MLGSA: Multi-Leader Gravitational Search Algorithm for Multi-Objective Optimization Problem

Mohd Riduwan Ghazali[1], Khairul Hamimah Abas[2], Badaruddin Muhammad[1],
Nor Azlina Ab. Aziz[3] and Kian Sheng Lim[2]

[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.*
[3]*Faculty of Engineering and Technology, Multimedia University, 75450 Melaka, Malaysia.*
riduwan@ump.edu.my

***Abstract*—Recently, we have introduced Multi-Leader Particle Swarm Optimization (MLPSO) algorithm for multi-objective optimization problem. Better convergence and diversity have been observed over the conventional Multi-Objective Particle Swarm Optimization. In this paper, the same concept is extended to Gravitational Search Algorithm (GSA). The performance was investigated by solving a set of ZDT test problem. An analysis was also performed by varying the value of initial gravitational constant.**

***Index Terms*—About Gravitational search algorithm; multi-objective optimization.**

I. 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 belongs to a class of population-based meta-heuristics. In GSA, agents are considered as an object and their performance is expressed by their masses. The position of a particle corresponds 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 Equation (2) and Equation (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;

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

whereas, for maximization problem,

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

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

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

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 a 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 the 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 the 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 Figure 1.

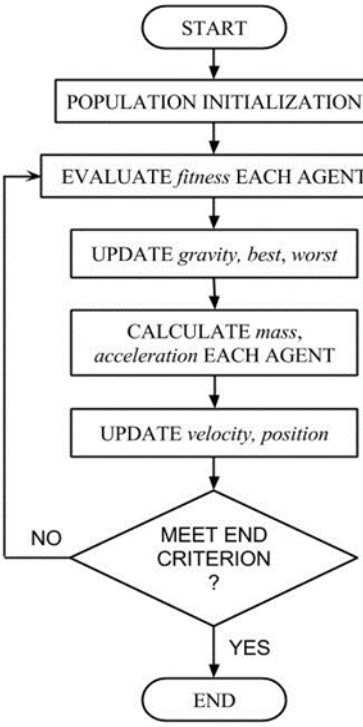

Figure 1: The gravitational search algorithm

Since 2009, the GSA algorithm has been extended extensively, for example, in solving multi-objective optimization problems [2-6], combinatorial optimization problems [7-9], and in solving engineering problems [10-12].

In solving multi-objective optimization problems, we have recently introduced Multi-Leader Particle Swarm Optimization (MLPSO) [13]. In MLSPO, the movement of a particle is determined by all leaders that dominate that particle. This concept allows for more information sharing between particles. As opposed to most of the algorithms, one or two leaders are used to guide the movement of every particle in a search space. The previous study has shown the superiority of MLPSO. In this study, similar concept is extended to GSA.

II. THE MULTI-LEADER GSA

In multi-leader concept [13], multiple leader set, $ML_{s,i}$, is defined as the set of non-dominated solutions that dominate the i-th agent from the s-th group as:

$$ML_{s,i} = \{\boldsymbol{\psi} \prec \boldsymbol{p}_{s,i} \mid \boldsymbol{\psi} \in \Psi\} \tag{13}$$

Consider the agents and the non-dominated solutions shown in Figure 2. For example, the agent labelled with ‘1’ is dominated by the non-dominated solution ‘A’, and thus, $ML_{s,1} = \{\boldsymbol{\psi}_A\}$. The second agent is dominated by the non-dominated solutions ‘A’ and ‘B’, and thus, $ML_{s,2} = \{\boldsymbol{\psi}_A, \boldsymbol{\psi}_B\}$. Similarly, $ML_{s,3} = \{\boldsymbol{\psi}_A, \boldsymbol{\psi}_B, \boldsymbol{\psi}_C\}$ because the non-dominated solutions ‘A’, ‘B’, and ‘C’ dominate the third agent. For the rest of the agent, $ML_{s,4} = \{\boldsymbol{\psi}_B, \boldsymbol{\psi}_C\}$ and $ML_{s,5} = \{\boldsymbol{\psi}_C\}$. Furthermore, if an agent is a member of the archive, then the other members in the archive are selected to be its leaders. In this case, $ML_{s,A} = \{\boldsymbol{\psi}_B, \boldsymbol{\psi}_C\}$. Hence, the non-dominated solutions $\boldsymbol{\psi}$ in the $ML_{s,i}$ are the leaders which dominate the $\boldsymbol{p}_{s,i}$. Also, if an agent is a non-dominated solution itself, then there will be no leader in the multi leader set.

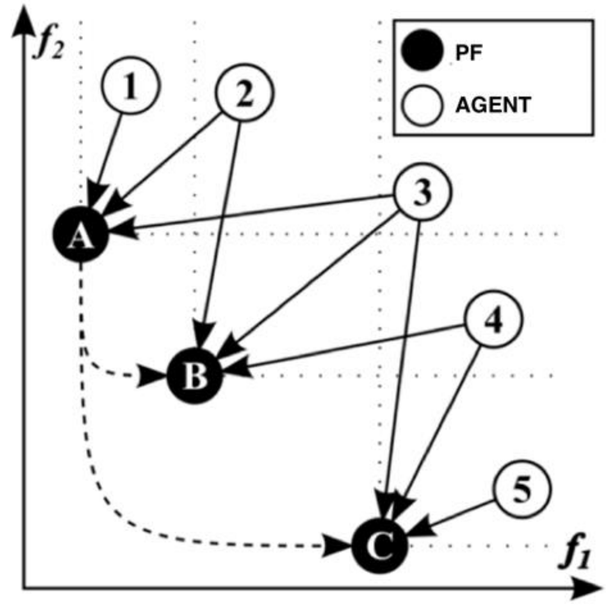

Figure 2: Multi-leader concept

The MLGSA algorithm includes two equal-size groups since only two objectives are considered in this study. Both groups employ the RANDOM selection method [14] to determine the leaders. The first group utilises the information from all leaders. The second group uses the information from one leader only. Figure 3 shows the flow chart of the proposed MLGSA algorithm. The solid line and the dotted line represent the flow of the first and the second groups, respectively.

Since multi-leader is introduced in MLGSA, the formulation for mass of i-th agent is modified as follows:

$$m_i(t) = \frac{MOFitness(x_i(t) - worst(t))}{best(t) - worst(t)} \tag{14}$$

To calculate the mass according to Equation (14), $best(t)$, $worst(t)$, and $MOFitness(x_i)$ are formulated as follows:

$$best(t) = \min_{j \in |1,\dots,N|} MOFitness(x_j(t)) \tag{15}$$

$$worst(t) = \max_{j \in |1,\dots,N|} MOFitness(x_j(t)) \tag{16}$$

$$MOFitness(x_i) = \sum_{j=1}^{|ML_i|} Euclidean(ML_{i,j}, \quad x_{i,2}) \tag{17}$$

where i represents the index of agent, the multi-leader set for each i-th agent is denoted as $ML_i = \{ML_{i,1}, ML_{i,2}, \ldots, ML_{i,|MLi|}\}$, and $Euclidean(a,b)$ is the Euclidean distance in objective space between solution a and b. Figure 4 illustrates this formulation. Agent with label 'A' is dominated by three non-dominated solutions and thus MOFitness(x_A) = $d_1+d_2+d_3$. On the other hand, agent labelled as 'B' is a non-dominated solution. Thus, MOFitness(x_B) = 0. Lastly, particle with label 'C' is dominated by two non-dominated solutions. Hence, MOFitness(x_A) = d_4+d_5.

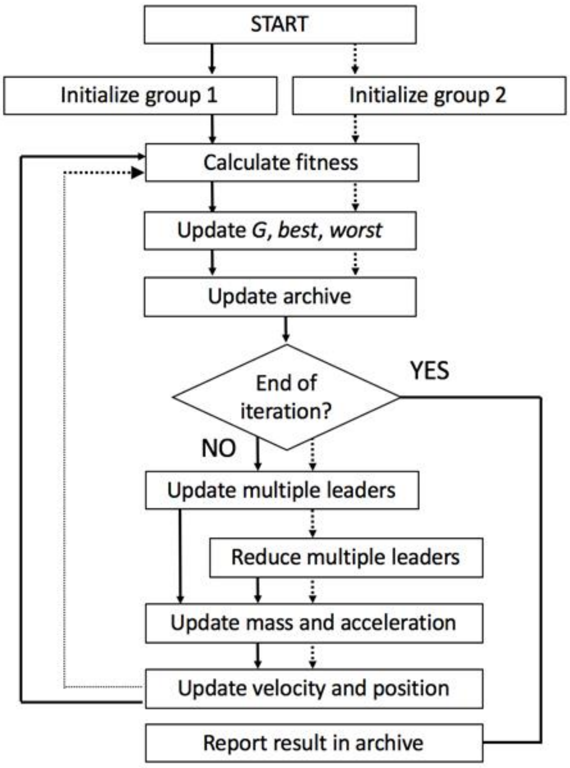

Figure 3: The multi-leader gravitational search algorithm

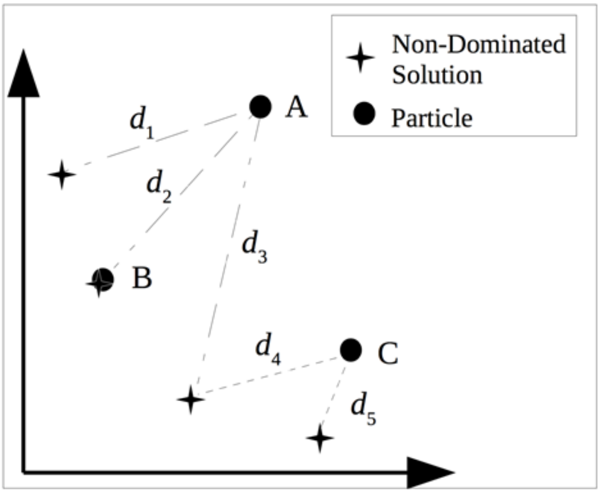

Figure 4: Agents and non-dominated solutions for two objective problems

Based on the mass calculation in MLGSA, the mass is evaluated based on the sum of Euclidean distance in objective space between an agent and its leaders. Agent which are far and has many leaders are considered as a weak solution; thus, having less mass and lesser overall attraction. In contrast, agents with short distance and less leaders are a better solution. If an agent has no leader, the agent itself is a non-dominated solution and will result in the largest overall attraction to other agents.

III. Experiment And Result

In this work, each experiment was repeated for 100 runs. Then, the average of the performance measures were calculated. The MLGSA used 100 agents and 1000 maximum number of iterations. Archive was limited to 100 solutions. The MLGSA's agents were equally divided into two groups. The performance measures in terms of number of solution (NS), generational distance (GD), spread (SP), and hypervolume (HV), were evaluated based on ZDT test problems [15]. The NS shows the number of solution in archive, SP indicates spread and HV indicates the area covered by non-dominated solutions. Thus, greater NS, SP, and HV values are desired. Meanwhile, GD and SP indicate the distance between non-dominated solutions to actual solution (Pareto front). Hence, smaller GD values are desired. Recent finding shows that the initial value of gravitational constant, G_0, contributes to the performance of GSA [16]. Hence, various G_0 values were considered in experiments as well.

Experimental results, which show the individual performance, in average, of MLGSA in terms of NS, GD, SP, and HV, were calculated as tabulated in Table 1, Table 2, Table 3, and Table 4, respectively. The best value of NS, GD, SP, and HV are written in bold. Then, statistical analysis was performed.

IV. Discussion

Friedman average ranked test was conducted using the results tabulated in Table 1, 2, 3 and 4. The null hypotheses of the Friedman test state that the performances of different values of G_0 are statistically identical to each other. If the statistical value of the test shows otherwise, then a post hoc procedure is conducted. The Holm post hoc procedure was chosen here.

The Friedman's average ranks for the different values of G_0 based on NS, GD, SP and HV are listed in Table 5. Based on the average ranks, it can be seen that G_0=1.00E+04 was ranked the highest in all measures with the exception for the HV. G_0=1.00E+05 was ranked the best for the HV measure.

The Friedman statistical value for NS shows significant difference between the different values of G_0. The Holm procedure shows that G_0=1.00E+04 was significantly better than G_0=1.00E+10.

Significant difference was also detected for GD measure. The result of Holm procedure shows that G_0=1.00E+04 and G_0=1.00E+06 were significantly better than G_0=1.00E+10. However, according to SP and HV measures, the performance of MLGSA using different value of G_0 was statistically identical to each other. Overall, G_0=1.00E+04 was recommended to ensure good performance of MLGSA.

Table 1
Performance based on NS (average values)

	G_0 = 1.00E+01	G_0 = 1.00E+02	G_0 = 1.00E+03	G_0 = 1.00E+04	G_0 = 1.00E+05	G_0 = 1.00E+06	G_0 = 1.00E+07	G_0 = 1.00E+08	G_0 = 1.00E+09	G_0 = 1.0E+010
ZDT1	44.98	50.28	54.9	57.12	62.04	74.92	73.82	78	**81.98**	50.04
ZDT2	**70.88**	68.52	64.16	70.64	67.26	55.76	46.94	39.88	29.92	16.38
ZDT3	88.44	84.5	78.84	91.38	88.52	88.76	92.52	**94.18**	89.66	54.04
ZDT4	99.94	99.9	**100**	**100**	99.94	**100**	**100**	**100**	97.68	15.86
ZDT6	14.08	14.74	15.16	15.88	16.8	**17.5**	15.58	14.82	13.04	11.22

Table 2
Performance based on NS (average values)

	G_0= 1.00E+01	G_0 = 1.00E+02	G_0 = 1.00E+03	G_0 = 1.00E+04	G_0 = 1.00E+05	G_0 = 1.00E+06	G_0 = 1.00E+07	G_0 = 1.00E+08	G_0 = 1.00E+09	G_0 = 1.0E+10
ZDT1	0.115791316	0.105874855	0.093888551	0.084888125	0.072544855	**0.056012272**	0.064507659	0.061400351	0.064865319	0.136939225
ZDT2	0.076452809	0.078056115	**0.071067565**	0.072842267	0.083126788	0.103826065	0.14499852	0.182063885	0.273121717	0.479869079
ZDT3	0.048244816	0.045857654	0.0434758	0.036203188	**0.035663001**	0.039353845	0.036083	0.042161391	0.05112472	0.090545517
ZDT4	**0.518834661**	0.874589207	0.904728599	0.785232629	0.911079538	0.960885892	1.09365798	1.256501249	1.678942133	10.24328026
ZDT6	1.418292612	1.261812056	1.200378313	1.172968823	1.102118007	**1.021186126**	1.066745185	1.211843502	1.344754495	1.768550523

Table 3
Performance based on SP (average values)

	G_0 = 1.00E+01	G_0 = 1.00E+02	G_0 = 1.00E03	G_0 = 1.00E+04	G_0 = 1.00E+05	G_0 = 1.00E+06	G_0 = 1.00E+07	G_0 = 1.00E+08	G_0 = 1.00E+09	G_0 = 1.0E+10
ZDT1	1.161836109	1.171298398	1.161979505	1.157957065	1.169180016	1.181774422	1.150313049	1.107985077	1.026377655	**0.911392796**
ZDT2	0.992196781	1.022425063	1.00312344	1.065315755	1.04709536	1.074282171	1.049827387	1.033858086	1.011032293	**0.947968153**
ZDT3	1.199944439	1.217386285	1.184204006	1.222116205	1.212792592	1.182526993	1.18115598	1.147644014	1.091744416	**0.960684213**
ZDT4	1.234614879	1.313699077	1.333295322	1.274984116	1.315022503	1.277046508	1.345645944	1.325045857	1.380917932	**1.117863813**
ZDT6	1.065925748	1.072137786	1.088359826	1.078718938	1.063264558	1.064064373	1.026683797	0.99246382	0.973068882	**0.941319412**

Table 4
Performance based on HV (average values)

	G_0 = 1.00E+01	G_0 = 1.00E+02	G_0 = 1.00E+03	G_0 = 1.00E+04	G_0 = 1.00E+05	G_0 = 1.00E+06	G_0 = 1.00E+07	G_0 = 1.00E+08	G_0 = 1.00E+09	G_0 = 1.0E+10
ZDT1	0.076576906	0.153793297	0.174082847	0.203995548	0.220267438	**0.228384002**	0.204197266	0.161659771	0.122852008	0.02964863
ZDT2	0.016644881	0.025527456	**0.027023884**	0.023177816	0.018065432	0.005285086	8.02E-04	4.69E-07	0	0
ZDT3	0.050481446	0.12267938	0.159302145	0.17872138	**0.197930891**	0.184794984	0.185295533	0.149478017	0.113492529	0.046910798
ZDT4	**0.045001147**	0.044014034	0.013207949	0.012744517	0.008140334	0.002585183	3.25E-05	0	0	0
ZDT6	0	0	0	0	0	0	0	0	0	0

Table 5
Friedman Average Rank

G_o	NS	GD	SP	HV
1.00E+01	6.5	6	6.2	5.9
1.00E+02	6.8	5.8	3.8	4.7
1.00E+03	5.8	4.6	4.2	3.9
1.00E+04	3.4	3.4	3.8	4.1
1.00E+05	4.7	3.8	4.2	3.5
1.00E+06	3.6	3.6	4	4.3
1.00E+07	4	4.2	5.2	4.9
1.00E+08	4	5.8	6.6	6.9
1.00E+09	6.4	7.8	7	8
1.00E+10	9.8	10	10	8.8

V. CONCLUSION

Previously, multi-leader concept has been introduced in PSO for multi-objective optimization problems. In this study, the same multi-leader concept has been employed in GSA. Different G_0 values were considered as well and G_0=1.00E+04 was recommended for implementing Multi-Leader GSA. The next step of this study is to compare the performance of Multi-Leader GSA with Multi-Leader PSO. Finally, the concept of multi-leader could be incorporated into other optimization algorithms such as simulated Kalman filter [17].

ACKNOWLEDGMENT

This work is supported by the Fundamental Research Grant Scheme (FRGS) awarded by the Ministry of Higher Education (MOHE) to Universiti Malaysia Pahang (UMP) (RDU130117), Multimedia University (FRGS/1/2015/ICT02/MMU/03/1), and Universiti Teknologi Malaysia (UTM) (R.J130000.7823.4F615).

REFERENCES

[1] Rashedi, E., Nezamabadi-pour, H., and Sayyazdi, S., "GSA: A Gravitational Search Algorithm", *Information Science*, vol.179, no.13, 2009, pp. 2232–2248.

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

[3] Muhammad, B., Ibrahim, Z., Ghazali, K.H., Ghazali, M.R., Lim, K.S., Nawawi, S.W., Aziz, N.A.A., Mubin, M., and Mokhtar, N., "Performance Evaluation of Vector Evaluated Gravitational Search Algorithms Based on ZDT Test Functions. International Journal of Simulation – Systems", *Science and Technology*, vol.15, no.1, 2014, pp.1-6.

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

[5] Nobahari, H., Nikusokhan, M., and Siarry, P., "A Multi-Objective Gravitational Search Algorithm Based on Non-Dominated Sorting", *International Journal of Swarm Intelligence Research*, vol.3, no.3, 2012, pp.32-49.

[6] Bhattacharya, A. and Roy, P.K., "Solution of Multi-Objective Optimal Power Flow using Gravitational Search Algorithm", *IET Generation, Transmission, and Distribution*, vol.6, no.8, 2009, pp.751-763.

[7] Ibrahim, I., Ibrahim, Z., Ahmad, H., and Yusof, Z.M., "Rule-based Multi-State Gravitational Search Algorithm for Discrete Optimization Problem", *4th International Conference on Software Engineering and Computer Systems*, 2015, pp.142-147.

[8] Shamsudin, H.C., Irawan, A., Ibrahim, Z., Abidin, A.F.Z., Wahyudi, S., Rahim, M.A.A., and Khalil, K., "A Fast Discrete Gravitational Search Algorithm", *Fourth International Conference on Computational Intelligence, Modelling and Simulation*, 2012, pp.24-28.

[9] Sudin, S., Nawawi, S.W., Abidin, A.F.Z., Rahim, M.A.A., and Khalil, K., "A Modified Gravitational Search Algorithm for Discrete Optimization Problem", vol.15, no.1, 2014. 51-55.

[10] Niknam, T., Narimani, M.R., Azizipanah-Abarghooee, R., and Bahmani-Firouzi, B., "Multiobjective Optimal Reactive Power Dispatch and Voltage Control: A New Opposition-Based Self-Adaptive Modified Gravitational Search Algorithm", *IEEE Systems Journal*, vol. 7, no.4, 2013, pp.742-753.

[11] Duman, S., Sönmez, Y., Güvenč, U. and Yörükeren, N., "Optimal reactive power dispatch using a gravitational search algorithm, *IET Generation, Transmission, and Distribution*, vol.6, no.6, 2012, pp.563-576.

[12] Ibrahim, I., Ibrahim, Z., Ahmad, H., Jusof, M.F.M., Yusof, Z.M., Nawawi, S.W., and Mubin, M., "An Assembly Sequence Planning Approach with a Rule-Based Multi-State Gravitational Search Algorithm", The International Journal of Advanced Manufacturing Technology, vol.79, no.5, 2015, pp.1-14.

[13] Ibrahim, Z., Lim, K.S., Buyamin, S., Satiman, S.N., Suib, M.H., Muhammad, B., Ghazali, M.R., Mohamad, M.S., and Watada, J., "MLPSO: Multi-Leader Particle Swarm Optimization for Multi-Objective Optimization Problems", *ARPN Journal of Engineering and Applied Sciences*, vol.10, no.23, 2015, pp.17533-17538.

[14] Alvarez-Benitez, J.E., Everson, R.M., and Fieldsend, J.E., "A MOPSO Algorithm Based Exclusively on Pareto Dominance Concepts", *Evolutionary Multi-Criterion Optimization*, 2005, pp.459-473.

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

[16] Aliman, M.N., Abas, K.H., Najib, M.S., Aziz, N.A.A., Mohamad, M.S., and Ibrahim, Z., "Gravitational Search Algorithm: R is better than R2?.", *ARPN Journal of Engineering and Applied Sciences* (in press), 2015.

[17] Ibrahim, Z., Abdul Aziz, N.H., Ab. Aziz, N.A., Razali, S., Shapiai, M.I., Nawawi, S.W., and Mohamad, M.S. A Kalman, "Filter Approach for Solving Unimodal Optimization Problems", *ICIC Express Letters*, vol.9, no.12, 2015, pp.3415-3422.