Dynamic Group Search Algorithm

Rui Tang, Simon Fong
Department of Computer and Information Science
University of Macau
Taipa, Macau SAR
e-mail: { yb57463, ccfong}@umac.mo

Suash Deb
Founding President of INNS-India
IT & Educational Consultant
Ranchi 834010,
Jharkhand, India
e-mail: suashdeb@gmail.com

Raymond Wong
School of Computer Science and Engineering
University of New South Wales
Sydney, NSW 2052, Australia
e-mail: wong@cse.unsw.edu.au

***Abstract*—Recently many researchers invented a wide variety of meta-heuristic optimization algorithms and they have achieved remarkable performance results. Through observing natural phenomena, clues were inspired and programmed into search logics, such as PSO, Cuckoo Search and so on. Although those algorithms have promising performance, there still exist a drawback – it is hard to find a perfect balance between the global exploration and local exploitation from the traditional swarm optimization algorithms. Like an either-or problem, algorithms that have better global exploration capability come with worse local exploitation capability, and vice versa. In order to address this problem, in this paper, we propose a novel Dynamic Group Search Algorithm (DGSA) with enhanced intra-group and inter-group communication mechanisms. In particular, we devise a formless "group" concept, where the vectors of solutions can move to different groups dynamically based on the group best solution fitness, the better group has the more vectors. Vectors inside a group mainly focus on the local exploitation for enhancing its local search. In contrast, inter group communication assures strong capability of global exploration. In order to avoid being stuck at local optima, we introduce two types of crossover operators and an inter-group mutation. Experiments using benchmarking test functions for comparing with other well-known optimization algorithms are reported. DGSA outperform others in most cases.**

Keywords-component; group search algorithm; optimization;

I. INTRODUCTION

In this paper, we propose a new optimization algorithm: dynamic group search algorithm (DGSA). Our proposed method is to simulate social behaviors of creatures in nature. Without strictly imposing the method on any specific animal, the dynamic grouping concept is generic. It fits the behaviour of a wide variety of living creatures or social structures that facilitate group collaboration [1]. Examples are birds, fish, wolves, human, and even working teams or units in a company. In a population or a company, it may has several small groups or divisions [2]. Each group is considered as a part of the whole system, which cooperates with the other groups to achieve a common goal. Members within a group do collaborate closely; members across other groups are loosely coupled, and their interactions often are in the form of information exchanges. A breed of birds, a family of people or a department of a company is considered as a small team in many scenarios. For example, although a bird's family has many birds, they team up as a unit when they prey, family members help each other and they concern about other families as a clan. It is worth pointing out that group sizes do vary dynamically, members may transfer from one group to another, or they just leave for good. New members could be recruited into a group from another group. New groups may emerge and existing groups may vanish when their members have ceased.

As the name suggests, DGSA takes the form of a number of groups and groups can communicate with each other. Each group contains a centroid and certain amount of members. The best solution of each group is stored in its corresponding centroid. In the same group, members can share all sorts of information among themselves, but they only exchange important information from the other groups. The number of members is changed dynamically in a group based on searching capability, the better solution, the more members, vice versa. Moreover, in our algorithm, the crossover is set, which helps the algorithm to escape from being trapped at local optimum.

We summarize the main activities of DGSA as follows:

- intra-group cooperation
- inter-group collaboration
- crossover
- group variation

There are four components in our proposed algorithm. Each component controls different behaviors. In the

978-1-5090-3488-8/16/$31.00 ©2016 IEEE

following sub-sections, the details of the components are described.

A. Data Structure

In our proposed DGSA, the data structure is distinct from other swarm optimization algorithms. First, we introduce the concept of centroid. Each group has a centroid which records the best solution of its group ever found. In other words, centroid replicates the best member's solution after one generation is processed. The centroid will not execute the intra-group cooperation, but it will communicate the other groups as a group delegate. So the centroid has two ways to update. One is that the centroid replicates the group best, the other is that the centroid gains information from other better groups. In the case of a group that has none member, the centroid still will run inter-group collaboration, crossover and group variation in despite of being empty. There are two particular scenarios in DGSA. The first one is that the whole population only has one group. In this case, the centroid stores the global best, and the inter-group collaboration doesn't work until group variation produces a new group. The second one is that the number of groups is equal to the number of population; in this case, each solution is a group. In the second case, the intra-group cooperation among the components is no longer possible. And the centroid can be treated just as a normal solution. For the abovementioned two cases, DGSA transforms them to a swarm of standard swarm optimization agents. Like a swarm, each agent searches for the global best without any cooperation between groups. However, the two cases can have different search strategies as a result of utilizing different components.

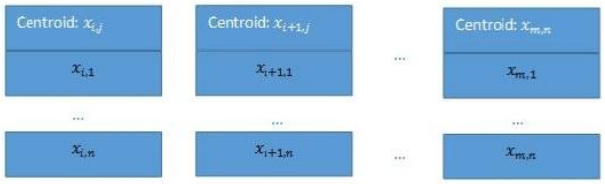

Figure 1. Data structure of DGSA

Fig. 1 illustrates the basic data structure of the so-called groups in DGSA, where $x_{i,j}$ means the j^{th} member of the i^{th} group. In the first column, the centroid stores the j^{th} member's solution, because the j^{th} member possess the best group result. Besides, by the configuration as in Fig.1, the centroid may potentially have different solutions from the group members. Via the inter-group communication, this centroid will communicate with other groups about the best solution which it has. The centroids of all the groups would update each other about the best solution.

B. Intra-Group Cooperation

The intra-group cooperation simulates the solution of group members using updating and mutation procedures. In this component, members of each group are updated by obtaining the group best solution and global best solution. To certain extend, an individual group is similar to a swarm in PSO. Moreover, the mutation is used here. The mutation plays an important role in our algorithm, it guarantees population diversity. The mutation is controlled by mutation probability Mr which is a random number generated between 0 and 1, the updating equation can be described as follow:

$$x_{i,j,k+1} = \begin{cases} x_{i,j,k} + rand(C_i - x_{i,j,k}) + w(G_{best} - x_{i,j,k}) & rand < Mr \\ (C_i - x_{i,j,k}) + \mu & else \end{cases} \quad (1)$$

where the $x_{i,j,k}$ is the j^{th} member of the i^{th} group in the k^{th} generation. w is a weight to control the proportion of movement direction. The $rand$ is a random number between 0 and 1. The C_i is the centroid of i^{th} group, G_{best} means the global best. μ is a random number which is drawn from $\mu \sim (0, s^2)$. s^2 is the step size factor to control the movement distance.

From (1), it is clearly shown that in the intra-group level, members are updated mainly through obtaining the locations of the peers within the group. The global best solution only carries the movement direction. In the case of a single group, the direction points to the group best. Members move towards group best. In the mutation, members only get information from their own group. It produces a new solution which is near the group best for alternative solutions which are probably better than the group best. This component does not only enhance the capability of local searching ability, but strengthen the speed of convergence. In addition to the ordinary information update, the mutation provides capability of avoiding sinking into local optimum.

C. Inter-Group Collaboration

Centroids are responsible delegates for communicating with the other groups in this phase. Only the group centroids get updated for saving communication costs. The movement is following the Lévy flight random walk. In 30s', French mathematician Lévy indicated that the time and path of Lévy flight satisfy the Lévy distribution. In the past few years, many researchers used the Lévy flight to explain the random prey and walk phenomenon as animal behaviors in the nature. This particular walking pattern, for independent and scattered search agents, is believed to be the best for doing searching in the nature. In recent studies scientists prove that the fruit flies and honey bees do move/search in patterns satisfying the statistical properties of the Lévy flight. The Lévy flight is one type of random walk. The step size obeys the heavy-tailed distribution, long distance search and short distance search occur intermittently. In Shlesinger M study [3], the Lévy flight is suitable for swarm optimization algorithms. It provides the capabilities of population diversity and expanding the scope of search.

The mathematical update equation of Lévy flight walk is formulated by Yang [4] as follow:

$$C_i^{k+1} = C_i^k + \alpha \oplus \text{Lévy}(\lambda) \quad (2)$$

where $\alpha > 0$ is the step size related to the scales of the problem, which is fixed in Cuckoo algorithm [5] and others. The $\oplus$ is the entry-wise multiplications. C_i^{k+1} and C_i^k means the i_{th} group in the k+1 and k generations. Lévy(λ) is a random number, which is drawn from Lévy distribution. The exponential form of probability function is:

$$\text{Lévy} \sim \mu = t^{-\lambda}, \ (1 < \lambda \leq 3) \tag{3}$$

Mantegna R [6] proposed the Lévy search equation in 1992 as follow:

$$s = \frac{\emptyset \times \mu}{|v|^{1/\beta}} \tag{4}$$

where $\lambda = 1 + \beta$, $\beta \in (0,2]$. In the cuckoo search algorithm, the $\beta = 1.5$, which is a constant number. The μ and v are normal distribution random number.

$$\emptyset = \left[\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\beta-1)/2}}\right]^{1/\beta} \tag{5}$$

From the (3)-(5), s is decided by two normal distribution numbers μ and v. Their values could be positive or negative, could be large or small. Therefore, the step size of Lévy is highly stochastic. The optimization algorithm which uses this random walk can anticipate a promising solution. In the situation of a lone group in our algorithm, centroids are updated by just using Lévy flight.

D. Crossover

Crossover is first used in GA as an effective strategy for global optimization. It solves the problem of trapping into local optimum easily. So a numbers of optimization algorithms use this operator, such as, DE, GA, GS, etc. In our proposed method we also set 2 crossover operators which are similar to DE and GS. The crossover is controlled by a crossover probability, Cr, and the crossover can be performed in two types: randomly chosen operator and biased random walk.

$$x_{i,j,k+1} = \begin{cases} x_{i,j,k=} r \times x_{m,n,k} & rand < Cr \\ x_{i,j,k} & else \end{cases} \tag{6}$$

where the $x_{m,n,k}$ is the member which is drawn randomly from the whole populations. r is the random number with the range [0,1]. $rand$ is the random number generator. The (6) is the random chosen operator. The second one is biased random walk. It is built into CS. The equation can be formulated as:

$$x_{i,j,k+1} = \begin{cases} x_{i,j,k=} x_{i,j,k} + r \times (x_{m,n,k} - x_{i,j,k}) & rand < Cr \\ x_{i,j,k} & else \end{cases} \tag{7}$$

In the biased random walk, the fitness is related to the difference in solutions. Likewise the random walk works in a biased way.

E. Group Variation

In this component, the group members, centroids change by ranking the fitness. The fittest one becomes the centroid. The better fitness there is in the group, the more members it attracts. In many cases, algorithms cannot find the best solution directly. When the solutions are stuck at a local optimum, they all just do useless work until the mutation is triggered and get them jumped out of the local optimum. Hence, massive computing power is wasted in the entrapment situations. In order to avoid wasting computing resources, we set up a group variation component to control the sizes of the groups. The flow chart shows how it works as follow.

Fig. 2 shows the procedure of group variation. First, the algorithm checks whether the members of a group have improved their solutions or not. If it gets improvement, it will do ordinary search. Otherwise, it further checks whether it has reached a given rounds of trail. If yes, it indicates a high possibility that the members have stuck, unable to jump out the local optimum. When this is detected, the group members would be relinquished from the current group and be transferred to other randomly selected group.

Our proposed algorithm is supposed to be equipped with sufficient mechanisms to avoid sinking into local optimum. Moreover, DGSA enhances the capability of local exploitation and accelerates the speed of convergence. The pseudo code is listed in Table 1.

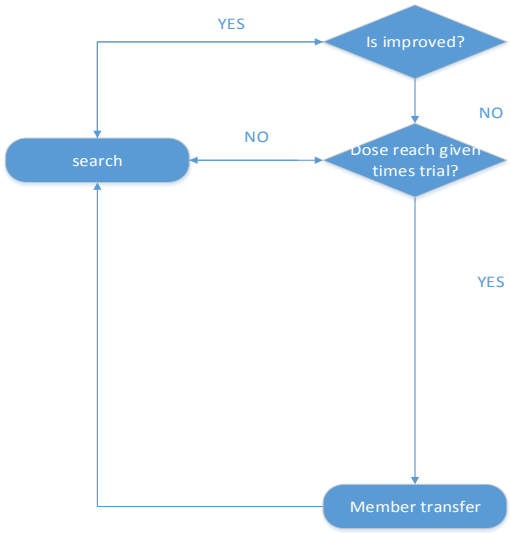

Figure 2. Procedure of group variation

TABLE I. PSEUDO CODE

```
Objective function f(x), x=(x1,x2,..xd)^T
Initialize the populations, xi where i=1,2,..,W
Define and initialize parameters
WHILE (t<generations && stopping criteria not
met)
      Inter-group collaboration
      Intra-group cooperation
      Crossover
      Evaluating groups best and global best
      Group variation
      Inter-group collaboration
  END
END WHILE
```

II. EXPERIMENTS

Different experiments have been carried out to assess the performance of DGSA. Two different crossover operators are tested into DGSA in the experiments. The first GS uses the bias random walk crossover (GS), the second one is the random operator (GSR). A total of 14 popular testing functions are used for performance assessment. The focus of the experiments is to compare the DGSA algorithms with the original algorithms. Two types of evaluation criteria are used, one is quantitative statistical performance results, and the other is in fitness curves showing how the fitness converges in iterations. We compared the performance of DGSA algorithms with original algorithms for the test suite using the function error value. The function error value for a solution is defined as $f(x) - f(x^*)$, where $f(x^*)$, is the global optimum of the function. The maximum number of fitness evaluations that we allowed for each algorithm to minimize this error was $1000 \times D$, where D is the dimension of the problem. The fitness evaluation criteria were as follows.

- Error: The minimum function error value that each algorithm can find is also recorded in 50 runs, and the average and the standard deviation of the error value are calculated. The average of errors, the standard deviations and the medians of errors are recorded in the simulation runs.
- Convergence graph: the convergence graph shows the average error performance.

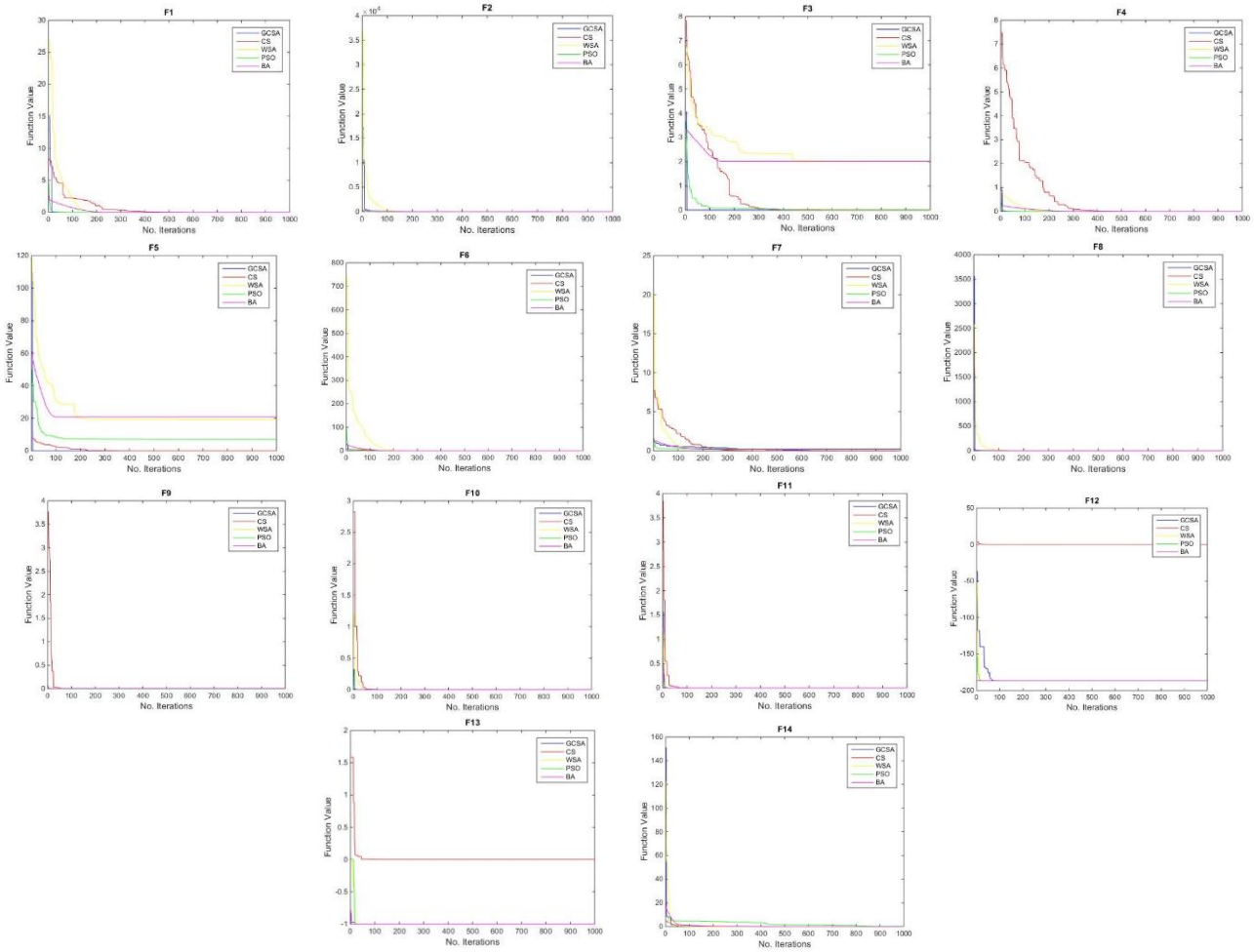

Figure 3. Convegence Graphs

A. Experiment Setting

To verify the efficacy of our new algorithm, we used four similar metaheuristic algorithms such as PSO, Cuckoo Search (CS) [5], Bat Algorithm (BA) [7], Wolf Search Algorithm (WSA) [8] for comparison. In the experiments, the same random approach is used to initialize the populations for all the algorithms. In order to obtain unbiased results, the same computer with the configuration listed in Table II is used for the simulation runs.

TABLE II. COMPUTER SETTING

Processor: Intel(R) Core i7-4790 @ 3.60Hz 3.60GHz RAM: 8.00GB System type: 64-bit Operating System Operating system: MS Window 7 Service Pack 1 Matlab: R2014b

The parameter settings for the algorithms are listed in Table III.

TABLE III. PARAMETER SETTING

Algorithm	Parameter setting
CS	*Popsize*=30, p_a=0.25
BA	*Popsize*=30, *loudness*=0.5, *pulse rate*=0.5
WSA	*Popsize*=30, p_a=0.25
PSO	w_1=1.5, w_2=1.5
DGSA	*Group*=5, p_a=0.25, *Popsize*=30

For fairness, the same values for the common parameters are used for unbiased results. The probability of mutation p_a=0.25 is default as in the setting of published literature [9] [10], the population size is chosen at 30, which are the same as studied in [9][10]. 14 different benchmark functions are used to evaluate our proposed DGSA method, which are suggested in [9] for rigorous evaluation. The benchmark functions contain many significant physical properties and shapes in the mathematical models. For example, they include bowl-shaped search space (bohachevsky function), search plane with many local minima (Ackley function, Griewank function and so on), search plane with steep ridges (Easom function) etc. The detailed reference can be found in [9].

B. The Effect of DGSA

The best error values using three different algorithms and their DGSA methods are tabulated in Table 6. From Table 4, it is found that our proposed method performed better than the existing algorithms. In the unimodal functions, our proposed algorithm at least has two orders of magnitude of superiority than the other algorithms. For instance, in the function Sphere, our GS reached 1^{e-08} while the WSA and CS only get 1e-4 and 1^{e-3}. For the multimodal functions, our proposed method also shows higher accuracy than that of the rest.

Table 5 records the standard deviations of function error values. DGSA generally has smaller standard deviations in most benchmark functions. It shows better results, and it has convergence of very small value compared to others.

Table 6 presented the medians of function error values which are relatively less affected by outliers than means. DGSA prevails by achieving the lowest median errors in most cases, similar to Table 5. Some exceptions are the cases of Rosenbrock, Lévy, and Colville functions

Fig. 3 shows a collection of convergence graphs. From the patterns of the convergence graphs we can observe operational information about convergence speed and the overall goodness of the search strategy. It is evident that our proposed method performs better than the other popular algorithms. It can be seen that the convergence speed of DGSA is the fastest among all.

TABLE IV. THE AVERAGE FUNCTION ERROR VALUES

Function	PSO	WSA	CS	BA	GS	GSR
F1	0.0260	0.0022	0.6528	1.7144e-05	1.5992e-05	4.3650e-07
F2	30.4996	29.1583	0.8547	25.8665	28.9316	27.8223
F3	0.2850	6.3805	0.7451	2.0641	0.0029	1.3378e-04
F4	0.0026	0.0013	0.8821	1.0148e-06	1.0012e-07	2.1791e-08
F5	24.2858	191.4855	0.8754	39.8011	0.0034	1.2722e-05
F6	4.6983	4.4108	0.7336	0.0027	0.0011	4.7923e-06
F7	0.7115	14.2613	0.9279	1.2355	1.8361	2.2626
F8	0.9857	9.2229	0.8293	0.0131	2.5289e-04	1.9355e-05
F9	4.1498e-08	1.1013e-07	6.8164e-12	0.0077	2.8756e-13	2.3909e-10
F10	1.2571e-07	9.1425e-08	2.1084e-12	0.0453	5.9133e-14	4.4264e-09
F11	1.1165e-06	6.7918e-08	8.1693e-12	0.1652	1.1102e-16	4.9802e-09
F12	-186.7308	-186.7309	3.9984e-12	-174.1001	-186.4777	-149.3822
F13	-1.0000	-1.0000	9.1245e-08	-1.0000	-1.0000	-0.9546
F14	0.0480	0.0150	1.6565e-07	1.6548	0.4562	12.7843

TABLE V. THE STANDARD DEVIATION FUNCTION ERROR VALUES

Function	PSO	WSA	CS	BA	GS	GSR
F1	0.0107	6.1505e-04	0.2135	2.5508e-06	3.1175e-06	7.8667e-07
F2	0.9365	0.4214	0.2692	2.0050	0.0409	2.4608
F3	0.2279	0.2001	0.3978	0.2523	3.2552e-04	5.2995e-05
F4	5.6407e-04	6.4257e-04	0.3563	1.2860e-07	3.7121e-07	4.7446e-08
F5	4.8721	18.4036	0.4347	4.0412	5.6517e-04	2.6528e-05
F6	1.9715	0.9825	0.5140	6.1393e-04	2.2835e-04	7.5384e-06
F7	0.1437	2.1514	0.4993	0.1602	0.0780	0.9158
F8	0.5130	5.2132	0.6884	0.0051	1.1225e-04	2.3659e-05
F9	4.3652e-08	1.3407e-07	7.3447e-12	0.0043	5.2126e-13	4.3203e-10
F10	7.1808e-08	7.2721e-08	1.2408e-12	0.1012	6.7890e-14	4.1145e-09
F11	1.2923e-06	4.6578e-08	9.9720e-12	0.2262	1.9230e-16	1.0888e-08
F12	9.4077e-05	2.6656e-06	4.0177e-12	28.2434	0.4745	22.1165
F13	1.2335e-07	1.8166e-07	5.7199e-08	2.1995e-10	3.1644e-12	0.0394
F14	0.0586	0.0159	1.4907e-07	3.4812	0.4706	6.7514

TABLE VI. THE MEDIAN FUNCTION ERROR VALUES

Function	PSO	WSA	CS	BA	GS	GSR
F1	0.0254	0.0019	0.7277	1.7501e-05	1.5707e-05	2.6152e-08
F2	30.2984	29.0116	0.8604	25.1915	28.9465	28.9420
F3	0.2115	6.2891	0.6040	2.0133	0.0027	1.4066e-04
F4	0.0026	0.0015	0.9913	9.7744e-07	1.1788e-07	8.2906e-10
F5	24.5891	192.7267	0.7753	39.8014	0.0036	6.1430e-07
F6	4.4025	4.9123	0.5588	0.0025	0.0011	2.2736e-07
F7	0.7298	14.5401	0.8424	1.2534	1.8272	2.6361
F8	0.7360	7.1056	0.5091	0.0114	2.1953e-04	6.7183e-06
F9	3.7443e-08	5.8522e-08	2.4629e-12	0.0096	3.3296e-13	1.1590e-12
F10	1.5147e-07	6.7712e-08	1.7630e-12	2.1047e-09	4.4264e-09	4.0121e-09
F11	6.2250e-07	7.3191e-08	1.8483e-12	8.7952e-09	0	1.8429e-10
F12	-186.7309	-186.7309	2.3990e-12	-186.7309	-186.7309	-161.1678
F13	-1.0000	-1.0000	5.2272e-08	-1.0000	-1.0000	-0.9337
F14	0.0062	0.0125	6.6199e-08	1.4352e-04	0.4069	11.4808

III. CONCLUSION

In this paper, a novel metaheuristic algorithm is proposed which takes literately no form, but just search members that operate in groups. It is similar to the classical PSO which contains some local best and global best information within a single swarm which scouts the search space and converge at a global best position (as global optimum). Unlike PSO, our new algorithm called Dynamic Group Search algorithm (DGSA) takes a very flexible formation of search "groups" which dynamically evolves themselves for changing the memberships of the search agents on the fly. Without adhering rigidly to any particular so-called nature-inspired phenomenon [11], DGSA is shifting its form from one to many groups which are driven by the search results during operation. Group sizes and the directions of their search movements do change iteratively. DGSA takes on the proven effective mechanisms to facilitate its search, such as Levy flight, mutation, inter- and intra-group communications etc. In particular the design of DGSA adaptively covers both local intensification and global exploration in the search operation, without the need of balancing these two contradicting capabilities. As the groups merge and scatter far apart, these two functions are dynamically adjusted when the solution vectors (search members) form the groups by themselves. Small number of groups with many members imply local intensification are at work. Many groups but few members inside mean that global exploration is preferred, given the current search progress on hand. The groups change dynamically as the search goes on without worrying how much local intensification and how much global exploration are needed to be set. Once some initial group number is defined at the beginning, through mutation and evolutions of group sizes and numbers, the balance of local/global search patterns is automatically tuned by itself. By using a rigorous set of 14 testing functions for benchmarking, DGSA is shown to outperform the classical algorithms in most cases, with the lowest errors and fastest convergence speed.

ACKNOWLEDGMENT

The authors are thankful for the financial support from the Research Grant Temporal Data Stream Mining by Using Incrementally Optimized Very Fast Decision Forest (iOVFDF), Grant no. MYRG2015-00128-FST, offered by the University of Macau, FST, and RDAO.

REFERENCES

[1] Noa Pinter-Wollman, Elizabeth A. Hobson, Jennifer E. Smith, Andrew J. Edelman, Daizaburo Shizuka, Shermin de Silva, James S. Waters, Steven D. Prager, Takao Sasaki, George Wittemyer, Jennifer Fewell and David B. McDonald, "The dynamics of animal social networks: analytical, conceptual, and theoretical advances", Behavioral Ecology (2013), June 14, 2013

[2] John Bratton, "Introduction to Work and Organizational Behaviour", Palgrave Macmillan, 2015.

[3] Micheal F. Shlesinger, George M. Zaslavsky, Uriel Frisch, "Lévy Flights and Related Topics in Physics", Proceedings of the International Workshop Held at Nice, France, 27–30 June 1994, Lecture Notes in Physics, Volume 450, 1995

[4] Xin-She Yang, T. O. Ting, "Random Walks, Lévy Flights, Markov Chains and Metaheuristic Optimization", Future Information Communication Technology and Applications, Volume 235 of the series Lecture Notes in Electrical Engineering, pp.1055-1064

[5] Xin-She Yang and Suash Deb, "Cuckoo Search via Lévy flights", World Congress on Nature & Biologically Inspired Computing, 9-11 Dec 2009, pp.210-214

[6] Rosario N. Mantegna and H. Eugene Stanley, "Stochastic Process with Ultraslow Convergence to a Gaussian: The Truncated Lévy Flight", Phys. Rev. Lett. Vol. 73, Iss. 22—28 November 1994

[7] Xin-She Yang, "Bat algorithm for multi-objective optimisation", Int. J. Bio-Inspired Computation, Vol. 3, 2011, pp.267-274

[8] Simon Fong, Suash Deb, Xin-she Yang, "A heuristic optimization method inspired by wolf preying behavior", Neural Computing and Applications, Volume 26, Issue 7, October 2015, pp.1725-1738

[9] Noman, Nasimul, and Hitoshi Iba. "Accelerating differential evolution using an adaptive local search", Evolutionary Computation, IEEE Transactions on 12.1 2008, pp: 107-125.

[10] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization", Nanyang Technol. Univ., Singapore, IIT Kanpur, India, KanGAL Rep. 2005005, May 2005Yang, Xin-She. "Review of meta-heuristics and generalised evolutionary walk algorithm." International Journal of Bio-Inspired Computation 3.2 (2011): 77-84.

[11] Beheshti, Zahra, and Siti Mariyam Hj Shamsuddin. "A review of population-based meta-heuristic algorithms." Int J Adv Soft Comput Appl 5.1 (2013): 1-35.