# Convergence and Diversity Evaluation for Vector Evaluated Particle Swarm Optimization

Lim Kian Sheng, Zuwairie Ibrahim, Salinda Buyamin, Anita Ahmad, Ismail Ibrahim, Mohd Falfazli Mat Jusof, Faradila Naim, Mohd Zaidi Mohd Tumari, Kamarul Hawari Ghazali

***Abstract*—Multi-objective optimization can be commonly found in many real world problems. In computational intelligence, Particle Swarm Optimization (PSO) algorithm has increasing popularity in solving optimization problems. An extended PSO algorithm called Vector Evaluated Particle Swarm Optimization (VEPSO) has been introduced to solve multi-objective optimization problems. However, VEPSO quantitative performance measure has not been investigated. Hence, in this study, the performance of VEPSO algorithm is investigated by measuring the convergence and diversity by using standard test functions. In addition, comparisons with other optimization algorithms are also conducted. The results show that the VEPSO algorithm performs weakly in solving problems with concave, mixed, and disconnected Pareto frontier and performs badly in solving multi-modal problems.**

***Index Terms*—Convergence, Diversity, Vector evaluated particle swarm optimization**

## I. Introduction

Multi-Objective Optimization (MOO) problem can be found in many real world problems [1] where there is more than one objective need to be optimized. Particle Swarm Optimization (PSO) algorithm, which has been proposed by James Kennedy and Russell Eberhart in 1995 [2], has getting more attentions recently due to its simplicity in solving optimization problems [3-4]. PSO algorithm is inspired by the social behavior of bird flocking and fish schooling to find the optimum solution.

Since the original PSO algorithm is basically introduced to solve Single-Objective Optimization (SOO) problems, a number of extended PSO algorithms such as Dynamic Neighborhood PSO [5], Multi-Objective PSO (MOPSO) [6], Another MOPSO (AMOPSO) [7], and Vector Evaluated Particle Swarm Optimization (VEPSO) [8] have been introduced for solving MOO algorithms. The VEPSO algorithm, which is based on Vector Evaluated Genetic Algorithm (VEGA) [9], requires multiple swarms where each swarm optimizes one objective and the best solution found in one swarm is transferred to the neighboring swarm. As a result, more non-dominated solutions can be found by every swarm.

To date, the VEPSO algorithm has been successfully applied in various MOO problems such as supersonic ejector [1], antenna design [10], composite structure [11], performance of power system [12], and machine scheduling [13]. Even though the usefulness of VEPSO algorithm in solving these problems has been shown by many researchers, however, no quantitative performance evaluation has been carried out at present. Thus, the effectiveness of VEPSO in producing solutions to MOO problems is still questionable. In addition, the VEPSO algorithm shares similar mechanism as in VEGA. Note that the main drawback of VEGA is that the algorithm is more favor to obtain the extreme solutions for each objective [9]. Hence, VEPSO is expected to perform badly in solving most MOO problems due to this weakness.

Hence, the main objective of this paper is to provide a quantitative performance measure of the VEPSO algorithm in term of convergence and diversity. In this work, the performance measure for convergence and diversity will be evaluated using the Generalize Distance (GD) [14] and Spread [15], respectively. Besides, the standard benchmark test functions, which are ZDT [16], DTLZ [17], and WFG [18], were chosen for validating the performance of VEPSO algorithm and for the purpose of benchmarking with Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [15] and Speed-constrained Multi-objective PSO (SMPSO) [19]. It is also worth to note that this paper will be focusing on continuous or real valued solution which has a continuous search space.

The remaining of this paper is organized as follows. The next section contains a brief description on MOO and VEPSO. In Section 3, the performance measure of the MOO algorithm will be explained. The result and discussion will be presented in Section 4. Finally, Section 5 concludes the findings of this paper.

## II. Multi-objective Optimization Algorithm

Most real problems involve more than one objective to be optimized. Usually, those objectives are conflicting with each other and hence, there will be no single solution exist that satisfies all the objectives. Consider a minimization MOO problem, which has a $j$ -dimensional search space of $x=\{x_1,\cdots,x_j\}$ contain all the possible solutions for a $k$

Manuscript received October 15, 2013. This work was supported by the UMP-MOHE RAGS Fund (RDU 121405), Fundamental Research Grant Scheme (VOT 78533), and MyPhD Scholarship from Ministry of Higher Education of Malaysia (MOHE).

Zuwairie Ibrahim, Ismail Ibrahim, Mohd Falfazli Mat Jusof, Faradila Naim, Mohd Zaidi Mohd Tumari, and Kamarul Hawari Ghazali are with the Faculty of Electrical & Electronic Engineering, Universiti Malaysia Pahang, 26600 Pekan, Malaysia (e-mail: zuwairie@ump.edu.my).

Lim Kian Sheng, Salinda Buyamin, and Anita Ahmad are with the Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia.

-objective functions of $f(x)=\{f_1(x),\cdots,f_k(x)\}$ that fulfill an $l$ -inequality constrains, $g_i(x)\le 0$ where $i=1,\cdots,l$ . The MOO problem is to find a vector, $x^*=\{x_1^*,\cdots,x_j^*\}\in x$ that is optimized for $f(x)$ while satisfying all the constraints. The conflicting objectives cause difficulty to obtain a global minimum. As a result, a concept non-dominated solution is employed to obtain a set of solutions which considers the trade-off among the objectives.

Non-dominated solutions are defined as follows. Given $u=\{u_1,\cdots,u_k\}$ and $v=\{v_i,\cdots,v_k\}$ as two vectors, $u$ *dominates* $v$ if and only if $u_i\le v_i$ for all $i$-objectives and $u_i<v_i$ for at least one objective. A solution $x$ of MOO problem is a non-dominated solution if and only if there is no other solution $x'$ that has $f(x)$ *dominate* $f(x')$. A set of non-dominated solutions in a search space is usually refered as *Pareto Optimal Set*. While, the set of objective vectors with respect to the *Pareto Optimal Set* is known as the *Pareto Optimal Front* or *Pareto Frontier*.

### A. Particle swarm optimization

Kennedy and Eberhart [2] have introduced the PSO algorithm, which is a stochastic population based optimization algorithm, for solving SOO problems. PSO algorithm is a bio-inspired algorithm by mimicking the social behavior of bird flocking and fish schooling [20]. In PSO, a swarm of individuals known as particles will fly around in a search space that contains all the possible solutions. Hence, the position of each particle represents the solution for the problem found. Each particle in the swarm will use its own and social information to move in the search space. Thus, the particles are collaborating with each other to find the optimum solution [21].

The PSO algorithm is shown in Figure 1. During the initialization, all the particles are randomly positioned in the search space and its velocity is set to zero. Then the particles' fitness is calculated before the particle own and global best positions are updated. The algorithm proceeds by updating the velocity and position based on the Eq. (1) and Eq. (2).

$$v_i^n(t+1)=\omega v_i^n(t)+c_1r_1\left(pBest_i^n+p_i^n(t)\right)+c_2r_2\left(gBest^n+p_i^n(t)\right) \quad (1)$$

$$p_i^n(t+1)=p_i^n(t)+v_i^n(t+1) \quad (2)$$

where $i$ as the number of particles in an $n$-dimensional search space. The velocity and position of the particle is denoted as $v_i(t)$ and $p_i^n(t)$, respectively. The $\omega$ is the weight inertia; while the $r_1$ and $r_2$ in Eq. (1) are random numbers range between zero and one. Besides, the $c_1$ and $c_2$ are the cognitive and social constant, respectively, that determine the influence of the own and social information toward the velocity update. The velocity update also considers two variables, which are the particle own best position, $pBest_i^n(t)$ , and the swarm best position, $gBest_i^n(t)$, respectively. After the position is updated, the fitness is calculated again and the particle's own and global best position are updated until the stopping criteria meet.

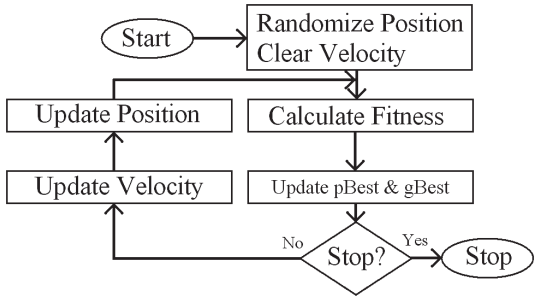



Fig. 1. The PSO algorithm.

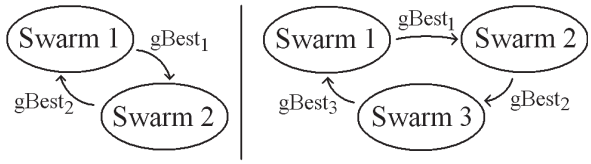



Fig. 2. Information sharing in VEPSO.

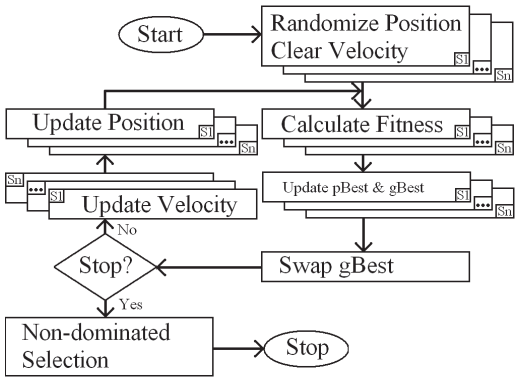



Fig. 3. The VEPSO algorithm.

### B. Vector Evaluated Particle Swarm Optimization

The main difference of VEPSO compared to the original PSO algorithm is information sharing. Specifically, the position update in one swarm is influenced by its neighbor swarm's best positions as shown in Figure 2. The VEPSO algorithm is shown in Figure 3. The steps are repeated for all swarm concurrently.

Even though each swarm searches the best solution based on its own objective, however, because of the information exchange between swarms, multiple 'trade-off' solutions could be found. After that, in order to make sure that all the solutions satisfies the *Pareto Dominance* concept, a non-dominated selection process is applied.

## III. Performance Measure for MOO Algorithm

### A. Performance measure

The first performance measure used in this study is

Generalized Distance (GD). GD is commonly used for measuring algorithm convergence ability [7, 15, 22]. GD measures the distance between the obtained *Pareto Optimal Front* (*PF*), $PF_{\text{obtained}}$, and the true *PF*, $PF_{\text{true}}$. By considering *K*-objectives problem, Eq. (3) shows how the minimum distance between the obtained solutions with all the true solutions is calculated.

$$GD = \frac{\left(\sum_{q=1}^{\|PF_{obtain}\|} d_q^K\right)^{\frac{1}{K}}}{\|PF_{obtained}\|} \tag{3}$$

where $d_q$, which is the minimum distance from $q$-th solution to the $PF_{\text{true}}$, is formulated as follows:

$$d_q = \min_{1 \le m \le \|PF_{true}\|} \sqrt{\sum_{k=1}^{K} \left(PF_{obtained}^k - PF_{true}^k\right)^2} \tag{4}$$

where $q$ represent the solution in $PF_{obtained}$ and $m$ as the solution in the $PF_{true}$ while $k$ is the index for each objective. Let the modulus, $\|\cdots\|$, be the count for the element, $(\cdots)$, the GD can be formulated as in Eq. (4).

In order to measure the diversity of non-dominated solutions, a performance measure called *Spread* [15, 19, 22] has been considered in this study. *Spread* evaluates the distance difference between all the solutions as follows:

$$Spread = \frac{d_f + d_l + \sum_{q=1}^{\|PF_{obtain}\|-1} \left|d_q - \bar{d}\right|}{d_f + d_l + \left(\|PF_{obtain}\| - 1\right)\bar{d}} \tag{5}$$

$$d_q = \sqrt{\left(PF_{obtain}{}_q^1 - PF_{obtain}{}_{q+1}^1\right)^2 + \left(PF_{obtain}{}_q^2 - PF_{obtain}{}_{q+1}^2\right)^2} \tag{6}$$

$$\bar{d} = \frac{\sum_{q=1}^{\|PF_{obtain}\|-1} d_q}{\|PF_{obtain}\| - 1} \tag{7}$$

where $d_f$ and $d_l$ are the Euclidean distance between the extreme solutions in $PF_{obtained}$ and $PF_{true}$. When all the solutions in $PF_{true}$ is arranged in descending, $d_q$ is the distance between one solution to the next solution while $\bar{d}$ is the mean distance for all the solution in the $PF_{obtained}$. Note that the calculation of *Spread* also includes the extreme solutions from the $PF_{true}$. The extreme solution is the solution that is best for one objective but is worst for another.

In both convergence and diversity measures, the obtained *PF* is produced by the VEPSO algorithm. However, the $PF_{true}$ requires well-defined non-dominated solutions for each MOO problems. Therefore, the $PF_{true}$ used in this work will be based on the standard database from the jMetal (http://jmetal.sourceforge.net/problems.html).

### B. Test problems

Zitzler, Deb and Thiele [16] have designed six MOO test problems where each problem focuses on one kind of problem feature. These problems were abbreviated as ZDT1 to ZDT6. However, in this study, the ZDT5-based evaluation is not considered since the ZDT5 is a binary coded test problem for discrete optimization problems.

In this study, another common test problems called DTLZ [17] is considered as well. The DTLZ, which is abbreviated from Deb, Thiele, Laumanns, and Zitzler, consists of seven MOO test problems in order to extensively evaluate different features of MOO problems.

A disadvantage of ZDT and DTLZ is that both test problems are separable and degenerate [24]. Hence, another test problem called WFG has been proposed by Huband *et al*. [18]. WFG test problem is also chosen in this study.

## IV. Experiment, Results and Discussion

The parameters used for the test problems followed the original papers of ZDT [16] and DTLZ [17], whereas for the WFG [18], similar parameters in [24] were used [23]. Meanwhile, the number of objective for all test problems was restricted to two objectives and the VEPSO parameters were set as in Table 1. The experiments for each problem were repeated for 100 runs and then the average convergence and diversity values are calculated.

For better analysis, the NSGA-II and SMPSO were used as benchmarking since these two algorithms have showed good convergence and diversity performance [19, 24]. The parameters used in these two algorithms were similar to the paper by Durillo *et al.* [19].

TABLE I
VEPSO Parameters

| Parameter | Value |
|---|---|
| Function Evaluation | 250000 |
| × Number of Swarm | 2 |
| × Particle for each Swarm | 50 |
| × Iterations | 250 |
| $c_1$ & $c_2$ | 0.5 |
| $\omega$ | Linearly degrade from 0.9 to 0.4 |

Table 2 shows the convergence performance of VEPSO, NSGA-II, and SMPSO based on ZDT, DTLZ, and WFG test problems. For every problem, the value in bold indicates the worst performance of a particular algorithm. The results, which are based on convergence, show that VEPSO produces the worst solutions compared to NSGA-II and SMPSO.

Figure 4, Figure 5, and Figure 6 show the non-dominated solutions obtained using the VEPSO, NSGA-II and SMPSO

TABLE II
CONVERGENCE RESULTS

| Problem | VEPSO | NSGA-II | SMPSO |
|---|---|---|---|
| ZDT1 | 0.597924 | 0.000223 | 0.000123 |
| ZDT2 | 0.960806 | 0.000171 | 0.000051 |
| ZDT3 | 0.361527 | 0.000212 | 0.000205 |
| ZDT4 | 39.781259 | 0.000469 | 0.000128 |
| ZDT6 | 1.678357 | 0.001031 | 0.012306 |
| DTLZ1 | 128.249513 | 0.000379 | 0.051440 |
| DTLZ2 | 0.091085 | 0.000275 | 0.000238 |
| DTLZ3 | 298.489315 | 0.147489 | 2.729057 |
| DTLZ4 | 0.096655 | 0.000235 | 0.000258 |
| DTLZ5 | 0.091237 | 0.000268 | 0.000241 |
| DTLZ6 | 4.074105 | 0.206694 | 0.054544 |
| DTLZ7 | 1.179065 | 0.000125 | 0.000099 |
| WFG1 | 0.159581 | 0.016281 | 0.047240 |
| WFG2 | 0.061234 | 0.000934 | 0.004538 |
| WFG3 | 0.030675 | 0.000784 | 0.001793 |
| WFG4 | 0.023618 | 0.000705 | 0.003042 |
| WFG5 | 0.054786 | 0.002766 | 0.002660 |
| WFG6 | 0.063283 | 0.002414 | 0.001671 |
| WFG7 | 0.028939 | 0.000459 | 0.001703 |
| WFG8 | 0.054860 | 0.006657 | 0.009110 |
| WFG9 | 0.039078 | 0.004573 | 0.001455 |

algorithm, respectively. For each algorithm, three different test problems, ZDT4, DTLZ1 and WFG1 were randomly selected to compare the non-dominated solutions obtained. In ZDT4 and DTLZ1 test problems, notice that the non-dominated solutions obtained using VEPSO were located far away from the actual *PF* or $PF_{true}$ whereas the NSGA-II and SMPSO solutions fall perfectly on the $PF_{true}$. Besides, the distance of the non-dominated solutions obtained by the VEPSO is the largest when compared with the other two algorithms. Due to the larger distance of the obtained non-dominated solutions from the $PF_{true}$, the GD of the VEPSO was larger when compared to NSGA-II and SMPSO. This large GD has show the convergence weakness in VEPSO where the solutions obtained have a large fitness error between the obtained and true solutions for the problem.

Table 4 shows the diversity performance of VEPSO, NSGA-II, and SMPSO based on ZDT, DTLZ, and WFG test problems. For every problem, the value in bold indicates the worst performance of a particular algorithm. The results, which are based on diversity, show that VEPSO produces the worst solutions compared to NSGA-II and SMPSO except in DTLZ3, DTLZ6 and WFG2.

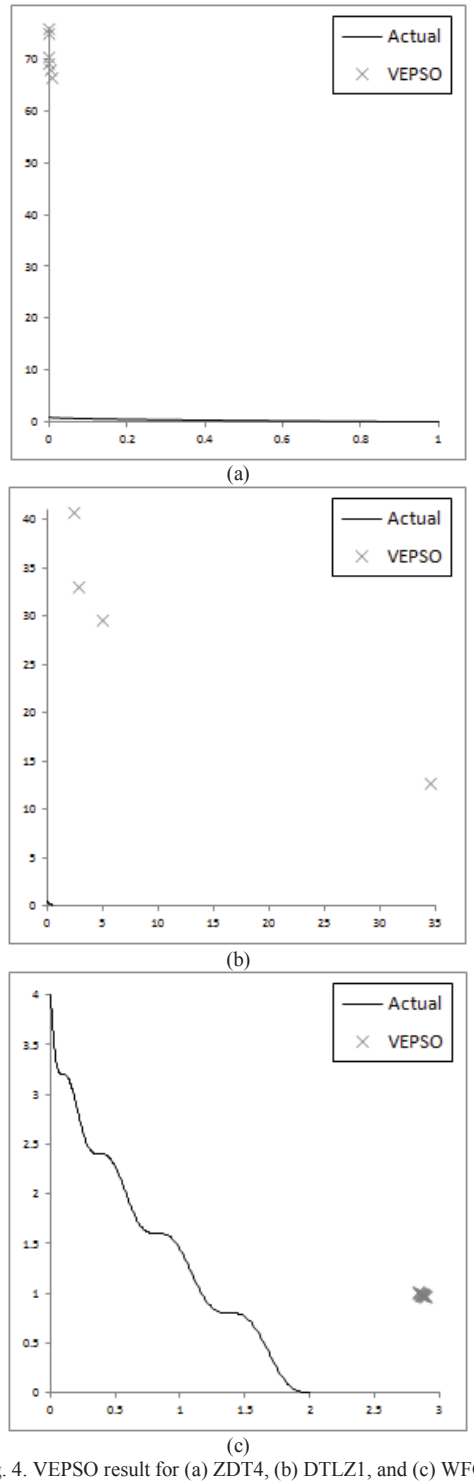
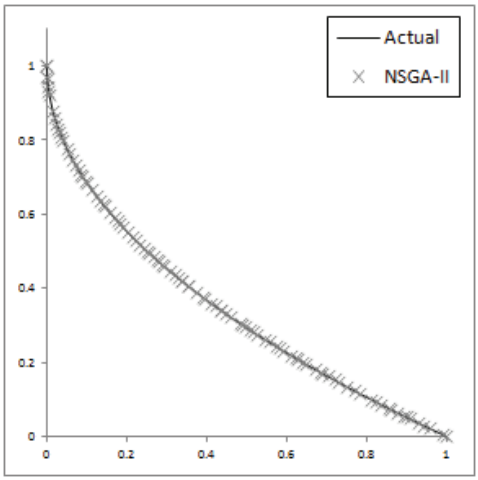
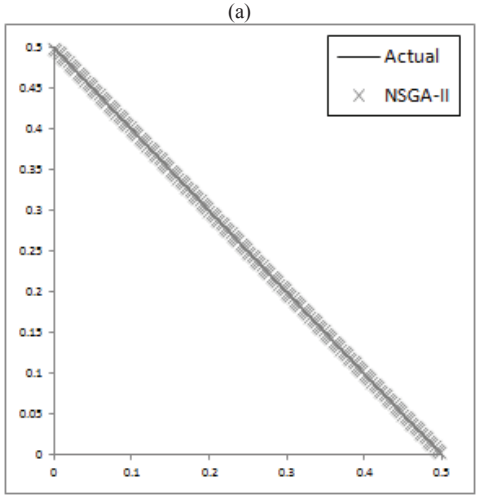


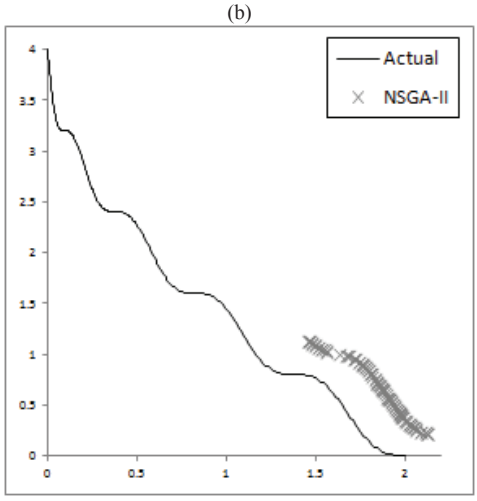

Fig. 4. VEPSO result for (a) ZDT4, (b) DTLZ1, and (c) WFG1.




(a)



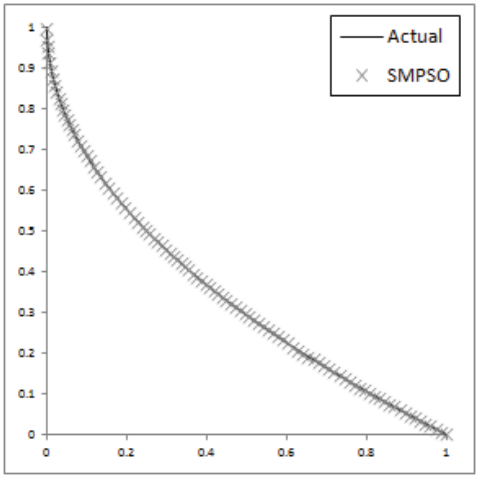

(b)



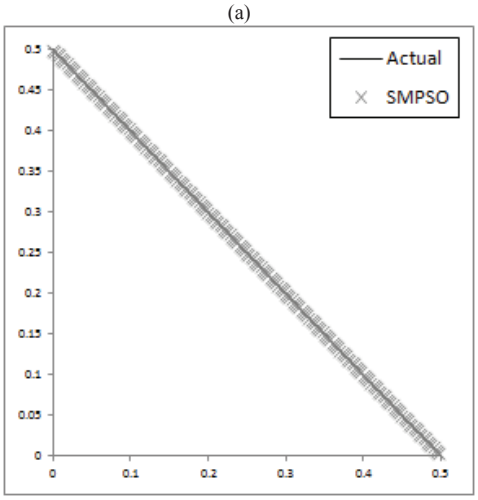

(c)

Fig. 5. NGSA-II result for (a) ZDT4, (b) DTLZ1, and (c) WFG1.

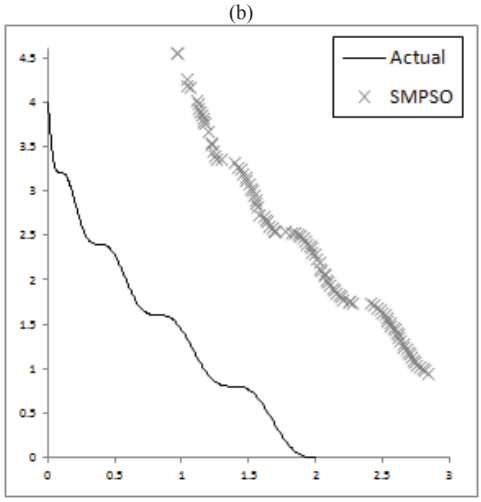
Referring back to Figure 3(a), and Figure 3(c), the non-dominated solutions obtained using VEPSO were accumulated nearer to the extreme solution which left the middle region having less solutions. Thus, the distance difference between obtained solutions become small while having larger distance between the extreme solution from $PF_{true}$ and $PF_{obtained}$, which result in a larger *Spread* or poor diversity performance. Besides, the solutions in Figure




(a)




(b)




(c)

Fig. 6. SMPSO result for (a) ZDT4, (b) DTLZ1, and (c) WFG1.

3(b) are widely spread but having inconsistence distance from one solution to the next, so the Spread measure is larger when comparing to the solutions obtained in Figure 4 and Figure 5. Therefore, the larger *Spread* measure show a weak diversity performance for the VEPSO which mean the obtained solutions does not cover the possible solution well for all the objectives.

TABLE II
SPREAD RESULTS

| Problem | VEPSO | NSGA-II | SMPSO |
|---|---|---|---|
| ZDT1 | 0.895581 | 0.366227 | 0.075971 |
| ZDT2 | 0.971432 | 0.381633 | 0.072996 |
| ZDT3 | 0.879105 | 0.748008 | 0.715525 |
| ZDT4 | 0.920244 | 0.399302 | 0.091404 |
| ZDT6 | 0.959756 | 0.358824 | 0.419699 |
| DTLZ1 | 0.790765 | 0.401470 | 0.083125 |
| DTLZ2 | 0.699262 | 0.374665 | 0.130057 |
| DTLZ3 | 0.771138 | 0.854892 | 0.397225 |
| DTLZ4 | 0.979001 | 0.485996 | 0.117719 |
| DTLZ5 | 0.697678 | 0.368575 | 0.126805 |
| DTLZ6 | 0.806422 | 0.846856 | 0.336035 |
| DTLZ7 | 0.944714 | 0.622253 | 0.518938 |
| WFG1 | 0.995679 | 0.868861 | 0.666307 |
| WFG2 | 0.883157 | 0.799965 | 0.955189 |
| WFG3 | 0.852420 | 0.374224 | 0.426410 |
| WFG4 | 0.788151 | 0.564849 | 0.372959 |
| WFG5 | 0.794173 | 0.402397 | 0.150457 |
| WFG6 | 0.788419 | 0.391880 | 0.208574 |
| WFG7 | 0.736291 | 0.379704 | 0.261381 |
| WFG8 | 0.817689 | 0.442193 | 0.394980 |
| WFG9 | 0.710053 | 0.437133 | 0.255571 |

## V. CONCLUSIONS

In this study, the original VEPSO's convergence and diversity performance have been quantitatively evaluated using *Generalize Distance* (GD) and *Spread* measurement, respectively. The original VEPSO algorithm has difficulty in producing good non-dominated solutions as shown by the large fitness error between the obtained and true solutions. Besides, the diversity performance of VEPSO is limited where the solutions obtained do not cover most of the possible solutions but concentrate at one area only. Therefore, there are still rooms for improvement, in term of convergence and diversity performance, for the VEPSO algorithm.

## REFERENCES


[1] S. M. V. Rao and G. Jagadeesh, *Vector Evaluated Particle Swarm Optimization (VEPSO) of Supersonic Ejector for Hydrogen Fuel Cells.* Journal of Fuel Cell Science and Technology, 2010. 7(4): p. 041014-7.

[2] J. Kennedy and R. Eberhart, *Particle swarm optimization*, in *Neural Networks, 1995. Proceedings., IEEE International Conference on*. 1995. p. 1942-1948.

[3] D. Besozzi, P. Cazzaniga, G. Mauri, D. Pescini, and L. Vanneschi, *A Comparison of Genetic Algorithms and Particle Swarm Optimization for Parameter Estimation in Stochastic Biochemical Systems*, in *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, C. Pizzuti, M. Ritchie, and M. Giacobini, Editors. 2009, Springer Berlin / Heidelberg. p. 116-127.

[4] R. Hassan, B. Cohanim, O. De Weck, and G. Venter. *A Comparison Of Particle Swarm Optimization And The Genetic Algorithm*. in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2005.

[5] X. Hu and R. Eberhart, *Multiobjective optimization using dynamic neighborhood particle swarm optimization*, in *Proceedings of the Evolutionary Computation on 2002. CEC '02*. 2002, IEEE Computer Society. p. 1677-1681.

[6] C. A. C. Coello and M. S. Lechuga. *MOPSO: a proposal for multiple objective particle swarm optimization*. in *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*. 2002.

[7] G. T. Pulido and C. A. Coello Coello, *Using Clustering Techniques to Improve the Performance of a Multi-objective Particle Swarm Optimizer*, in *Genetic and Evolutionary Computation – GECCO 2004*. 2004, Springer Berlin / Heidelberg. p. 225-237.

[8] K. E. Parsopoulos and M. N. Vrahatis, *Particle swarm optimization method in multiobjective problems*, in *Proceedings of the 2002 ACM symposium on Applied computing*. 2002, ACM: Madrid, Spain. p. 603-607.

[9] J. D. Schaffer, *Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition)*. 1984, Vanderbilt University. p. 166.

[10] D. Gies and Y. Rahmat-Samii, *Vector evaluated particle swarm optimization (VEPSO): optimization of a radiometer array antenna*, in *Antennas and Propagation Society International Symposium, 2004. IEEE*. 2004. p. 2297-2300.

[11] S. N. Omkar, D. Mudigere, G. N. Naik, and S. Gopalakrishnan, *Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures.* Computers & Structures, 2008. 86(1-2): p. 1-14.

[12] J. G. Vlachogiannis and K. Y. Lee, *Review: Multi-objective based on parallel vector evaluated particle swarm optimization for optimal steady-state performance of power systems.* EXPERT SYSTEMS WITH APPLICATIONS, 2009. 36(8): p. 10802-10808.

[13] J. Grobler, *Particle swarm optimization and differential evolution for multi objective multiple machine scheduling*, in *Department of Industrial and Systems Engineering* 2009, University of Pretoria: Pretoria, South Africa. p. 159.

[14] D. A. V. Veldhuizen, *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. 1999, Air Force Institute of Technology. p. 249.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, *A fast and elitist multiobjective genetic algorithm: NSGA-II.* IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, 2002. 6(2): p. 182-197.

[16] E. Zitzler, K. Deb, and L. Thiele, *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results.* EVOLUTIONARY COMPUTATION, 2000. 8(2): p. 173-195.

[17] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multi-Objective Optimization*, in *KanGAL Report 2001001*. 2001, Kanpur Genetic Algorithms Laboratory, Indian Institute of Technology Kanpur: Kanpur, India. p. 27.

[18] S. Huband, L. Barone, L. While, and P. Hingston, *A Scalable Multi-objective Test Problem Toolkit*, in *Evolutionary Multi-Criterion Optimization*, C.A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Editors. 2005, Springer Berlin / Heidelberg. p. 280-295.

[19] J. Durillo, J. García-Nieto, A. Nebro, C. Coello, F. Luna, and E. Alba, *Multi-Objective Particle Swarm Optimizers: An Experimental Comparison*, in *Evolutionary Multi-Criterion Optimization*, M. Ehrgott, et al., Editors. 2009, Springer Berlin / Heidelberg. p. 495-509.

[20] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*. The Morgan Kaufmann Series in Evolutionary Computation, ed. D.B. Fogel. 2001, San Francisco: Morgan Kaufmann Publishers. 512.

[21] H. El-Sayed, M. Belal, A. Almojel, and J. Gaber, *Swarm Intelligence*, in *Handbook of Bioinspired Algorithms and Applications*, S. Olariu and A.Y. Zomaya, Editors. 2006, Taylor AND Francis Group: Boca Raton, FL, USA. p. 55 - 63.

[22] S.-K. S. Fan and J.-M. Chang, *A parallel particle swarm optimization algorithm for multi-objective optimization problems.* Engineering Optimization, 2009. 41(7): p. 673 - 697.

[23] S. Huband, P. Hingston, L. Barone, and L. While, *A review of multiobjective test problems and a scalable test problem toolkit.* IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, 2006. 10(5): p. 477-506.

[24] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba. *SMPSO: A new PSO-based metaheuristic for multi-objective optimization*. in *Computational intelligence in miulti-criteria decision-making, 2009. mcdm '09. ieee symposium on*. 2009.