Comparative Study of Knee-Based Algorithms for Many-Objective Optimization Problems

Saad M. Alzahrani[1] and **Naruemon Wattanapongsakorn**[2]

ABSTRACT

Nowadays, most real-world optimization problems consist of many and often conflicting objectives to be optimized simultaneously. Although, many current Multi-Objective optimization algorithms can efficiently solve problems with 3 or less objectives, their performance deteriorates proportionally with the increasing of the objectives number. Furthermore, in many situations the decision maker (DM) is not interested in all trade-off solutions obtained but rather interested in a single optimum solution or a small set of those trade-offs. Therefore, determining an optimum solution or a small set of trade-off solutions is a difficult task. However, an interesting method for finding such solutions is identifying solutions in the Knee region. Solutions in the Knee region can be considered the best obtained solution in the obtained trade-off set especially if there is no preference or equally important objectives. In this paper, a pruning strategy was used to find solutions in the Knee region of Pareto optimal fronts for some benchmark problems obtained by NSGA-II, MOEA/D-DE and a promising new Multi-Objective optimization algorithm NSGA-III. Lastly, those knee solutions found were compared and evaluated using a generational distance performance metric, computation time and a statistical one-way ANOVA test.

Keywords: Decision making, Many-Objective optimization, Multi-Objective optimization, Knee-based MOEAs

1. INTRODUCTION

Multi-Objective optimization problems, known as MOP, are optimization problems with two or more incommensurable objective functions, often conflicted, and need to be optimized simultaneously. Similarly, Many-Objective optimization problems, known as MaOP, are optimization problems with more than three objectives [1]. In general, Multi-Objective optimization does not have a single optimum solution but rather there is a set of alternative trade-off solutions, generally known as Pareto optimal solutions. These solutions are optimal in the sense that no other solutions in the search space are superior to them when all objectives are considered [24]. An example of a Many-Objective optimization problem is a power plant expansion problem. The objectives of such problem may include minimizing cost and greenhouse gas emissions while maximizing system reliability and capacity. Fig.1 illustrates this example.

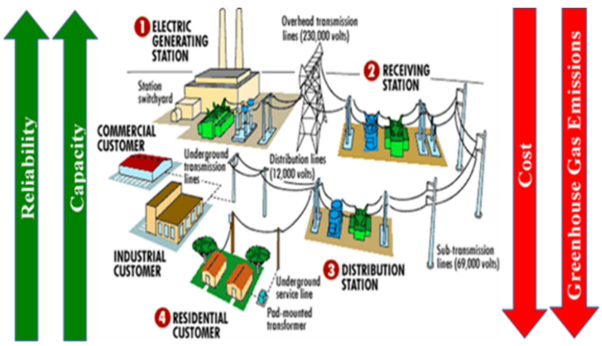

Fig.1: *Power plant expansion problem.*

Multi-Objective Evolutionary Algorithms, well-known as MOEA, have been proven to be quite efficient in solving problems with two or three objectives, but that is not the case with Many-Objective optimization problems. Recent studies [1, 4, 25] and have shown that MOPs face many difficulties when tackling problems involving a larger number of objectives.

In general, both MaOP and MOP Evolutionary Algorithms (EA) yield numerous Pareto optimal solutions. However, the Decision Maker (DM) is not interested in all Pareto optimal solutions but rather interested in a single optimum solution or a preferred region of the Pareto Front (PF) [6].

The optimization and the decision process are often combined in the literature [26], and therefore Multi-Objective optimization approaches can be generally classified into three types:

1. *Decision making before search:* is often described as a Priori technique where the objectives of the MOP are aggregated into a single objective which implicitly comprises the preference information given by the DM.
2. *Search before decision making:* is often described as a Posteriori technique. The optimization is performed without any preference information where the search process yields a set of (ideally Pareto-

---

Manuscript received on September 13, 2017 ; revised on January 7, 2018.

Final manuscript received on January 25, 2018.

[1,2] The authors are with Department of Computer Engineering King Mongkut's University of Technology Thonburi, Bangkok, Thailand., E-mail: almufdhali@gmail.com and naruemon.wat@kmutt.ac.th

optimal) solutions from which the DM must choose a solution.

3. *Decision making during search:* is often described as an Interactive technique where the DM can articulate preferences during the optimization process to guide the search. After each iteration or number of iterations, several Pareto optimal solutions are presented to the DM. Then, the DM further specifies more preference information to guide the search until the DM is satisfied.

Consequently, many methods have been proposed and used to deal with the decision making in the optimization problems. One of the most interesting methods is finding solutions in the "Knee" region. A Knee region can be described as the "bulge" in the Pareto front as illustrated in Fig.2 [3]. Knee solutions are defined as a subset of Pareto optimal solutions for which an improvement in one objective will result in a deterioration in at least another one. In general, the MOP solutions in the knee region of the Pareto front are logically preferred to the DM if there is no user-specific or problem specific preferences [2-3].

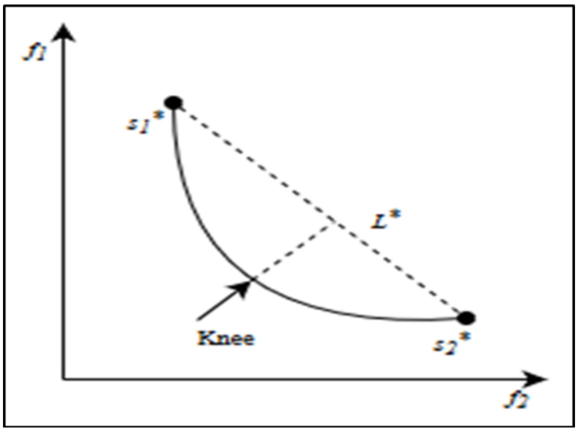

Fig.2: *Knee region [3].*

These issues have been and will continue to be a very active research area in the field of Evolutionary Computing and its applications. Therefore, the main goal of this paper in summary is to find knee solutions from the obtained Pareto optimal fronts by pruning Algorithm, called Angle-based with Specific bias parameter (ASA), and then evaluate and compare the the found knee solutions with the generational distance (GD) metric, one-way ANOVA statistical significance and the overall computation time for each algorithm. Also, different number of objectives, and number of knee regions were considered.

The rest of this paper is organized as follows: section 2 presents a literature review, a background study and a description of the Many-Objective evolutionary algorithms. Section 3 defines the research framework, the benchmark problems, the performance metrics and the parameter settings. Section 4 presents and discusses the obtained results. Finally, section 5 summarizes the paper and presents a conclusion and recommendations.

2. LITERATURE REVIEW AND BACKGROUND

The first implementation of a MOEA was in the mid-80s. Since then, several methods of MOEAs have been introduced in the literature and gradually been improved in both effectiveness and efficiency to solve MOPs. However, most of these algorithms have been applied to problems with only two or three objectives [1]. While many real-world problems have more than three objectives which may make some of these algorithms inefficient when dealing with Many-Objective optimization problems.

2.1 Multi-Objective Optimization Problems

A Multi-Objective optimization problem can be defined as follows:

Minimize $f(x) = [f_1(x), f_2(x) \dots f_k(x)]^T$
subject to $x \in D$ where
$f(x)$ a k-vector of objective functions
$x = (x_1, x_2 \dots x_n)$ decision vector
$X \subseteq R^n$ feasible decision space
$f(D)$ feasible objective space (solution space)
R^n might be restricted with constraints of the following types to form D:
$g(x) \geq 0$ inequality type constraints
$h(x) = 0$ equality type constraints

A solution x that satisfies all constraints and variable bounds is a **feasible solution**. Otherwise, it is an infeasible solution. Also, a **feasible space** is a set of all feasible solutions. The objective function $f(x)$ defines a multidimensional objective space [1-2].

2.2 Many-Objective Optimization Problems

MaOPs are defined as problems with four or more objectives where the resulting Pareto front cannot be visualized by conventional graphical means [4]. Deb and Jain [4] argued that the current state-of-the-art EMO algorithms that work under the concept of "dominance" face many difficulties. Deb et al. [4] summarized these difficulties as follows:

1. A large fraction of population is non-dominated
2. Evaluation of diversity measures are computationally expensive
3. Recombination operation may be inefficient
4. Representation and visualization of the Pareto fronts are difficult
5. Performance metrics are computationally expensive

These difficulties have been and will continue to be a very active research area in the field of Evolutionary Computing and its applications.

2.3 Pareto-optimality

In Multi-Objective optimization problems, the notion of "dominance" is used to determine if a solution is better than others. A solution x dominates solution y if: (i) x is no worse than y in all objectives and (ii) x is better than y in at least one objective. Specifically, y is said to be "dominated" by x, or alternatively, x is "non-dominated" by y. Therefore, the non-dominance relationship defines the Pareto optimality concept. This concept of dominance is exemplified in a two-objective minimization example shown in Fig.3.

Moreover, a solution is **Pareto optimal** if it is not dominated by any other solution. In other words, an improvement in one solution cannot be achieved without losing quality in another objective. Furthermore, the set of all Pareto-optimal solutions is called the **Pareto set (PS)**. Also, the objective values of the Pareto set in the objective space define the **Pareto front (PF)** [1-2].

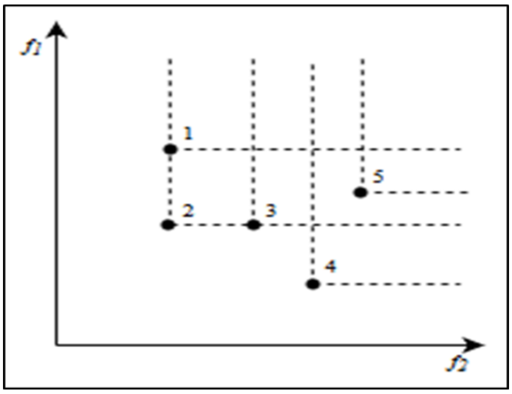

Fig.3: *Pareto dominance example [6].*

2.4 Knee Region

MOEAs in general yield numerous Pareto optimal solutions. However, in the real-world problems, the Decision Maker (DM) is not interested in the whole Pareto optimal solutions but rather interested in a single optimum solution or a preferred region of the Pareto Front (PF). Such preferred region is the "Knee" region. In practice, given the Pareto front for a MOP, the DM usually picks a point 'in the middle' of the front where the Pareto surface "bulges out the most" [3]. Das [3] took advantage of this realization and proposed a method based on the Normal-Boundary Intersection (NBI) to locate the knee of the PF. Das characterized the knee solutions in terms of the Convex Hull of Individual Minima (CHIM). Basically, the knee regions could be recognized by the furthest point from the CHIM. However, this technique requires a prior approximation of the extreme solutions for each objective. In Fig.2, the knee of the PF corresponds to the furthest point from the extreme line L^*. The extreme line is the line defined by the extreme solutions s_1^* and s_2^*.

2.5 Knee-based MOEAs

MOEAs that use the word "Knee" in them can be classified as Kneed-based algorithms or Knee-driven algorithms. Knee-based algorithms can be described as a decision-making tool such as in [2, 6]. Knee-driven algorithms can be described as an optimization search strategy such as in [8, 9, 25, 27]. This paper focuses only on the Knee-based algorithms.

NSGA-II is one of the most used and cited evolutionary algorithms for MOPs in the literature. Since Deb et al. [7] proposed NSGA-II in 2002, researchers continued to improve and propose variants of the algorithm such as [6, 8, 9]. NSGA-II extensive use was due to its low computational complexity, elitist approach and method of diversity [8].

Bechikh et al. [8] proposed a knee-based version of R-NSGA-II [11] which was a modified version of the NSGA-II that focused the search on the region of interest (**ROIs**) based on a user-provided reference points set. The reference points in KR-NSGA-II are chosen from the first non-dominated front automatically by an updating strategy in each generation. This strategy is called *Mobile Reference Points Updating Strategy (MRPUS)*.

Bechikh et al. [9] proposed a new version of KR-NSGA-II. KR-NSGA-II had a main disadvantage which was the need of considering the extreme solutions as mobile reference points for estimating the CHIM line [9]. Therefore, the algorithm results depended on the success of discovering the Pareto optimal extreme solutions, which was not an easy task when estimating the nadir point [9]. TKR-NSGA-II differs from the original KR-NSGA-II by modifying the MRPUS strategy to avoid stagnation in local optima. This modified strategy is called *Trade-off-based Mobile Reference Point Updating Strategy (T-MRPUS)*. T-MRPUS assigns a trade-off worth value to each solution from the best front FF instead of the distance as in the original MRPUS.

Sudeng and Wattanapongsakorn [6] proposed a knee-based version with a pruning strategy to the NSGA-II. The pruning strategy was called *Angle-based with Specific bias parameter pruning Algorithm (ASA)*. The k-ASA-NSGA-II basically consisted of two steps. First, it approximated a rough set of Pareto-optimal solutions using a NSGA-II algorithm. Then, the k-ASA algorithm was applied to eliminate non-preferred solutions.

In general, the k-ASA algorithm calculates an angle (θ) of each pair of solutions. Then, the angle (θ) is compared to a threshold angle (δ). If the angle (θ) between solutions i and j is smaller than the threshold (δ), solution i is kept and solution j is discarded. The threshold angle (δ) can be adjusted using a user defined parameter called τ which is used to give the

DM the ability to prioritize the objectives. The value of τ is between 0 and 1. The number of the knee solutions that can be found can be increased by decreasing the value of τ and decreased by increasing the value of τ.

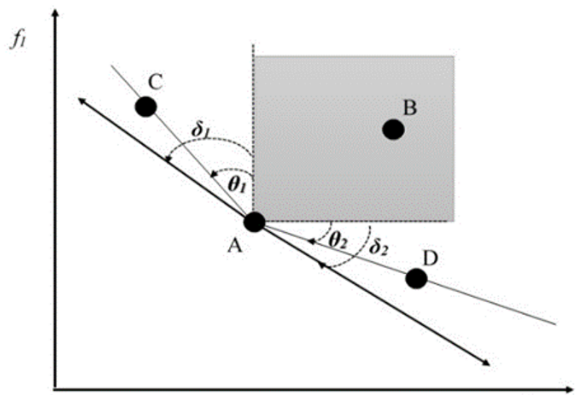

Fig.4: *ASA concept [6].*

The angle between two non-dominated solutions is calculated using the following equation. The geometric angle is denoted by θ_n where n is the nth objective. For the minimizing objective context, θ_n is given by:

$$\theta_n = \tan^{-1}\left[\frac{\sqrt{\sum_{m=1,m\neq n}^{N}(\Delta f_m)^2}}{\Delta f_n}\right]$$

where N denotes the number of objectives and n denotes the nth objective function. Δf_n denotes the difference between the nth objective values of the two non-dominated solutions.

The threshold angle δ_n of each objective value can be calculated as follows:

$$\delta_n = \left[\tan^{-1}\left(\frac{IQS_n}{AV_n}\right)\right]^{\tau}$$

where n is the objective number and τ is bias intensity of each objective, ranging from 0.0 - 1.0. The IQS_n is the inter-quartile range of sorted data of each objective and AV_n is the average distance of the nth objective value between two consecutive non-dominated solutions.

2.6 Knee-based MOEA/D-DE

Sudeng and Wattanapongsakorn [2] proposed a knee-based Multi-Objective Evolutionary Algorithm Based on Decomposition with Differential Evolution k-MOEA/D-DE. Generally, MOEA/D-DE used the concept of neighbourhood to maintain population diversity which consumed high computation effort to evaluate neighbourhood for every individual [2].

The k-MOEA/D-DE calculates only the portion of solutions located at a specific region and does not need neighborhood relationship. Thus, a lot of computing effort can be eliminated. Moreover, the number of function evaluations (FFs) of individuals is eliminated because the population size is decreased when the algorithm runs along the optimization process. The k-MOEA/D-DE algorithm works as follows: First, an initial population is generated randomly. Then, a reference point z is initialized using the Tchebychev method where the minimum value for each objective is identified. Then, each single objective function and population are evaluated. After that, the knee-center is identified. Next, the maximum distance between individuals and weight vectors is found. When the problem has more than one knee to discover, the *multi_knees* algorithm is invoked. The density of each knee region is controlled by the r parameter. Lastly, the reference point z is updated before repeating the loop.

2.7 NSGA-III

In 2014, Deb and Jain [4] proposed a new version of the famous NSGA denoted as NSGA-III. The main motivation of the new algorithm was to deal with Many-Objective optimization problems. The main principle behind NSGA-III was that, instead of searching the entire search space for Pareto-optimal solutions, multiple predefined targeted searches can be used. The basic framework of NSGA-III was like the original NSGA-II algorithm with changes in its selection operator. Also, the maintenance of diversity among population members in NSGA-III was aided by providing and adaptively updating several well-spread reference points. Instead of using a crowding distance strategy as in NSGA-II, NSGA-III applied five strategies as follows:

1− Classification of Population into Non-Dominated Levels
2− Determination of Reference Points on a Hyper-Plan
3− Adaptive Normalization of Population Members
4− Association Operation
5− Niche-Preservation Operation

3. RESEARCH METHODOLOGY

The framework of this research is basically first extending the ASA algorithm to work with MOEA/D-DE and NSGA-III, and then evaluating and comparing the selected MOEA algorithms using well-known benchmark problems. The used MOEAs are taken from the literature.

3.1 Benchmark Problems

Since the beginning of MOEAs, several benchmark problems were proposed to challenge MOEA capabilities of approximating the Pareto front. Two of the most cited benchmark problem suites were DTLZ and WFG suites. Deb et al. [12] and Huband et al. [13]

introduced the Scalable Multi-Objective Optimization Test Problems to challenge MOEAs to demonstrate their ability to solve Multi-Objective problems efficiently and effectively. DTLZ and WFG suites both distinguished themselves by their scalability. This feature has enabled the recent research on Many-Objective problems. Generally, seven test problems were included in each test suite [12, 13]. However, in this research few test problems are considered.

Problems WFG1 and WFG2 were taken from the WFG test suite. WFG1 was chosen because the Pareto front is continuous and has a convex shape. On the other hand, WFG2 has a discontinuous Pareto front with a convex shape. The DTLZ problems were modified based on Branke et al. in [14]. The modified DTLZ problems are henceforward denoted as DEB(M)DK_K, where M indicates the number of objectives and K indicates the number of knees. The DEB(M)DK-K problems were designed specifically to have a knee region in the Pareto front based on the parameter K, and were chosen for that feature. The number of the objectives for each problem varied as 3, 5, 10 and 15.

3.2 Performance Metrics

Generally, there are two goals in Multi-Objective optimization: 1) convergence to the true Pareto optimal front and 2) maintenance of diversity in solutions of the Pareto-optimal front. However, these two goals cannot be measured reasonably with one performance metric. Many performance metrics have been suggested in the literature [7].

The general performance measures for the Multi-Objective optimization algorithms can be classified into three main categories: convergence, coverage, and success metrics. The first class of the metrics measures the closeness of the solutions obtained to the true Pareto front, and the second class of the metrics defines how well the solutions obtained "cover" the range of each of the objectives.

However, not all the above-mentioned metrics can be used in this research. That is because the used MOEAs in this research yield a small number of solutions obtained from the knee region of the Pareto front. Therefore, only the generational distance (GD) can be used for measuring the convergence of the solution. Thus, the GD was used to measure the performance of the used MOEAs with addition of one-way ANOVA and computation time.

Simply, GD measures the distance of the obtained Pareto optimal solutions from a selected reference set of the true Pareto optimal front.

The One-way Analysis of Variance (ANOVA) is a procedure for testing if the means of K groups are equal, where $K > 2$. The One-way ANOVA is also called a single factor analysis of variance because there is only one independent variable or factor. Normally, in a one-way ANOVA, there is one measurement variable and one nominal variable (independent variable). The one-way ANOVA only determines if there is a statistical significant difference between the groups or not. It does not indicate which group differs from the rest if there is a significant difference among the groups. To find out which group or groups differ, a further analysis is required. Such analysis is called a post hoc test. There are many choices of post hoc tests to perform this analysis. In this research, Tukey's HSD (honest significant difference) test or simply Tukey test is used. The main idea of this test is to compare all possible pairs of means. Tukey test is based on a studentized range distribution (q) which is like the distribution of t from the t-test. The main advantage of Tukey test is that it performs all pairwise comparisons to determine which group or groups are significantly different from the rest.

3.3 Parameters Settings

A set of 10 runs was conducted on each test problem for each different number of objectives. Table 1 shows the parameter settings for the used MOEAs. The population size used for each problem was set the same to evaluate how each MOEA performed under the same conditions. The number of the knees and the number of the objectives for each problem were varied. Table 2 shows the number of generations for each algorithm. The number of the generations in NSGA-III differed from the other algorithms due to its high computation time. Finally, the parameter τ in the ASA algorithm was set to 0.9.

Table 1: *Parameter settings*

Problem	Number of Objectives M	Number of Knees *K*	Population size
WFG1	3	Not specified	92
	5		212
	10		276
	15		136
WFG2	3	Not specified	92
	5		212
	10		276
	15		136
DEB3DK	3	1, 2, 3	92
DEB5DK	5	1, 2, 3	212
DEB10DK	10	1, 2, 3	276
DEB20DK	15	1, 2, 3	136

The main source code used for this research was jMetal 5.2. The jMetal is an object-oriented Java-based framework for Multi-Objective optimization with metaheuristics [21]. jMetal stands for Meta-heuristic Algorithms in Java. The source code can be obtained from [22].

4. SIMULATION RESULTS AND DISCUSSION

A set of 10 runs was conducted on each algorithm for each benchmark problem. Then, the obtained

Table 2: *Number of generations for each algorithm*

Number of Objectives	Number of Generations		
M	NSGA-II	MOEA/D-DE	NSGA-III
3	25000	25000	1000
5	50000	50000	2000
10	100000	100000	3000
15	200000	200000	5000

Pareto front from each run was normalized to be processed by the ASA algorithm to find knee solutions. Those knee solutions were then restored to their original values. After the restoration, the GD values were calculated using the true optimal Pareto fronts. Finally, the average GD was calculated and recorded.

4.1 Experimental Results

The average GD values of the benchmark problems used are shown in Table 3. An algorithm is considered better than another when the GD value of the former is less than that of the latter.

Table 3: *Average GD results.*

Problem	M	Average GD		
		NSGA-II	MOEA/D-DE	NSGA-III
WFG1	3	2.84619E-01	4.88330E-01	**1.04097E-01**
	5	5.14531E-01	1.28926E+00	**9.05657E-02**
	10	1.80275E+00	1.69497E+00	**3.62081E-01**
	15	1.57569E+00	3.24465E+00	**6.36703E-01**
WFG2	3	1.24186E-01	2.85894E-01	**1.16036E-01**
	5	4.75537E-01	1.36280E+00	**2.25697E-01**
	10	1.35677E+00	1.16552E+00	**7.20268E-01**
	15	1.94309E+00	9.65733E-01	**5.16299E-01**
DEB(M)DK_1	3	**3.85781E+00**	7.91808E+00	4.17352E+00
	5	3.36091E+00	4.20733E+00	**2.17509E+00**
	10	2.29637E+00	1.64340E+00	**1.08051E+00**
	15	7.39517E+00	**3.94963E+00**	7.09684E+00
DEB(M)DK_2	3	**3.36609E+00**	3.86440E+00	4.30687E+00
	5	2.51543E+00	2.75106E+00	**1.30590E+00**
	10	6.59197E+00	1.33181E+01	**4.27732E+00**
	15	1.58030E+01	**4.20668E+00**	4.27510E+00
DEB(M)DK_3	3	3.74577E+00	**2.82260E+00**	3.57990E+00
	5	2.94950E+00	2.18837E+00	**1.47039E+00**
	10	4.64778E+00	4.34024E+00	**2.30016E+00**
	15	1.17285E+01	8.89294E+00	**2.63049E+00**

In general, results obtained by NSGA-III had the lowest or near lowest GD values and hence are considered the best in terms of the convergence metric. However, it could not be said the same in terms of the CPU time. NSGA-III is computationally expensive comparing to other algorithms as shown in Table 4. Also, the results showed that NSGA-III performed almost better when the objectives were more than 3 objectives. In other words, NSGA-III was better dealing with Many-Objective problems.

According to Table 4, MOEA/D-DE had the lowest computation complexity and therefore was the fastest running time. In contrast, NSGA-III had the slowest running time due to its computation complexity which was in the worst-case scenario has a complexity of $O(N^2 \log^{M-2} N)$ or $O(N^2 M)$, whichever is larger [4].

The one-way ANOVA analysis was conducted on the GD results to determine if there was a statistical significant difference between the results. The one-way ANOVA is a relatively robust procedure against violations of the normality assumption [23]. Furthermore, it is not very sensitive to heteroscedasticity

Table 4: *Average CPU time.*

Problem	M	Average CPU time (milliseconds)		
		NSGA-II	MOEA/D-DE	NSGA-III
WFG1	3	420.20	222.00	1804.10
	5	1491.10	556.70	17108.20
	10	5338.90	1632.90	59861.50
	15	9812.40	4015.50	41244.50
WFG2	3	318.70	137.60	1544.80
	5	1286.80	360.00	16341.10
	10	4618.30	1167.10	58645.70
	15	7498.10	2814.80	39252.30
	3	274.60	94.70	1245.00
	5	1122.60	271.90	15127.40
	10	4015.70	863.00	54090.90
	15	6189.50	1718.30	33011.50
DEB(M)DK_2	3	266.50	92.70	1238.20
	5	1118.80	269.40	15164.90
	10	4491.70	838.10	52013.60
	15	7007.60	1642.20	29920.80
DEB(M)DK_3	3	254.50	92.70	1326.90
	5	1115.00	269.80	14943.80
	10	4476.30	814.20	51494.60
	15	7077.70	1732.30	30344.40

when we have a balanced design, meaning that the number of the observations is the same in each group [23]. In this research, each group had 10 observations (or 10 GD values specifically). Therefore, it could be concluded that both assumptions were met in this research.

The null hypothesis of this test was that there was no significant difference between the means of the groups. Otherwise, it could be concluded that there was at least 1 difference between groups' means. The significant level was set to be 5%. The null hypothesis was rejected if:

Sig. $< p - value$ or F > F_{critical}

Also, a post hoc test was conducted in each case of the rejected null hypothesis. The post hoc test was conducted to find which algorithm performed better than the others. The post hoc test used here was Tukey's honestly significant difference (HSD) or simply Tukey test. The same null hypothesis was assumed here also. The null hypothesis was rejected if Sig. $< p - value$. If the null hypothesis was rejected in the Tukey test, a negative mean difference between the pairs indicated that the value of the first variable was lower than the second variable, and therefore the first variable was the best. Table 5 shows the F_{critical} value obtained from the F-distribution table based on the degree of freedom between groups and within groups.

Table 5: *Significance level and critical value.*

	df (degree of freedom)	F_{critical}	p-value
Between Groups	2	3.354	0.05
Within Groups	27	-	-
Total	29	-	-

Table 6 summarizes the one-way ANOVA test results. The test was conducted on each problem for each case of the number of the objectives. The null hypothesis was rejected in almost every case except for a 10 objective WFG2 problem and DEB3DK_2. In these two cases, the null hypothesis was accepted, and it could be concluded that there was no significant difference among the 3 algorithms.

In case of a rejected null hypothesis in the one-way

Table 6: *Summary of one-way ANOVA test.*

Problem	M	F	Sig.	Reject H_o
WFG1	3	86.863	1.731E-12	Yes
	5	65.033	4.746E-11	Yes
	10	28.595	2.150E-7	Yes
	15	69.938	2.095E-11	Yes
WFG2	3	16.214	2.369E-5	Yes
	5	217.921	2.188E-17	Yes
	10	3.084	0.062	**No**
	15	11.770	0.0002	Yes
DEB(M)DK_1	3	36.750	1.969E-8	Yes
	5	16.664	1.934E-5	Yes
	10	55.912	2.513E-10	Yes
	15	12.460	0.0001	Yes
DEB(M)DK_2	3	1.826	0.180	**No**
	5	18.368	9.208E-6	Yes
	10	39.600	9.347E-9	Yes
	15	29.122	1.818E-7	Yes
DEB(M)DK_3	3	6.459	0.005	Yes
	5	14.882	4.399E-5	Yes
	10	38.531	1.230E-8	Yes
	15	30.877	1.054E-7	Yes

ANOVA test, a post hoc test was performed for a pair of the algorithms to determine the difference between the algorithms (average of the results). Table 7 shows the Tukey test result of the rejected null hypothesis in Table 6.

4.2 Discussion

In terms of the GD performance metric, NSGA-III clearly performed better according to Table 8. A lower score (number between parentheses) meant a higher rank. For each benchmark problem, the score values were the sum of the ranks of each algorithm in each designated number of objectives. For example, NSGA-III in WFG1 problem ranked first in 3, 5, 10 and 15 numbers of objectives respectively. Obviously, NSGA-III had the lower scores in all simulation cases and therefore had the highest ranking. In problem WFG1, NSGA-III ranked first in each case of the number of objectives while MOEA/D-DE ranked last in all 4 cases.

In problem WFG2, NSGA-III ranked first in the cases of 3, 5 and 15 objectives and came in second after MOEA/D-DE in the case of 10 objectives. NSGA-III did not perform very well in the case of 10 objectives. This was perhaps due to the discontinuity of the Pareto front in WFG2.

In problem DEB(M)DK_1, NSGA-III ranked first in 5 and 10 objective cases while coming in second in the remaining cases. In general, all DEB(M)DK problems had complex Pareto fronts. This complexity in the Pareto fronts might cause a slight shortfall in the NSGA-III performance because of the reference points strategy used in NSGA-III to find uniformly distributed solutions. NSGA-II and MOEA/D-DE had the same score and therefore ranked second together.

In problem DEB(M)DK_2, NSGA-III ranked first in 5 and 10 objective cases while coming in third and second in 3 and 15 objective cases respectively. NSGA-III slightly underperformed in this problem, which might be due to the complexity of the Pareto front. MOEA/D-DE ranked last in this problem although it came first in the case of 15 objectives. The underperformance of MOEA/D-DE could be attributed to the weakness of the Tchebycheff approach.

Lastly, NSGA-III ranked first in DEB(M)DK_3 problem. It came first in 5, 10 and 15 objective cases and ranked second in 3 objective case. MOEA/D-DE ranked second in general and came first in 3 objective case.

The statistical one-way ANOVA test indicated that NSGA-III ranked first in all cases according to Table 9 and therefore was the best algorithm. NSGA-III ranked first in all cases of problems WFG1, WFG2 and DEB(M)DK_2.

NSGA-III ranked first in all cases of problem DEB(M)DK_1 except in 15 objective case. That was expected because MOEA/D-DE had the lower average GD values in this case. NSGA-III did not perform better than the MOEA/D-DE in this case.

In problem DEB(M)DK_3, NSGA-III ranked first in all cases except in 3 objective case. This was also expected because MOEA/D-DE performed better than NSGA-III based on the average GD results.

MOEA/D-DE and NSGA-II almost performed equally based on this statistical test. In general, it could be concluded that there was no significant difference between them.

5. CONCLUSION

In most real-world problems, the number of objectives is often higher than 3 objectives, and therefore the problems can be identified as a Many-Objective optimization problems (MaOP). Moreover, the objectives of these problems are often in conflict and need to be optimized simultaneously. While the current state-of-the-art Multi-Objective optimization evolutionary algorithms (MOEAs) can efficiently solve Multi-Objective problems, their performance deteriorates when dealing with Many-Objective ones. Additionally, the decision maker (DM) is not interested in all obtained trade-off solutions. Thus, an interesting concept for identifying solutions in the Knee region is used to find few solutions that are near optimum and are significant for the decision maker. Solutions in the Knee region can be considered the best obtained solution in the obtained trade-off set, especially if there is no preference or equally important objectives. In this research, a newly introduced and promising Multi-Objective optimization algorithm NSGA-III was used where the obtained Pareto-optimal front was pruned to find Knee solutions. Also, the same pruning technique was used on Pareto-optimal Pareto fronts obtained by NSGA-II and MOEA/D-DE.

In general, NSGA-III performed very well in terms of the generational distance metric (GD) in almost all problems. Also, MOEA/D-DE slightly performed better than NSGA-II. In general, it was evident from the average GD results that both MOEA-DE and NSGA-II struggled in dealing with Many-Objective optimization problems. Although NSGA-III outperformed the other algorithms in terms of the GD, it

Table 7: *Summary of Tukey post hoc test.*

Problem	M	Pair		Mean Difference	Sig.	Reject H_o
WFG1	3	NSGA-III	MOEA/D-DE	-3.842E-01	5.114E-9	Yes
		NSGA-III	NSGA-II	-1.805E-01	3.778E-6	Yes
		MOEA/D-DE	NSGA-II	2.037E-01	4.935E-7	Yes
	5	NSGA-III	MOEA/D-DE	-1.199E+00	5.145E-9	Yes
		NSGA-III	NSGA-II	-4.240E-01	0.001	Yes
		MOEA/D-DE	NSGA-II	7.747E-01	2.442E-7	Yes
	10	NSGA-III	MOEA/D-DE	-1.333E+00	2.976E-6	Yes
		NSGA-III	NSGA-II	-1.441E+00	8.073E-7	Yes
		MOEA/D-DE	NSGA-II	-1.078E-01	0.868	**No**
	15	NSGA-III	MOEA/D-DE	-2.608E+00	5.127E-9	Yes
		NSGA-III	NSGA-II	-9.390E-01	0.001	Yes
		MOEA/D-DE	NSGA-II	1.669E+00	1.492E-7	Yes
WFG2	3	NSGA-III	MOEA/D-DE	-1.699E-01	7.734E-5	Yes
		NSGA-III	NSGA-II	-8.150E-03	0.968	**No**
		MOEA/D-DE	NSGA-II	1.617E-01	0.0001	Yes
	5	NSGA-III	MOEA/D-DE	-1.137E+00	5.113E-9	Yes
		NSGA-III	NSGA-II	-2.498E-01	0.0005	Yes
		MOEA/D-DE	NSGA-II	8.873E-01	5.113E-9	Yes
	10	NSGA-III	MOEA/D-DE	No Post hoc test		
		NSGA-III	NSGA-II			
		MOEA/D-DE	NSGA-II			
	15	NSGA-III	MOEA/D-DE	-4.494E-01	0.3093	Yes
		NSGA-III	NSGA-II	-1.427E+00	0.0002	Yes
		MOEA/D-DE	NSGA-II	-9.774E-01	0.0084	Yes
DEB(*M*)DK_1	3	NSGA-III	MOEA/D-DE	-3.745E+00	3.631E-7	Yes
		NSGA-III	NSGA-II	3.157E-01	0.822	**No**
		MOEA/D-DE	NSGA-II	4.060E+00	8.616E-8	Yes
	5	NSGA-III	MOEA/D-DE	-2.032E+00	1.207E-5	Yes
		NSGA-III	NSGA-II	-1.186E+00	0.007	Yes
		MOEA/D-DE	NSGA-II	8.464E-01	0.060	**No**
	10	NSGA-III	MOEA/D-DE	-5.629E-01	0.0001	Yes
		NSGA-III	NSGA-II	-1.216E+00	5.240E-9	Yes
		MOEA/D-DE	NSGA-II	-6.530E-01	1.463E-5	Yes
	15	NSGA-III	MOEA/D-DE	3.147E+00	0.0009	Yes
		NSGA-III	NSGA-II	-2.983E-01	0.9198	**No**
		MOEA/D-DE	NSGA-II	-3.446E+00	0.0003	Yes
DEB(*M*)DK_2	3	NSGA-III	MOEA/D-DE	No Post hoc test		
		NSGA-III	NSGA-II			
		MOEA/D-DE	NSGA-II			
	5	NSGA-III	MOEA/D-DE	-1.445E+00	1.564E-5	Yes
		NSGA-III	NSGA-II	-1.210E+00	0.0002	Yes
		MOEA/D-DE	NSGA-II	2.356E-01	0.632	**No**
	10	NSGA-III	MOEA/D-DE	-9.041E+00	1.552E-8	Yes
		NSGA-III	NSGA-II	-2.315E+00	0.091	**No**
		MOEA/D-DE	NSGA-II	6.726E+00	2.342E-6	Yes
	15	NSGA-III	MOEA/D-DE	6.842E-02	0.999	**No**
		NSGA-III	NSGA-II	-1.153E+01	1.340E-6	Yes
		MOEA/D-DE	NSGA-II	-1.160E+01	1.212E-6	Yes
DEB(*M*)DK_3	3	NSGA-III	MOEA/D-DE	7.573E-01	0.027	Yes
		NSGA-III	NSGA-II	-1.659E-01	0.818	**No**
		MOEA/D-DE	NSGA-II	-9.232E-01	0.006	Yes
	5	NSGA-III	MOEA/D-DE	-7.180E-01	0.035	Yes
		NSGA-III	NSGA-II	-1.479E+00	2.616E-5	Yes
		MOEA/D-DE	NSGA-II	-7.611E-01	0.024	Yes
	10	NSGA-III	MOEA/D-DE	-2.040E+00	4.530E-7	Yes
		NSGA-III	NSGA-II	-2.348E+00	3.824E-8	Yes
		MOEA/D-DE	NSGA-II	-3.075E-01	0.548	**No**
	15	NSGA-III	MOEA/D-DE	-6.262E+00	4.105E-5	Yes
		NSGA-III	NSGA-II	-9.098E+00	9.170E-8	Yes
		MOEA/D-DE	NSGA-II	-2.836E+00	0.060	**No**

Table 8: *Ranking based on average GD values.*

Problem	Rank (Score)		
	NSGA-II	MOEA/D-DE	NSGA-III
WFG1	2 (9)	3 (11)	1 (4)
WFG2	2 (10)	2 (10)	1 (4)
DEB(M)DK_1	2 (9)	2 (9)	1 (6)
DEB(M)DK_2	3 (9)	2 (8)	1 (7)
DEB(M)DK_3	3 (12)	2 (7)	1 (5)

Table 9: *Ranking based on average GD values.*

Problem	Rank (Score)		
	NSGA-II	MOEA/D-DE	NSGA-III
WFG1	2 (8)	3 (11)	1 (4)
WFG2	2 (7)	3 (8)	1 (4)
DEB(M)DK_1	3 (8)	2 (7)	1 (5)
DEB(M)DK_2	2 (6)	2 (6)	1 (4)
DEB(M)DK_3	3 (11)	2 (7)	1 (5)

was the most computationally expensive algorithm in all cases due to its computational complexity. In contrast, MOEA/D-DE was the fastest algorithm due to its decomposition strategy. This decomposition strategy significantly reduced the computation complexity and therefore lowered the CPU time.

In dealing with Many-Objective optimization problems, NSGA-III is clearly the better choice if the computation time is acceptable. A machine with high processing power is recommended to be used when running NSGA-III especially when the number of objectives is higher than 5. On the other hand, MEAD/D-DE is the second choice when the processing power is low or when the results are needed very fast.

The population sizes in this research were restricted by the number of reference points used in NSGA-III. For any other algorithm, a bigger population size and a higher number of generations are recommended to have a better chance of finding a good near-optimum solution.

References

[1] A. L. Jaimes and C. A. C. Coello, 2015, "Many-Objective Problems: Challenges and Methods," *Springer Handbook of Computational Intelligence*, Springer, pp. 1033-1046.

[2] S. Sudeng and N. Wattanapongsakorn, 2016, "A Decomposition-based Approach for Knee Solution Approximation in Multi-Objective Optimization," *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, July 24 - 29, 2016.

[3] I. Das, 1999, "On Characterizing the 'Knee' of the Pareto Curve Based on Normal-Boundary Intersection," *Structural Optimization*, Vol. 18, No. 2-3, pp. 107-115.

[4] K. Deb and H. Jain, 2014, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-point-based Nondominated Sorting Approach, Part I: Solving Problems with Box Constraints," *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 4, pp. 577-601.

[5] Q. Zhang and H. Li, 2007, "MOEA/D: A Multi-Objective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 6, pp. 712-731.

[6] S. Sudeng and N. Wattanapongsakorn, 2015, "A Knee-based Multi-Objective Evolutionary Algorithm: An Extension to Network System Optimization Design Problem," *Cluster Computing*, Vol. 19, No. 1, pp. 411-425.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, 2002, "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182-197.

[8] S. Bechikh, L. Ben Said, and K. Ghdira, 2010, "Searching for Knee Regions in Multi-Objective Optimization Using Mobile Reference Points," *Proceedings of the 2010 ACM Symposium on Applied Computing*, Sierre, Switzerland, March 22 - 26, 2010, pp. 1118-1125.

[9] S. Bechikh, L. B. Said, and K. Ghdira, 2011, "Searching for Knee Regions of the Pareto Front Using Mobile Reference Points," *Soft Computing*, Vol. 15, No. 9, pp. 1807-1823.

[10] H. Li and Q. Zhang, 2009, "Multi-Objective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 2, pp. 284-302.

[11] K. Deb and J. Sundar, 2006, "Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms," *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, Seattle, WA, USA, July 08 - 12, 2006, pp. 635-642.

[12] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, 2002, "Scalable Multi-Objective Optimization Test Problems," *2002 IEEE Congress on Evolutionary Computation (CEC 2002)*, Honolulu, HI, USA, May 12-17, 2002, pp. 825-830.

[13] S. Huband, P. Hingston, L. Barone, and L. While, 2006, "A Review of Multi-Objective Test Problems and A Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 5, pp. 477-506.

[14] J. Branke, K. Deb, H. Dierolf, and M. Osswald, 2004, "Finding Knees in Multi-Objective Optimization," *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature - PPSN VIII*, Birmingham, UK, September 18-22, 2004, pp. 722-731.

[15] L. Bradstreet, L. Barone, L. While, S. Huband, and P. Hingston, 2007, "Use of the WFG Toolkit and PISA for Comparison of MOEAs," *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, Honolulu, HI,

USA, April 01 - 05, 2007, pp. 382-389.

[16] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, Eds., 2007, "MOEA Test Suites," *in Evolutionary Algorithms for Solving Multi-Objective Problems: Second Edition*, Boston, MA: Springer US, pp. 175-232.

[17] S. Mirjalili and A. Lewis, 2015, "Novel Performance Metrics for Robust Multi-Objective Optimization Algorithms," *Swarm and Evolutionary Computation*, Vol. 21, pp. 1-23.

[18] J. J. Durillo and A. J. Nebro, 2011, "jMetal: A Java Framework for Multi-Objective Optimization," *Advances in Engineering Software*, Vol. 42, No. 10, pp. 760-771, Oct. 2011.

[19] H. Seada and K. Deb, 2015, "U-NSGA-III: A Unified Evolutionary Algorithm for Single, Multiple, and Many-Objective Optimization," *COIN Report*, No. 2014022.

[20] K. Li, *Research Projects* [Online], Available: `http://www.cs.bham.ac.uk/~likw/projects.html.[22-Jun-2017].`

[21] A. J. Nebro, J. J. Durillo, and M. Vergne, "Redesigning the jMetal Multi-Objective optimization framework," *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, Madrid, Spain, July 11 - 15, 2015, pp. 1093-1100.

[22] A. J. Nebro, J. J. Durillo, and M. Vergne, *jMetal 5 Web site* [Online]. Available: `https://jmetal.github.io/jMetal/.[07-Aug-2017].`

[23] J. H. McDonald, "One-way anova," *Handbook of Biological Statistics.* [Online]. Available: `http://www.biostathandbook.com/onewayanova.html.[07-Aug-2017].`

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

[25] J. Maltese, B. M. Ombuki-Berman, and A. P. Engelbrecht, "Pareto-based many-objective optimization using knee points," *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3678-3686. November 2016.

[26] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32-49, 2011.

[27] X. Zhang, Y. Tian and Y. Jin, "A Knee Point-Driven Evolutionary Algorithm for Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761-776, December 2015.

Saad M. Alzahrani received his Bachelor of Science degree in Mechanical Engineering from King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia, and is currently a master degree student at the Department of Computer Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi (KMUTT), Thailand. His research interests include Multi-Objective and Many-Objective optimization and Evolutionary Algorithms.

Naruemon Wattanapongsakorn received her Ph.D. in Computer Engineering from University of Pittsburgh, Pennsylvania, USA in 2000. She received her M.S. and B.S. in Computer Engineering from The George Washington University, Washington D.C., USA. Currently, she is an Associate Professor and Ph.D. Program Chair at Computer Engineering Department, King Mongkut's University of Technology Thonburi, Bangkok, Thailand. Her research areas are multi-objective optimization, computational intelligence, and network security. Dr. Wattanapongsakorn serves as editorial board and reviewer for several outstanding international journals, steering committee and technical program committee for many IEEE international conferences.