Vector Evaluated Gravitational Search Algorithm Assisted by Non-dominated Solutions in Solving Multiobjective Optimization Problems

Badaruddin Muhammad[1], Khairul Hamimah Abas[2], and Mohd Riduwan Ghazali[1]

[1]Faculty of Electrical and Electronics Engineering, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
[2]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

Abstract—Previously, non-dominated solutions have been employed to improve the performance of particle swarm optimization (PSO). In this paper, we re-implement the same concept to gravitational search algorithm (GSA). The performance is investigated by solving a set of ZDT test problem. An analysis also is performed by varying the value of initial gravitational constant.

Keywords—gravitational search algorithm; non-dominated solution; ZDT

1. INTRODUCTION

The Gravitational Search Algorithm (GSA) has been introduced in 2009 as an alternative approach for solving optimization problems [1]. GSA is inspired by nature (gravity) and it is belonging to a class of population-based meta-heuristics. In GSA, agents are considered as an object and their performance are expressed by their masses. The position of particle is corresponding to the solution of the problem. Consider a population consists of *N* agents, the position of *i*th agent can be presented by:

$$X_i = \left(x_i^1 \dots x_i^d \dots x_i^n\right) \; for \; i = 1,2,\dots,N \tag{1}$$

The mass of *i*th particle at time *t* is derived from Eq. (2) and Eq. (3), denoted as $M_i(t)$.

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

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{3}$$

where *N* is a population size, $m_i(t)$ is an intermediate variable in agent mass calculation, $fit_i(t)$ is the fitness value of *i*th agent at time *t*, $best(t)$ and $worst(t)$ denote the best and the worst fitness value of the population at time *t*. The best and the worst fitness for the case of minimization problem are defined as follows;

$$\begin{gathered} best(t) = \min_{j\in\{1,\dots,N\}} fit_j(t) \\ worst(t) = \max_{j\in\{1,\dots,N\}} fit_j(t) \end{gathered} \tag{4}$$

whereas for maximization problem,

$$\begin{gathered} best(t) = \max_{j\in\{1,\dots,N\}} fit_j(t) \\ worst(t) = \min_{j\in\{1,\dots,N\}} fit_j(t) \end{gathered} \tag{5}$$

At specific time "*t*", the force acting on agent "*i*" from agent "*j*" in *d*th dimension can be represented as the following:

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

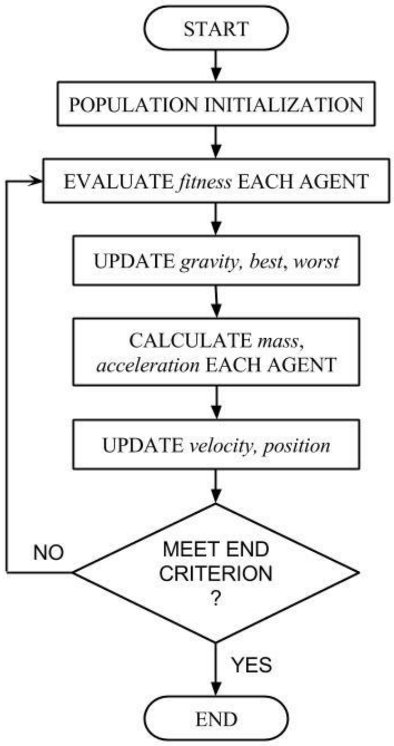

Figure 1 The gravitational search algorithm

where $M_{pi}(t)$ is the passive gravitational mass of agent "*i*", $M_{aj}(t)$ is the active gravitational mass of agent "*i*", $G(t)$ is the gravitational constant, ε is a small constant, and $R_{ij}(t)$ is the Euclidian distance between agent "*i*" and "*j*". The distance is calculated using a standard formula as follows:

$$R_{ij}(t) = \left\|X_i(t), X_j(t)\right\|_2 \tag{7}$$

while gravitational constant is defined as a decreasing function of time, which is set to G_0 at the beginning and decreases exponentially towards zero with lapse of time.

$$G(t) = G_0 \times e^{-\alpha\frac{t}{t_{max}}} \tag{8}$$

The total force acted on agent "*i*" in "*d*" dimension is a randomly weighted sum of *d*th components of the forces exerted from other agents;

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_i F_{ij}^d(t) \tag{9}$$

where, $rand_i$ is a random number in the interval of [0,1].

According to law of motion, the current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation or acceleration of any mass is equal to the force acted on the system divided by mass of inertia, which is shown in the following formula.

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}\ for\ M_{ai} = M_{pi} = M_{ii} \tag{10}$$

Therefore, the new agent velocity and position are calculated using these equations:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{11}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{12}$$

Finally, the next iteration is executed until the maximum number of iterations, t_{max}, is reached. The principle of standard GSA is shown in Fig. 1.

In solving multi-objective optimization problems, previously, non-dominated solutions have been exploited to assist vector evaluated particle swarm optimization (VEPSO) [2]. In this study, similar concept is introduced in VEGSA [3].

2. Vector Evaluated Gravitational Search Algorithm Assisted by Non-dominated Solutions

The vector evaluated GSA (VEGSA) assumes M populations of $P_1, P_2, P_3, \ldots, P_M$, 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, has been introduced to promote trade-off between objectives. In VEGSA, particles with minimum and maximum fitness are transferred to the neighboring population. Let m be the index of a population, $m = \{1, 2, \ldots, M\}$ and fit_j^m be the fitness of jth particle of the mth population. For function minimization problem, *best*(t) and *worst*(t) are redefined as follows:

$$best_m(t) = min\left(\min_{j\in\{1,\ldots,N\}} fit_j^m(t), min_{m-1}\right) \tag{13}$$

$$worst_m(t) = max\left(\max_{j\in\{1,\ldots,N\}} fit_{jj}^m(t), max_{m-1}\right) \tag{14}$$

For function maximization problem:

$$best_m(t) = max\left(\max_{j\in\{1,\ldots,N\}} fit_j^m(t), max_{m-1}\right) \tag{15}$$

$$worst_m(t) = min\left(\min_{j\in\{1,\ldots,N\}} fit_j^m(t), min_{m-1}\right) \tag{16}$$

where min_{m-1} and max_{m-1} are the minimum and the maximum fitness of neighboring population.

Consider a minimization of 2 objective functions. Figure 3 shows an objective space which includes agents from population (swarm) 1 and agents from population (swarm) 2. In this case, the *best* agent is selected from population (swarm) 2. Figure 4 shows the objective space when non-dominated solutions are considered. With respect to objective 2, the *best* agent is still selected from population (swarm) 2. However, in certain condition, the *best* agent is could be selected from non-dominated solutions, as shown in Figure 5. The same concept is applied to the *worst* solution.

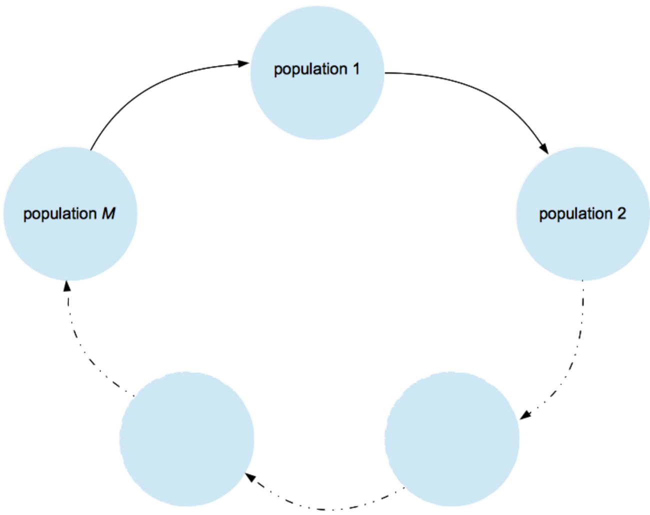

Figure 2: Information transfer between populations in VEGSA algorithm.

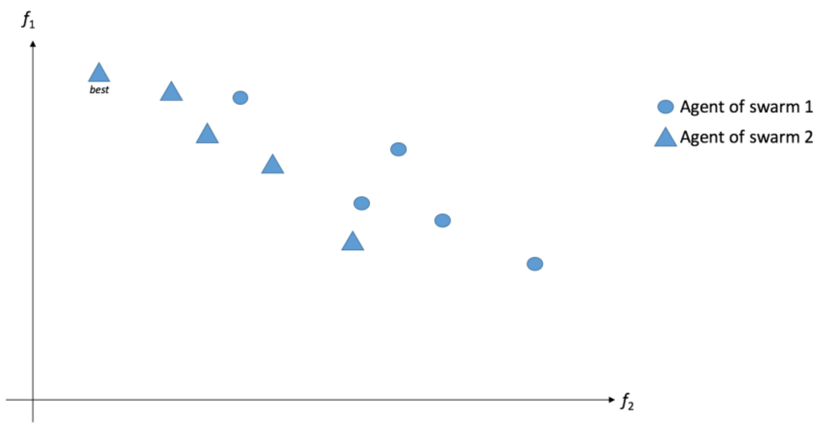

Figure 3: Selection of “best” agent in population (swarm) 1 considering agents from population (swarm) 2.

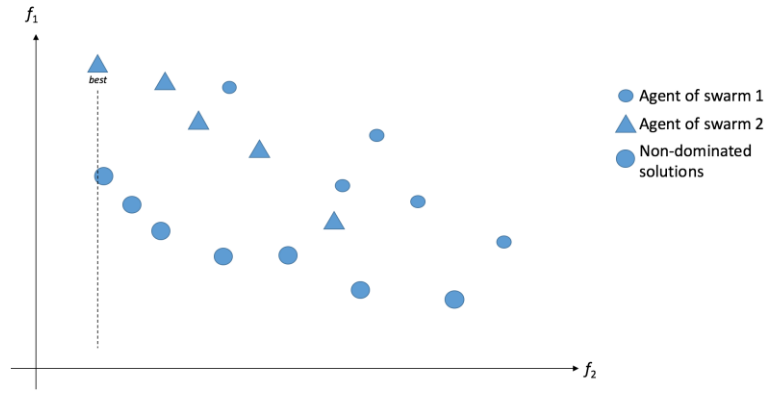

Figure 4: Selection of "best" agent in population (swarm) 1 considering agents from population (swarm) 2 and non-dominated solutions. The "best" agent is chosen from population (swarm) 2.

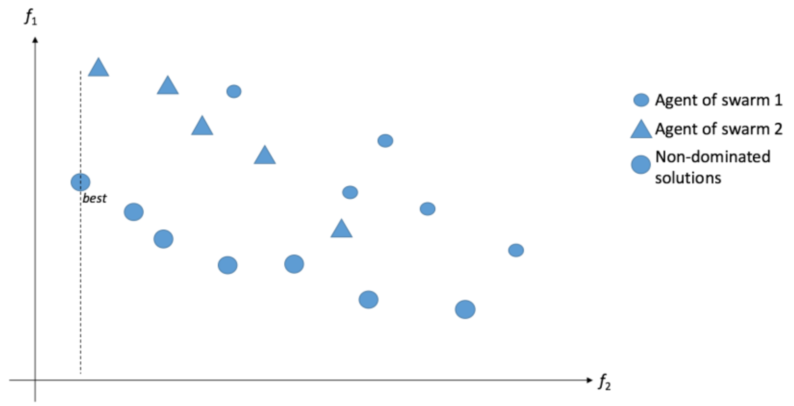

Figure 5: Selection of "best" agent in population (swarm) 1 considering agents from population (swarm) 2 and non-dominated solutions. The "best" agent is chosen from non-dominated solutions.

3. EXPERIMENT, RESULT, AND DISCUSSION

The parameter values used in the experiments are shown in Table 1. In this study, an archive is introduced to maintain and to update the non-dominated solutions at every iteration. The size of the archive chosen in this study is 100. Also, the performance of the VEGSAnds algorithm based on different values of initial gravitational constant, G_o, is also investigated. The ZDT 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.

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 represents the average distance between the Pareto front obtained, PF_o, and the true Pareto front, PF_t, as formulated in Eq. (17) and Eq. (18).

$$GD = \frac{\left(\sum_{i=1}^{|\mathcal{PF}_o|} {d_i}^M\right)^{1/M}}{|\mathcal{PF}_o|} \tag{17}$$

$$d_i = \min_{1\le k\le|\mathcal{PF}_t|} \sqrt{\sum_{j=1}^{M}\left(f_{j,i} - f_{j,k}\right)^2} \tag{18}$$

This measure estimates how close the PF_o lies to the PF_t. Hence, a smaller *GD* value represents better performance. The *Spread* is used to measure the extent of the PF_o distribution of the along the PF_t. The Eq. (19), Eq. (20), and Eq. (21) show the calculation of *Spread*.

Table 1: The parameters and its value used in experiments

Parameter	**Value**
Iteration	250
Number of agent per swarm, N	50
Archive size	100
G_0	10, 100, 1000, 10000, 100000
ε	2^{-52}
β	20
Number of run	30

Table 2: Results (number of solution)

Problem	**VEGSAnds (G_o = 10)**	**VEGSAnds (G_o = 100)**	**VEGSAnds (G_o = 1000)**	**VEGSAnds (G_o = 10000)**	**VEGSAnds (G_o = 100000)**
ZDT1	**68.66**	28.2	21.78	22.3	22.08
ZDT2	**14.82**	14.02	11.28	11.12	12.36
ZDT3	**68.46**	32.06	26.34	25.08	25.2
ZDT4	**27.62**	13.34	13.96	13.5	15.86
ZDT6	25.86	**28**	15.86	13.54	12.2

Table 3: Results (GD)

Problem	**VEGSAnds (G_o = 10)**	**VEGSAnds (G_o = 100)**	**VEGSAnds (G_o = 1000)**	**VEGSAnds (G_o = 10000)**	**VEGSAnds (G_o = 100000)**
ZDT1	**0.462971942**	0.607635565	0.590220744	0.570666537	0.568108635
ZDT2	1.023069803	0.991721258	1.031533305	1.000435525	**0.929691293**
ZDT3	**0.261349174**	0.304842662	0.304927251	0.289915755	0.281959937
ZDT4	18.16152696	16.13121131	**15.68148364**	20.57683949	22.33879306
ZDT6	1.561454307	**1.504907169**	1.905413113	1.973092885	2.094675769

$$Spread = \frac{d_f + d_l + \sum_{i=1}^{|\mathcal{PF}_o|-1} |d_i - \bar{d}|}{d_f + d_l + (|\mathcal{PF}_o|-1)\bar{d}} \tag{19}$$

$$\bar{d} = \frac{\sum_{i=1}^{|\mathcal{PF}_o|-1} d_i}{|\mathcal{PF}_o|-1} \tag{20}$$

$$d_i = \sqrt{\left(f_{1,\mathrm{i}} - f_{1,\mathrm{i}+1}\right)^2 + \left(f_{2,\mathrm{i}} - f_{2,\mathrm{i}+1}\right)^2} \tag{21}$$

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.

The experimental results are shown in Table 2, Table 3, and Table 4, which show the number of solutions, *GD*, and *SP* values, respectively. In terms of number of solutions, it is found that low G_o value able to increase the number of solution. However, G_o values does not correlate to *GD*. On the other hand, higher G_o values are more preferable to obtain lower *SP*.

Table 4: Results (SP)

Problem	VEGSAnds (G_o = 10)	VEGSAnds (G_o = 100)	VEGSAnds (G_o = 1000)	VEGSAnds (G_o = 10000)	VEGSAnds (G_o = 100000)
ZDT1	1.058438888	0.915064363	**0.849577476**	0.857401645	0.853398717
ZDT2	0.989110196	0.936550066	0.90507099	**0.899899343**	0.913010521
ZDT3	1.083917774	0.903042827	**0.839756996**	0.840216416	0.858439223
ZDT4	1.058164372	0.920216084	0.957837863	**0.911424922**	0.92832957
ZDT6	1.027297073	1.029958963	0.969478366	0.943521569	**0.934589823**

4. CONCLUSION

Following the existing vector evaluated PSO assisted by non-dominated solutions for multi-objective optimization problems, in this study, the same concept has been employed in GSA. Different G_o values were considered and we found that low G_o values is good to increase the number of solution, higher G_o values are more preferable to obtain lower *SP*, and G_o values does not effect the *GD*.

ACKNOWLEDGEMENT

This work is financially supported by the Fundamental Research Grant Scheme awarded by the Ministry of Education (MOE) to Universiti Teknologi Malaysia (UTM) (Vote No. 4F615).

REFERENCES

1. Rashedi, E., Nezamabadi-pour, H., and Saryazdi, S., GSA: A Gravitational Search Algorithm, Information Sciences, 2009, Vol. 179, No. 13, pp. 2232-2248.
2. Kian Sheng Lim, Zuwairie Ibrahim, Salinda Buyamin, Anita Ahmad, Faradila Naim, Kamarul Hawari Ghazali, and Norrima Mokhtar, Improving Vector Evaluated Particle Swarm Optimisation by Incorporating Nondominated Solutions, The Scientific World Journal, 2013, Vol. 2013, pp. 1-19.
3. Badaruddin Muhammad, Zuwairie Ibrahim, Kamarul Hawari Ghazali, Mohd Riduwan Ghazali, Kian Sheng Lim, Sophan Wahyudi Nawawi, Muhammad Arif Abdul Rahim, Nor Azlina Ab. Aziz, Marizan Mubin, and Norrima Mokhtar, Performance Evaluation of Vector Evaluated Gravitational Search Algorithm, ICIC Express Letters, 2015, Vol. 9, No. 3, pp. 649-654.