MOGSA, A Gravity Inspired Multi-Objective Meta-heuristic

Hamid Reza Hassanzadeh ha.hassanzadeh@ieee.org
Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

Seyyed Modjtaba Rouhani rouhani@ieee.org
Department of Electrical Engineering, Gonabad Azad University, Gonabad, Iran

Abstract
Recently there has been a great research conducted on diverse variations of multi-objective swarm optimization algorithms each of which has its own strengths and weaknesses. Due to the high complexity of multi-objective problems the efficiency of these methods has become a matter of concern. In this paper a new multi-objective meta-heuristic algorithm based on gravitational forces is proposed and applied to different test benches. The obtained results prove the superiority of the algorithm compared with other pioneering techniques such as the MOPSO (Multi-Objective Particle Swarm Optimization).

Keywords
Multi-objective Particle Swarm Optimization (MOPSO), Multi-objective Gravitational Search Algorithm (MOGSA), Pareto Optimality, Indirect Niching

1 Introduction

Optimization is considered as one of the most challenging ends of meta-heuristic algorithms. One of the main concerning aspects of such methods is how close the obtained results are to the global optimum. The problem becomes more prominent when dealing with optimization tasks in which several conflicting objectives are to be optimized. In such cases there is no global optimum but rather a set of non-dominated solutions exist. Thus, unless the optimization process is properly moderated, the algorithm's outcome may be adversely affected. In this respect, several multi-objective evolutionary algorithms (MOEAs) have been suggested by Deb et al. (2000); Schxxxxxxxxxxxxxxxxxx (1985); Zitzler et al. (2000); Knowles and Corne (2000b); Laumanns et al. (2000) to tackle this issue which appear quite successful albeit they have their own drawbacks including their high computational burden.

PSO, introduced by Kennedy and Eberhart (1995), inspires from the behavior of flocks of birds and school of fish around the food. PSO incorporates both the individual attitudes towards food and the corresponding social one. That is to say, each individual (which is considered as a solution) keeps its best experience among the solutions in mind while at the same time behaves under the influence of the best experience (the so-called global best) found by the whole flock. Accordingly, the flight direction associated to each particle gets updated based on a linear combination of the personal and global bests in the following sense,

$$v_d^i(t+1) = w(t)v_d^i(t) + c_1 r_1(\text{pbest}_d^i(t) - x_d^i) + c_2 r_2(\text{gbest}_d^i(t) - x_d^i) \quad (1)$$

$$x_d^i(t+1) = x_d^i(t) + v_d^i(t+1) \tag{2}$$

where w is an inertia factor, x_d^i, v_d^i, c_1 and c_2 are in order the position and speed of ith element in d^{th} dimension, cognitive and social factor gains. In the same sense, r_1 and r_2 are random variables uniformly distributed between 0, 1 and finally pbest and gbest are the personal and global best, in order.

PSO has some benefits over other heuristics such as Genetic Algorithm in that it does not need any complicated operators such as crossover and selection which simplifies the its implementation. Furthermore, Kennedy and Spears (1998) show that compared to genetic algorithm, PSO benefits a faster convergence rate, needs less parameter tuning and is less affected by changes in dimension of the underlying problem.

In practice, the optimization problems are not aimed at finding a solution which maximizes (minimizes) a single objective but rather a set of simultaneous objectives exist which have conflicts with each other. Therefore there will be no unique solution any longer. To date, multi-objective algorithms gained an immense interest and consequently dufferent techniques have been put forward to handle problems of this kind. Among hese techniques, MOPSO based methods are of the most promising ones. Till now, different by-products of MOPSO have been proposed among which the one introduced by Cagnina and Coello Coello (2005) is selected as the basis for our future comparisons. In that article the strategy to hybridize the multi-objective concept and the PSO meta-heuristic is to form an archive (also called repository) for storing the non-dominated solutions found by the swarm through an iterative process. During the course of each iteration, one of the particles already stored in the archive is employed selected as a *leader* which serves as the global best. The swarm particles are then set up their flight directions according to this selected solution and their own personal history of exploration. Finally after each iteration those solutions in the archive that already got dominated are omitted. The algorithm proceeds until it reaches its maximum iteration number or an acceptable solution found, either.

The gravitational search algorithm (GSA) proposed by Rashedi et al. (2009), mimics one of the most restrictive natures rules, the gravity. The gravity is one of the most tangible and ubiquitous forces amongst the other physics rules. The aforementioned emerges between two particles as soon as they come into existence. The inducted amount of force follows Newtons law of gravity which states that gravitational forces between two entities is directly proportional to the product of their masses and inversely proportional to their distance squared. That is, $\vec{F} = G(t).\frac{m_a m_p}{r^2}$ where G is the gravitational constant, changing during the course of time, m_a is the active mass (the strength of a gravitational field due to its mass), m_p is the passive mass (the strength of an object's interaction with gravitational field) and r is the distance between the two particles. When there are more than one forces applied to a mass, the overall force is the vector sum of each individually, as depicted in Fig. 1. Note also that $G(t)$ is a decreasing constant which changes according to the following formula, a fact that is considered in cosmology (see Mansouri et al. (1999)),

$$G(t) = G(t_0) \times (\frac{t_0}{t_1})^{\beta} \quad \beta < 1 \tag{3}$$

Figure 2 depicts the flowchart of the foregoing method for a minimization problem

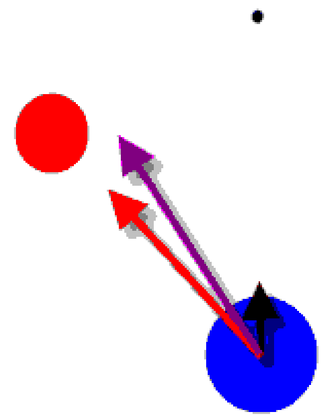


Figure 1: The biggest particle attracted by the two others.

where m, M and F obtained as follows,

$$\begin{aligned}
m_i(t) &= \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad i = 1, 2, \ldots, N \\
M_i(t) &= \frac{m_i(t)}{\sum\limits_{j=1}^{N} m_j(t)} \quad i = 1, 2, \ldots, N \\
best(t) &= \min_{j \in \{1,2,\ldots,N\}} fit_j(t) \\
worst(t) &= \max_{j \in \{1,2,\ldots,N\}} fit_j(t) \\
F_i^d(t) &= \sum_{j=1, j \neq i}^{N} rand_j F_{ij}^d(t) \quad i = 1, 2, \ldots, N \,,\, d = 1, 2, \ldots, D \\
F_{ij}^d(t) &= G(t).\frac{M_{ip}(t) M_{ja}(t)}{R_{ij}(t) + \epsilon}.\left(x_j^d(t) - x_i^d(t)\right) \quad i, j = 1, 2, \ldots, N \,,\, d = 1, 2, \ldots, D
\end{aligned} \tag{4}$$

A word under notation is that GSA is a stochastic algorithm that incorporates a random term in calculation of forces between individual solutions. Also note that decrementing G by lapse of time corresponds to make the search process finer, which amounts to a local search around near optimal solutions especially in the last generations of the search operation. Finally to compromise between the exploration and exploitation, the number of particles that generate gravitational fields should be decreased during the time to benefit higher exploration rates in the beginning and better exploitation with less exploration in the final iterations.

The rest of this paper is organized as follows. In Section 2 the new algorithm is presented. Section 3 demonstrates different test functions used to validate our method. In Section 4 two important metrics for evaluation of the algorithm's performances are introduced. Section 5 presents the experiments and a comparison between the new method and other competitive ones is given. Finally Section 6 concludes by illustrating the main strengths of the proposed method and following that few suggestions for future researches are mentioned.

2 Multi-objective Gravitational Search Algorithm

As opposed to the adopted approach in single objective PSO where there is a unique global best, in Multi-objective PSO there is no unique best solution in each iteration but rather there exist a set of non-dominated solutions. The key to handle this problem in diverse variations of MOPSO is to pick out an individual as a leader in each iteration so that other individuals seek following it. Yet, the policy based on which the leader is selected differs in different proposed methods. Cagnina and Coello Coello (2005) choose the random selection method while Wickramasinghe and Li (2008) introduce a new differential evolution (DE) operator to generate a leader. Regardless of what strategy is appointed for choosing leaders, this movement of pursuing as single particle in population is a drawback especially in problems where there are many local optimal fronts Wickramasinghe and Li (2008). Mostly in these situations the mutli-objective PSO converges prematurely into a local front rather than the global one Li (2003a,b). In addition, selecting only one particle through the whole group of particles means depriving oneself from the amount of information hidden in them all and to chase a probably misleading leader. In this proposed method however, instead of overlooking the whole solutions but one, we let other particles as well to have their shares in directing the whole population towards a better position based on a gravitational perspective. In order to hybridize the GSA and the multi-objectivity we incorporate some of the well known operators to our method as follows,

Uniform Mutation Operator (Backxxxxxxxx et al. (1997)): it selects one of the particles and adds a signed random value to one of its dimensions. The total value should be in the admissible range otherwise it will be truncated into its nearest feasible value.

Elitist Policy (Cagnina and Coello Coello (2005)): The non-dominated solutions found are stored in an archive with a grid-like structure (as is the case for PAES proposed by Knowles and Corne (2000b)). The grid structure is created such that each dimension in the objective space is divided into 2^{n_i} equal divisions where i denotes dimension index and hence for a k-objective optimization problem there will be $2^{\sum_{i=1}^{k} n_i}$ different segments. As long as the archive is not full, new non-dominated solutions are added to it. As soon as the number of elements in the archive gets to its limit one of the elements should be omitted from it in order to make enough room for inclusion of a new element. The strategy to remove an element from the archive is to find the most crowded hypercube, randomly select one of the solutions in it extract it from the repository. Note that before inserting a new element into the archive, the set is fully searched in case there appear some a dominated solutions which should be removed beforehand.

Algorithm 1 depicts the proposed approach in pseudo code. After allocating memory to each of the corresponding variables, the population is initialized randomly in the search space (line 2), then the velocity vector corresponding to each particle is initialized to zero (line 3). After that among the generated individuals those which are non-dominated are added to the archive explained already. The algorithm next starts iterating for a given maximum iteration number; in each iteration the gravitational constant is updated on an exponential rate, then the mass corresponding to each solution in the archive is calculated (the active mass). For each element in the archive, the aforementioned value is the distance to its nearest neighbor among other elements, in the objective space. That is the farther an element is from its neighbors the heavier it will be. This corresponds to an intrinsic uniform distribution of the solutions along the Pareto front which is usually done by means of niching techniques such as fitness sharing.

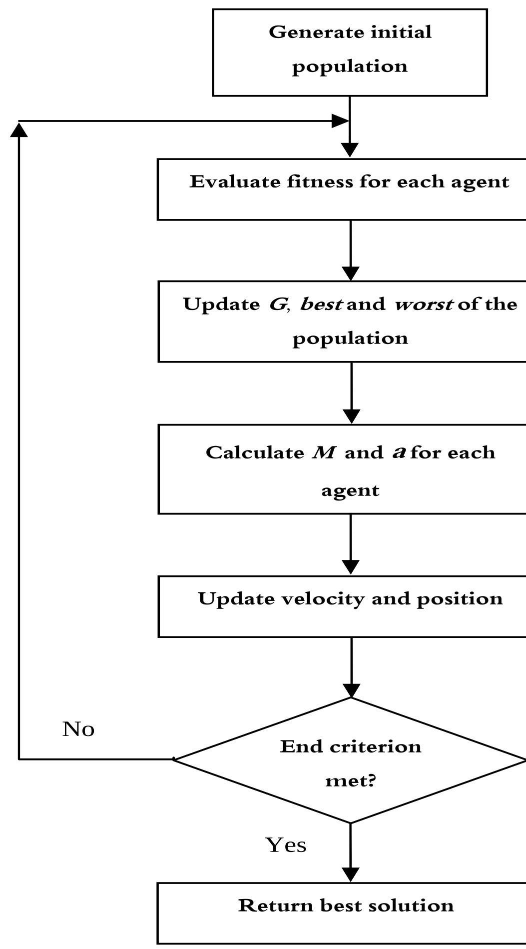


Figure 2: The flowchart diagram of GSA

Note that as opposed to the repository particles through the whole execution process the searching particles are assigned a unit mass (passive mass). After calculation of masses, the top K best non-dominated solutions are selected from the archive (line 8). These are in fact the ones considered as the sources of gravitational fields. This means that they are the only elements among all the archive members that absorb particles in the search space. Initially K equals the size of the archive but during the time this factor is decremented for the sake of a better compromise between the exploration and exploitation rates Rashedi et al. (2009). Exploration is defined as the ability to surf unexplored regions in the search space while exploitation denotes finding better solutions around a local optimum. As initially K-best takes on its largest values the exploration rate is high while by decrementing this factor the particles tend to search around the best solutions. Then for each of the population individuals the total amount of force that it undergoes is calculated from which the acceleration, speed and next positions are found respectively (line 9-15).

After all the new individuals are exposed to uniform mutation operator and those

Algorithm 1 The pseudo code corresponding to MOGSA

```
procedure MOGSA(MaxIter, PopNum)
    Init_Pop();
    Init_Velocity();
    Add_Nondom2Rep();
    for i=1:MaxIter do
        Update_Grav_Constant();
        Calc_Mass();
        Select_Kbest();
        for k=1:PopNum do
            Update_Force();
            Update_Accel();
            Update_Velocity();
            Update_Pos();
            Add_Nondom2Rep();
        end for
        Mutate();
        Keeping();
    end for
end procedure
```

individuals among the resulting population that are non-dominated are added to the archive. Finally the keeping functionality removes those solutions that recently become dominated in the archive.

3 Test Functions

To evaluate the performance of our approach and comparing results with other similar works, three well-know test functions are selected from Coello-Coello et al. (2002):

MOP5: This is an unconstrained three-objective function having asymmetric and disconnected Pareto front in the search space. The Pareto front is however connected in the objective space. MOP5 is defined as follows,

$$
\begin{aligned}
&F = (f_1(x,y), f_2(x,y), f_3(x,y)) \quad -30 \leq x, y \leq 30 \\
&f_1(x,y) = 0.5 \times (x^2 + y^2) + \sin(x^2 + y^2) \\
&f_2(x,y) = \frac{(3x-2y+4)^2}{8} + \frac{(x-y+1)^2}{27} + 15 \\
&f_3(x,y) = \frac{1}{x^2+y^2+1} - 1.1 \times e^{-(x^2+y^2)}
\end{aligned}
\tag{5}
$$

MOP6: This is an unconstrained two-objective function constructed from Debs methodology. Its Pareto front is comprised of four disconnected curves in objective space. The Pareto front in search space is also disconnected. The definition is as follows:

$$
\begin{aligned}
&F = (f_1(x,y), f_2(x,y)) \quad 0 \leq x, y \leq 1 \\
&f_1(x,y) = x \\
&f_2(x,y) = (1 + 10y) \times \left[1 - (\tfrac{x}{1+10y})^2 - \tfrac{x}{1+10y} \times \sin(2\pi 4x)\right]
\end{aligned}
\tag{6}
$$

MOPC1: This is a constrained two-objective function introduced by Binh and Korn. The Pareto front in search space and the objective space form an area and a convex curve, respectively. The definition of this function is as follows:

$$F = (f_1(x, y), f_2(x, y)) \quad 0 \le x \le 5,\ 0 \le y \le 3$$

$$\begin{aligned} f_1(x, y) &= 4x^2 + 4y^2 \\ f_2(x, y) &= (x-5)^2 + (y-5)^2 \end{aligned} \tag{7}$$

$$s.t. \quad \begin{cases} (x-5)^2 + y^2 - 25 \le 0 \\ -(x-8)^2 - (y+3)^2 + 7.7 \le 0 \end{cases}$$

4 Performance Measures

According to Cagnina and Coello Coello (2005), to evaluate the performance of a multi-objective algorithm there are important criteria suggested in the relevant literature signifying: 1) how well the solutions in objective space are distributed along the found Pareto curve 2) how well the recognized Pareto matches the real Pareto front (also called true Pareto), in objective space. Doing so, the following two metrics introduced by Veldhuizen (1999); Schott (1995) are selected for further comparisons, as follows,

Spacing (S): This metric demonstrates the variance of d_is (to be defined shortly). In other words it states how well the found solutions are uniformly distributed. More specifically it is defined as:

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \tag{8}$$

with d_i defined as,

$$\begin{aligned} &d_i = \min_j \left(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|\right) \\ &i, j = 1, 2, \ldots, n \end{aligned} \tag{9}$$

where $\bar{d}$ is the mean value over all d_is. Note that the lower this metric becomes, the solutions more uniformly spaced.

Generational Distance (GD): This criterion expresses the average distance between the solutions found by the algorithm and the true Pareto front. Specifically, it is defined as follows,

$$GD = \frac{1}{n}\sqrt{\sum_{i=1}^{n} d_i^2} \tag{10}$$

Where n is the number of solutions found and d_i is the Euclidean distance between each solution and its nearest point in the true Pareto front. The lower this value becomes the better known Pareto approaches the real one.

5 Simulation Results

To evaluate our new method three experiments were designed based on the test functions introduced in Section 3. The results are then compared and contrasted with that of SMOPSO, PAES and MOGA2 (the facts and figures corresponding to last three techniques are quoted and clipped from the work published by Cagnina and Coello Coello (2005)) which are powerful techniques for handling multi-objective problems.

SMOPSO introduced by Cagnina and Coello Coello (2005) is a well-knwon multi-objective PSO which takes advantage of a grid based archive and the operators derived

Table 1: Parameter settings for MOGSA

Parameters	MOGSA		
	MOP5	MOP6	MOPC1
Iterations	7000	3000	2000
Archive Size	799	799	799
G_0	1.5	1.5	1.5
β	7	7	7
Mutation Prob.	0.5	0.0335	0.3
Particles/Individuals	30	20	20
Number of segments in each dimension	5	5	5

Table 2: Parameter settings for SMOPSO

Parameters	SMOPSO		
	MOP5	MOP6	MOPC1
Iterations	7000	3000	2000
Archive Size	799	799	799
Mutation Prob.	0.5	0.0335	0.3
Particles/Individuals	30	20	20
Number of segments in each dimension	5	5	5
$C_1 = C_2$	1.5	1.6	1.5
W	0.5	0.6	0.5

Table 3: Parameter settings for PAES

Parameters	PAES		
	MOP5	MOP6	MOPC1
Iterations	210000	60000	40000
Archive Size	799	799	799
Mutation Prob.	0.03	0.05	0.05
Particles/Individuals	1	1	1
Number of segments in each dimension	5	6	5

from evolutionary algorithms. On the other hand, PAES proposed by Fonseca and Fleming (1992), is a [1+1] - evolution strategy (i.e. it applies the mutation operator on a child and generates only one child from one parent) which conducts a local search and keeps the previously found solutions in an archive for identification of the approximate dominance ranking of the current and candidate solution vectors. Finally MOGA2, suggested by Fonseca and Fleming (1992), is a multi-objective version of genetic algorithm which is based on a Pareto ranking technique. The rank of each individual in this method is the number of times it becomes dominated by others. In other words the less an individual gets dominated the lower its rank and correspondingly a higher priority it would be attributed to among others.

Tables 1-4 demonstrate the parameter settings for each algorithm through different scenarios. These are the best parameters which are empirically found by means of a trial and error approach.

The resulting solutions are then evaluated in regard of the true Pareto fronts. These

Table 4: Parameter settings for MOGA2

Parameters	**MOGA2**		
	MOP5	MOP6	MOPC1
Iterations	7000	3000	2000
Archive Size	799	799	799
Crossover Prob.	0.8	0.8	0.8
Mutation Prob.	0.25	0.25	0.25
Particles/Individuals	30	20	20

Table 5: GD measure resulted from different algorithms for different testbenches

Testbench Functions	**GD**			
	MOGSA	SMOPSO	PAES	MOGA2
MOP5	0.0011	0.011083	0.167133	0.727717
MOP6	2.4631e-005	0.000298	0.02204	0.000999
MOPC1	0.0032	0.002687	0.01986	2.441103

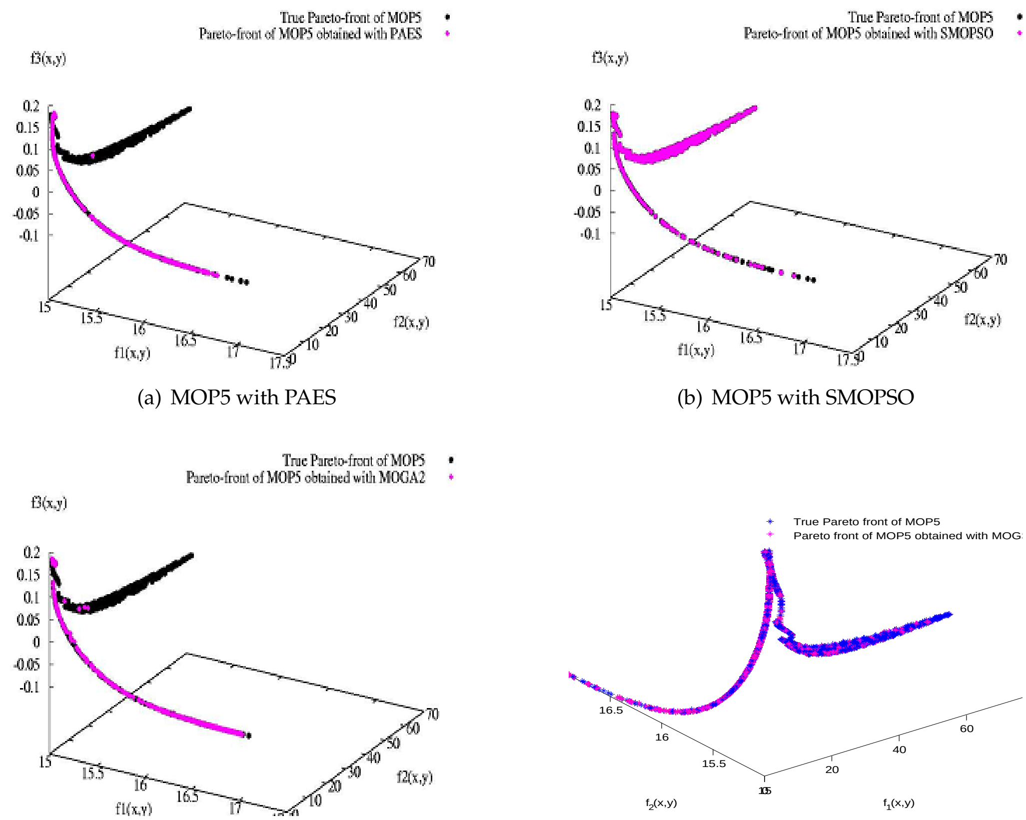
Table 6: S measure resulted from different algorithms for different test benches

Testbench Functions	**S**			
	MOGSA	SMOPSO	PAES	MOGA2
MOP5	0.1352	0.39566	0.378167	0.200790
MOP6	0.0010	0.003402	0.011853	0.000908
MOPC1	0.2043	0.116149	0.32289	0.942198

true fronts are acquired in advance by means of a multi-objectie meta-heuristic with large values assigned to their parameters (ie. size of population, maximum number of iterations). Tables 5-6 illuminate resulting values of the two metrics already mentioned for each of the foregoing methods.

As easily observed from these tables, the outcomes of MOGSA appear either to be rather the same or to be in a better position when considering the other three. As an evidence, Figure 3 depicts the graphs of known and true Pareto fronts corresponding to MOP5 benchmark function during the course of different scenarios for each method. As inferred by the Tables 5-6, for MOP5, the GD factor in our case outperforms by an order of magnitude compared to best GD experienced in other three. The S factor has also been greatly improved. Figure 4 portrays the graphs of known Pareto front and the true Pareto front of the MOP6 benchmark function for each of the four methods. Notice how obtained solutions for MOGSA are uniformly distributed on the true Pareto. This is in harmony with the results noted in Table 6. Finally, Figure 5 shows the results obtained for the MOPC1 test function. According to the aforementioned tables, the results of MOGSA compete with that of SMOPSO and perform better than the other two.

Overall, MOGSA emerges to offer a significant contribution to the multi-objective literature. Note that MOGSA is still in its infancy era and future variations will undoubtedly improve this algorithm to more matured ones.



(a) MOP5 with PAES

(b) MOP5 with SMOPSO

(c) MOP5 with MOGA2

(d) MOP5 with MOGSA

Figure 3: MOP5 with PAES, SMOPSO, MOGA2, MOGSA

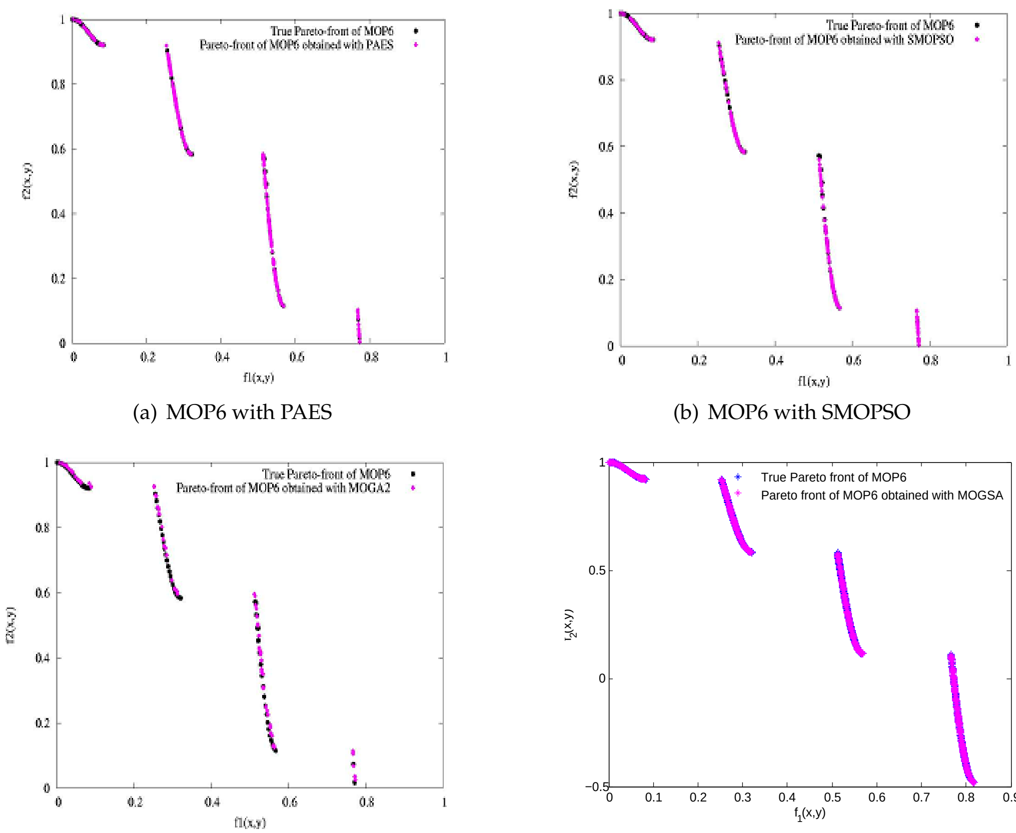
6 Conclusion & Future works

In this article we introduced a new hybridized method for handing multi-objective optimization problems and inspired from gravitational forces in the universe. The approach is based on Pareto optimality and takes advantage of operators derived from evolutionary algorithms. Comparing the results with the recent successful methods, MOGSA emerges to outperform promisingly. In order to come to a better conclusion some notes about the proposed hybrid should be underlined,

1. Since each particle in the search space is aware of the others, the information is shared among all the moving ones.

2. As the MOGSA exploits the gravitational fields, the particles are attracted to heavier masses which represent those regions around the Pareto front that the particles are not distributed uniformly (as the masses were defined in this way) and hence gives rise to a better distribution of solutions without directly applying any niching method.

3. By tuning the K-best parameter, during the time, the exploration fades out and exploitation fades in and hence it serves as a degree of freedom in tuning a balance



(a) MOP6 with PAES

(b) MOP6 with SMOPSO

(c) MOP6 with MOGA2

(d) MOP6 with MOGSA

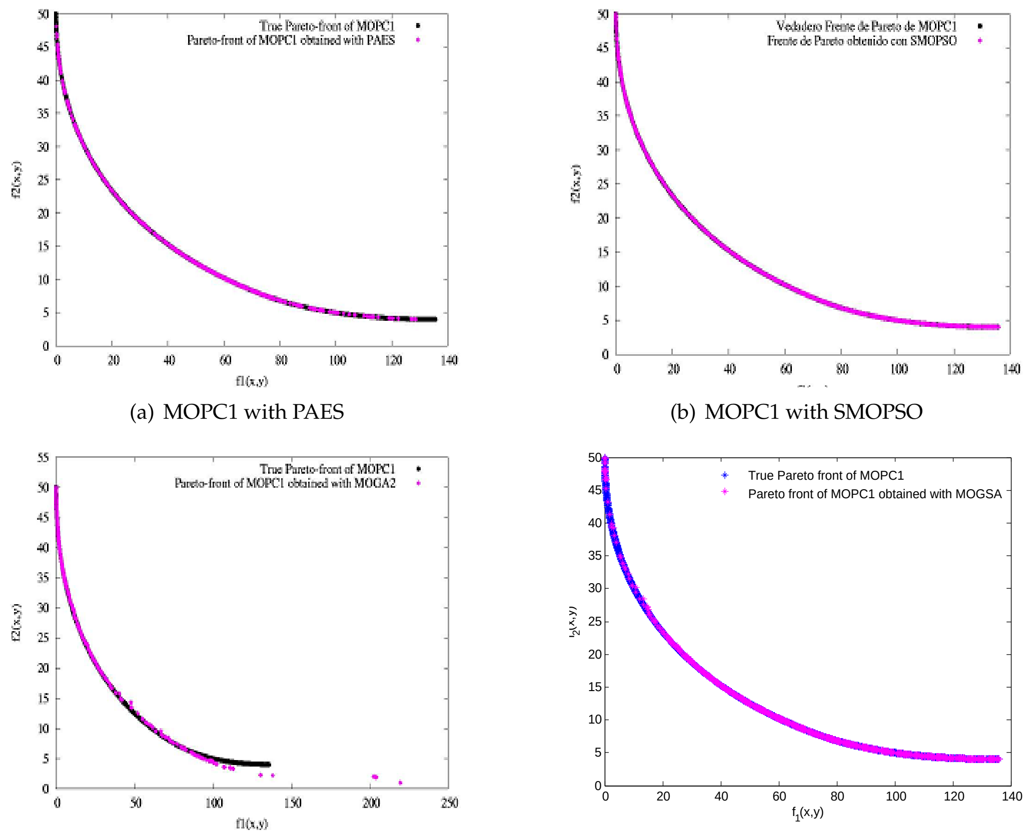
Figure 4: MOP6 with PAES, SMOPSO, MOGA2, MOGSA

between these two important factors.

4. Decreasing the gravitational constant during the time implies a finer search around the optimums in the last iterations (similar to that of simulated annealing)

Finally it is worthy of noting that although gravitational based algorithms have shown their efficiency but it should be remarked that the time taken for execution of one iteration in MOGSA is slightly higher than its counterparts for other conventional methods such as PSO, a side-effect remedied as the K-best decreases during the time.

As a future work one can introduce the notion of fuzzy multi-objective gravitational search algorithm (FMOGSA). The fuzziness may be applied to the Pareto front Kppen et al. (2005) such that for each solution in the archive there is be a degree of membership denoting to what extent the solutions are considered as being non-dominated. Moreover, in this article the procedure to selecting the leaders is simply based on a K-best mechanism. Paying attention to the fact that choosing appropriate particles as leaders has a considerable effect on the generations' average fitness, makes it clear that better selection policy improves the algorithm's performance. Therefore, as another future work one may consider a new leader selection mechanism for MOGSA.



(a) MOPC1 with PAES

(b) MOPC1 with SMOPSO

(c) MOPC1 with MOGA2

(d) MOPC1 with MOGSA

Figure 5: MOPC1 with PAES, SMOPSO, MOGA2, MOGSA

References

Backxxxxxxxx, T., Fogel, D., and Michalewicz, Z., editors (1997). *Handandbook of Evolutionary Computation*. IOP Publishing Ltd and Oxford University Press.

Cagnina, L.and Esquivel, S. and Coello Coello, C. (2005). A particle swarm optimizer for multi-objective optimization. *JCS&T*, 4(5):204–210.

Carlos, C., Gregprop, T., and Maximino, S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8:256–279.

Coello-Coello, C., Vedhuizen, D. V., , and Lamont, G., editors (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, USA.

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Ltd., England.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGAII. *xxxxxxxxxxxxxxxxxxxxxx*, xxxxxxxxxxxxxxxxxx(xxxxxxxxxxxxxx):849–858.

Fonseca, C. and Fleming, P. (1992). Genetic algorithms for multiobjective optimization formulation, discussion and generalization. In *Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, CA. Morgan Kaufmann Publishers.

Hassanzadeh, H. and Rouhani, S. M. (2010). A multi-objective gravitational search algorithm. In *xxxxxxxxxxxxxxxxxx*.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *International Conference on Neural Networks*, pages 1942–1948, Piscataway, NJ. IEEE Service Center.

Kennedy, J. and Spears, W. (1998). Matching algorithms to problems: An experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator. In *IEEE World Congress on Computational Intelligence*, pages 74–77.

Knowles, J. and Corne, D. (2000a). M-paes: A memetic algorithm for multi-objective optimization. In *Congress on Evolutionary Computation*, pages 325–332, Piscataway, NJ. IEEE Service Center.

Knowles, J. D. and Corne, D. W. (2000b). Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8:149–172.

Kppen, M., Vicente-Garcia, R., and Nickolay, B. (2005). Fuzzy-pareto-dominance and its application in evolutionary multi-objective optimization. In *EMO2005*, pages 399–412.

Laumanns, M., Zitzler, E., and Thiele, L. (2000). A unified model for multi-objective evolutionary algorithms with elitism. In *Congress on Evolutionary Computation*, pages 46–53, Piscataway, NJ. IEEE Service Center.

Li, S.-T., Chen, C.-C., and Li, J. W. (2007). A multi-objective particle swarm optimization algorithm for rule discovery. In *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP2007*, pages 597–600.

Li, X. (2003a). *Better Spread and Convergence: Particle Swarm Multi-objective Optimization Using the xxxxxxxxxxxxxxxx Fitness Function*, pages 117–128. xxxxxxxxxxxxxx.

Li, X. (2003b). *A Non-dominated Sorting Particle Swarm Optimizer for Multi-objective Optimization*, pages 37–48. xxxxxxxxxxxxxx.

Mansouri, R., Nasseri, F., and Khorrami, M. (1999). Effective time variation of g in a model universe with variable space dimension. *Physics Letters*, 259:194–200.

Rashedi, E., Nezamabadi-pour, H., and Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Science*, 179:2232–2248.

Schott, J. (1995). Fault tolerant design using single and multi-criteria genetic algorithms. Master's thesis, Massachusetts Institute of Technology, Boston.

Schxxxxxxxxxxxxxxxxxxx, J. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In *First International Conference on Genetic Algorithms*, page xxxxxxx.

Veldhuizen, D. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Air Force Institute of Technology, Dayton.

Wickramasinghe, U. K. and Li, X. (2008). Choosing leaders for multi-objective pso algorithms using differential evolution. In *Proceedings of SEAL2008*, pages 249–258.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparision of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195.