Flavours of Group Search Optimizer: A Survey

Joshil Raj
Ph.D Scholar, Dept of CSE, Pondicherry University
joshlion89@gmail.com

S.SivaSathyaa
Associate Professor, Dept of CSE, Pondicherry University
ssivasathya@gmail.com

Sebabrata Ghosh
M.Tech Student, Dept of CSE, Pondicherry University
sebabrataghsh@gmail.com

Abstract

Group search optimizer (GSO) is a novel bio inspired optimization algorithm imitating the foraging behavior of animals. Producer–Scrounger (PS) model has been used for achieving the best searching strategies in the execution of the algorithm thereby making GSO simple and powerful optimizing technique and hence several researchers in the recent past have come up with different versions of GSO for varied applications. This paper provides a broad overview of GSO and then delves into an exhaustive survey of the different variations & modifications of GSO proposed in various published research works. The paper also covers the different application domains where GSO & its variations/modifications have been successfully used till now.

Keywords: Group search optimization, PSO, Multi-Producer, Cooperative learning, Multi-objective, Artificial Fish Swarm.

1. Introduction

Bio-inspired population-based methods are increasingly becoming popular in solving large number of optimization problems. The most commonly used ones are Genetic Algorithm (GA), Differential Evolution Algorithm (DE), Artificial Neural Network (ANN), Ant Colony Optimization (ACO), Bee Colony Optimization (BCO) and Particle Swarm Optimization (PSO). Group Search Optimizer (GSO) is a novel bio-inspired heuristic algorithm imitating the behaviour of a population of animals foraging for food resources & the same has been widely used for solving several optimization problems [1]. In the animal kingdom, among a population of animals, individual animal member benefits from the process of sharing of meaningful information between the members of the group. GSO uses the Producer-Scrounger (PS) methodology for the search of food by the members of the population [1] [2]. Since the successful implementation of GSO algorithm, there have been several variations & modifications of the GSO algorithm that has been bought forward by researchers and this paper surveys the use of GSO & its variations/modifications in solving several real life problems in different application domains.

The paper is organized in the following sections for easier understanding and assimilation: Section 2 provides an overview of GSO. The different variations/modifications of GSO are given in section 3 and the use of GSO in different application domains is explored in Section 4. Conclusion is drawn in section 5.

2. Group Search Optimizer

All The members of the animal population are organized in the form of groups in the GSO algorithm and each individual of the group is referred to as a member of the group. Three different kinds of members form the group namely:

1. Producer are members that performs the producing strategies in the search for food. In every iteration, the producer member has the best fitness value and is thus located in the most promising area;

2. Scrounger members are involved in task of performing scrouning strategies i.e. joining the food resources discovered by others members;

3. Ranger members are involved in performing random walks for the search of resources or search for randomly distributed food resources in the search space.

The producer will scan randomly in its field of scanning [3] and will move to a location with a better resource or it will stay in its current location and turn the head angle. All scroungers will move towards the producer to find a location having better food resources. If the food resource found by the scrounger is better than the one held by the current producer, then scrounger will become producer for
the next iteration thereby maintaining the requirement of having the best member as the producer. The rangers move to a random distance with a random head angle. Whenever a GSO member moves outside the search space, it is turned back to its previous position inside the bounded search space.

3. Different Types of GSO

Some modifications and variations on the basic algorithm of GSO has been proposed by different researchers to take advantage of the search capability of GSO algorithm and to make it amenable for performance of a particular task. The modifications/variations proposed over the GSO algorithm can be broadly classified as Hybrid GSO algorithms and Modified GSO algorithms. Hybrid GSO algorithms are combinations of GSO algorithm with other algorithms for example the FSGSO and GSPSO. FSGSO is the combination of GSO algorithm and Fish Swarm Algorithm and likewise GSPSO is the combination of GSO and PSO algorithms. In the GSPSO, the PSO algorithm is used for global optimization and the GSO algorithm is employed for optimization in the local search space. Switching between the two algorithms is also done. Modified GSO algorithms are simple modifications over the GSO algorithm. Some of the modified GSO algorithms reported in the research publications are MPGSO, NMGSO, CGSO and QGSO. Traditional GSO has a single producer however MPGSO uses multi producers. In NMGSO, the scanning strategy is performed by the limited pattern search thereby increasing the computational efficiency and improving the performance. CGSO uses cooperative behaviour between different GSO groups to improve the performance. QGSO increase the number of rangers in the algorithm and uses the search strategy of PSO algorithm. The hierarchy of different types of GSO presented in Figure 1. This section deals with brief description of the different types GSO.

![Figure 1: Different Types of GSO algorithms](Image)

3. FSGSO

Oliveira, Pacifico and Ludermir [4] proposed Fish Swarm Group Search Optimizer (FSGSO) which is a hybrid Group Search Optimization based on the behaviors of fish swarms. The hybridization was done by effecting a change in the method of scrouninging procedures. The group members nominated as scroungers moved in the search space following the behaviors of fish swarms like Swarm, Follow, Prey and Leap from the Artificial Fish Swarm Algorithm (AFSA) [5]. In FSGSO, the producer maintains its search procedures similar to the original GSO, however the search strategy of scroungers follows the search procedures of fish swarm. In the traditional GSO scrouning strategy, the scroungers move towards the producer randomly in each iteration, however in FSGSO the scroungers moves around in the search space taking into consideration the location of the producer, the center of the group and also performing random walks. In this method, the scrouning is performed through the execution of two behaviors from the AFSA: Swarm and Follow. The Follow and Swarm behaviors are executed and the one that achieves better fitness is used to update the position of the scrounger. The follow behavior represents the ability to follow the producer given that the current state conditions are accepted. In the Follow behavior the selected scrounger will follow the producer if the region is not crowded ($\frac{n_f}{N} < \sigma$) and the best position has better fitness than the present location ($Y_p > Y_i$), then the position is updated using eq.(1) & if the condition is not satisfied then eq (2) is used for updating the positions of the scrounger.

$$X_{i}^{k+1} = X_{i}^{k} + \text{rand} \cdot \text{step} \cdot \frac{X_p - X_{i}^{k}}{||X_p - X_{i}^{k}||} \tag{1}$$

$$X_{i}^{k+1} = \text{Prey} (X_{i}^{k}) \tag{2}$$

The swarm behavior will gather information from all the members of the group in order to determine a central position, which will influence the movement of the scrounger. In the Swarm behavior the update process is done if the region is not crowded ($\frac{n_f}{N} < \sigma$) and the central position is in a better food concentration region than the current position ($Y_p > Y_i$) then the position is updated using eq.(3) and if the condition is not satisfied then the scrounger uses the prey procedure as per eq (4) for updating its position.

$$X_{i}^{k+1} = X_{i}^{k} + \text{rand} \cdot \text{step} \cdot \frac{X_c - X_{i}^{k}}{||X_c - X_{i}^{k}||} \tag{3}$$

$$X_{i}^{k+1} = \text{Prey} (X_{i}^{k}) \tag{4}$$
In the Prey behaviour, the scrounger tries to find a better position inside its visual field. If after a number
of tries (trynumber), the scrounger succeeds, the update will be performed as shown in eq. (5). If the
scrounger does not find a neighbor with better fitness after a number of tries, it will move randomly as
shown in eq. (6).

\[ X_{i}^{k+1} = X_{i}^{k} + \text{rand} \cdot (r_{1} \cdot \text{step}) \frac{x_{j} - x_{i}^{k}}{\|x_{j} - x_{i}^{k}\|} \]  
\[ X_{i}^{k+1} = \text{Leap}(X_{i}^{k}) \]  

The Leap behaviour is characterized by random movements from the scrounger (eq.7)

\[ X_{i}^{k+1} = X_{i}^{k} + \text{rand} \cdot (r_{1} \cdot \text{step}) \]  

All the other movements of producers and rangers are the same as used in the traditional GSO.

The FSGSO is evaluated over eight benchmark functions and compared to DE, PSO and standard
GSO algorithms. The FSGSO achieved the best overall performance in comparison with the GSO,
PSO and DE.

3. 2FSGSO

Yan and Shi [6] are introduced a novel hybrid algorithm called Group Search Particle Swarm
Optimization (GSPSO) using both the Particle Swarm Optimization (PSO) and the Group Search
Optimization (GSO) algorithms. The PSO algorithm finds a good local search space for the problem and
in the GSO algorithm uses the convergence by scroungers and rangers to update the local search
space. PSO algorithm is again used to find a smaller and better local search space iteratively and the use of
GSO or PSO is used by switching. The best position is found out by the process of mutual correction by
both PSO and GSO. The GSPSO algorithm is present in Figure 2.

The GSPSO is run over four benchmark functions for 30 and 300 dimension dataset. It has been compared
with GSO, PSO, QGSOPC and CBPSO algorithms.

3. 3MPGSO

A. B. M. Junaed, M. A. H. Akhand, Al-
Mahmud and K. Murase [7] proposed a multi-
producer GSO (MPGSO) based on multiple
producers. In MPGSO, at first a top x% members
are selected as producers and a roulette wheel is made
using these producers. For each of the scrounger, a
producer is selected from the roulette wheel, and the
scrouning strategy is performed towards this
producer. MPGSO algorithm produced optimal
solutions within a less number of iteration than the
original GSO algorithm while solving benchmark
functions. The MPGSO algorithm is presented in
Figure 3.

1. Randomly initialize the members of the
population & generate their fitness values.
2. while terminal conditions are not met do
3. for each members i in the group do
   a. Find the top x% members as producers of
      the group.
   b. Create roulette wheel for selecting a
      producer from the group of scroungers.
   c. The rest of the members (rangers) will
      perform ranging.
   d. Calculate the fitness value for each
      member again.
4. end for
5. end while

Figure 3: Pseudo code for the MPGSO algorithm

The MPGSO is tested by eight benchmark functions for 2 and 30 dimension dataset and compared to
traditional GSO algorithms. MPGSO is achieved best results for each of the problems.

3. 4 NMGSO

Wang, Zhong and Liu [8] proposed a novel
modification to GSO for multi-objective optimization
problems as NMGSO. Unlike the original GSO,
where each member has to keep the record of its head
angle and do a Polar to Cartesian coordinate
transformation while updating its position, in
NMGSO, the scanning strategy for each member is
replaced with a limited pattern search (LPS) [8][9]
which is more efficient the high-dimensional
optimization problems. The limited pattern search
procedure simplifies the search behavior and also
enhances the capability of local search. In traditional
GSO, the ranger search for the new position based on
a random head angle and distance but in NMGSO,
the global search behavior of the rangers is modified.
with a controlling probability thereby introducing randomness and enhancing the diversity of members.

In multi-objective optimization problems, it is not easy to choose the best member to be the producer since there is no dominance relationship between the various objective functions and may require multiple producers to guide the groups of members. Hence more number of producers is set as per the number of the objectives in the multi objective function. For every objective in multi objective function problem, the one group member with the single-objective extreme is selected as the producer and these producers optimize the respective objective with the limited pattern search. The NMGSO is tested by seven benchmark functions with two objective functions and compared to MOGSO and MOPSO algorithms. NMGSO has achieved best results for each of these problems and the convergence property of the NMGSO along with its ability to find a diverse set of solutions is the best among the other algorithms. NMGSO is more suitable as compared to GSO for solving the multi-objective optimization problems.

3. 5CGSO

L. D. S. Pacifico and T. B. Ludermir[10] introduced a novel GSO approach based on cooperative behaviour among groups, called Cooperative Group Search Optimizer (CGSO). The modified algorithm works with a set of groups of members, where each group manipulates a limited set of the problem variables and the solutions of each group is combined to get the optimized solution for composite problem. The population members are initially divided into k independent groups and each group is then associated with d dimensions from the search space (where d x k = n). Each group will execute local searches seeking to minimize its own set of variables. The current fitness for each member of the groups is evaluated as a concatenation of its own variables with the best contributions by all the groups, at the corresponding positions. The CGSO algorithm is present in Figure 4.

The CGSO was evaluated in nine well known benchmark functions for 30-dimension search space problems and compared to standard GSO, PSO and DE algorithms. Experimental results display that the CGSO is more efficient than the standard GSO algorithm.

3. 6QGSO

Li and Liu [11] proposed a quick group search optimizer (QGSO) for structural optimization. The QGSO has three main features:

(a) Increase in quantity of rangers at certain points of the algorithm, when the optimization stops going forward.

(b) The search strategy is similar to that of PSO algorithm wherein the group best and the personal best are considered. Every scrounger moves forward by a random walk and does not just follow the producer in every iteration. They also consider the best position they have ever been during the move.

(c) The rangers are produced according to the group best and the personal best.

The QGSO algorithm is present in Figure 5.

The QGSO was evaluated in five benchmark pin-connected structures for Truss Structures with Discrete Variables and Truss Structures with Continuous Variables and compared to standard GSO, HPSO algorithms. QGSO algorithm has less computational complexity, better convergence rate and accuracy as compared to other algorithms that it was compared with such as standard GSO and HPSO algorithms.

---

1. for each group $G_i$ do
   a. Randomly generate the members & their fitness values.
2. end for
3. while the terminal conditions are not met do
4. for each group $G_i$ do
   a.Nominate the producer.
   b. Nominate scroungers & perform scrounging.
   c. Nominate rangers & perform ranging.
   d. Regenerate fitness function for each member.
5. endfor
6. end while

Figure 4: Pseudo code for the CGSO algorithm

---

IJCTA | Mar-Apr 2015
Available online@www.ijcta.com

ISSN:2229-6093
1. Randomly initialize velocities and positions of all member.
2. for each particle i in the group do
3. while the conditions are met do
   a. Randomly regenerate the current particle $X_i$
4. end while
5. end for
6. while the terminal conditions are not met do
7. for each member i in the group do
   a. Calculate the fitness value.
   b. Nominate the producer, scroungers & rangers.
   c. Use the pbest update the value of the member.
   d. Calculate the fitness value $f(X_{ik})$ of the particle.
   e. Update pbest.
   f. Update gbest.
8. endfor
9. end while

Figure 5: Pseudo code for the QGSO algorithm

4. Application of GSO

GSO has effectively been used for solving various real life problems & has proved to be an efficient algorithm for solution of some practical applications domain problems. In He et al. [12], the weights of ANN was training using the GSO algorithm and hybrid algorithm was used for the classification of breast cancer dataset. Wu et al. [13] used a GSO with multiple producers (GSOMP) for the optimal placement of multi-type Flexible AC Transmission System (FACTS) devices in a power system, thereby successfully solving a multi-objective optimization problem. He and Li [14] used a hybrid combination of ANN algorithm trained with GSO algorithm for the monitoring of machine condition. Silva et al. [15] used GSO to optimize input weights and hidden biases of Extreme Learning Machine (ELM) neural network. The other major applications of the GSO algorithm are listed in Table I.

5. Conclusion

Several computational intelligence solutions have been proposed to solve and optimize the difficult continuous optimization problems. But algorithms inspired from the natural behavior yields special attention for its performance. This paper provides an overview of basic GSO and different types of modified GSO (FSGSO, CGSO, MPGSO, GSPSO, NMGSO and QGSO).

This paper also lists some of the applications of bio-inspired population-based algorithm GSO in various fields for accomplishing various tasks. Some modifications and variations on the basic algorithm of GSO have been proposed by different researchers to improve the performance of GSO algorithm and solving several real life problems in different application domains. GSO technique and its numerous modifications provide a number of ways for solving the real world problems more efficiently and quickly.

6. References

<table>
<thead>
<tr>
<th>Algorithms Used</th>
<th>Application</th>
<th>Data used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group Search Optimizer and ANN [14]</td>
<td>GSO was effective in optimizing ANN used for Machine Condition Monitoring.</td>
<td>Ultrasound Data Records.</td>
</tr>
<tr>
<td>GSO and Discrete operators [16]</td>
<td>GSO was used for solving Hybrid Flow shop Scheduling Problem.</td>
<td>Ten benchmark problems</td>
</tr>
<tr>
<td>GSO [17]</td>
<td>GSO was used for Clustering of Complex Networks and Community Detection</td>
<td>Five social network datasets</td>
</tr>
<tr>
<td>Swap Operator (SO), Swap Sequence (SS), GSO [18]</td>
<td>GSO is used to solve the Traveling Salesman Problem.</td>
<td>15 benchmark problems</td>
</tr>
<tr>
<td>PSO and GSO [19]</td>
<td>GSO is used for optimal structural design like 72-bar spatial structure etc.</td>
<td>Structural design dataset.</td>
</tr>
<tr>
<td>Group Search Optimizer Algorithm [20]</td>
<td>GSO is used for complex structural optimization problems. GSO was used for solving structural design problems in civil construction designing.</td>
<td>Five typical benchmark structural optimization problems</td>
</tr>
<tr>
<td>GSO and harmony search mechanism [21]</td>
<td>The application of improved GSO for structural designing of trusses.</td>
<td>Structural truss design dataset</td>
</tr>
<tr>
<td>GSO [22]</td>
<td>GSO is used for the Multicast Routing Problem</td>
<td>Network topology consisting of 60 nodes</td>
</tr>
<tr>
<td>Evolutionary strategy and GSO [23]</td>
<td>GSO is used for mechanical design optimization problems.</td>
<td>Spring design and mechanical design data.</td>
</tr>
<tr>
<td>GSO [24]</td>
<td>GSO is used for supervised Classification &amp; feature selection.</td>
<td>Colon Cancer dataset, Leukemia dataset, Breast cancer dataset</td>
</tr>
<tr>
<td>GSO and Invasive Weed Optimization (IWO) [26]</td>
<td>GSO is used for Multimodal optimization problem</td>
<td>21 multimodal general functions, 9 composition functions</td>
</tr>
<tr>
<td>PSO &amp; GSO [27]</td>
<td>GSO is used for optimizing location and capacity of distributed electricity generation &amp; distribution problem.</td>
<td>Electricity distribution &amp; generation dataset.</td>
</tr>
<tr>
<td>GSO [28]</td>
<td>GSO is used to train the neural networks.</td>
<td>23 benchmark functions.</td>
</tr>
<tr>
<td>GSO [29]</td>
<td>GSO is used for solve the Optimal Power Flow Problem.</td>
<td>Electricity distribution network data.</td>
</tr>
<tr>
<td>GSO and limited pattern search procedure [30]</td>
<td>Group search optimizer is used for multi-objective optimization</td>
<td>7 benchmark functions</td>
</tr>
<tr>
<td>Group Search Optimizer and ANN [31]</td>
<td>GSO is used for diagnosis of breast cancer.</td>
<td>breast cancer dataset</td>
</tr>
</tbody>
</table>


[17] G. Kishore Kumar, Dr. V. K. Jayaraman, “Clustering of Complex Networks and Community Detection Using Group Search Optimization”


[27] Qi Kang, Tian Lan, Yong Yan, Lei Wang, Qidi Wu, “Group search optimizer based optimal location and capacity of distributed generations” Neurocomputing 78 pp. 55–63, 2012


