New Trends in Software Methodologies, Tools and Techniques
H. Fujita et al. (Eds.)
IOS Press, 2014

doi:10.3233/978-1-61499-434-3-170

Performance Evaluation of Vector Evaluated Gravitational Search Algorithm II

Badaruddin MUHAMMAD[a1], Zuwairie IBRAHIM[a], Kamarul Hawari GHAZALI[a], Mohd Riduwan GHAZALI[a], Muhammad Salihin SAEALAL,[a] Kian Sheng Lim[b], Sophan Wahyudi NAWAWI,[b] Nor Azlina AB. AZIZ,[c] Marizan MUBIN,[d] and Norrima MOKHTAR[d]

[a]*Faculty of Electrical and Electronics Engineering*
Universiti Malaysia Pahang, Malaysia
[b]*Faculty of Electrical Engineering*
Universiti Teknologi Malaysia, Malaysia
[c]*Faculty of Electrical Engineering*
Multimedia University, Malaysia
[d]*Applied Control and Robotics Laboratory*
Faculty of Engineering
University of Malaya, Malaysia

Abstract. This paper presents a performance evaluation of a novel Vector Evaluated Gravitational Search Algorithm II (VEGSAII) for multi-objective optimization problems. The VEGSAII algorithm uses a number of populations of particles. In particular, a population of particles corresponds to one objective function to be minimized or maximized. Simultaneous minimization or maximization of every objective function is realized by exchanging a variable between populations. The results shows that the VEGSA is outperformed by other multi-objective optimization algorithms and further enhancements are needed before it can be employed in any application.

Keywords. Gravitational search algorithm, optimization, vector evaluated

Introduction

The Gravitational Search Algorithm (GSA) has been firstly introduced by Rashedi *et al.* in 2009 [1]. The population-based optimization algorithm is derived based on the Newtonion Law of Gravity and the law of motion. GSA has been found superior to some well-established optimization algorithms, such as Central Force Optimization (CFO) [2] and Particle Swarm Optimization (PSO) [3].

In order to apply the GSA algorithm to multi-objective optimization problems, a number of variants of GSA algorithms have been reported. The first variant is called Multi-Objective GSA (MOGSA) [4]. Later, Nobahari *et al.* have proposed a Non-dominated Sorting GSA (NSGSA) [5]. Recently, a novel approach for handling multiple objectives using GSA is proposed. The proposed approach is called Vector Evaluated

[1] Corresponding authors: Faculty of Electrical and Electronic Engineering, Universiti Malaysia Pahang, Malaysia; Email: zuwairie@ump.edu.my

GSA (VEGSAII) [6]. VEGSAII uses a number of populations of particles in which a population corresponds to one objective and the multi-objective optimization is realized by exchanging a variable between populations. Specifically, the direction of a particle is not only determined by all the particles in its population, but also with the addition of the best particle of its neighboring population.

The purpose of this paper is to investigate the performance of the VEGSAII algorithm. ZDT test function is used as the benchmark multi-objective optimization problem and the performance of the VEGSAII algorithm is evaluated in terms of *Number of Solutions* (*NS*), *Generational Distance* (*GD*), and *Spread*. To conclude the finding, the performance of VEGSAII algorithm is compared to the existing NGSA-II, SPEA2, and SMPSO algorithms.

1. Gravitational Search Algorithm

Searching in GSA is performed by a set of *N* particles. The position of the *i*th particle in *n* dimensions is denoted as $X_i = (x_i^1, x_i^2, \dots, x_i^n)$ for i = 1, 2, …. , *N*. In a particular *d*th dimension, the position of *i*th particle can be represented as x_i^d. In GSA, every particle has its own mass. The mass of *i*th particle is influenced by fitness value, fit_i, which is subjected to the position of the particle in a search space. The mass of *i*th particle is updated as follows:

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{1}$$

where $m_i(t)$ is defined as

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{2}$$

The *best*(*t*) and *worst*(*t*) are defined subjected to the optimization problem.

The flowchart of GSA is shown in Fig. 1. During initialization, the positions of particles are randomly positioned in the search space. The velocity of each particle and the iteration number, *t*, are set as zero. Gravitational constant is also initialized. Then, the fitness of each particle, fit_i, is calculated according to objective function. After that, the gravitational constant, *G*(*t*), is updated based on the following equation:

$$G(t) = G_o \times e^{\beta \frac{t}{T}} \tag{3}$$

where *T* is the total number of iterations. The next step is to calculate the mass, *M*, and acceleration, α, for each particle. The mass, *M*, is calculated based on Eq. (2). Acceleration of *i*th particle in *d*th dimension, $\alpha_i^d(t)$, determines the direction of a particle and can be calculated based on the law of motion, as follows:

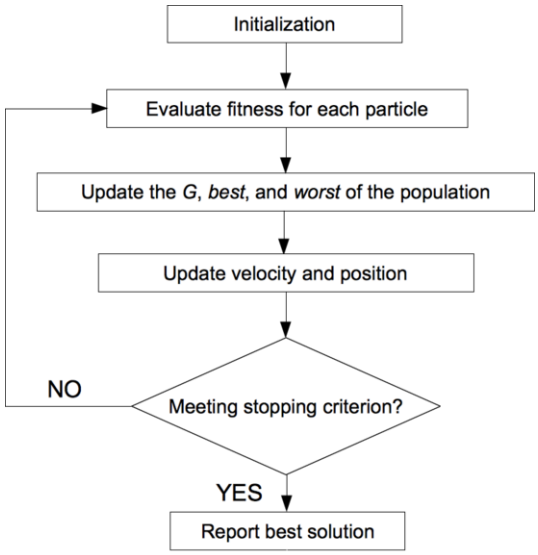

Figure 1. Flowchart of Gravitational Search Algorithm

$$\alpha_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{4}$$

where M_{ii} is called inertia mass of *i*th particle. Note that $M_{ii} = M_i$. The total force, F_i^d, that act on particle *i*th in a dimension *d*, is calculated as follows:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t) \tag{5}$$

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}(x_j^d(t) - x_i^d(t)) \tag{6}$$

where M_{pi} is the passive gravitational mass related to particle *i*, M_{aj} is the active gravitational mass related to particle *j*, ε is a small constant, R_{ij} is the Euclidian distance between particle *i* and *j*, and $rand_j$ is a uniform random variable in the interval [0,1]. Finally, the velocity, v_i^d, and position, x_i^d, of *i*th mass are updated as follows:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + \alpha_i^d(t) \tag{7}$$

$$x_i^d(t+1) = x_i^d(t) \times v_i^d(t+1) \tag{8}$$

where $rand_i$ is another uniform random variable in the interval [0,1]. The algorithm ends if the stopping criterion is met.

2. Vector Evaluated Gravitational Search Algorithm (VEGSA)

The VEGSA assumes M populations of P_1, P_2, P_3, … , P_M, of size N aim to simultaneously optimize M objective functions. Each population optimizes one objective function. Information transfer between populations, as shown in Fig. 2, is introduced to promote trade-off between objectives. Two versions of VEGSA algorithm are proposed in this paper, namely, VEGSAI and VEGSAII. Both algorithms occupy an archive to store the non-dominated solutions and this archive is updated at every iteration. However, this study focuses on VEGSAII only. In VEGSAII, the direction of a particle is not only determined by all the particles in its population, but also with the addition of the best particle of its neighbouring population.

Let m be the index of a population, $m = \{1, 2, \dots, M\}$ and fit_j^m be the fitness of jth particle of the mth population. Eq. (6) is modified as follows:

$$F_i^{d,m}(t) = rand \times F_{i,m-1}^d(t) + \sum_{j=1, j\neq i}^{N} rand_j \times F_{ij}^{d,m}(t) \tag{9}$$

$$F_{i,m-1}^d(t) = G(t)\frac{M_{pi}^m \times M_{a,m-1}^m}{R_{i,m-1}^k + \varepsilon}\left(x_{m-1}^d(t) - x_i^d(t)\right) \tag{10}$$

where $F_i^{d,m}$ is the total force that act on particle ith in a dimension d of population m. The force that act on ith particle by the best particle in a neighboring population is denoted by $F_i^{d,m-1}$.

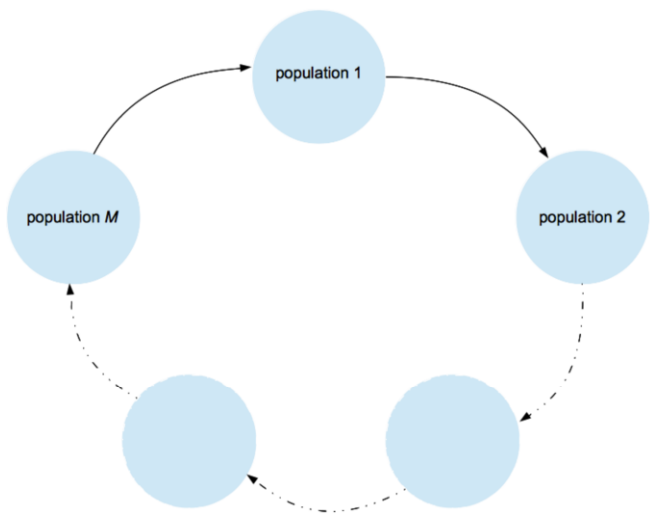

Figure 2. Information transfer between populations in Vector Evaluated Gravitational Search Algorithm

Table 1. The parameter and its value used in experiments

Parameter	Value
Iteration	250
Number of agent per swarm, *N*	50
Archive size	100
G_0	100
ε	2^{-52}
β	20
Number of run	30

3. Performance Evaluation

The parameter values used in the experiments are shown in Table 1. In VEGSAII algorithm, an archive is introduced to maintain and to update the non-dominated solutions at every iteration. The size of the archive chosed in this study is 100. In this study, the ZDT [7] benchmark test problems were used to validate the performance of the algorithm. The ZDT includes six test problems. However, the ZDT5, which is used for binary evaluation, has been excluded because this study focuses on the continuous search space problem. The parameters used for the test problems are based on [7].

Three quantitative performance measures, which are *Number of Solutions* (*NS*), *Generational Distance* (*GD*), and *Spread* have been used to evaluate the performance of the VEGSA. The *NS* is calculated based on the number of nondominated solutions found at the end of the iteration. The *GD* measure [8] represents the average distance between the Pareto front obtained, PF_o, and the true Pareto front, PF_t, as formulated in Eq. (11) and Eq. (12).

$$GD = \frac{\left(\sum_{i=1}^{|\mathcal{PF}_o|} {d_i}^M\right)^{1/M}}{|\mathcal{PF}_o|} \tag{11}$$

$$d_i = \min_{1 \le k \le |\mathcal{PF}_t|} \sqrt{\sum_{j=1}^{M} \left(f_{j,i} - f_{j,k}\right)^2} \tag{12}$$

This measure estimates how close the PF_o lies to the PF_t. Hence, a smaller *GD* value represents better performance. The *Spread* [9] is used to measure the extent of the PF_o distribution of the along the PF_t. Eq. (13), Eq. (14), and Eq. (15) show the calculation of *Spread*.

$$Spread = \frac{d_f + d_l + \sum_{i=1}^{|\mathcal{PF}_o|-1} |d_i - \bar{d}|}{d_f + d_l + (|\mathcal{PF}_o| - 1)\bar{d}} \tag{13}$$

$$\bar{d} = \frac{\sum_{i=1}^{|\mathcal{PF}_o|-1} d_i}{|\mathcal{PF}_o| - 1} \tag{14}$$

$$d_i = \sqrt{\left(f_{1,\mathrm{i}} - f_{1,\mathrm{i}+1}\right)^2 + \left(f_{2,\mathrm{i}} - f_{2,\mathrm{i}+1}\right)^2} \tag{15}$$

where d_f is the Euclidean distance between the first extreme member in PF_o and PF_t, and d_l is the Euclidean distance between the last extreme member in PF_o and PF_t. A smaller *Spread* value shows better performance.

Three different MOO algorithms were selected for performance comparison. These algorithms are the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [9], the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [10], and the Speed-constrained Multi-objective PSO (SMPSO) [11]. The parameter settings of these algorithms are as follows:

- NSGA-II has been frequently used for performance comparison because of its excellent performance. The population size of NSGA-II was set to 100. The Simulated Binary Crossover (SBX) and polynomial mutation operators were used with the crossover probability ρ_c = 0:9 and the mutation probability ρ_m = 1/N. The distribution index for both operators was set to $\mu_c = \mu_m = 20$.
- The SPEA2 algorithm used the same parameters as NSGA-II.

Table 2. VEGSAII performance based on ZDT Test Problems

Performance Measure		ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
NS	Average	55.36	96.03	99.66	97.43	84.13
	Std.	32.97	8.85	0.82	6.85	20.53
GD	Average	0.045	0.010	0.010	0.015	0.009
	Std.	0.010	0.002	0.0003	0.001	0.005
Spread	Average	0.94	0.74	0.65	0.77	0.71
	Std.	0.22	0.14	0.05	0.07	0.18

Table 3. VEGSAII performance againts NGSA-II, SPEA2, and SMPSO Algorithms

ZDT Test Problem	Algorithm	*NS*	*GD*	*Spread*
ZDT1	NSGA-II	100	0.000223	0.379129
	SPEA2	100	0.000220	0.148572
	SMPSO	100	0.000117	0.076608
	VEGSAII	55.36	0.045	0.94
ZDT2	NSGA-II	100	0.000176	0.378029
	SPEA2	100	0.000182	0.158187
	SMPSO	100	0.000051	0.071698
	VEGSAII	96.03	0.002	0.74
ZDT3	NSGA-II	100	0.000211	0.747853
	SPEA2	100	0.000230	0.711165
	SMPSO	99.9	0.000203	0.717493
	VEGSAII	99.66	0.10	0.65
ZDT4	NSGA-II	100	0.000486	0.392885
	SPEA2	100	0.000923	0.298269
	SMPSO	100	0.000134	0.092281
	VEGSAII	97.43	0.015	0.77
ZDT6	NSGA-II	100	0.001034	0.357160
	SPEA2	100	0.001761	0.226433
	SMPSO	100	0.012853	0.390481
	VEGSAII	84.13	0.009	0.71

- The population size of the SMPSO was set to 100, and a maximum of 250 iterations was employed. The C_1 = C_2 = Random[1.5, 2.5] and the ω = Random[0.1, 0.5]. Additionally, 15% of the particles in the SMPSO algorithm are subjected to polynomial mutation with $\rho_m = 1/N$ and $\mu_m = 20$.

The performance of VEGSAII algorithm in terms of *NS*, *GD*, and *Spread* is tabulated in Table 2. Table 3 shows the performance of VEGSAII algorithm against NSGA-II, SPEA2, and SMPSO. These results show that the number of non-dominated solution obtained by VEGSAII is the worst for the ZDT1 problem. For the case of ZDT2, ZDT3, ZDT4, and ZDT6, the number of non-dominated solutions obtained are less than the number of non-dominated solutions obtained by the state-of-the-art algorithms. Note that the size of archieve is limited to 100 in this study. The value of *GD* and *Spread* measures are also significantly higher than NSGA-II, SPEA2, and SMPSO in most cases. The VEGSAII algorithm exhibits better *Spread* values than the state-of-the-art algorithms only for the case of ZDT3.

4. Conclusions

The primary objective of this study is to perform performance evaluation of the newly introduced VEGSAII algorithm. The VEGSAII algorithm requires a number of populations of agents. The number of population is equal to the number of objective. Simultaneous minimization or maximization of every objective function is realized by exchanging a variable between populations. For the case of VEGSAII, the direction of a particle is not only determined by all the particles in its population, but also with the addition of the best particle of its neighboring population. Based on ZDT test problem and by examining its performance in terms of *NS*, *GD*, and *Spread*, it is found that the current VEGSAII algorithm is still immature and further enhancements are needed before it is ready to be employed in any application.

Acknowledgment

The authors are grateful to the Ministry of Higher Education (MOHE) and Universiti Malaysia Pahang (UMP) for the financial support through the Fundamental Research Grant Scheme (FRGS) (RDU130117). This work is also funded by Universiti Malaya under UM-UMRG Scheme (CG031-2013). The first author is thankful to Universiti Malaysia Pahang (UMP) for granting him an opportunity to further his study in PhD degree.

References

[1] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, pp. 2232-2248, June 2009.

[2] R.A. Formato, "Central Force Optimization with Variable Initial Probes and Adaptive Decision Space", *Applied Mathematic and Computation*, vol. 217, pp. 8866-8872, July 2011.

[3] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[4] H.R. Hassanzadeh, M. Rohani, "A Multi-Objective Gravitational Search Algorithm," *Second International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 7-12, 2010.

[5] H. Nobahari, M. Nikusokhan, P. Siarry, "Non-dominated Sorting Gravitational Search Algorithm," *International Conference on Swarm Intelligence*, June 2011.

[6] Ibrahim, Z., Muhammad, B., Ghazali, K.H., Kian Sheng Lim, Nawawi, S.W., Yusof, Z.M., "Vector Evaluated Gravitational Search Algorithm (VEGSA) for Multi-objective Optimization Problems," *Fourth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 13-17, 2012.

[7] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results", *Evolutionary Computation*, vol. 8, no. 2, pp. 173-195, 2000.

[8] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations", in *Faculty of the Graduate School of Engineering*. 1999, Air Force Institute of Technology, Air University. pp. 249.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[10] E. Zitzler, M. Laumanns, and L. Thiele. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pp. 95-100, 2002.

[11] J. Durillo, J. García-Nieto, A. Nebro, C. A. Coello Coello, F. Luna, and E. Alba, "Multi-Objective Particle Swarm Optimizers: An Experimental Comparison", *Evolutionary Multi-Criterion Optimization*, Springer Berlin / Heidelberg, pp. 495-509, 2009.