Multi Algorithmic Approach to Galactic Swarm Optimization (MAGSO)

Vasanthakumar Kathirvelu¹, Venkataraman Muthiah-Nakarajan¹
¹School of Electrical Engineering, Vellore Institute of Technology, Vellore - 632 014, India.
E-mail: mnvenkataraman@vit.ac.in

Abstract. Multi population algorithms have noted advantages over single population algorithms when optimizing over multimodal function space. The existence of multiple attractors allows exploration on multiple optimality regions which in turn provide more exploration compared with single population algorithms. This paper is a novel utilization of Galactic Swarm Optimization (GSO) framework in which the subpopulations are evolved with multiple variants of Particle Swarm Optimizers (PSOs) rather than single PSO. The proposed algorithm is called Multi Algorithmic Galactic Swarm Optimization (MAGSO). MAGSO utilizing multivariant PSOs in subswarm level combined with the unique superswarm tailored for exploitation has shown significant improvement of performance over GSO. Experiments done on Congress on Evolutionary Computation 2013 (CEC 2013) and other standard general benchmark suite indicate that the proposed MAGSO algorithm outperforms the previously proposed state-of-the-art swarm algorithms such as Bollinger bands approach on boosting ABC (ABCBB) and Dynamic Neighbourhood Learning Particle Swarm Optimizer (DNLPSO) besides GSO. Detailed statistical evidence over 51 independent trials on the benchmark functions indicate the best performance of MAGSO. The main research goal is to propose a superior performing algorithm by exploiting the GSO framework. In order to show its efficacy, the algorithm is compared with state-of-the-art swarm algorithms such as ABCBB, DNLPSO and unmodified GSO. The method proposed here is quite general and demonstrates that more ability and extension work possible from the basic GSO framework.

1. Introduction
Global optimization of multimodal functions occurs in problems of many scientific [1] [2] [3], engineering [4] [5] [6] and economic [7] [8] [9] disciplines. In general, the global optimization problem of multimodal functions can be mathematically expressed as:

\[ \mathbf{x}^* = \arg\min_{\mathbf{x}} f(\mathbf{x}) \]  

where \( f \) is called the objective function, \( \mathbf{x} \in [x_{\min}, x_{\max}]^D \) is a vector of \( D \) dimensions and \( \mathbf{x}^* \) is the global minimum location. \( x_{\min} \) and \( x_{\max} \) are the lower and upper limits of each components of the vector \( \mathbf{x} \). In Eq.(1) optimization is assumed to be a minimization operation.
On the other hand if maximization is desired it can be achieved by simply negating the objective function.

In these problems the soft computing approach is to propose suitable heuristic algorithms that can fetch the best local minimum. Multimodal functions require algorithms to have sound exploration for wide global search and escape from being stuck at local minimum. Differential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Algorithms (GA) and other nature inspired algorithms are proven to be useful in multimodal search space where the traditional derivative based algorithms fail. Unlike classical algorithms, DE, PSO and GA operate with a population of particles and use random numbers to enhance search capability.

No free lunch theorems proven by Wolpert and McCreary in 1997 [10] has dictated that on an average any two algorithms will have the same performance when optimizing over all function space. The claims laid down by no free lunch theorems should discourage the development of new algorithms on global optimization. However, the synthesis of new and powerful algorithms are proposed incessantly. The proponents are rather more interested in developing new algorithms that perform well in practically important functions. It may be impossible to design an algorithm that does well on entire function space but algorithms that can work well on practical problems will continue to receive attention.

Multipopulation algorithms work under the premise that many populations instead of one, acting simultaneously can bring in the benefits of parallel search. Multiswarms though framed on this vision, however while implementing the search multiswarms are focussed towards a single global exemplar [11]. This defeats the main purpose of multiswarms. Galactic Swarm Optimization (GSO) is the first algorithmic framework which enables multiple swarms focus around different attractors [12]. GSO showed superior performance over the state-of-the-art PSOs on DeJong benchmarks [12]. Recent studies by [13] have shown that GSO yields best results if Artificial Bee Colony Algorithm is substituted in place of the PSO.

In this paper, the authors propose a Multi Algorithmic Galactic Swarm Optimization (MAGSO) an optimizer that improves the performance of a previously proposed Galactic Swarm Optimization (GSO) algorithm. GSO is a physics inspired framework which uses Particle Swarm Optimization as the main algorithm to solve global optimization problems [12]. There is more freedom and flexibility available for better utilization of the GSO framework rather than just changing the basic algorithm. MAGSO is one such better exploitation of GSO framework.

The presence of non-interacting subswarms was underlined to be the main contribution of GSO [12]. Instead of using PSO as the main algorithm the variants of PSO are tried in the subswarm level in MAGSO. Finally, a superswarm which is better in local search is used to improve the search ability.

The paper is organized as follows. In Section 2, GSO is reviewed briefly. Section 3 explains the construction of MAGSO. Experiments conducted on Congress of Evolutionary Computation 2013 (CEC 2013) benchmark functions, and set of 10 demanding general functions followed by statistical comparison suggests that the proposed MAGSO gives superior performance over the existing GSO and the other leading swarm algorithms such as Bollinger bands approach on boosting ABC (ABCBB) and Dynamic Neighbourhood Learning Particle Swarm Optimizer (DNLPSO). The results are discussed in Section 4. Finally, Section 5 concludes the paper and discusses possible future research directions.
2. GSO Review

GSO is a global optimization algorithmic framework that can fit in any population-based metaheuristic to search best local optimum. The movement equations in GSO depend on galactic motion [12]. In GSO the population is divided into subpopulations, each of which serves as a subswarm [12]. The subswarms undergo PSO movement, and the global bests are collected to form the superswarm, which also undergoes PSO movement [12].

At the algorithmic level, GSO is a framework that provides a facility for non-interacting metaheuristic subswarms to explore the search space independently. Exploration is a crucial feature when dealing with multimodal problems. The subswarms do not exchange information in GSO. The subswarms must be aggressive in global search, and the superswarm should have high local search ability. Alternating subswarm and superswarm movements by multiple epochs provide an alternation of global and local search yielding better solutions when compared with a single population-based algorithms.

The movement equations of GSO subswarm is given below:

\[
\begin{align*}
\mathbf{v}_j^i &= \omega_1 \mathbf{v}_j^i + c_1 r_1 (\mathbf{p}_j^i - \mathbf{x}_j^i) + c_2 r_2 (\mathbf{g}_i - \mathbf{x}_j^i); \\
\mathbf{x}_j^i &= \mathbf{v}_j^i + \mathbf{x}_j^i;
\end{align*}
\]

where \(x_j^i, \mathbf{v}_j^i, \mathbf{p}_j^i\) are the position of particle \(j\), velocity of the particle \(j\), personal best of particle \(j\) belonging to subswarm \(i\). \(\mathbf{g}_i\) is the global best of the subswarm \(i\). Unlike multiswarm algorithms proposed earlier GSO does not allow leakage of information via regrouping or retention of particle memory.

The superswarm is obtained by collecting all \(\mathbf{g}_i\) from the subswarm stage and it undergoes movement as given below:

\[
\begin{align*}
\mathbf{v}_i &= \omega_2 \mathbf{v}_i + c_3 r_3 (\mathbf{p}_i - \mathbf{x}_i) + c_4 r_4 (\mathbf{g} - \mathbf{x}_i); \\
\mathbf{x}_i &= \mathbf{v}_i + \mathbf{x}_i;
\end{align*}
\]

where \(\mathbf{x}_i, \mathbf{v}_i, \mathbf{p}_i\) are the position, velocity and personal best of particle \(i\) in superswarm. \(\mathbf{g}\) is the galactic best. In GSO the communication from \(\mathbf{g}\) to subswarms is arrested because it is malicious to the search process. This reasoning is in line with the general agreement that sharing of global best exemplar with the members of population dilutes the search rather than enhancing [14]. For a detailed explanation of GSO, the interested reader is referred to [12].

3. MAGSO Construction

MAGSO is a multipopulation algorithm like GSO; however, the four different PSO variants drive the subpopulations. In this work, MAGSO has four PSO components.
They are:

(i) Regular PSO (RPSO),
(ii) Comprehensive Learning Particle Swarm Optimizer (CLPSO)
(iii) Fully Informed Particle Swarm Optimizer (FIPSO)
(iv) Fitness-Distance Ratio based Particle Swarm Optimization (FDR-PSO)

Algorithm 3.1: RPSO($x_{j}^{(i)}, v_{j}^{(i)}, p_{j}^{(i)}, g^{(i)}$)

\[
\text{for } k \leftarrow 1 \text{ to } L_1 \\
\quad \text{for } j \leftarrow 1 \text{ to } N \\
\quad \quad \text{do } \begin{cases} 
\quad \quad v_{j}^{(i)} \leftarrow v_{j}^{(i)} + c_1 r_1 (p_{j}^{(i)} - x_{j}^{(i)}) + c_2 r_2 (g^{(i)} - x_{j}^{(i)}) \\
\quad \quad x_{j}^{(i)} \leftarrow x_{j}^{(i)} + v_{j}^{(i)} \\
\quad \end{cases} \\
\quad \text{for } j \leftarrow 1 \text{ to } N \\
\quad \quad \text{do } \begin{cases} 
\quad \quad \text{if } f(x_{j}^{(i)}) \leq f(p_{j}^{(i)}) \\
\quad \quad \quad \text{then } p_{j}^{(i)} \leftarrow x_{j}^{(i)} \\
\quad \quad \text{if } f(x_{j}^{(i)}) \leq f(g^{(i)}) \\
\quad \quad \quad \text{then } g^{(i)} \leftarrow x_{j}^{(i)} \\
\quad \quad \text{if } f(x_{j}^{(i)}) \leq f(g) \\
\quad \quad \quad \text{then } g \leftarrow x_{j}^{(i)} \\
\end{cases}
\]

RPSO is the unmodified form of the classical PSO, whose details are available below. CLPSO was proposed by Liang et al., in the year 2006 [15]. In CLPSO the concept of gbest attraction was abandoned, and the movement equations depend on swarm’s own or other personal bests [15]. In FIPSO, instead of just moving towards the local neighbor’s influence, the swarm is guided by the informed movement of all members of the swarm [16] (2004). In FDR-PSO, the particles move towards higher fitness, which is computed by fitness distance ratio rather than the immediate past best exemplar [17] (2003).

Because different PSO variants can have a different effect on the function, the performance result after the subswarm stage has varying degrees of exploration and exploitation tendencies for different functions. An algorithm exploring aggressively on a function could have low exploitation tendency when acting on the same function. This varying degree of exploration/exploitation search gives MAGSO an advantage over differing landscapes as normally encountered by optimizers on multimodal functions. This also allows MAGSO to work effectively on multiple types of functions over competing algorithms.
Algorithm 3.2: MAGSO\((f)\)

Level 1 Initialization: \(x_j^{(i)}, v_j^{(i)}, p_j^{(i)}, g^{(i)}\) within \([x_{\text{min}}, x_{\text{max}}]^D\) randomly.

Level 2 Initialization: \(v^{(i)}, p^{(i)}, g\) within \([x_{\text{min}}, x_{\text{max}}]^D\) randomly.

\[
\text{for } EP \leftarrow 1 \text{ to } EP_{\text{max}} \begin{cases} \text{Begin Level 1 - Subswarms} \\
\text{for } i \leftarrow 1 \text{ to } \left\lceil \frac{M}{4} \right\rceil \\
\text{ } \text{do } RPSO(x_j^{(i)}, v_j^{(i)}, p_j^{(i)}, g^{(i)}, g) \\
\text{for } i \leftarrow \left\lceil \frac{M}{4} \right\rceil + 1 \text{ to } \left\lceil \frac{2M}{4} \right\rceil \\
\text{ } \text{do } CLPSO(x_j^{(i)}, v_j^{(i)}, p_j^{(i)}, g^{(i)}, g) \\
\text{for } i \leftarrow \left\lceil \frac{2M}{4} \right\rceil + 1 \text{ to } \left\lceil \frac{3M}{4} \right\rceil \\
\text{ } \text{do } FIPSO(x_j^{(i)}, v_j^{(i)}, p_j^{(i)}, g^{(i)}, g) \\
\text{for } i \leftarrow \left\lceil \frac{3M}{4} \right\rceil + 1 \text{ to } M \\
\text{ } \text{do } FDR-PSO(x_j^{(i)}, v_j^{(i)}, p_j^{(i)}, g^{(i)}, g) \\
\text{Begin Level 2 - Superswarm} \\
\text{Initialize Swarm } x^{(i)} = g^{(i)} : i = 1, 2, \cdots, M; \\
\text{for } i \leftarrow 1 \text{ to } M \begin{cases} \text{for } k \leftarrow 0 \text{ to } L_2 \begin{cases} \text{do } \begin{cases} \text{if } f(x^{(i)}) < f(p^{(i)}) \\
\text{then } \begin{cases} p^{(i)} \leftarrow x^{(i)}; \\
\text{if } f(p^{(i)}) < f(g) \text{ then } g \leftarrow p^{(i)}; 
\end{cases} \end{cases} \end{cases} \\
\end{cases}
\end{cases}
\end{cases}
\]

MAGSO is a opportunistic better utilization of the GSO framework, by using the multiple algorithms effectively. All subswarms are initialized randomly within the search space. At the subswarm level, the MAGSO is built identically to GSO excepting variants of PSO are used instead of the single PSO version. Considering \(M\) subswarms, each of \(\frac{M}{4}\) subswarms evolve RPSO, CLPSO, FIPSO, and FDR-PSO, sequentially. The parameters in subswarm stage are chosen in such a way to encourage exploration.

This parameter study is included in experimental section of the paper. For the algorithm to make best use of the subswarm information the superswarm parameters are chosen to focus on a relatively smaller region. This boosts the exploitation capability. The varying degrees of search as provided by different optimizers in subswarm and superswarm stages results in a synergistic performance as noted by superior experimental results.

Algorithm 3.1 represents the implementation of the regular PSO (RPSO) used as a member in the subswarm stage of MAGSO. The movement equations of RPSO are different from the superswarm PSO as used in the original GSO. The complete psuedocode of MAGSO is illustrated in Algorithm 3.2.

The difference is in an iteration, all particles are driven towards the same global best, whereas in superswarm, the global best keeps on updating after every swarm member movement. All
particles in subswarm get synchronized to the same global best. Keeping the global best fixed in
time over an iteration avoids fast reaction of swarm members avoiding premature convergence.
All PSO variants employed in the subswarm adopt this way of gbest update. By the same
reasoning, the asynchronous PSO possessing better global best tracking ability yields better
exploitation, hence is useful in the superswarm stage. The combination of synchronous PSOs
in subswarm stage and asynchronous PSO in the superswarm stage as employed in MAGSO is
experimentally confirmed to give better solutions.

Note the inertial weight to speed up convergence does not appear in RPSO. Absence of inertial
weight does not narrow the search range and hence understandably promotes exploration, as
the search range is now not restricted with time in the subswarm stage. Whereas inertial weight
is introduced in superswarms to encourage exploitation.

4. Results and discussion
In our work, we have demonstrated the robust performance of MAGSO by comparing with
Bollinger bands approach on boosting ABC (ABCBB), Dynamic Neighbourhood Learning
Particle Swarm Optimizer (DNLPSO) and the original version of GSO.

These algorithms are chosen as they are all recently developed and motivated by biological
swarm movements. The GSO though an physics inspired framework, its members belong to the
swarm algorithm (PSO) category. The authors consider that choosing the competing algorithms
from a pool of swarm intelligence makes the comparison fair.

To cover more on the function space, the general functions given in Table.1 are also considered
for comparison.

Each of the benchmark function $f$ has 6 entries corresponding to mean error $\mu$, standard
deviation $\sigma$, median $m_d$, best value $b$, worst value $w$ and relative rank $r$ secured by the algorithm
for 51 independent trials.

The statistical analysis presented in this paper supports the claim that multiple variants of
PSO in place of the single version along with ranking procedure produces best results. Table.2
indicates a model entry of such statistical information for each algorithm.

Each algorithm is run for 51 independent trials and the essential statistical information are
given in Table 3 and Table 4 for 10 dimensions. Similarly, Table 5 and Table 6 showcases the
results for 30 dimensions.

The result tables shows that for unimodal functions $f_1$-$f_5$ MAGSO leads except for $f_4$.
Though unimodal minimization is not considered a special achievement, it must be noted that
among the swarm category this is competitive.

In the space of basic multimodal functions from $f_6$-$f_{26}$ MAGSO performance is superior than
the other algorithms. The performance of MAGSO on hybrid functions such as $f_{27}$ and $f_{28}$ is
notable. Though, GSO proves to be competitive in the general function space, on an overall
space among 38 functions MAGSO stands out with least average rank (A.R.).

In moving to 30 dimensions the lead position is continuously maintained which implies that
MAGSO performance is not hindered by effects such as curse of dimensionality. In both $10D$
and $30D$, using the average rank as performance metric, we see MAGSO is followed next by
ABCBB, DNLPSO and GSO.

CEC 2013 benchmark functions as well as a certain general functions are used as the hallmark
for comparison in this work. CEC 2013 benchmark [18] is adopted as a standard in many
competetions [19] [20] [21] and literature [22] [23] [24].
Table 1. Details of general test functions used for comparison in this study.

| Function       | Definition                                                                 |
|----------------|-----------------------------------------------------------------------------|
| Weierstrass    | $f_{29}(x) = \sum_{i=1}^{D} \left( \sum_{j=0}^{K} \left( a^j \cos(2\pi b^j (x_i + 0.5)) \right) \right) - D \sum_{j=0}^{K} \left( a^j \cos(\pi b^j) \right); \quad a = 0.5, b = 3, K = 20$ |
| Rastrigin      | $f_{30}(x) = 10D + \sum_{i=1}^{D-1} (x_i^2 - 10 \cos(2\pi x_i))$             |
| Rosenbrock     | $f_{31}(x) = 10D + \sum_{i=1}^{D} \left( 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right)$ |
| Griewangk      | $f_{32}(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos \left[ \frac{x_i}{\sqrt{1}} \right] + 1$ |
| Ackley         | $f_{33}(x) = 20 - \exp \left[ \frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i) \right] + e - 20 \exp \left[ -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right]$ |
| Rotated Weierstrass | $f_{34}(y) = \sum_{i=1}^{D} \left( \sum_{j=0}^{K} \left( a^j \cos(2\pi b^j (y_i + 0.5)) \right) \right) - D \sum_{j=0}^{K} \left( a^j \cos(\pi b^j) \right); \quad a = 0.5, b = 3, K = 20$ |
| Rotated Rastrigin  | $f_{35}(y) = 10D + \sum_{i=1}^{D} (y_i^2 - 10 \cos(2\pi y_i)) \quad y = M*x$ |
| Rotated Rosenbrock   | $f_{36}(y) = 10D + \sum_{i=1}^{D-1} (100(y_{i+1} - y_i^2)^2 + (1 - y_i)^2) \quad y = M*x$ |
| Rotated Griewangk   | $f_{37}(y) = \sum_{i=1}^{D} \frac{y_i^2}{4000} - \prod_{i=1}^{D} \cos \left[ \frac{y_i}{\sqrt{1}} \right] + 1 \quad y = M*x$ |
| Rotated Ackley    | $f_{38}(x) = 20 - \exp \left[ \frac{1}{D} \sum_{i=1}^{D} \cos(2\pi y_i) \right] + e - 20 \exp \left[ -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} y_i^2} \right] \quad y = M*x$ |

Table 2. Statistical Model Entry

| $f$ | $\mu$ | $\sigma$ | $m_d$ | $b$ | $w$ | $r$ |

CEC 2013 has a total of 28 functions out of which 5 are unimodal functions, 15 are basic multimodal functions and 8 are composition functions [18]. The maximum function evaluations are limited to 10000$D$, $D$ being the dimensionality of the problem. Tests are carried out for 10$D$ and 30$D$ cases to show that the results are robust at higher dimensions. The methods as recommended by CEC 2013 comparison is also adapted for general functions. Table 1 shows the function definition of general functions.
Table 3. Comparison table for 10 dimensions ($f_1 - f_{19}$).

| $f$ | MAGSO | ABCBB | GSO | 6.615E-02 | 4.358E-01 | 2.353E+00 |
|-----|-------|-------|-----|------------|------------|------------|
| $f_1$ | 0     | 0     | 0   | 5.500E+00 | 3.393E+01 | 6.615E-02 |
| $f_2$ | 6.882E+01 | 5.986E+04 | 5.164E+04 | 2.181E+06 | 1.045E+06 | 2.036E+03 |
| $f_3$ | 5.896E+05 | 2.290E+06 | 3.812E+04 | 9.175E+06 | 1.315E+07 | 5.502E+06 |
| $f_4$ | 1.507E+03 | 1.297E+03 | 2.143E+03 | 8.177E+07 | 1.422E+10 | 1.640E+03 |
| $f_5$ | 1.270E+03 | 5.854E+02 | 1.195E+03 | 1.896E+04 | 1.743E+03 | 3.047E+03 |
| $f_6$ | 3.458E+02 | 2.741E+03 | 5.342E+03 | 9.371E+03 | 9.015E+03 | 2.056E+03 |
| $f_7$ | 0     | 0     | 0   | 2.353E+01 | 3.472E+01 | 4.491E+00 |
| $f_8$ | 3.403E+00 | 5.818E+00 | 2.448E+00 | 2.240E+00 | 3.351E+00 | 2.322E+00 |
| $f_9$ | 5.935E+00 | 6.491E+00 | 2.448E+00 | 1.251E+00 | 4.900E+00 | 8.888E+00 |
| $f_{10}$ | 2.035E+01 | 1.371E+00 | 4.765E+00 | 1.098E+00 | 2.036E+00 | 1.200E+00 |
| $f_{11}$ | 3.035E+00 | 1.097E+00 | 3.699E+00 | 3.947E+00 | 6.578E+00 | 5.655E+00 |
| $f_{12}$ | 3.818E+01 | 2.321E+00 | 2.957E+00 | 1.362E+00 | 5.435E+00 | 9.884E+00 |
| $f_{13}$ | 5.414E+02 | 1.202E+01 | 6.487E+00 | 2.154E+00 | 6.254E+00 | 1.370E+00 |
| $f_{14}$ | 2.358E+02 | 1.064E+00 | 1.990E+00 | 3.047E+00 | 7.088E+00 | 5.821E+00 |
| $f_{15}$ | 3.872E+01 | 1.293E+00 | 1.422E+00 | 4.388E+00 | 1.407E+01 | 3.021E+01 |
| $f_{16}$ | 5.977E+00 | 2.557E+01 | 4.385E+00 | 2.378E+01 | 8.636E+00 | 9.164E+00 |
| $f_{17}$ | 6.161E+01 | 7.002E+01 | 1.661E+00 | 3.074E+00 | 8.013E+00 | 3.995E+01 |
| $f_{18}$ | 2.128E+00 | 1.314E+01 | 3.143E+00 | 1.234E+01 | 5.167E+00 | 3.101E+01 |
| $f_{19}$ | 3.554E+02 | 1.084E+02 | 3.786E+02 | 1.239E+01 | 5.976E+02 | 1.249E+01 |
| $f_{20}$ | 6.237E+02 | 1.084E+02 | 3.786E+02 | 1.239E+01 | 5.976E+02 | 1.249E+01 |
| $f_{21}$ | 5.972E+02 | 5.972E+02 | 3.786E+02 | 1.239E+01 | 5.976E+02 | 1.249E+01 |
| $f_{22}$ | 6.506E+02 | 5.972E+02 | 3.786E+02 | 1.239E+01 | 5.976E+02 | 1.249E+01 |
| $f_{23}$ | 6.519E+02 | 6.519E+02 | 3.786E+02 | 1.239E+01 | 5.976E+02 | 1.249E+01 |
| $f_{24}$ | 7.146E+01 | 2.450E+00 | 8.130E+00 | 1.190E+00 | 2.013E+01 | 7.532E+01 |
| $f_{25}$ | 6.003E+01 | 3.976E+01 | 6.003E+01 | 1.190E+00 | 2.013E+01 | 7.532E+01 |
| $f_{26}$ | 3.132E+01 | 2.183E+00 | 6.426E+00 | 1.746E+00 | 2.124E+01 | 3.746E+01 |
| $f_{27}$ | 1.709E+01 | 1.371E+00 | 1.103E+00 | 1.014E+01 | 1.144E+00 | 6.916E+00 |
| $f_{28}$ | 4.311E+01 | 6.190E+00 | 3.211E+00 | 4.417E+00 | 6.065E+00 | 3.873E+01 |
| $f_{29}$ | 5.791E+01 | 4.161E+00 | 2.961E+00 | 5.962E+00 | 4.204E+00 | 6.630E+00 |
| $f_{30}$ | 8.585E+00 | 8.585E+00 | 4.078E+00 | 1.751E+00 | 7.912E+00 | 1.024E+00 |
| $f_{31}$ | 3.576E+01 | 1.347E+00 | 1.072E+00 | 6.829E+00 | 1.218E+00 | 2.161E+00 |
## Table 4. Comparison table for 10 dimensions ($f_20 - f_{38}$) and A.R.

| $f$ | MAGSO | ABCBB | GSO | DNSFO |
|-----|-------|-------|-----|-------|
| $f_{20}$ | $2.565E+00$ | $5.128E+00$ | $2.871E+00$ | $3.831E+00$ | $2.518E+00$ |
| $f_{21}$ | $1.328E+00$ | $3.958E+00$ | $2.805E+00$ | $4.288E+00$ | $1.594E+01$ |
| $f_{22}$ | $3.503E+00$ | $9.120E+00$ | $4.002E+00$ | $2.346E+00$ | $1.978E+01$ |
| $f_{23}$ | $4.691E+00$ | $2.114E+00$ | $4.256E+00$ | $1.34E+00$ | $3.831E+00$ |
| $f_{24}$ | $7.291E+00$ | $1.131E+00$ | $4.167E+00$ | $1.03E+00$ | $2.912E+00$ |
| $f_{25}$ | $8.815E+00$ | $2.882E+00$ | $9.012E+00$ | $1.30E+00$ | $2.734E+00$ |
| $f_{26}$ | $2.399E+00$ | $1.610E+00$ | $6.356E+00$ | $1.664E+00$ | $5.734E+00$ |
| $f_{27}$ | $1.644E+00$ | $3.274E+00$ | $1.340E+00$ | $2.313E+00$ | $1.55E+00$ |
| $f_{28}$ | $1.135E+00$ | $2.138E+00$ | $1.239E+00$ | $2.197E+00$ | $2.734E+00$ |
| $f_{29}$ | $1.952E+00$ | $2.346E+00$ | $2.028E+00$ | $1.992E+00$ | $2.188E+00$ |
| $f_{30}$ | $1.308E+00$ | $2.142E+00$ | $1.356E+00$ | $2.245E+00$ | $1.650E+00$ |
| $f_{31}$ | $1.136E+00$ | $6.058E+00$ | $1.119E+00$ | $1.41E+00$ | $1.706E+00$ |
| $f_{32}$ | $1.004E+00$ | $1.318E+00$ | $1.397E+00$ | $1.789E+00$ | $2.015E+00$ |
| $f_{33}$ | $3.343E+00$ | $4.008E+00$ | $3.928E+00$ | $4.008E+00$ | $3.935E+00$ |
| $f_{34}$ | $2.699E+00$ | $8.297E-01$ | $3.000E-01$ | $2.558E+00$ | $1.596E+00$ |
| $f_{35}$ | $1.000E+00$ | $9.008E+00$ | $7.885E+00$ | $8.000E+00$ | $7.861E+00$ |
| $f_{36}$ | $5.924E+00$ | $1.743E+00$ | $5.663E+00$ | $0.000E+00$ | $0.000E+00$ |
| $f_{37}$ | $1.350E+00$ | $9.018E+00$ | $3.395E+00$ | $2.656E+00$ | $1.305E+00$ |
| $f_{38}$ | $2.926E-01$ | $9.596E-01$ | $1.000E+00$ | $0.000E+00$ | $9.566E+00$ |
| $f_{39}$ | $0.000E+00$ | $5.978E+00$ | $0.000E+00$ | $0.000E+00$ | $8.479E+00$ |
| $f_{40}$ | $4.332E+00$ | $8.738E+00$ | $1.030E+01$ | $2.814E+00$ | $4.034E+00$ |
| $f_{41}$ | $1.034E+00$ | $4.480E+00$ | $1.050E+01$ | $1.656E+01$ | $2.000E+00$ |
| $f_{42}$ | $0.000E+00$ | $0.000E+00$ | $3.984E-03$ | $5.935E+00$ | $8.479E-03$ |
| $f_{43}$ | $0.000E+00$ | $0.000E+00$ | $0.000E+00$ | $0.000E+00$ | $0.000E+00$ |
| $f_{44}$ | $0.000E+00$ | $0.000E+00$ | $0.000E+00$ | $0.000E+00$ | $0.000E+00$ |
| $f_{45}$ | $1.978E+00$ | $5.813E+00$ | $2.006E+00$ | $3.405E+00$ | $2.132E+00$ |
| $f_{46}$ | $6.597E+00$ | $1.999E+00$ | $6.839E+00$ | $5.368E+00$ | $7.649E+00$ |
| $f_{47}$ | $3.889E+00$ | $9.185E+00$ | $3.452E+00$ | $7.028E+00$ | $5.324E+00$ |
| $f_{48}$ | $3.889E+00$ | $9.185E+00$ | $3.452E+00$ | $7.028E+00$ | $5.324E+00$ |
| $f_{49}$ | $1.972E+00$ | $2.017E+00$ | $1.594E+00$ | $2.012E+00$ | $1.706E+00$ |
| $f_{50}$ | $1.387E+00$ | $1.387E+00$ | $4.179E+00$ | $3.831E+00$ | $2.656E+00$ |
| $f_{51}$ | $1.372E+00$ | $6.608E+00$ | $1.399E+00$ | $7.885E+00$ | $8.000E+00$ |
| $f_{52}$ | $0.000E+00$ | $9.378E-03$ | $8.488E-03$ | $1.000E-02$ | $2.734E-02$ |
| $f_{53}$ | $0.000E+00$ | $1.871E+00$ | $2.892E+00$ | $1.650E+00$ | $1.305E+00$ |
| $f_{54}$ | $0.000E+00$ | $2.643E+00$ | $2.022E+00$ | $2.012E-01$ | $2.012E-01$ |

**References:**

1. doi:10.1088/1757-899X/1099/1/012021

---

**Note:**

- The comparison table includes 10 dimensions ($f_20 - f_{38}$) and an additional column for A.R.
- The values are rounded for clarity.
- The table is formatted to show the comparison between different optimization methods (MAGSO, ABCBB, GSO, DNSFO) for each dimension.

---

**Additional Information:**

- This table is part of a larger study comparing different optimization algorithms for multi-dimensional problems.
- The study focuses on the performance of these algorithms in terms of fitness values and convergence rates.
- The datasets and results are sourced from a recent research paper in the field of materials science and engineering.

---

**Further Reading:**

- For more details on the methodologies and results, refer to the original research paper available via the provided doi link.
- Additional insights and comparisons can be found in the study's supplementary materials or related literature on optimization techniques in materials science and engineering.
| Table 5. Comparison table for 30 dimensions ($f_1 - f_{19}$). |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| MMSO           | ABCBB          | GSO            | DNLPSO         | MMSO           | ABCBB          | GSO            | DNLPSO         |
| $f_1$          | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| $f_2$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_3$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_4$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_5$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_6$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_7$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_8$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_9$          | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{10}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{11}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{12}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{13}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{14}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{15}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{16}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{17}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{18}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
| $f_{19}$       | 0              | 0              | 1              | 0              | 1              | 0              | 0              |
Table 6. Comparison table for 30 dimensions ($f_{20} - f_{38}$) and A.R.

| $f$   | MAGSO  | ABCBB  | A.R.  | GSO    | DNLPSO  | A.R.  |
|-------|--------|--------|-------|--------|---------|-------|
| $f_{20}$ | 1.191E+01 | 9.916E+00 | 1.3157 | 1.391E+01 | 2.114E+00 | 1.3157 | 1.391E+01 | 2.114E+00 |
| $f_{21}$ | 2.081E+00 | 6.052E-01 | 1.191E+01 | 1.196E+01 | 1.501E+00 | 1.327E+00 | 1.199E+01 | 1.501E+00 |
| $f_{22}$ | 7.139E+00 | 2.071E+00 | 1.191E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{23}$ | 5.051E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{24}$ | 2.001E+00 | 3.987E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{25}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{26}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{27}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{28}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{29}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{30}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{31}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{32}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{33}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{34}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{35}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{36}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{37}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
| $f_{38}$ | 2.001E+00 | 2.046E+00 | 1.366E+01 | 1.366E+01 | 1.327E+00 | 1.199E+01 | 1.199E+01 | 1.327E+00 |
The parameters for experimentation are $M = 20$, $N = 5$, $EP_{\text{max}} = 5$, $L_1 = 100$ and $L_2 = 497$ for 10D and $L_1 = 290$ and $L_2 = 1547$ for 30D. The unified acceleration coefficients for all subswarms is $c_1 = c_2 = 2.9889$ and for superswarm is $c_3 = c_4 = 1.94$. The inertial weight of superswarm is the one suggested by Eberhart and Shi [25]:

$$\omega = 0.9 - 0.4999 \left( \frac{k - 1}{L_2 - 1} \right),$$  \hspace{1cm} (6)$$

where, $k$ is the count of iteration.

In the subswarm stage of MAGSO, the exploration level should be higher than the exploitation level. In order to aid high exploration, the projection operators restrict the particle velocity equal to $[-8, 8]^D$ for subswarms. CEC 2013 searches in the hypercube range $[-100, 100]^D$. The global minimum in CEC 2013 is located randomly in the $[-80, 80]^D$ range. Since superswarms should do more of local search, the velocity is restricted within $[-3.2, 3.2]^D$. This search range though appearing to be restrictive is experimentally found to yield better results than the full range. These findings bolster that subswarms do more of exploration and superswarms do more of exploitation as observed by the authors of GSO [12].

So we notice that MAGSO takes advantage of parameter tuning which yields performance advantage over the unmodified GSO. This is one way by which the GSO can be improved.

CEC 2013 identifies that the convergence has attained if the optimized value reaches lower than $10^{-8}$. In this work, all algorithms run until the termination criterion is satisfied. However, when a function has attained convergence, it is marked in the result table as 0. The same notion of convergence is observed for general functions.

If an algorithm secured rank #1 indicates that the corresponding algorithm demonstrated the best performance. The algorithms are assigned appropriate ranks depending on how well they can are separated by Ranksum test [26]. On running the ranksum test on two algorithms if the return value is $h = 1$, then the two algorithms are classified to have different performance otherwise they are treated as similar in performance.

The Figure.1, Figure.2, Figure.3, Figure.4 and Figure.5 indicate the convergence characteristics of competing algorithms for the 10D case. These are plots of single runs. When the historical best solution reaches $10^{-8}$ the convergence plots stop. This is in line with CEC 2013 convergence definition and is adopted for all 38 functions.

As can be seen in the convergence plots, for the functions $f_1$-$f_4$, the MAGSO shows rapid convergence. For the function $f_5$ ABCBB is the fastest. For $f_6$ the results are tied between MAGSO and ABCBB. MAGSO leads in $f_7$. All algorithms show comparable performance for $f_8$. DNLPSO leads in $f_9$. For $f_{10}$ results of MAGSO and ABCBB are the same.

ABCBB performs the best in $f_{11}$, $f_{14}$, $f_{19}$, $f_{21}$, $f_{22}$. For functions $f_{29}$-$f_{38}$ GSO performs the best except for $f_{30}$. When GSO performs the best its convergence is much quicker compared to the rate of convergence of other algorithms.
Figure 1. Convergence characteristics of function $f_1$-$f_8$. The $x$ axis indicates function evaluations and $y$ axis indicates historical best solution with respect to function evaluations.
Figure 2. Convergence characteristics of function $f_9$-$f_{16}$. The x axis indicates function evaluations and y axis indicates historical best solution with respect to function evaluations.
Figure 3. Convergence characteristics of function $f_{17}$-$f_{24}$. The $x$ axis indicates function evaluations and $y$ axis indicates historical best solution with respect to function evaluations.
Figure 4. Convergence characteristics of function $f_{25}$-$f_{32}$. The $x$ axis indicates function evaluations and $y$ axis indicates historical best solution with respect to function evaluations.
Figure 5. Convergence characteristics of functions $f_{33}$-$f_{38}$. The $x$ axis indicates function evaluations and $y$ axis indicates historical best solution with respect to function evaluations.

5. Conclusion

The proposed idea of MAGSO using multiple PSO variants in the subswarm stage in synergistic combination led to significant improvement in the performance over the existing GSO framework. CEC 2013 benchmark functions along with the general benchmark functions have been utilized to demonstrate that the performance of MAGSO. The statistical evidence presented in the paper convincingly shows that MAGSO is better than the unmodified GSO as well as the recent swarm based algorithms such as ABCBB and DNLPSO.

MAGSO is an outgrowth of the basic GSO framework, and similar work can be undertaken to improve the efficiency of the algorithm by embedding some other metaheuristic or combinations of metaheuristics. Future work in the same lines of thought of MAGSO could be investigating other global optimization algorithms such as GA, DA, or other heuristics employed in place of PSO variants. Usage of different heuristics could add more variety in the framework allowing
adaptation to the optimization problem covering a wide span of multimodal functions. The approach proposed in this paper is one of many ways to exploit the basic GSO framework. Many novel constructs can improve the performance of the GSO framework for a wide span of multimodal functions.

Acknowledgements
The authors would like to thanks Dr.Kocer and Dr.Suganthan for providing the MATLAB codes of ABCBB and DNLPSO, respectively. The MATLAB code of MAGSO will be available upon email request from the authors. The authors also would like to thank VIT, Vellore for extending all the facilities and providing conducive research atmosphere.

References
[1] R. Leardi 2001 Genetic Algorithms in Chemometrics and Chemistry: A Review Journal of Chemometrics Vol. 15, No.7, pp. 559-569.
[2] M. Jaberipour, E. Khorram and B. Karimi 2011 Particle swarm algorithm for solving systems of nonlinear equations, Computers and Mathematics with Applications Volume 62, Issue 2, pp. 566-576.
[3] D. Alizadeh and S. Ashraf 2019 New hybrid metaheuristic algorithm for scintillator gamma ray spectrum analysis, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment Volume 915, pp. 1-9.
[4] A. Babazadeh, H. Poorzahedy and S. Nikoosokhan 2011 Application of particle swarm optimization to transportation network design problem, Journal of King Saud University - Science Volume 23, Issue 3, pp. 293-300.
[5] M. Misaghi and M. Yaghoobi January 2019 Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller Journal of Computational Design and Engineering.
[6] F. Xu, W. Wu, F. Zhao, Y. Zhou, Y. Wang, R. Wu, T. Zhang, Y. Wen, Y. Fan and S. Jiang October 2019 A micro-marked module design for university demand-side management using self-crossover genetic algorithms Applied Energy, Volume 252.
[7] G. Notte, M. Pedemonte, H. Cancela and P. Chilibroste 2016 Resource allocation in pastoral dairy production systems: Evaluating exact and genetic algorithms approaches Agricultural Systems, Volume 148, pp. 114-123.
[8] V. N. Coelho, T. A. Oliveira, I. M. Coelho, B. N. Coelho, P. J. Fleming, F. G. Guimaraes, H. Ramalhinho, M. J. F. Souza, E. G. Talbi and T. Lust 2017 Generic Pareto local search metaheuristic for optimization of targeted offers in a bi-objective direct marketing campaign, Computers and Operations Research Volume 78, pp. 578-587.
[9] F. T. Nobibon, R. Leus and F. C. R. Spieksma 2011 Optimization models for targeted offers in direct marketing: Exact and heuristic algorithms European Journal of Operational Research, Volume 210, Issue 3, pp. 670-683.
[10] D. H. Wolpert and W. G. Macready 1997 No free lunch theorems for optimization IEEE Transactions on Evolutionary Computation Vol 1 pp. 67-82.
[11] S. Z. Zhao, J. J. Liang, P. N. Suganthan and M. F. Tasgetiren 2008 Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence).
[12] V. Muthiah-Nakaran and M. M. Noel 2016 Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion Applied Soft Computing, 38 pp. 771-787.
[13] E. Kaya, S. A. Uymaz and B. Kocer 2018 Boosting galactic swarm optimization with ABC International Journal of Machine Learning and Cybernetics, pp. 1-19.
[14] P. Angeline 1998 Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Differences The Seventh Annual Conf. on Evolutionary Programming.
[15] J.J. Liang, A.K. Qin, P.N. Suganthan and S. Baskar 2006 Comprehensive learning particleswarm optimizer for global optimization of multimodal functions IEEE Trans.Evolu. Comput. 10 (June (3)) pp. 281–293.
[16] R. Mendes, J. Kennedy and J. Neves 2004 The fully informed particle swarm: simpler, maybe better IEEE Trans. Evolut. Comput. 8 (3) pp 204–210.
[17] T. Peram, K. Veeramachaneni and C.K. Mohan 2003 *Fitness-Distance-Ratio Based Particle Swarm Optimization* Proceedings of the 2003 IEEE Swarm Intelligence Symposium. pp.174-181.

[18] J. J. Liang, B. Y. Qu, P. N. Suganthan and Alfredo G. Hernández-Díaz 2013 *Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization* Technical Report 2012, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China And Technical Report, Nanyang Technological University, Singapore.

[19] M. El-Abd 2013 *Local best Artificial Bee Colony algorithm with dynamic sub-populations* IEEE Congress on Evolutionary Computation, pp. 522-528.

[20] S. M. Elsayed, R. A. Sarker and D. L. Essam 2013 *A genetic algorithm for solving the CEC’2013 competition problems on real-parameter optimization* IEEE Congress on Evolutionary Computation, pp. 356-360.

[21] R. Tanabe and A. Fukunaga 2013 *Success-history based parameter adaptation for Differential Evolution* IEEE Congress on Evolutionary Computation, pp. 71-78.

[22] J. Li, S. Zheng and Y. Tan 2017 *The Effect of Information Utilization: Introducing a Novel Guiding Spark in the Fireworks Algorithm* IEEE Transactions on Evolutionary Computation, Vol. 21, Issue: 1, pp. 153-166.

[23] X. G. Zhou and G. J. Zhang 2017 *Abstract Convex Underestimation Assisted Multistage Differential Evolution* IEEE Transactions on Cybernetics, Volume 47, Issue 9, pp. 2730-2741.

[24] L. Zhang, K. Sun, H. Hu and Y. Xing 2014 *A System-Level Control Strategy of Photovoltaic Grid-Tied Generation Systems for European Efficiency Enhancement* IEEE Transactions on Power Electronics, Volume 29, Issue 7, pp 3445-3453.

[25] R.C. Eberhart and Y. Shi 1998 *A modified particle swarm optimizer* IEEE International Conference on Evolutionary Computation Proceedings, pp. 69-73.

[26] Matlab help: http://in.mathworks.com/help/stats/ranksum.html