

Fusion of Many-Objective Non-dominated Solutions Using Reference Points

Amin Ibrahim¹(✉), Shahryar Rahnamayan¹, Miguel Vargas Martin²,
and Kalyanmoy Deb³

¹ Faculty of Electrical Computer and Software Engineering,
University of Ontario Institute of Technology, Oshawa, Canada
{amin.ibrahim, shahryar.rahnamayan}@uoit.ca

² Faculty of Business and Information Technology,
University of Ontario Institute of Technology, Oshawa, Canada
miguel.vargasmartin@uoit.ca

³ Department of Electrical and Computer Engineering,
Michigan State University,
East Lansing, MI, USA
kdeb@egr.msu.edu

Abstract. With recent advancements of multi- or many-objective optimization algorithms, researchers and decision-makers are increasingly faced with the dilemma of choosing the best algorithm to solve their problems. In this paper, we propose a simple hybridization of population-based multi- or many-objective optimization algorithms called fusion of non-dominated fronts using reference points (FNFR) to gain combined benefits of several algorithms. FNFR combines solutions from multiple optimization algorithms during or after several runs and extracts well-distributed solutions from a large set of non-dominated solutions using predefined structured reference points or user-defined reference points. The proposed FNFR is applied to non-dominated solutions obtained by the Generalized Differential Evolution Generation 3 (GDE3), Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO), and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) on seven unconstrained many-objective test problems with three to ten objectives. Experimental results show FNFR is an effective way for combining and extracting (fusion) of well-distributed non-dominated solutions among a large set of solutions. In fact, the proposed method is a solution-level hybridization approach. FNFR showed promising results when selecting well-distributed solutions around a specific region of interest.

1 Introduction

For the last three decades, there has been a number multi- and many-objective algorithms capable of solving complex problems. However, despite recent advancements, researchers and decision-makers are increasingly faced with the difficulty of choosing an appropriate algorithm capable of solving their problem effectively. This is due to a well-established “no-free-lunch” theorem that an algorithm that may have proven to give good performance on a

particular class of problems may not provide the same level of performance on other classes of problems. As a result many researchers have shifted their focus in developing powerful problem-specific or instance-specific algorithms. One way to accomplish this task is the hybridizations of optimization algorithms where new algorithms are developed by combining one optimization algorithm with another, by combining a standard optimization algorithm with mathematical operators, or incorporating evolutionary operators (selection, mutation, and crossover) into non-evolutionary optimization algorithms [1]. The main hope is that hybridization combines the desirable properties of different approaches so that the hybrid algorithm exhibits improved exploration and exploitation capabilities.

For example, Mirjalili and Hashim [2] proposed a hybrid population-based algorithm called PSO-GSA by combining Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA). Their main aim was to integrate the exploitation ability of PSO with the exploration ability of GSA to synthesize both algorithms' strength. Similarly El-hossini et al. [3] proposed three hybrid algorithms based on the PSO and the strength Pareto evolutionary algorithm 2 (SPEA2) to solve multi-objective optimization problems. In all of these algorithms, strength Pareto fitness assignment is used to maintain an external archive; and the three algorithms are developed by alternating the evolutionary and PSO processes in different order. Experimental results showed that the proposed hybrid PSO algorithms have comparable performance to SPEA2. Also, Tang and Wang [4] proposed a novel hybrid multi-objective evolutionary algorithm (HMOEA) for real-valued multi-objective problems by incorporating the concepts of personal best and global best in PSO and evolutionary operators (multiple crossovers) to improve the robustness of evolutionary algorithms to solve variant kinds of optimization problems.

Wang et al. [5] proposed a hybrid evolutionary algorithm based on different crossover and mutation strategies combined with adaptive constrained-handling technique to deal with numerical and engineering constrained optimization problems. Zăvoianu et al. [6] proposed a hybrid and adaptive co-evolutionary optimization method that can efficiently solve a wide range of multi-objective optimization problems. Their approach combines Pareto-based selection for survival, differential evolution's crossover and mutation operators, and decomposition-based strategies. Recently, an ensemble strategy was proposed to benefit from both the availability of diverse approaches and to overcome the difficulty of fine tuning associated parameters. Some of these work include ensemble of ϵ parameter values and an ensemble of external archives in a multi-objective PSO algorithm [7], ensemble of constraint handling methods to tackle constrained multi-objective optimization problems [8], and ensemble of different neighborhood sizes in multi-objective evolutionary algorithm based on decomposition (MOEA/D) with online self-adaptation [9].

Tan et al. [10] proposed a hybrid multi-objective evolutionary algorithm (HMOEA) featured with specialized genetic operators, variable-length representation and local search heuristic to find the Pareto optimal routing solutions for the truck and trailer vehicle routing problem (TTVRP). Experimental results showed the HMOEA is effective in solving a multi-objective and multi-modal combinatorial optimization problems. Similarly, Xia and Wu [11] proposed a hybrid multi-objective algorithm by combining the PSO algorithm for its explorative power and simulated annealing (SA) for its exploitations to solve flexible job-shop scheduling problem (FJSP). Tavakkoli-Moghaddam et al. [12] also proposed a hybrid

multi-objective algorithm based on the features of a biological immune system (IS) and bacterial optimization (BO) to find Pareto optimal solutions for flow shop scheduling problem. Their algorithm uses the clonal selection principle in IS with highest affinity antibodies and criterion distinguishing between antigens and antibodies in BO for Pareto dominance relationship among solutions. Karthikeyan et al. [13] proposed a hybrid discrete firefly algorithm (HDFFA) to solve the multi-objective FJSP problem. They have combined the discrete firefly algorithm with local search (LS) method to enhance the searching accuracy and information sharing among fireflies.

In the effort of developing a powerful general-purpose hybridization framework, propose a novel hybridization of population-based multi- or many-objective optimization algorithms called fusion of non-dominated fronts using reference points (FNFR), to gain combined benefit of several algorithms in solution-level. The FNFR method uses well-structured or user supplied reference points to extract targeted solutions from non-dominated solutions gathered during several runs of any multi-objective optimization algorithms (MOOA).

The rest of the paper is organized as follows. Section 2 outlines the FNFR framework in detail. Section 3 presents experimental studies conducted to verify the efficacy the proposed algorithm on 3- to 10-objective benchmark test problems. Concluding remarks are provided in Sect. 4.

2 Proposed Method: Fusion of Non-dominated Solutions Using Reference Points

Generally speaking, hybridization of optimization algorithms can be grouped into two main components. The first includes hybridization of algorithms during the optimization process and the second being the hybridization of the algorithms after the optimization process. Hybridization during the optimization process can also further be grouped into two main categories: algorithms created by combining multiple metaheuristics and algorithms created by combining a metaheuristic algorithm with multiple mathematical operators and/or evolutionary operators.

In this section, we present a novel hybridization technique called Fusion of Non-dominated Fronts using Reference points (FNFR) capable of extracting targeted solutions from a set of non-dominated results collected during several runs of multiple optimization algorithms. The skeleton for the FNFR hybridization procedure is shown as a flowchart in Fig. 1. The FNFR procedure consists of three modules:

Problem Formulation. In this module, the problem to be optimized is formulated along with the determination of various applicable algorithms that maybe included in the optimization process.

Approximate Front Evaluation. In this module, the selection process of applicable algorithms are narrowed to the most relevant set for solving the problem. At this point the number of iterations to run for each algorithm is determined by the user, with the foresight of generating enough solutions for a meaningful data set. Although any algorithm can be selected and combined together in this framework, we recommend

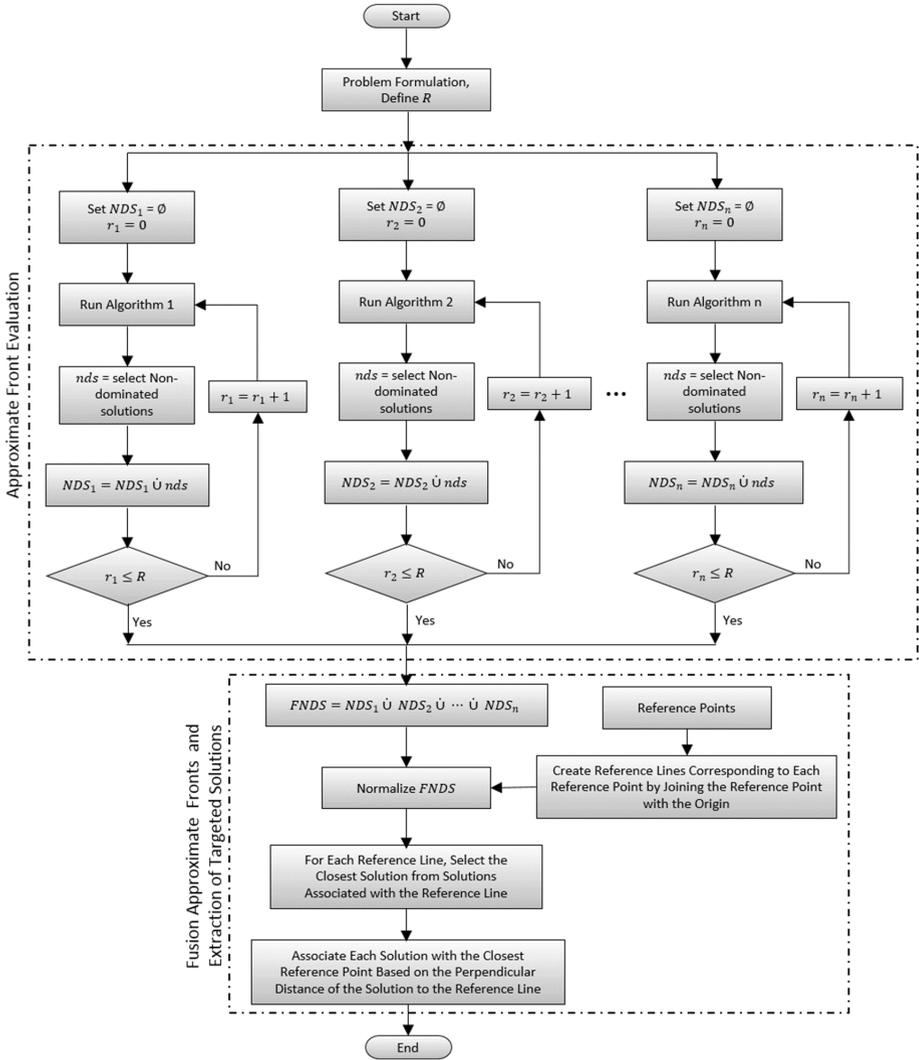


Fig. 1. Flowchart illustrating the FRFR procedure, where \cup indicate the extraction of non-dominated solutions from the union of two non-dominated solution sets.

combining algorithms complimentary to each other, that is, algorithms addressing different constructs to achieve a good representative solutions set on the entire Pareto-optimal front (PF). Note that each algorithm is run independently with the option of running each algorithm in parallel or serial. In every run, the found approximate non-dominated front is combined with previously obtained front to create new non-dominated front by removing dominated solutions from the combined front. This step is a crucial step, because we need a well-represented set of solutions on the entire PF before extracting preferred solutions from this set.

Fusion and Extraction of Targeted Solutions. In this module, targeted solutions are extracted from the union of non-dominated solutions obtained from several runs of multiple algorithms using a procedure similar to reference-point-based non-dominated sorting algorithm (NSGA-III) [14]. Given a union of M -objective non-dominated approximate front obtained from the algorithms, the first step is to combine these results and remove dominated solutions ($FNDS = NDS_1 \dot{\cup} NDS_2 \dot{\cup} \dots \dot{\cup} NDS_n$ where $\dot{\cup}$ indicate the extraction of non-dominated solutions from the union of two non-dominated solution sets). In the next step, a structured or target predefined set of reference points are selected on a normalized hyper-plane. In the case of structured reference points, one can use the Das and Dennis' [15] procedure to create well-distributed structured reference points. However, if the reference points are targeted points selected by the decision-maker, then M extreme points, one at each objective axis $\{z_1 = (1, 0, 0, \dots, 0), z_2 = (0, 1, 0, \dots, 0), \dots, z_M = (0, 0, \dots, 1)\}$, are needed to determine the right normalized hyper-plane. Thereafter, we construct an ideal point by finding the minimum value for each objective function from the fusion of non-dominated solutions extracted in the previous step ($FNDS$). Then, we translate each solution in $FNDS$ by the ideal point and find M extreme points from these translated solutions. These extreme points are used to construct M -dimensional linear hyper-plane. Special care is required when finding extreme points so that we construct the proper hyper-plane and be able to extract targeted solutions. We recommend verifying obtained extreme points by finding the individual optimum of each objective function to ensure an appropriate hyper-plane is constructed before moving to the next step. Thereafter, each solution is associated with a reference point by using the shortest perpendicular distance (d) of each solution with a reference line created by joining the reference point with the origin. Finally, for each reference point, we select a solution with the minimum d among solutions associated to each reference line.

The main advantages of the FNFR framework are: (1) we get the full benefit of all algorithms involved in the optimization process, (2) we don't need to investigate how and when to combine merits of different algorithms, (3) many algorithms can be used without the need of extra parameter tuning, and (4) we can run all algorithms in the optimization process in parallel. However, the FNFR framework requires a great amount of time commitment to run all algorithms in the optimization process (i.e., higher time complexity). As result a trade-off occurs from the onset when selecting the number of algorithms to use for the FNFR framework.

3 Experimental Setup and Results

In this section, we describe the optimization algorithms, parameter settings, used test problems, and simulation results of the proposed framework on 3- to 10-objective benchmark test problems.

3.1 Algorithms

In order to assess the capability of the proposed FNFR hybridization framework, we have considered three MOOAs that have considerable differences in their fitness assignment and diversity mechanism to gain combined benefits and achieve a good representative solutions set on the entire PF. These MOOAs are: the Generalized Differential Evolution Generation 3 (GDE3), Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO), and the Strength Pareto Evolutionary Algorithm 2 (SPEA2).

Generalized Differential Evolution Generation 3 (GDE3). GDE3 [16] is an extension of Differential Evolution (DE) for global optimization with an arbitrary number of objectives and constraints (Kukkonen & Lampinen, [16]). GDE3 with a single objective and without constraint is similar to the original DE. GDE3 improves earlier GDE versions in the case of multi-objective problems by giving better distributed solutions. GDE3 uses a growing population and non-dominated sorting with pruning of non-dominated solutions to decrease the population size at the end of each generation. This improves obtained diversity and makes the method more stable for the selection of control parameter values.

Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO). SMPSO [17] is similar to Particle Swarm Optimization (PSO) algorithm which inspired by the social behavior of birds flocking to find food. In PSO, particles move in the search space in a cooperative manner where movements are performed by the velocity operator. The velocity operator is guided by a local and a social behaviour of swarm. SMPSO uses the concept of crowding distance to filter out leader solutions when the leaders archive is full, mutation operator accelerates the convergence of the swarm and it incorporates a mechanism to limit the velocity of the particles which can result an erratic movements of particles towards the upper and lower position limits of particles.

Strength Pareto Evolutionary Algorithm 2 (SPEA2). SPEA2 [18] is an extension SPEA for solving multi-objective optimization problems. Both SPEA and SPEA2 use an external archive to store previously found non-dominated solutions. In SPEA, the external archive maintained based on each individual's strength in which the strength of an individual is measured according to the number of solutions this individual dominates. In every generation the fitness of each member of the current population is computed according to the strengths (closeness to the true PF and distribution of solutions) of all external non-dominated solutions that dominate it. On the other hand, in SPEA2, the external archive is maintained according to each individual's strength not only by the number of individuals that dominate it but also the number of individuals by which it is dominated. Moreover, SPEA2 uses a nearest neighbor density estimation method to guide the search process efficiently and it preserves boundary solutions.

3.2 Test Problems

In order to test the quality of the proposed algorithm, we have used seven many-objective benchmark test problems. The first set of test problems are the DTLZ (DTLZ1 – DTLZ4,

Convex DTLZ2) introduced by Deb et al. [19]. The number of variables are $(M + k - 1)$, where M is the number of objectives and $k = 5$ for DTLZ1, while $k = 10$ for DTLZ2, Convex DTLZ2, DTLZ3, and DTLZ4. The corresponding PFs lie in $f_i \in [0, 0.5]$ for the DTLZ1 problem and in $f_i \in [0, 1]$ for other DTLZ problems. The DTLZ1 problem has a linear PF, Convex DTLZ2 has convex PF, and DTLZ2 to DTLZ4 problems have concave PFs.

The second set of test problems utilized in this study are WFG1 and WFG2 test problems introduced by Huband et al. [20]. The number of position parameters is set to $k = M - 1$, and the number of distance parameters is set to $l = 3$ for all dimensions. The WFG1 has a mixed PF and WFG2 problem has a convex, disconnected PF. The PFs for WFG test problems used in this work lie in $f_i \in [0, 2i]$.

3.3 Parameters Setting

Here, we describe the parameter setting used in the three sample algorithms used in the FNFR method. The GDE3 algorithm has two control parameters: mutation ($F = 0.5$) and crossover ($CR = 0.1$) probabilities. The SMPSO algorithm has three control parameters: external archive size (same as population size), Polynomial mutation ($p_m = 1/n$, where n is the number of variables) and Mutation Distribution Index ($\eta_m = 20$). The SPEA2 algorithm has five control parameters: external archive size (same as population size), SBX probability ($p_c = 0.9$), Polynomial mutation ($p_m = 1/n$, where n is the number of variables), Crossover Distribution Index ($\eta_c = 20$), and Mutation Distribution Index ($\eta_m = 20$). In order to maintain a consistent and fair comparison, the optimal parameter settings reported in [16–18] are used. Table 1 presents the number of reference points (H), the population size (N), and the number of inner and outer divisions used for different dimensions of test problems. In this study we have utilized the Das and Dennis' [15] procedure to create structured reference points used in the FNFR method.

Table 1. Number of reference points and population sizes used in this study.

Number of objectives (M)	Divisions		Reference points (H)	Population size (N)
	Outer	Inner		
3	12	0	91	92
5	6	0	210	212
8	3	2	156	156
10	3	2	275	276

To evaluate the performance of the proposed fusion technique, first we have run each algorithm 20 times independently and the best, the worst, and the median results of each algorithm are recorded. For the performance measure, we have used the inverse generational distance (IGD) metric, which is capable of measuring the convergence and the diversity of the obtained Pareto-optimal solutions. The IGD measure has been predominantly used to evaluate the performance of evolutionary many-objective problems [21, 22]. In this study, the reference Pareto front utilized in the IGD metric is constructed by joining all results from all runs, and then selecting non-dominated solutions from this set.

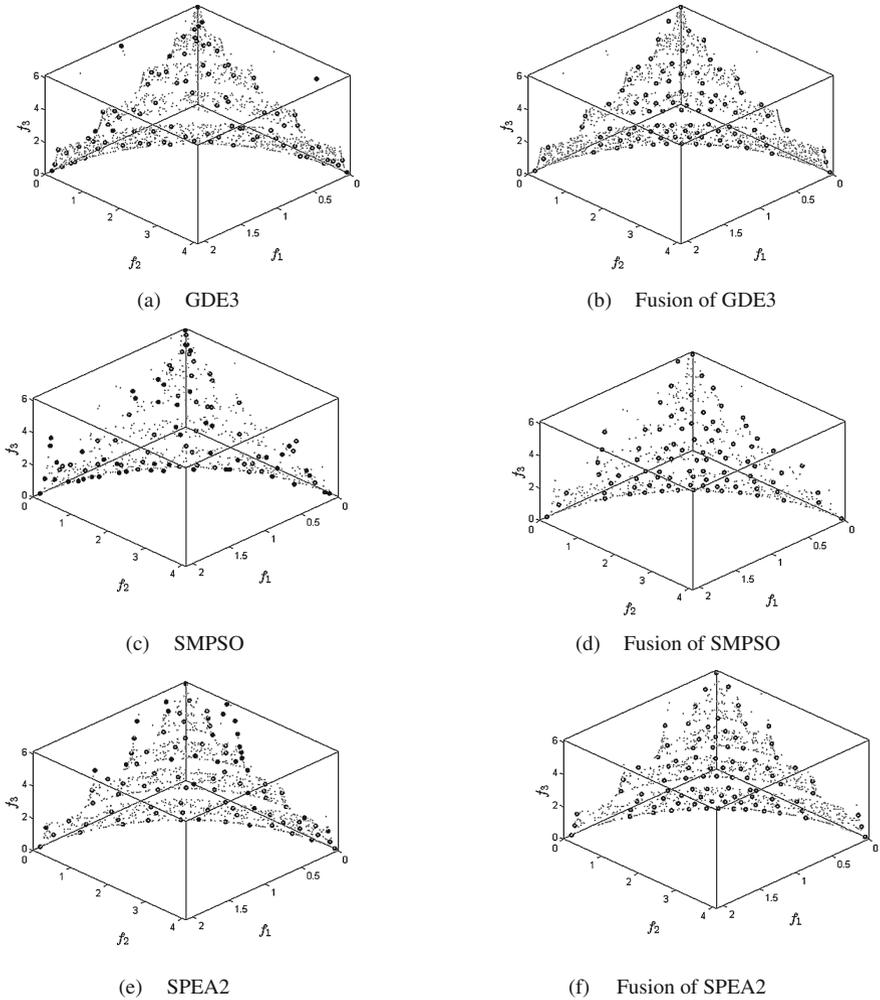


Fig. 2. The trade-off plots of obtained solutions by the GDE3, SMPSO and SPEA2 algorithms for three-objective WFG1 test problem. The grey dotted background indicated the non-dominated solution assembled during 20 runs of the GDE3 ((a) and (b)), SMPSO ((c) and (d)), and SPEA2 ((e) and (f)). The black dots in (a), (c), and (e) indicate the best solution set obtained by each algorithm based on the IGD metric. The black dots in (b), (d), and (f) show well-distributed solutions extracted from the non-dominated solutions assembled in 20 runs of each algorithm.

3.4 Experimental Results and Discussion

In this section, we describe experiments carried out to investigate the effectiveness of the FNFR method. Overall we have conducted three sets of experiments. The first experiment compares the quality of solutions obtained by each algorithm and the FNFR scheme using scatter plots (for three-objective problems) and 3D-RadVis [23] (for many-objective problems, $M > 3$). The second set of experiments involve numerical analysis to evaluate the

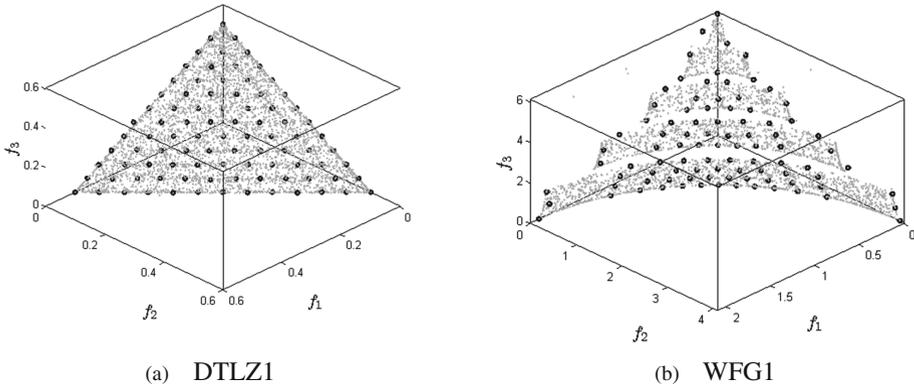


Fig. 3. The trade-off plots of solutions obtained by all algorithms for three-objective DTLZ1 and WFG1 test problems. The grey dotted background indicate non-dominated solutions assembled during 20 runs of the GDE3, SMPSO, and SPEA2 algorithms and the black dots illustrate well-distributed solutions extracted from these solutions using the proposed FNFR scheme.

quality of solutions obtained by each algorithm and the FNFR method. The last set of experiments involve extracting preferred solutions around a specific region.

Visual Analysis of Solutions obtained by each Algorithm against the FNFR Method.

Here we investigate solutions obtained by the three algorithms and solutions extracted by the FNFR method. We run each algorithm 20 times and assembled the non-dominated fronts from each run by removing dominated solution after combining non-dominated solutions from each run. Figure 2 show trade-off plots of solutions obtained by the GDE3, SMPSO, and SPEA2 for 3-objective WFG1 test problem. Figure 3 illustrates trade-off plots of solutions collectively obtained by the GDE3, SMPSO, and SPEA2 and solutions extracted by the FNFR procedure for 3-objective DTLZ1 and WFG1 and test problems.

The black dots in Fig. 2(a), (c), and (e) plots illustrate best solutions (based on the IGD metric) found by the GDE, SMPSO, and SPEA2 in 20 runs for 3-objective WFG1 test problem. From these plots we can see that none of the algorithms are able to find well-converged and well-distributed set of solutions in a single run. However, when we consider all non-dominated solutions collected during 20 runs (grey dots) by each algorithm, we see that they are able to find “satisfactory” solutions on the PF. This analysis indicate that the algorithms used in this experiment cannot consistently find well-distributed and well-converged solutions a single run. This phenomenon is in agreement with the no-free-lunch theorem.

The black dots in Fig. 2(b), (d), and (f) plots show well-distributed and well-converged solution extracted using the FNFR method from the non-dominated solutions gathered by each algorithm during the 20 runs of 3-objective WFG1 test problem. It can be seen that the FNFR method is an effective way of collecting and extracting well-distributed solutions after the optimization process is complete. We also observe that solutions extracted by the FNFR method are better than the best solution set found by any of the algorithms in a single run. Moreover, Fig. 3 shows trade-off plots of non-dominated solutions obtained by

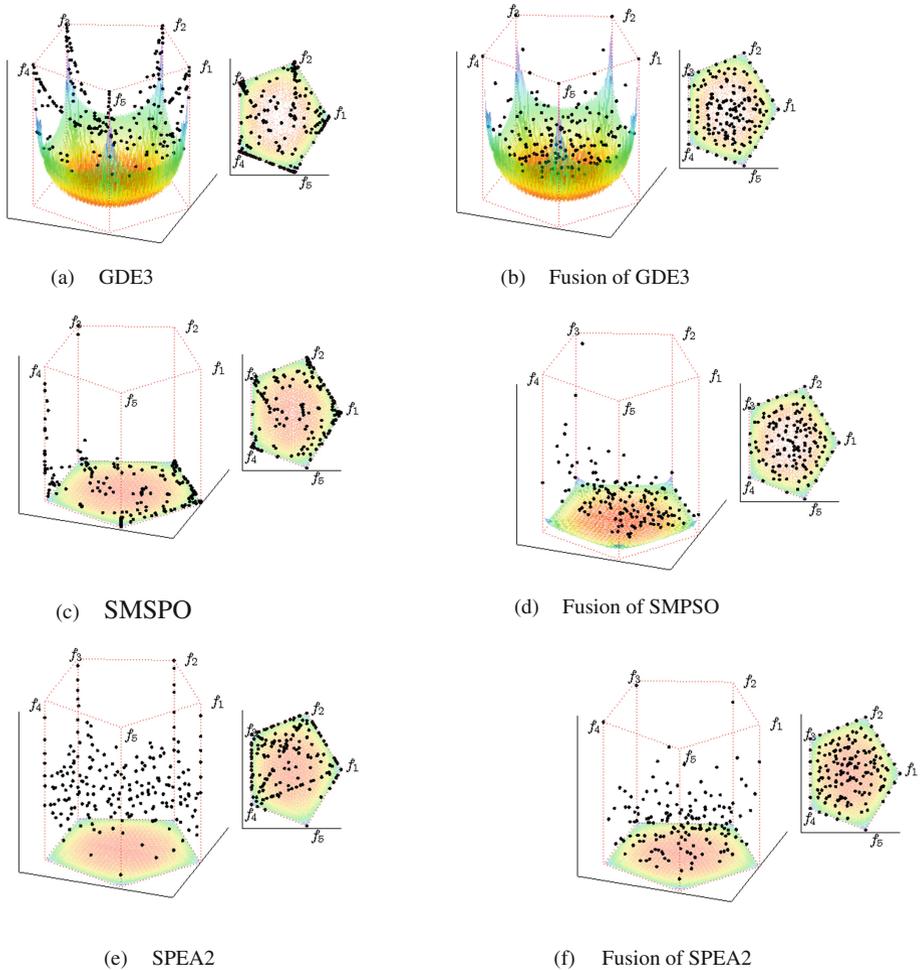


Fig. 4. 3D-RadVis plots of obtained solutions by GDE3, SMPSO and SPEA2 algorithms for five-objective convex DTLZ2 test problem. The surface for all plots are constructed from numerically generated PF. The black dots in (a), (c), and (e) indicate the best solution set obtained by each algorithm based on the IGD metric. The black dots in (b), (d), and (f) show well-distributed solutions extracted from the non-dominated solutions assembled in 20 runs of all algorithms using the FNFR method.

all algorithms for three-objective DTLZ1 and WFG1 test problems. The grey dotted background indicate the non-dominated solutions collected during 20 runs of GDE3, SMPSO, and SPEA2 algorithms and the black dots illustrate well-distributed solutions extracted from these solutions using the FNFR method.

The 3D-RadVis plots in Fig. 4 show the best non-dominated solutions obtained by GDE3, SMPSO, and SPEA2 algorithms (left plots) and solutions extracted using the FNFR method (right plots) for 5-objective convex DTLZ2 test problem. As it can be seen that as

the number of objective increases the quality of solutions found by each algorithm start to degrade in terms of solution diversity and accuracy. However, in Figs. 3 and 5, we see that these algorithms collectively are able to find well-distributed solutions on the entire PF and we are able to extract uniformly distributed non-dominated solutions using the FNFR method among solutions gathered during 20 runs of multiple algorithms.

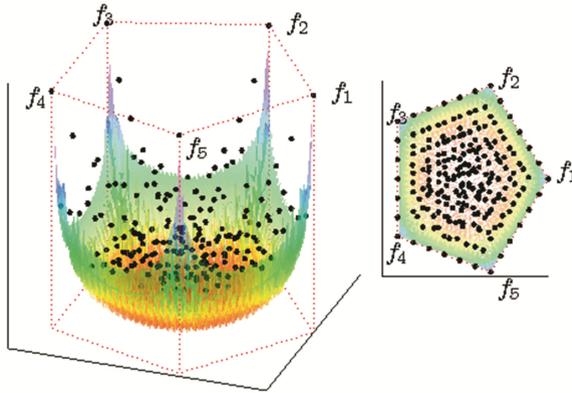


Fig. 5. 3D-RadVis plot of non-dominated solutions extracted using the proposed FNFR method. The black dots illustrate solutions extracted using the proposed scheme from a large set of non-dominated solutions generated by the GDE3, SMPSO, and SPEA2 algorithms in 20 runs.

Numerical Analysis of Solutions obtained by each Algorithm against the FNFR Method.

Here, we investigate the performance of GDE3, SMPSO, and SPEA2 on seven benchmark test problems with 3-, 5-, 8-, and 10-objectives. Table 2 provides the best, median, worst IGD values for GDE3, SMPSO, and SPEA2 on 3- and 5-objective DTLZ1, convex DTLZ2 and WFG1 test problems in 20 runs. In all instances the solution set obtained through the FNFR method from each algorithm have better IGD values than the best IGD values attained by a single run. Moreover, the IGD value of the solution set extracted by the FNFR method from a large set of solutions collected during each algorithm's run is also better than the three above-mentioned IGD values. The grey shades in Table 2 indicate the IGD values of solutions extracted by the FNFR method. When it comes to higher dimensional test problems, the FNFR scheme further proved its efficacy when comparing the IGD values of solutions extracted by the FNFR method against the best IGD values obtained by each algorithm in a single run. Table 3 shows the best, median, and worst IGD values for GDE3 and SMPSO algorithms against the IGD value of a set solutions extracted by the FNFR method on 8- and 10-objective DTLZ3, convex DTLZ4, and WFG2 test problems.

Table 2. Best, median, and worst IGD values for GDE3, SMPSO, and SPEA2 algorithms against the IGD value of solutions extracted by the FNFR method on 3- and 5-objective DTLZ1, convex DTLZ2, and WFG1 test problems.

M	DTLZ1		Convex DTLZ2		WFG1	
	3	5	3	5	3	5
Max Generation	400	600	400	600	400	600
GDE3 Best	7.43E-04	1.23E-03	7.09E-04	5.96E-04	1.23E-03	8.24E-04
GDE3 Median	8.10E-04	1.27E-03	7.70E-04	6.23E-04	1.40E-03	9.23E-04
GDE3 Worst	8.43E-03	1.31E-03	8.48E-04	7.09E-04	2.36E-03	1.26E-03
GDE3 Fusion	5.68E-04	9.27E-04	6.94E-04	4.96E-04	1.96E-04	7.49E-04
SMPSO Best	9.26E-04	2.21E-03	8.92E-04	1.29E-03	7.45E-03	6.04E-03
SMPSO Median	9.83E-04	2.48E-03	1.00E-03	1.83E-03	7.94E-03	6.44E-03
SMPSO Worst	1.05E-03	2.97E-03	1.15E-03	2.71E-03	8.11E-03	6.74E-03
SMPSO Fusion	6.99E-04	1.75E-03	8.27E-04	1.05E-03	6.41E-03	5.02E-03
SPEA2 Best	6.16E-04	2.23E-01	5.79E-04	2.19E-02	8.13E-04	7.48E-04
SPEA2 Median	6.35E-04	6.38E-01	6.03E-04	3.16E-02	1.04E-03	1.15E-03
SPEA2 Worst	6.90E-04	1.03E+00	6.39E-04	4.02E-02	3.56E-03	1.44E-03
SPEA2 Fusion	5.63E-04	2.23E-01	5.39E-04	2.48E-02	6.05E-04	8.21E-04
Overall Fusion	4.54E-04	8.92E-04	7.00E-04	4.21E-04	1.42E-04	7.64E-04

Table 3. Best, median, and worst IGD values for GDE3, SMPSO, and SPEA2 algorithms against the IGD value of solutions extracted by the FNFR method on 8- and 10-objective DTLZ3, convex DTLZ4, and WFG2 test problems.

M	DTLZ3		DTLZ4		WFG2	
	8	10	8	10	8	10
GDE3 Best	1.26E-03	9.30E-04	2.83E-03	2.35E-03	2.23E-03	1.84E-03
GDE3 Median	2.95E-03	1.24E-03	2.97E-03	2.43E-03	2.64E-03	2.02E-03
GDE3 Worst	7.59E-03	1.42E-03	3.04E-03	2.53E-03	3.26E-03	2.24E-03
GDE3 Fusion	1.08E-03	7.10E-04	3.02E-03	2.40E-03	2.05E-03	1.71E-03
SMPSO Best	3.90E-03	1.11E-03	2.85E-03	2.59E-03	2.73E-03	1.97E-03
SMPSO Median	8.88E-03	2.05E-03	3.95E-03	2.85E-03	3.12E-03	2.18E-03
SMPSO Worst	8.77E-03	1.98E-03	3.67E-03	3.28E-03	2.42E-03	1.93E-03
SMPSO Fusion	3.77E-03	1.83E-03	3.01E-03	2.32E-03	2.55E-03	1.80E-03
Overall Fusion	3.34E-03	1.07E-03	1.99E-03	2.25E-03	2.09E-03	1.51E-03

Visual Analysis of Preferred Solutions Found by the FNFR Method. Here we investigate the effectiveness of the proposed FNFR method when dealing with extracting solutions close to preferred region. In a practical multi- of many-objective optimization problems decision-makers may be interested in visualizing (when possible) the entire PF before selecting the one or more solutions for further investigation. In such scenario, the FNFR scheme is an effective tool to construct the entire PF and select Pareto-optimal points that are close to the supplied reference points. Figure 6 show preferred solutions extracted using the proposed FNFR method. The grey dotted background indicates non-dominated solutions obtained by the GDE3, SMPSO, and SPEA2 algorithms in 20 runs

and the black dots illustrate preferred solutions extracted from these solutions using the proposed scheme.

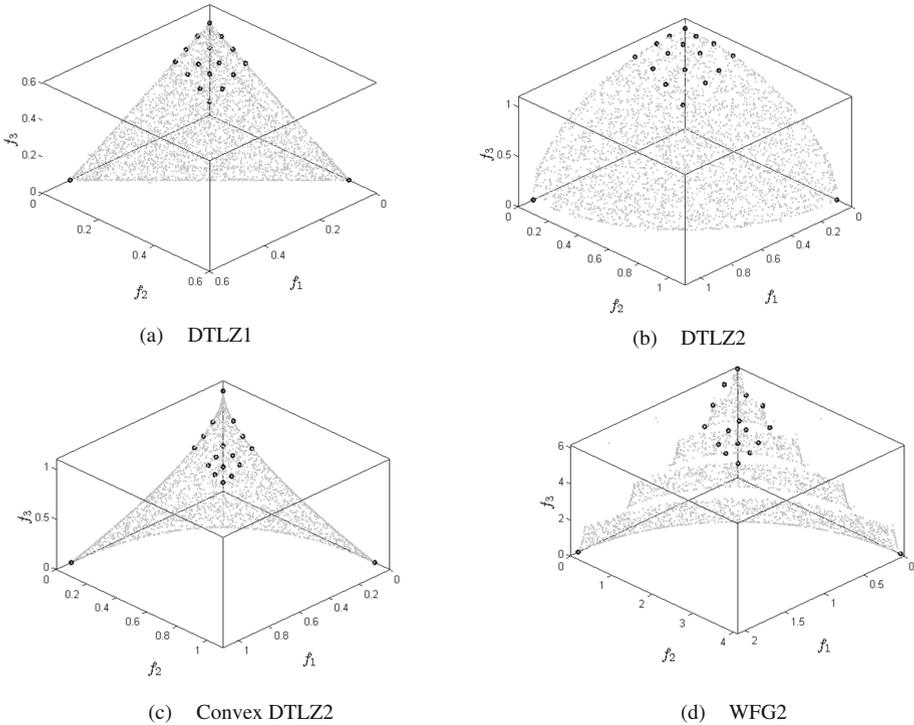


Fig. 6. Preferred solutions extracted using the proposed FNFR method. The grey dotted background indicate non-dominated solutions obtained by the GDE3, SMPSO, and SPEA2 algorithms in 20 runs and the black dots illustrate preferred solutions extracted from these solutions using the proposed FNFR approach.

4 Concluding Remarks

In this paper, we proposed a novel hybridization of population-based metaheuristic algorithms called fusion of non-dominated fronts using reference points (FNFR) to gain combined benefit of several algorithms. The hybridization step in FNFR occurs after the optimization process of multiple runs of several algorithms. In every run, FNFR assembles and constructs the entire Pareto-optimal front from a large set of non-dominated solutions found using multiple algorithms. This step is crucial because no single algorithm is capable of producing well-distributed solutions for all types of problems all the time. Thereafter, FNFR uses predefined structured reference points or user selected reference points to extract preferred solutions from this set.

Experimental results showed the effectiveness of the proposed FNFR scheme with numerical experiments by considering three widely used optimization algorithms, GDE3,

SMPSO, and SPEA2. The FNFR method is able to extract well-distributed solutions in a specific region of interest among thousands of non-dominated solutions collected after every run of multiple algorithms. Therefore the FNFR scheme can effectively be used by decision-makers to select and examine preferred solutions among a large set of solution which can be astronomically difficult to manage. In future, we would like to extend this study further by applying the FNFR scheme during the optimization process so that preferred solutions can be extracted and inserted back to the current population to boost the search process.

References

1. Nwankwor, E., Nagar, A.K., Reid, D.: Hybrid differential evolution and particle swarm optimization for optimal well placement. *Comput. Geosci.* **17**(2), 249–268 (2013)
2. Mirjalili, S., Hashim, S.Z.M.: A new hybrid PSO-GSA algorithm for function optimization. In: 2010 International Conference on Computer and Information Application (ICCIA), pp. 374–377 (2010)
3. Elhossini, A., Areibi, S., Dony, R.: Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evol. Comput.* **18**, 127–156 (2010)
4. Tang, L., Wang, X.: A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **17**(1), 20–45 (2013)
5. Wang, Y., Cai, Z., Zhou, Y., Fan, Z.: Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. *Struct. Multi. Design Optim.* **37**, 395–413 (2009)
6. Zăvoianu, A.-C., Lughofer, E., Bramerdorfer, G., Amrhein, W., Klement, E.P.: DECMO2: a robust hybrid and adaptive multi-objective evolutionary algorithm. *Soft. Comput.* **19**, 3551–3569 (2015)
7. Zhao, S.-Z., Suganthan, P.N.: Multi-objective evolutionary algorithm with ensemble of external archives. *Int. J. Innovative Comput. Inf. Control* **6**, 1713–1726 (2010)
8. Qu, B.Y., Suganthan, P.N.: Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods. *Eng. Optim.* **43**(4), 403–416 (2011)
9. Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes. *IEEE Trans. Evol. Comput.* **16**, 442–446 (2012)
10. Tan, K.C., Chew, Y., Lee, L.H.: A hybrid multi-objective evolutionary algorithm for solving truck and trailer vehicle routing problems. *Eur. J. Oper. Res.* **172**, 855–885 (2006)
11. Xia, W., Wu, Z.: An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Comput. Ind. Eng.* **48**(2), 409–425 (2005)
12. Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., Mirzaei, A.H.: A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. *Inf. Sci.* **177**, 5072–5090 (2007)
13. Karthikeyan, S., Asokan, P., Nickolas, S., Page, T.: A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems. *Int. J. Bio-Inspired Comput.* **7**, 386–401 (2015)
14. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans. Evol. Comput.* **18**, 577–601 (2014)
15. Das, I., Dennis, J.E.: Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **8**, 631–657 (1998)

16. Kukkonen, S., Lampinen, J.: GDE3: the third evolution step of generalized differential evolution. In: The 2005 IEEE Congress on Evolutionary Computation, pp. 443–450 (2005)
17. Nebro, A.J., Durillo, J.J., Garcia-Nieto, J., Coello, C.C., Luna, F., Alba, E.: Smpso: a new pso-based metaheuristic for multi-objective optimization. In: 2009 IEEE Symposium on Computational Intelligence in Multi-criteria Decision-Making, pp. 66–73 (2009)
18. Zitzler, E., Laumanns, M., Thiele, L., Zitzler, E., Zitzler, E., Thiele, L. et al.: SPEA2: improving the strength pareto evolutionary algorithm. Ed: Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) (2001)
19. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the Congress on Evolutionary Computation (CEC-2002), (Honolulu, USA), pp. 825–830 (2002)
20. Huband, S., Barone, L., While, L., Hingston, P.: A scalable multi-objective test problem toolkit. In: Evolutionary Multi-criterion Optimization, pp. 280–295 (2005)
21. Yang, S., Li, M., Liu, X., Zheng, J.: A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **17**, 721–736 (2013)
22. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach. *IEEE Trans. Evol. Comput.* **18**, 602–622 (2014)
23. Ibrahim, A., Rahnamayan, S., Vargas Martin, M., Deb, K.: 3D-RadVis: visualization of pareto front in many-objective optimization. In: 2016 IEEE Congress on Evolutionary Computation (CEC), pp. 736–745 (2016)