Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

A STUDY ON MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION WITH WEIGHTED SCALARIZING FUNCTIONS

Loo Hay Lee
Ek Peng Chew
Qian Yu
Haobin Li
Yue Liu

National University of Singapore.
Department of Industrial and Systems Engineering
1 Engineering Drive 2, SINGAPORE 117576

ABSTRACT

In literature, multi-objective particle swarm optimization (PSO) algorithms are shown to have great potential in solving simulation optimization problems with real number decision variables and objectives. This paper develops a multi-objective PSO algorithm based on weighted scalarization (MPSOws) in which objectives are scalarized by different sets of weights at individual particles while evaluation results are shared among the swarm. Various scalarizing functions, such as simple weighted aggregation (SWA), weighted compromise programming (WCP), and penalized boundary intersection (PBI) can be applied in the algorithm. To improve the diversity and uniformity of the Pareto set, a hybrid external archiving technique consisting of both KNN and ε-dominance methods is proposed. Numerical experiments on noise-free problems are conducted to show that MPSOws outperforms the benchmark algorithm and WCP is the most preferable strategy for the scalarization. In addition, simulation allocation rules (SARs) can be further applied with MPSOws when evaluation error is considered.

1 INTRODUCTION

The Particle Swarm Optimization (PSO) originates from the simulation of the movement of flocks of birds (Parsopoulos and Vrahatis 2002). PSO has become a popular metaheuristic especially for solving problems involving real number decision variables and objective functions (Margarita and Coello 2006).

From an abstract level, in the PSO algorithm, a set (swarm) of candidate solutions (particles) to the optimization problem "fly" through the decision space. Particles move according to a pre-defined velocity function. The velocity function usually takes the following as arguments: the current velocity of the particle, the current position of the particle, the best position visited so far by the particle (**lbest**) and the best position visited by the swarm (**gbest**). Selection of **lbest** and **gbest** usually occurs according to a predefined leader selection strategy (Nebro et al. 2013).

In this paper, we apply the PSO concept in solving a multi-objective optimization problem that has the following form (Margarita and Coello 2006):

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f^{(1)}(\mathbf{x}), f^{(2)}(\mathbf{x}), \dots, f^{(J)}(\mathbf{x})], \text{ s.t. } \mathbf{l} \le \mathbf{x} \le \mathbf{u},$$

where $\mathbf{x}$ is the vector of decision variables, $f^{(j)} : \mathbb{R}^n \to \mathbb{R}$ for $j = \{1, \dots, J\}$ are the objective functions, and $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bounds of the decision variables.

Since the J objective functions usually conflict with each other, the identification of a single global minimum at the same point is impossible. The goal of multi-objective optimization is to detect the highest possible number of Pareto optimal solutions that correspond to an adequately spread Pareto front, with the smallest possible deviation from the true Pareto front (Parsopoulos and Vrahatis 2008).

We propose the new multi-objective particle swarm optimization algorithm based on weighted scalarizing functions (MPSOws) for solving multi-objective global optimization problems. Under appropriate weighted scalarizing schemes, particles each having a unique weight vector tend to be attracted towards different regions in the objective space, resulting in a diverse set of solutions. Different weighted scalarizing functions are tested, namely Simple Weighted Aggregation (SWA), Weighted Compromise Programming (WCP), Tchebycheff Method (TCH) and Penalized Boundary Intersection (PBI). All the four scalarizing schemes convert multiple objectives into a scalar value and use it to select leaders and guide particles towards the Pareto front. Unlike SWA which often fails when the Pareto front is non-convex, WCP, TCH and PBI methods can deal with non-convex problems. Furthermore, WCP and PBI yield more consistent performance compared to TCH. A k-nearest neighbour-based crowding measure coupled with ε-dominance is used to maintain the external archive. The archiving method is empirically shown to outperform the conventional crowding distance archiving in terms of diversity and uniformity preservation. MPSOws is compared against the well-known multi-objective optimization algorithm NSGA-II using 2-dimensional ZDT and 3-dimensional DTLZ test problems. For all the twelve test problems, MPSOws performs better or similarly well compared to NSGA-II in terms of both convergence and diversity measures.

The rest of this section introduces some basic concepts such as multi-objective optimization and particle swarm optimization. Section 2 lists some relevant work done by other researchers. Section 3 introduces the four weighted scalarizing functions which have been tested in this paper. This section also discusses about the use of velocity function and external archive in MPSOws. Experiment results are presented in Section 4, which also benchmarks MPSOws against NSGA-II.

2 LITERATURE REVIEW

A comprehensive survey of different approaches of PSO in solving multi-objective problems was conducted by Parsopoulos and Vrahatis (2008). The paper summarized a few main approaches. Conventional Weighted Aggregation (CWA), Bang-Bang Weighted Aggregation (BWA), Dynamic Weighted Aggregation (DWA), Pareto Dominance Approaches and Objective Function Ordering Approaches were among the methods being discussed.

Jin et al. (2001) provided a schematic theory illustrating why CWA methods fail to solve multi-objective problems with non-convex Pareto front. It is shown empirically that Evolutionary Dynamic Weighted Aggregation (EDWA) can deal with multi-objective optimization problems with a concave Pareto front. EDWA dynamically changes the weights of all particles to force them to trace along the Pareto front.

Weighted compromise programming (WCP) was discussed by Athan and Papalambros (1996) as a method for converting multi-objective formulation into a substitute problem with a scalar objective. WCP raises each objective value to a large integer power m before aggregating them using weights raised to the same power. The WCP method is further discussed in Section 3.1.2.

A decomposition-based multi-objective evolutionary algorithm named MOEA/D was introduced by Zhang and Li (2007). This algorithm decomposes a multi-objective optimization problem into a number of single-objective scalar optimization sub-problems and optimizes them simultaneously. It was demonstrated that MOEA/D performed outstandingly on a number of multi-objective test problems. Several scalarizing schemes were introduced, namely Weighted Sum Approach, Tchebycheff Approach and Boundary Intersection Approach.

Some research has been done to incorporate weighted scalarization, e.g., Tchebycheff method, into PSO by Al Moubayed et al. (2010) in the algorithm SDMOPSO which was reported to be competitive against five other standard algorithms.

Instead of focusing on a specific scalarizing function, in this paper we first propose a general framework of MPSOws that works well with a wide range of scalarizing functions. Then, four of the scalarizing functions, which have mainly been applied in evolutionary algorithm, are introduced and incorporated in to the MPSOws. Through the numerical experiments, we identified that the WCP is the best among the four and outperforms the Tchebycheff method as in Al Moubayed et al. (2010). In addition, we also suggest that the constrained velocity function and a new archiving method can be applied in the algorithm to improve the performance. The numerical experiment is conducted to show the comparison results.

3 METHODOLOGY

In MPSOws, particles with unique weight vectors are used to search for solutions in the objective space. During each iteration, particles share their own objective values among the swarm so as to make better use of the computation effort. At the same time, since particles have different sets of weights, their measures about the fitness of the same particle usually differ from each other. In other words, the fittest from the perspective of one particle may not be the fittest from the perspective of another because particles use different weights to assess the swarm. The procedure of MPSOws is shown in Algorithm 1.

Algorithm 1: Multi-objective PSO with weighted scalarizing function (MPSOws)

Set the iteration count $k = 1$;
forall $h \in \{1, \ldots, N\}$ **do**
    assign a weight vector $\mathbf{w}_h$ *to the particle* h *;*
    initialize $\mathbf{x}_h \leftarrow \mathbf{x}$ *by randomly selecting* $\mathbf{x} \in [\mathbf{l}, \mathbf{u}]$ *and evaluate* $\mathbf{f}(\mathbf{x}_h)$ *;*
    $\mathbf{v}_h \leftarrow \mathbf{0}$, $\mathbf{lbest}_h \leftarrow \mathbf{x}_h$, $\mathbf{gbest}_h \leftarrow \mathbf{x}_h$
end
while $k < K$ **do**
    $k \leftarrow k + 1$ *;*
    insert $\mathbf{x}_h$ *into the external archive* ζ *which stores the non-dominated solutions;*
    forall $h \in \{1, \ldots, N\}$ **do**
        update $\mathbf{lbest}_h \leftarrow \mathbf{x}_h$ *if* $g(\mathbf{f}(\mathbf{x}_h), \mathbf{w}_h) < g(\mathbf{f}(\mathbf{lbest}_h), \mathbf{w}_h)$ *;*
        update $\mathbf{gbest}_h \leftarrow \mathbf{x}_{i^*}$ *if* $g(\mathbf{f}(\mathbf{x}_{i^*}), \mathbf{w}_h) < g(\mathbf{f}(\mathbf{gbest}_h), \mathbf{w}_h)$ *where* $i^* = \arg\min_{i \in \{1,\ldots,N\}} g(\mathbf{f}(\mathbf{x}_i), \mathbf{w}_h)$ *;*
        update $\mathbf{v}_h$ *by velocity function, considering* $\mathbf{lbest}_h$ *and* $\mathbf{gbest}_h$ *;*
        $\mathbf{x}_h \leftarrow \mathbf{x}_h + \mathbf{v}_h$, *mutate according to* γ, *and evaluate* $\mathbf{f}(\mathbf{x}_h)$ *;*
    end
end

Note that as the input setting of the algorithm, N is the number of particles, K is the maximum number of iterations, and γ is the mutation rate. Emphasized terms, i.e., the weighted scalarizing function $g(\mathbf{f}, \mathbf{w}_h)$, the velocity function, and the archiving method in the algorithm will be further discussed in later sections. Empirical results are presented in Section 4 .

3.1 Weighted Scalarizing Function

A number of weighted scalarizing functions have been tested during the construction of MPSOws. Simple Weighted Aggregation (SWA), Weight Compromise Programming (WCP), Tchebycheff Method (TCH) and Penalized Boundary Intersection (PBI) will be discussed. In Section 4.1, it will be demonstrated that WCP and PBI are preferred in the context of MPSOws because of their consistency in achieving good convergence and diversity.

3.1.1 Simple Weighted Aggregation (SWA)

When the weight vector $\mathbf{w}_h$ is set, the SWA function (Jin et al. 2001) returns a scalar value which is the weighted sum of each objective value $f^{(j)}$ with respect to $w^{(j)}$. This can be considered the simplest and most intuitive way of converting a multi-objective problem into N single objective optimization sub-problems. For any particle h, the weighted scalarizing function by SWA has the form

$$g^{\text{swa}}(\mathbf{f}, \mathbf{w}_h) = \mathbf{f} \cdot \mathbf{w}_h = \sum_{j}^{J} f^{(j)} \cdot w_h^{(j)}.$$

Since g^{swa} is essentially a convex combination of the elements of $\mathbf{f}$, it suffers from the disadvantage of not being able to find a diverse set solutions if the Pareto front is non-convex (Jin et al. 2001). As shown in Figure 1 , for the convex ZDT1 Pareto front, as $w^{(1)}$ changes from 0 to 1, $f^{*(1)}$ which minimizes g^{swa} changes as well. Using different weights can therefore lead to a diverse set of non-dominated solutions. However, this is not the case when the Pareto front is non-convex, as in ZDT2. Regardless of the weights used, the particles are drawn towards either extremes of the Pareto front. This weakness of SWA is further demonstrated in Section 4.1 where empirical results comparing SWA and its alternatives are presented.

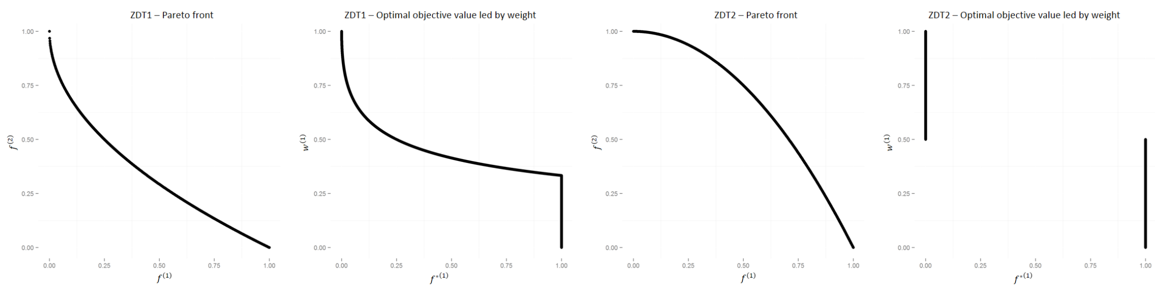

Figure 1: The Pareto front and optimal $f^{*(1)}$ led by weight $w^{(1)}$ for SWA on ZDT1 & ZDT2.

3.1.2 Weighted Compromising Programming (WCP)

The disadvantage of SWA can be dealt with using a simple but powerful strategy as discussed by Athan and Papalambros (1996). For particle h, the WCP weighting function has the form:

$$g^{\text{wcp}}(\mathbf{f}, \mathbf{w}_h) = \sum_{j}^{J} \left[f^{(j)} \cdot w_h^{(j)} \right]^m, \text{ where } m = 3, 5, 7, 9, \text{ or } 11, \ldots.$$

The structure of the WCP is similar to that of SWA. The only difference is that WCP raises the product $f^{(j)} \cdot w_h^{(j)}$ to an integer power m before the summation. In fact, SWA is just a special case of WCP with $m = 1$. Two features of this scalarizing function are worth mentioning. Firstly, the WCP weighting function is sufficient for obtaining Pareto optimality because g^{wcp} increases monotonically with respect to $f^{(j)}$ for every dimension j. Secondly, Athan and Papalambros (1996) proves that, when integer m is large enough, a set of weights exists for every point on the Pareto front. This implies that the whole Pareto front can be covered when a large number of different weight vectors are used. An intuitive understanding of WCP is shown in Figure 2 that, for every point $\mathbf{f} = \left[f^{(1)}, f^{(2)}\right]$ on the non-convex Pareto front of ZDT2, a corresponding weight vector can be found at $\left[w^{(1)}, 1 - w^{(1)}\right]$. Moreover, if the curve on the right is smooth and close to a straight line, using equally-spaced weight vectors can potentially lead to a set of almost uniformly spaced Pareto solutions.

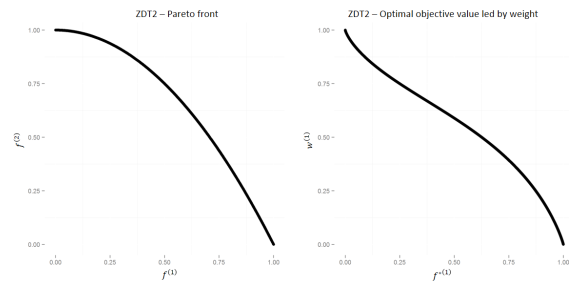

Figure 2: The Pareto front and optimal $f^{*(1)}$ led by weight $w^{(1)}$ for WCP on ZDT2.

3.1.3 Tchebycheff Method (TCH)

The TCH method (Zhang and Li 2007) converts multi-objective optimization into a number of single-objective ones using weighting function in this form:

$$g^{\text{tch}}(\mathbf{f}, \mathbf{w}_h) = \max_{1 \le j \le J} \left| f^{(j)} - r^{*(j)} \right| \cdot w_h^{(j)}$$

where $\mathbf{r}^*$ is the "ideal point", i.e. the vector containing the best objective value in each dimension found so-far. This scalarizing scheme is adopted in a number of decomposition-based PSO algorithms such as SDMOPSO (Al Moubayed et al. 2010) and MOPSO-PD (Liu and Niu 2013). In Section 4.1, TCH method is benchmarked against other schemes and it is found that TCH is not competitive in the context of MPSOws.

3.1.4 Penalized Boundary Intersection (PBI)

The PBI method (Zhang and Li 2007) is a modification of the Normal Boundary Intersection (NBI) method (Das and Dennis 1998). It uses a penalty-based method to handle the equality constraint in the NBI sub-problem (see Figure 3). The PBI weighting function for the particle h has the form

$$g^{\text{pbi}}(\mathbf{f}, \mathbf{w}_h) = d_1 + \theta \cdot d_2$$

in which

$$d_1 = \frac{\left\| (\mathbf{r}^* - \mathbf{f})^{\mathrm{T}} \times \mathbf{w}_h \right\|}{\|\mathbf{w}_h\|} \text{ and } d_2 = \|\mathbf{f} - (\mathbf{r}^* - d_1 \mathbf{w}_h)\| .$$

Again, $\mathbf{r}^*$ is the "ideal point"; θ is an arbitrary constant set to 5.0 according to Zhang and Li (2007).

In g^{pbi}, d_1 can be thought of as the length of the projection of the $(\mathbf{f} - \mathbf{r}^*)$ vector onto the intersecting line. d_2 is then the distance of $\mathbf{f}$ from the intersecting line. The PBI method has the advantage of being able to deal with both convex and non-convex problems. Furthermore, the resulting solutions tend to be uniformly distributed when equally spaced weight vectors are used. The empirical comparisons of PBI and WCP methods are presented in Section 4.1.

3.2 Velocity Function

The speed constrained velocity function developed by Nebro et al. (2009) in SMPSO is adopted for MPSOws due to its proven performance. According to the method, for a particle h, the velocity $\mathbf{v}_h$ can be updated as in Algorithm 2. The main feature of this velocity function is that it constricts the velocity using a constriction factor χ determined according a probability distribution. It also limits the velocity between the lower and upper bound of the velocity allowed, i.e., $\mathbf{v}^{\mathrm{l}}$ and $\mathbf{v}^{\mathrm{u}}$. Note that, ω is a constant indicating the tendency of particle to continue with its current velocity.

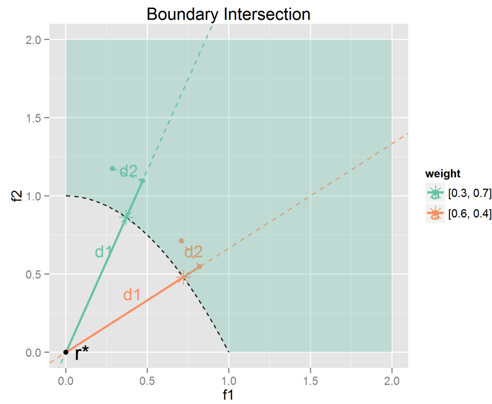

Figure 3: Illustration of Penalized Boundary Intersection (PBI).

Algorithm 2: The speed constrained velocity function (Nebro et al. 2009)

Generate r_1 and r_2 from the uniform distribution $(0,1)$;
Generate c_1 and c_2 from the uniform distribution $(1.5,2.5)$;
if $c_1 + c_2 \leq 4$ **then**
  $\chi \leftarrow 1$;
end
else
  $\rho \leftarrow c_1 + c_2$;
  $\chi \leftarrow \frac{2.0}{2.0-\rho-\sqrt{\rho^2-4\rho}}$;
end
$\mathbf{v}_h \leftarrow \chi\left(\omega \mathbf{v}_h + c_1 r_1 \left(\mathbf{lbest}_h - \mathbf{x}_h\right) + c_2 r_2 \left(\mathbf{gbest}_h - \mathbf{x}_h\right)\right)$;
$\mathbf{v}_h \leftarrow \min\left(\max\left(\mathbf{v}_h, \mathbf{v}^{\mathrm{l}}\right), \mathbf{v}^{\mathrm{u}}\right)$;
return $\mathbf{v}_h$.

3.3 External Archive

The non-dominated solution obtained by the algorithm is stored in a solution set called "external archive". To save the computational resources needed by the optimizer, not every non-dominated solution found during the iterations need to be stored in the archive. Since scalarizing methods such as WCP and PBI cannot guarantee the diversity and uniformity of solutions, careful selection of non-dominated solutions into the external archive is important for MPSOws.

An archiving method that is widely used for multi-objective PSO is the crowding distance archiving (Raquel and Naval Jr. 2005). This method works very well for 2-d problems but performance quickly deteriorates when dealing with higher dimensional problems (Kukkonen and Deb 2006).

In our study, we combine the advantages of k-nearest neighbour (KNN) archiving (Kukkonen and Deb 2006) and ε-dominance archiving (Laumanns et al. 2002). We name the hybrid method as ε-KNN archive. As described by Algorithm 3, when a new solution $\mathbf{x}$ is to be inserted to the archive ζ, it checks the existing solutions and removes those that are ε-dominated by the new solution. If the archive is over capacity after the insertion, it performs a shrinking. It calculates the sum of the Euclidean distance to k-nearest neighbours for each solution and removes the one having the smallest sum of distance.

Algorithm 3: The ε-KNN external archive procedure

forall $\mathbf{z} \in \zeta$ **do**
    if $\mathbf{f}(\mathbf{x})$ *ε-dominates* $\mathbf{f}(\mathbf{z})$ *and they are in different boxes* **then**
        Remove $\mathbf{z}$ from ζ ;
    end
    else if $\mathbf{f}(\mathbf{z})$ *ε-dominates* $\mathbf{f}(\mathbf{x})$ **then**
        Terminates.
    end
end
Add $\mathbf{x}$ into ζ ;
if $\|\zeta\| >$ *external archive capacity* **then**
    Calculate the KNN distance for each $\mathbf{z} \in \zeta$;
    Remove the solution with the minimum sum of KNN distance ;
end

Empirical evidence has shown that ε-KNN archive performs better than the traditional crowding distance archive in terms of dominated hypervolume of Pareto front. The representative experiment results comparing crowding distance and ε-KNN archive are shown in Section 4.2.

4 RESULTS AND DISCUSSION

All the experimental settings below make comparisons by running each configuration 20 times using different random seeds each time. The test problems, i.e., ZDTs and DTLZs, are defined in Zitzler et al. (2000) and Deb et al. (2005) respectively.

4.1 Comparison of Weighted Scalarizing Functions

WCP and PBI clearly outperform SWA and TCH in most of the test problems. The advantage of WCP and PBI over SWA is clear especially when the Pareto front is non-convex. The advantage of PBI over WCP is seen in the case of DTLZ4 which poses the difficulty that solutions are denser along the edges of the true Pareto front . Despite the challenge, the PBI method still finds a set of fairly uniformly distributed solutions. When using TCH on problems such as ZDT4 and DTLZ4, particles seem to have strong preference towards certain regions on the Pareto front. The performance of TCH does not seem to be consistent. Plots of solutions obtained by MPSOws using different weighted scalarizing functions for some representative test problems are shown in Figure 4.

The boxplots of total dominated hypervolume achieved by MPSOws using the four scalarizing schemes are shown in Figure 5.

4.2 Comparison of Crowding Distance and ε-KNN Archive

The use of ε-KNN archive helps improve the diversity, uniformity and total hypervolume of the results of MPSOws. The comparison of results obtained by MPSOws using crowding distance archive and ε-KNN archive is presented in Figure 6. For the ε-KNN archive, the number of nearest neighbours is set to $k+1$ with k being the number of objectives. For example, for 3-d problems, KNN distance of an archived solution is found by summing the distance to its 4 nearest neighbours in the archive. As previously mentioned, the advantage of ε-KNN archive is clear especially for 3-d DTLZ problems.

4.3 Benchmarking Against NSGA-II

To evaluate the performance of MPSOws, it is benchmarked against the well-known multi-objective evolutionary algorithm NSGA-II (Deb et al. 2002). The configurations used for MPSOws is WCP weighted scalarizing function ($m = 9$), equally-spaced weight vectors and ε-KNN archive. Both MPSOws and NSGA-II evaluate the objective function $30,000$ times for 2-d problems and $31,500$ times for 3-d problems (A multiple of 105 is used because MPSOws needs to have such number of particles in order to use equally-spaced 3-d weight vectors. NSGA-II is set accordingly for fair comparison). The population size of NSGA-II is set to 100. The external archive size of MPSOws is set to 100. Each algorithm is executed 20 times on each test problem. It is demonstrated that MPSOws performs better than NSGA-II for most of the twelve test problems in terms of total dominated hypervolume. The visual appearance of most of the problems' Pareto optimal solutions is better than that obtained by NSGA-II most likely due to the use of ε-KNN archive. Graphical plots of solutions for some representative test problems are shown in Figure 7. Statistics of total dominated hypervolume, i.e., the mean and standard deviation from the 20 runs, is shown in Table 1.

Table 1: Dominated hypervolume (μ, σ) achieved by NSGA-II and MPSOws on twelve test problems.

Problem	NSGA-II	MPSOws
ZDT1	$(0.660357,\ 2.33 \times 10^{-4})$	$(0.662068,\ 2.09 \times 10^{-5})$
ZDT2	$(0.327298,\ 2.95 \times 10^{-4})$	$(0.328715,\ 2.31 \times 10^{-5})$
ZDT3	$(0.515081,\ 7.75 \times 10^{-4})$	$(0.515807,\ 3.20 \times 10^{-5})$
ZDT4	$(0.657277,\ 2.26 \times 10^{-3})$	$(0.660276,\ 1.33 \times 10^{-3})$
ZDT6	$(0.395814,\ 7.26 \times 10^{-4})$	$(0.401449,\ 1.83 \times 10^{-5})$
DTLZ1	$(0.747002,\ 2.53 \times 10^{-2})$	$(0.758574,\ 7.57 \times 10^{-3})$
DTLZ2	$(0.388222,\ 4.97 \times 10^{-3})$	$(0.401466,\ 2.36 \times 10^{-3})$
DTLZ3	$(0,0)$	$(0.371043,\ 1.85 \times 10^{-2})$
DTLZ4	$(0.389151,\ 3.76 \times 10^{-3})$	$(0.395365,\ 3.92 \times 10^{-3})$
DTLZ5	$(0.094131,\ 1.52 \times 10^{-4})$	$(0.093817,\ 1.10 \times 10^{-4})$
DTLZ6	$(0,0)$	$(0.094414,\ 1.45 \times 10^{-4})$
DTLZ7	$(0.293703,\ 3.64 \times 10^{-3})$	$(0.304843,\ 9.88 \times 10^{-4})$

5 SIMULATION ALLOCATION RULES

All experiment results shown above assume that solutions can be evaluated with sufficient accuracy, in the sense that no false selection of **gbest** and **lbest** is made in each iteration. However, in simulation optimization where evaluation error occurs, this assumption can be violated. Therefore, some simulation allocation rule (SAR) needs to be applied to allow one solution to be evaluated multiple times, such that we can increase the chance of making the correct decision. A simple SAR is to equally allocate simulation runs to each solution visited by particles, until the measure of selection quality (Lee et al. 2010) is observed to reach the desired level. Alternatively, when computing budget is limited, Chen and Lee (2011), Lee et al. (2010) suggest that the MOCBA (multi-objective optimal computing budget allocation) concept can be applied to utilize simulation runs most efficiently, in the sense that the probability of correct selection (PCS) of the Pareto set can be maximized.

However, with MPSOws, the previous works about MOCBA cannot be directly applied, because instead of finding a unique Pareto set, we are more interested in comparing weighted objectives of **gbest** and **lbest** simultaneously from each particle's perspective. By setting the corresponding PCS as the main objective to be maximized and applying the general MOCBA framework, we can develop an efficient SAR for MPSOws in future research.

6 CONCLUSION

The proposed algorithm MPSOws uses weighted scalarizing functions to breakdown a multi-objective optimization problem into a number of single objective problems and solves them simultaneously. Particles are assigned different weight vectors which make them prefer different regions in the objective space. WCP achieves better total hypervolume for most of the test problems but PBI is able to generate a more uniformly distributed solution set for some problems. The external archive is maintained by means of KNN distance coupled with ε-dominance which has significant advantages over the conventional crowding distance archiving especially for higher dimensional problems.

The proposed algorithm has been shown to perform better or similarly compared to NSGA-II over a number of test problems in terms of diversity and convergence. The work in this paper provides a framework which facilitates the testing of different weighted scalarizing functions in the context of PSO.

Future work can explore other scalarizing schemes other than the ones already introduced. More work can be done on finding a better weight generation mechanism to deal with Pareto fronts with disconnected or complex geometric shapes. Adaptive weight update can be developed to improve the efficiency when dealing with new problems having complex Pareto fronts. Other archiving methods such as hypervolume archive can be readily incorporated especially if the complexity for the computation of hypervolume improves. Additional benchmarking can be performed using MPSOws against other standard multi-objective algorithms to better understand its performance.

REFERENCES

Al Moubayed, N., A. Petrovski, and J. McCall. 2010. "A Novel Smart Multi-Objective Particle Swarm Optimisation Using Decomposition". In *Parallel Problem Solving from Nature - PPSN XI. 11th International Conference. Proceedings*, Volume pt.2, 1 – 10. Berlin, Germany.

Athan, T. W., and P. Y. Papalambros. 1996. "Note on weighted criteria methods for compromise solutions in multi-objective optimization". *Engineering Optimization* 27 (2): 155 – 176.

Chen, C.-H., and L. H. Lee. 2011. *Multi-objective Optimal Computing Budget Allocation*, 117 – 140. Singapore, Singapore: World Scientific.

Das, I., and J. Dennis. 1998, 8. "Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems". *SIAM Journal on Optimization* 8 (3): 631 – 657.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002, 4. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation* 6 (2): 182 – 197.

Deb, K., L. Thiele, M. Laumanns, and E. Zitzler. 2005. "Scalable Test Problems for Evolutionary Multiobjective Optimization". In *Evolutionary Multiobjective Optimization*, edited by A. Abraham, L. Jain, and R. Goldberg, Advanced Information and Knowledge Processing, 105 – 145. Springer London.

Jin, Y., M. Olhofer, and B. Sendhoff. 2001. "Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization: Why Does It Work and How?".

Kukkonen, S., and K. Deb. 2006. "A fast and effective method for pruning of non-dominated solutions in many-objective problems". In *Parallel Problem Solving from Nature - PPSN IX. 9th International Conference. Proceedings (Lecture Notes in Computer Science Vol. 4193)*, 553 – 562. Berlin, Germany.

Laumanns, M., L. Thiele, K. Deb, and E. Zitzler. 2002. "Combining convergence and diversity in evolutionary multiobjective optimization". *Evolutionary Computation* 10 (3): 263 – 282.

Lee, L. H., E. P. Chew, and S. Teng. 2010, 12. "Computing budget allocation rules for multi-objective simulation models based on different measures of selection quality". *Automatica* 46 (12): 1935 – 1950.

Liu, Y., and B. Niu. 2013. "A multi-objective particle swarm optimization based on decomposition". In *Communications in Computer and Information Science*, Volume 375, 200 – 205. Nanning, China.

Margarita, R.-S., and C. A. C. Coello. 2006. "Multi-Objective particle swarm optimizers: A survey of the state-of-the-art". *International Journal of Computational Intelligence Research* 2 (3): 287 – 308.

Nebro, A., J. Durillo, and C. Coello. 2013. "Analysis of leader selection strategies in a multi-objective particle swarm optimizer". In *2013 IEEE Congress on Evolutionary Computation*, 3153 – 3160. Piscataway, NJ, USA.

Nebro, A., J. Durillo, J. Garcia-Nieto, C. Coello Coello, F. Luna, and E. Alba. 2009. "SMPSO: a new PSO-based metaheuristic for multi-objective optimization". In *2009 IEEE Symposium on Computational Intelligence in Milti-Criteria Decision-Making*, 8 pp. –. Piscataway, NJ, USA.

Parsopoulos, K. E., and M. N. Vrahatis. 2002. "Recent approaches to global optimization problems through particle swarm optimization". *Natural Computing* 1 (2-3): 235 – 306.

Parsopoulos, K. E., and M. N. Vrahatis. 2008. "Multi-Objective Particles Swarm Optimization Approaches". In *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*, 20 – 42. IGI Global.

Raquel, C. R., and P. C. Naval Jr.. 2005. "An effective use of crowding distance in multiobjective particle swarm optimization". In *GECCO 2005 - Genetic and Evolutionary Computation Conference*, 257 – 264. Washington, D.C., United states.

Zhang, Q., and H. Li. 2007. "MOEA/D: A multiobjective evolutionary algorithm based on decomposition". *IEEE Transactions on Evolutionary Computation* 11 (6): 712 – 731.

Zitzler, E., K. Deb, and L. Thiele. 2000. "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results". *Evolutionary Computation* 8 (2): 173 – 195.

AUTHOR BIOGRAPHIES

LOO HAY LEE is Associate Professor and Deputy Head in the Department of Industrial and Systems Engineering, National University of Singapore. He received his B. S. (Electrical Engineering) degree from the National Taiwan University in 1992 and his S. M. and Ph. D. degrees in 1994 and 1997 from Harvard University. He is currently a senior member of IEEE, a committee member of ORSS, and a member of INFORMS. His research interests include production planning and control, logistics and vehicle routing, supply chain modeling, simulation-based optimization, and evolutionary computation. His email address is iseleelh@nus.edu.sg.

EK PENG CHEW is Associate Professor and Deputy Head in the Department of Industrial and Systems Engineering, National University of Singapore. He received his Ph. D. degree from the Georgia Institute of Technology. His research interests include logistics and inventory management, system modeling and simulation, and system optimization. His email address is isecep@nus.edu.sg.

QIAN YU is undergraduate student in the Department of Industrial and Systems Engineering, National University of Singapore. He is expected to graduate with B. Eng and B.B.A from National University of Singapore in 2014. His research interests include multi-objective optimization, particle swarm optimization and decentralized portfolio optimization. His email address is yuqian1990@gmail.com.

HAOBIN LI is Research Fellow at Centre for Maritime Studies, National University of Singapore. He received his B. Eng. (1st Class Hons) and Ph. D. degrees in 2009 and 2014 from the Industrial and Systems Engineering Department of National University of Singapore. His research interests include system modeling, multi-objective optimization and interactive information system. His email address is i@li-haobin.net.

YUE LIU is Ph. D. candidate in the Department of Industrial and Systems Engineering at National University of Singapore. She received her B. Eng in Financial Engineering from Nanjing University. Her research interest focuses on simulation optimization, especially issues related to optimal computing budget allocation. Her email address is liu.yue@nus.edu.sg.

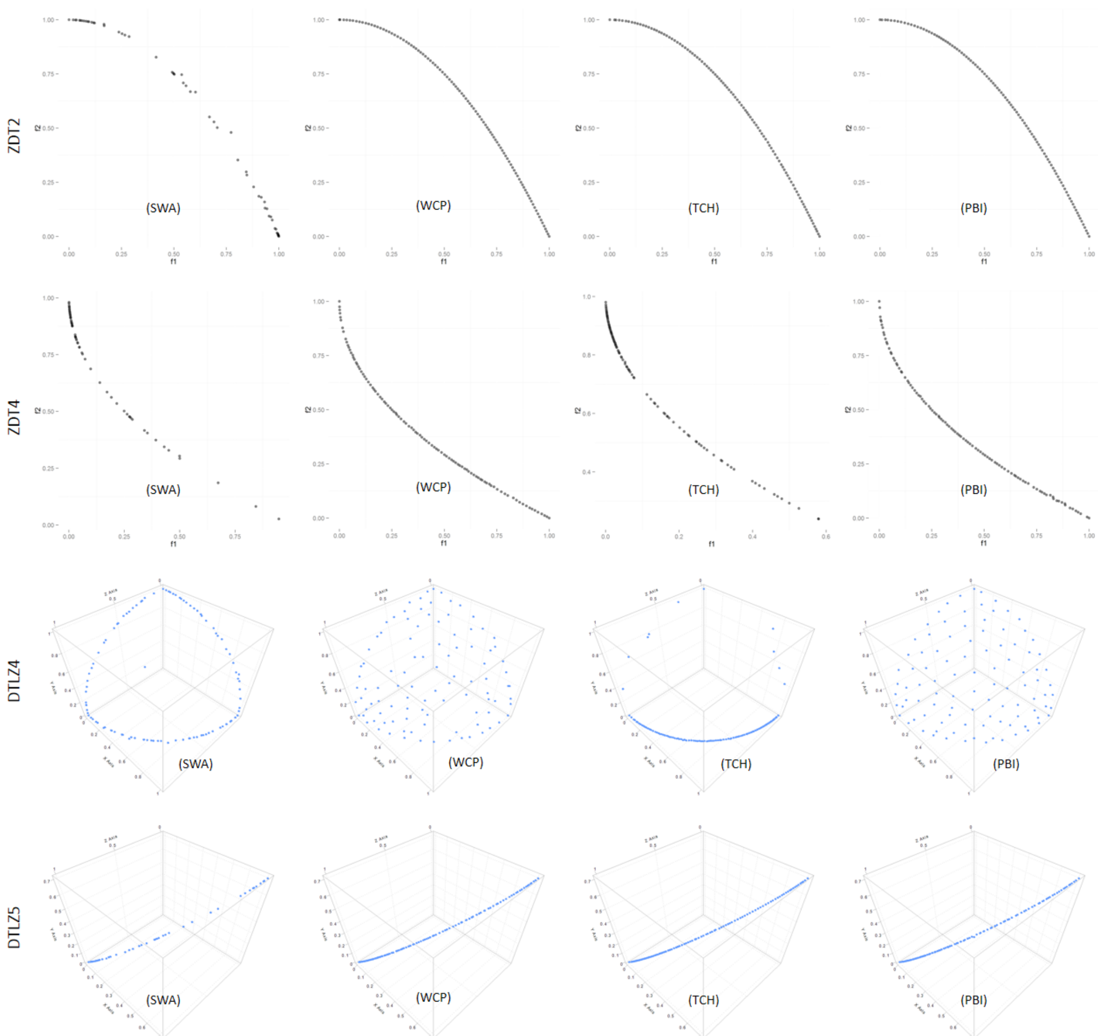

Figure 4: Pareto fronts obtained by MPSOws with four scalarization schemes.

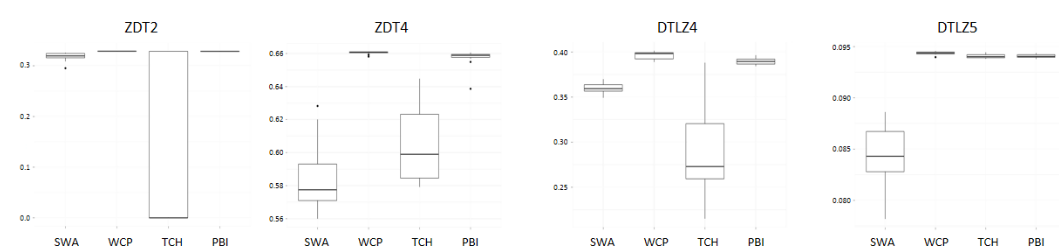

Figure 5: Dominated hypervolume achieved by MPSOws with four scalarizing schemes.

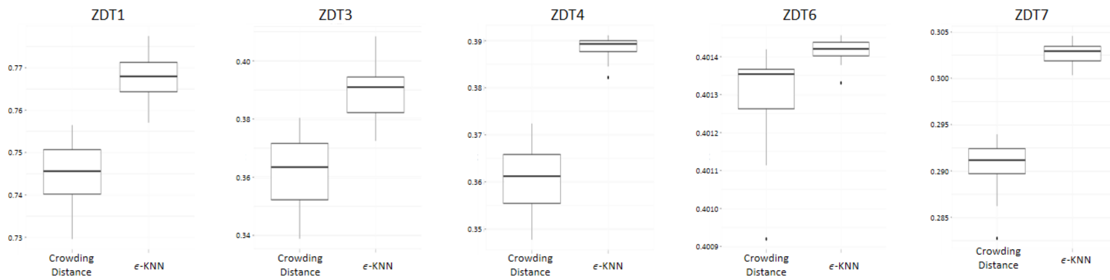

Figure 6: Dominated hypervolume (boxplot) achieved by MPSOws with two archiving strategies.

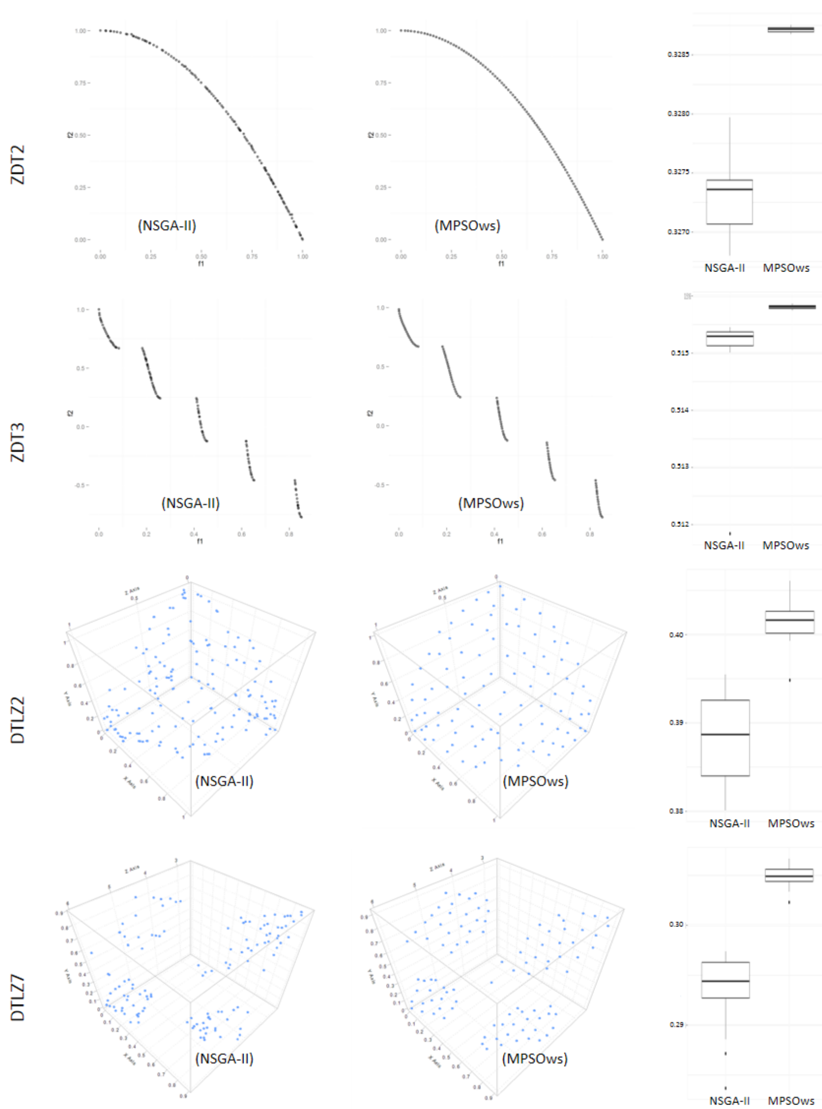

Figure 7: Pareto front and dominated hypervolume (boxplot) achieved by NSGA-II and MPSOws.