E," *Goldberg.-[USA]: Addison-Wesley*, 1989.

- [33] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in Proceedings of the 8th annual conference on Genetic and evolutionary computation, pp. 635-642, 2006.
- [34] J. C. Bednarz, "Cooperative hunting in Harris' hawks (*Parabuteo unicinctus*)," *Science*, vol. 239, pp. 1525, 1988.
- [35] I. Rechenberg, "Evolution Strategy: Nature's Way of Optimization," in Optimization: Methods and Applications, Possibilities and Limitations: Proceedings of an International Seminar Organized by Deutsche Forschungsanstalt für Luft- und Raumfahrt, Bonn, June 1989, H. W. Bergmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 106-126, 1989.
- [36] H. P. P. Schwefel, "Evolution and Optimum Seeking: The Sixth Generation," *John Wiley & Sons, Inc.*, pp. 456, 1993.
- [37] T. Desell, et al., "Validating evolutionary algorithms on volunteer computing grids," in IFIP International Conference on Distributed Applications and Interoperable Systems, Springer, pp. 29-41, 2010.
- [38] C. A. C. Coello, et al., "Evolutionary algorithms for solving multi-objective problems," *Springer*, 2007.
- [39] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evolutionary Computation*, vol. 18, pp. 577-601, 2014.
- [40] K. Li, et al., "R-Metric: Evaluating the Performance of Preference-Based Evolutionary Multiobjective Optimization Using Reference Points," *IEEE Transactions on Evolutionary Computation*, vol. 22, pp. 821-835, 2018.
- [41] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," *Citeseer*, 1999.
- [42] D. Brockhoff, et al., "Analyzing hypervolume indicator based algorithms," in International Conference on Parallel Problem Solving from Nature, Springer, pp. 651-660, 2008.

BIOGRAPHIES OF AUTHORS



Shaymah Akram Yasear holds a Bachelor in Computer Sciences from University of Benghazi (Garyounis), Libya in 2008. Her Masters degree in Information Technology was obtained from Universiti Utara Malaysia, Malaysia in 2016. Currently, she is doing a PhD in Information Technology at Universiti Utara Malaysia, Malaysia. As an academic, her research interest includes swarm intelligence optimization.



Ku Ruhana Ku-Mahamud holds a Bachelor in Mathematical Sciences and a Masters degree in Computing, both from Bradford University, United Kingdom in 1983 and 1986 respectively. Her PhD in Computer Science was obtained from Universiti Pertanian Malaysia in 1994. As an academic, her research interests include ant colony optimization, pattern classification and vehicle routing problem.