Enhancement of Gravitational Search Algorithm using A Differential Mutation Operator and Its Application on Reconstructing Gene Regulatory Network

Chukiat Worasucheep[1]

ABSTRACT

Gravitational Search Algorithm (GSA) is a recent stochastic search algorithm that is inspired from the concepts of gravity rule and law of motion in physics. Despite its success and attractiveness, it has some coefficients and parameters that should be properly tuned to improve its performance. This paper studies the performance of GSA by varying the parameters that controls its gravitational force. Then a new differential mutation operator is proposed to enhance performance of GSA by accelerating its convergence. The proposed algorithm, namely DMGSA, is evaluated using 15 well-known benchmark functions from the special session of CEC2013 with different characteristics including randomly shifted optimum, rotation and non-separability. The results obviously confirms the performance achieved from the proposed mutation operator outperforms that from the attempts of parameter tuning in the original GSA. Lastly, DMGSA is applied for optimizing a small-scale gene regulatory network. The result demonstrates that its performance is highly competitive and clearly surpasses original GSA.

Keywords: Gravitational Search Algorithm, Hybrid Algorithm, Differential Mutation, Gene Regulatory Network

1. INTRODUCTION

Nature-inspired metaheuristic algorithms are algorithms that uses an iterative heuristic process to search for optimal solutions from the search space. Among them, evolutionary algorithms and swarm intelligence are very popular during the last few decades due to its performance and ease of implementation. Gravitational Search Algorithm (GSA) is another metaheuristic algorithm that is inspired from the concepts of gravity rules of physics [1]. Since proposed in 2009, GSA has been well in focus and its variants have been developed and successfully applied in various optimization problems in many fields. Some applications of GSA are in power engineering [2], civil engineering [3], chemical industry [4], bioinformatics [5], face recognition [6], clustering [7], feature selection [8], control engineering [9], etc. More applications and improved variants can be seen in a recent comprehensive survey [10].

Despite its success so far, original GSA has some inherent weaknesses that limit its performance. First, GSA has a few constants governing its gravitational coefficient and number of best solutions, called K_{best}, for calculating the driving force. The size of $K_{best}$is recommended to be high at the beginning and decreases to 1 during the search progress. Both gravitational coefficient and K_{best}have a strong effect performance of GSA and deserve a thorough investigation. Second, GSA does not maintain and fully utilize the best solutions in the current population to guide the search direction [11].

The first part of this research is to comprehensively study these in-built parameters using 6 well-known benchmark functions for minimization. This part aims to discover the optimal set of parameters for further improvement using a directional mutation operator that is inspired from differential evolution algorithm [12]. Differential evolution (DE) has been proved to be one of the most efficient algorithms for optimization in continuous domain [13]. In the second part of this research, the performance of the proposed algorithm and the considered variants has been validated on 15 well-known benchmark functions of IEEE Congress on Evolutionary Computation (CEC), 2013. The comparative experiment is conducted in terms of mean and standard deviation of final fitness obtained and uses non-parametric Wilcoxon signed-rank test for ranking the variants. Lastly the proposed algorithm is applied to reconstruct a small-scale hypothetical gene regulatory network (GRN). GRN is a nonlinear ordinary differential equation model for analyzing the gene expression to understand regulatory mechanism among genes. Performance of the proposed algorithm is statistically compared to original GSA and another efficient hybrid algorithm.

The remaining of this paper are as follows. Section 2 briefly reviews GSA and its variants. Section 3 analyzes GSA's parameters and proposes the en-

Manuscript received on July 17, 2018 ; revised on March 1, 2019.

Final manuscript received on March 13, 2019.

[1] The author is with Applied Computer Science Program, Department of Mathematics Faculty of Science, King Mongkut's University of Technology Thonburi, Thailand., E-mail: chukiat.wor@kmutt.ac.th

hancement. Section 4 explains the experimentation, results and discussion. Section 5 investigates performance of the proposed algorithm in reconstructing a GRN. Finally, Section 6 concludes this paper with some possible future works.

2. OVERVIEW OF GSA

GSA is an iterative population-based stochastic search algorithm inspired by the Newton's law of gravity and law of motions. GSA simulates mass interactions in a multi-dimensional search space under the influence of gravitational forces. In GSA, a population of NP agents of $x_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,D}]$, where $i = 1, 2, 3, \ldots, NP$, are candidate solutions moving in a search space of D dimensions. Each agent i has its own velocity $v_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,D}]$. During the initialization, all the agents are created and their positions are uniformly randomized within the search boundary of each dimension. Then the agents are moved by the force that are proportional to their masses and their distance. In iteration t, the total gravitational force on the ith agent is calculated from

$$F_{i,j} = \sum_{k \in K_{best}, k \neq i} rand_k \cdot G^t \cdot \frac{M_k \cdot M_i}{R_{i,k} + \varepsilon} \cdot (x_{k,j} - x_{i,j}) \quad (1)$$

where $j = 1, 2, \ldots, D$; $rand_k$ is random value within the range [0, 1]; M_i and M_k are the masses of agent i and k, respectively; $R_{i,k}$ is the Euclidean distance between agent i and k; ϵ represents a tiny value to avoid division of zero. This gravitational force is calculated from the top K agents called K_{best}, where K is defined by:

$$K = NP - (NP - 1) \times \frac{t}{t_{max}} \quad (2)$$

t_{max} is the maximum iteration allowed. G^t is the gravitational coefficient which is calculated from

```
t ← 0
Create and randomize an initial population of NP agents
Compute f(X) for fitness value of every agent
while t < t_max do
    Calculate K for K_best with equation (2)
    for each agent do
        Step 1. Calculate gravitational coefficient with equation (3)
        Step 2. Calculate masses with equation (4)
        Step 3. Calculate gravitational force with equation (1)
        Step 4. Calculate velocity with equation (5)
        Step 5. Update position with equation (6)
        Step 6. Compute f(X) for fitness value of the agent
    end for
    t ← t + 1
end while
```

Fig.1: *Original Gravitational Search Algorithm.*

$$G^t = G^0 \times \exp(-\alpha \times \frac{t}{t_{max}}), \quad (3)$$

where G^0 equals to 100, α equals to 20, as recommended in [1]. The mass M_i is defined by $M_i = q_i / \sum_{n=1}^{NP} q_n$ and

$$q_i = \frac{f(x_i) - f(x_{worst})}{f(x_{best}) - f(x_{worst})} \quad (4)$$

where x_{best} and x_{worst} are positions of the agent with the best (lowest) and the agent with the worst (highest) fitness values, respectively, for a minimization problem. $f(x)$ represents the fitness function on agent x. After the gravitational force is computed, the acceleration of each agent is calculated by $a^t_{i,j} = F_{i,j}/M_i$ Then the velocity and new position of each agent are computed as in (5) and (6).

$$v^{t+1}_{i,j} = rand_i \times v^t_{i,j} + a^t_{i,j} \quad (5)$$

$$x^{t+1}_{i,j} = x^t_{i,j} + v^{t+1}_{i,j} \quad (6)$$

All the above procedure are repeated for $t_{\max}$ iterations. Then the position of the heaviest mass, or the agent with lowest fitness value, is considered as the optimal solution. The algorithm of GSA can be outlined as in Fig 1.

Since GSA was introduced in 2009, many researchers have proposed improved variants of GSA with many different techniques. Some of them are inspired from physics phenomena such as disruption [14] and black hole [15], whereas some others are mathematical techniques such as orthogonality [3] and K-harmonic means [7]. Sarafrazi *et al.* [14] proposed a disruption operator for GSA in which agents can scatter or disrupt under the force exerted by the heaviest agent. Doraghinejad *et al.* [15] introduced astronomical black hole operator for GSA in which the heaviest agent is considered as the black hole that attracts other agents. Khatibinia *et al.* [3] introduced an orthogonal crossover with a quantization technique and a local search into the GSA for accelerating convergence and avoiding local optima simultaneously. Yin *et al.* [7] hybridized GSA with K-harmonic means algorithm for clustering problem.

Hybridization with other algorithms have been successfully reported including opposition-based learning concept [2], particle swarm optimization [4][5][11] and DE [16][17]. DE is an evolutionary algorithm in which the population of members are iteratively processed through mutation and crossover operations. In each iteration, a mutant member is created using a mutation operator on a target member. Then a trial member is built from the components of either the mutant member or the target member depending on a probability called crossover rate. If the trial member has an equal or better fitness value, it replaces the corresponding target member for next generation, otherwise the target member is retained. This selection policy of DE guarantees that the overall fitness value of the population will never deteriorate. Detailed description of DE can be reached from [12][13].

3. THE PROPOSED ALGORITHM

For all metaheuristic search algorithms, balancing exploration and exploitation is a key success factor. Exploration or diversification deals with searching for promising subspaces with a high potential of optima, while exploitation or intensification focuses on local search for an optimum in the promising subspaces. In GSA parameter K denotes the number of other best agents for calculating the gravitational force which is the main interaction between agents that drives them to the optima. The parameter K is initialized to NP (total number of agents) and is decreased linearly to 1, so as to support more exploration in early search stage and gradually change to exploitation during the search.

The gravitational coefficient G is another important parameter of GSA controlling the magnitude of force in (1) and decreases with time to control the search accuracy. This G directly influences convergence speed and, like K, balances the exploration and exploitation. By analyzing eq. (3), it is obvious that the value of G is highly dependent on G^0 and the rate of decent is controlled with α. The constants G^0 and α, are recommended to be equal to 100 and 20, respectively [1]. Fig 2 illustrates the changing rate of G for α equals to 3, 7, and the original value of 20, while keeping G^0 equal to 100.

At the same time, it is interesting to examine the performance if K is changed non-linearly against linearly as originally proposed [1]. To do this, we mimic the exponential decay rate from G and we have

$$K = NP \times \exp(-\beta \times \frac{t}{t_{\max}}) \cdot \tag{7}$$

A new parameter β controls the acceleration rate of decent of K. As shown in Fig 3, the higher value of β, the faster exponential term increases the decay rate in the early stage than the original linear term in (2).

Therefore the first goal of this research is to study the performance if we change the values of G^0, α and β to other values. An experiment will be conducted on a small subset of well-known benchmark functions for minimization to discover the combinations of $[G^0, \alpha, \beta, NP]$ that present the best performance. NP is also included here since it is widely known that population size generally has effects on performance of a metaheuristic algorithm. Such best combinations will be used for further improvement.

The improvement proposed in this research focuses on the velocity and position update of the agent. Although GSA identifies the best agent and the worst

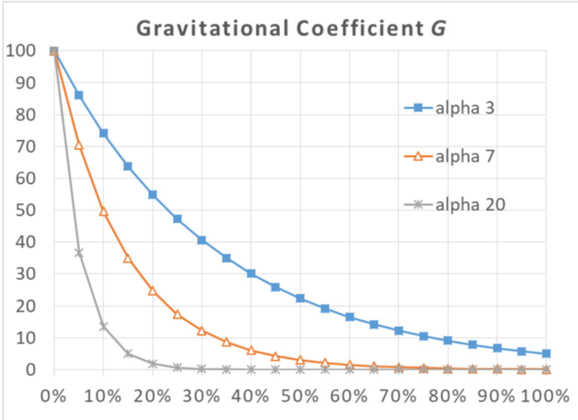

Fig.2: *Gravitational Coefficient G.*

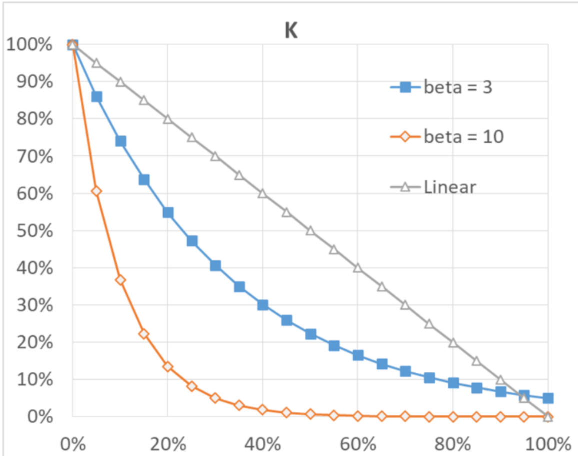

Fig.3: *The number of best agents (K).*

agent for calculating masses, it does not maintain and fully utilize them in the current population to guide the search direction [11]. Inspired from the DE [12], the best agent and the worst agent together provide fitness information useful for guiding the population toward the optima. The proposed GSA in this work modifies the steps 4 to 6 of the original GSA in Fig 1 by adding a novel differential mutation operator using the best agent and the worst agent. The difference of positions of the best agent and the worst agent can be employed as a direction guide. This difference when being multiplied with a random value assists in bringing the current agent toward a better position. This mutation would enhance the convergence as can be seen in the experimentation thereafter. To avoid excessive convergence that might cause undesirable premature convergence, the proposed mutation will be used on only particular dimensions regulated by a parameter called crossover rate (CR). The higher CR value, the more chance a dimension of the trial agent, comes from the differential mutation, rather than the original velocity and position update of GSA.

In addition, the selection policy of DE is utilized

```
k = randint (1..D)
for each dimension d do
  if rand(0, 1) < CR or d = k do
    trial.x[d] = target.x[d] + rand(0,1) × (best.x[d] – worst.x[d])
  else
    calculate velocity with equation (5)
    update position with equation (6)
  end if
  check boundary of trial.x[d]
end for
compute fitness value trial.fitness of the temp agent
if trial.fitness is better than target.fitness then
  update position of target with position of trial agent
  target.fitness ← trial.fitness
end for
```

Fig.4: *Velocity and position update of the proposed DMGSA.*

in which the new trial agent will replace the current agent only if the new trial agent's fitness value is better. Therefore steps 4 to 6 in the original GSA will be replaced with the algorithm as shown in Fig 4, where $randint(a, b)$ denotes a random integer between a and b inclusively; $rand(m, n)$ denotes a random value between m and n. trial and target denote the trial agent and target agent, respectively; $best$ and $worst$ denote the agent with the best fitness value and with the worst fitness value, respectively, in the current iteration. Note that, to guarantee at least one dimension gets the differential mutation like in DE, the conditional statement has a conjunction term for checking if the dimension d is equal to the pre-randomized index k. The Experiment II in the next section studies performance of the proposed algorithm, namely differential mutated GSA or DMGSA, with regard to this new parameter CR.

4. EXPERIMENTATION

The experiment in the work has two steps. Experiment I is a combination test to study the performance of GSA by varying four parameters $[G^0, \alpha, \beta, NP]$. Experiment II studies the performance improvement of the proposed strategies on the three most efficient variants obtained from Experiment I.

4.1 Experiment I: Setup

In this experiment, the performance of GSA are tested using six widely-accepted benchmark functions, namely Sphere, Schwefel, Rosenbrock, Quartic with noise, Ackley and Rastrigin at dimension (D) of 30. The search ranges are as shown in Table 1. The combination of 4 parameters, $G0, \alpha, \beta, NP$, for testing are as follows:

G^0 :30, 100, 300
α:3, 7, 20
β:3, 10, li (li stand for linear)
NP : 20, 40, 60

Values 3 and 10 of β are for calculating the number K in eq. (7) while the linear β means that the number K is changed linearly using eq. (2). There are 4 parameters to test, each of which has 3 values, therefore a total of $3^4 = 81$ parameter cases or combinations are to be tested. For each test case, 30 independent runs will be executed. Each run allows a maximum number of objective function calls ($maxnfc$) of 100,000 as stopping criterion. The mean and s.d. of the final optimal values obtained from all runs will be computed. Tables 2 and 3 report all 81 sets of means and s.d. for the Sphere function and the Ackley function respectively. To conserve space, those results of other functions can be obtained upon request. The algorithm name is encoded with G*ggg*-A*aa*-B*bb*-N*nn* to conserve space as well; *ggg* denotes the G^0, aa denotes the α, *bb* denotes the β and the *nn* denotes the NP. Note again that *bb* equal to '*li*' means the linearly changing as in eq. (2).

Table 1: *Ranges of the benchmark functions in Experiment I.*

Function	Range
F1 Sphere	[-100, 100]
F2 Schwefel	[-5.12, 5.12]
F3 Rosenbrock	[-30, 30]
F4 Quartic with noise	[-1.28, 1.28]
F5 Ackley	[-32, 32]
F6 Rastrigin	[-5.12, 5.12]

Non-parametric Wilcoxon's signed rank test is conducted to draw a statistically comparison at 0.05 significant level between the two algorithms in the same function. An algorithm j is ranked higher than algorithm k (i.e. $r_j < r_k$) if the Wilcoxon signed-rank test result of algorithms j against k gets a p-value below 0.05. Any two algorithms are ranked equally if the Wilcoxon signed-rank test result is not significant.

Then the ranks of all six functions run with each parameter case are collected and used to compute the mean ranks as shown in the in Table 4. Column *mean rank for unimodal* is the mean rank for all unimodal functions (F1 to F4), while the *mean rank for multimodal* column is the mean rank for two multimodal functions (F5 and F6). The last column is the mean rank for all 6 tested functions. To conserve space, only cases with top 5 ranks in any of the last 3 columns are presented here. The algorithms with the top rank (=1) for three categories (unimodal, multimodal and all functions) are highlighted with italic and bold font.

Table 2: *Means and S.D. of the fitness values obtained for function F1 Sphere.*

Algorithm	Mean±S.D.	Algorithm	Mean±S.D.	Algorithm	Mean±S.D.
G030-A03-B03-N20	4.969E+04±5.167E+03	G030-A03-B03-N40	5.261E+04±6.977E+03	G030-A03-B03-N60	5.532E+04±4.949E+03
G030-A03-B10-N20	6.156E+04±1.034E+04	G030-A03-B10-N40	6.224E+04±7.220E+03	G030-A03-B10-N60	5.715E+04±6.415E+03
G030-A03-B*li*-N20	4.038E+04±6.278E+03	G030-A03-B*li*-N40	5.261E+04±5.471E+03	G030-A03-B*li*-N60	5.462E+04±3.929E+03
G030-A07-B03-N20	5.830E+04±7.071E+03	G030-A07-B03-N40	5.821E+04±6.897E+03	G030-A07-B03-N60	5.913E+04±6.830E+03
G030-A07-B10-N20	6.082E+04±5.687E+03	G030-A07-B10-N40	6.356E+04±4.060E+03	G030-A07-B10-N60	6.177E+04±4.971E+03
G030-A07-B*li*-N20	5.552E+04±6.800E+03	G030-A07-B*li*-N40	6.205E+04±7.149E+03	G030-A07-B*li*-N60	5.867E+04±3.311E+03
G030-A20-B03-N20	6.202E+04±7.070E+03	G030-A20-B03-N40	6.066E+04±4.467E+03	G030-A20-B03-N60	5.855E+04±6.624E+03
G030-A20-B10-N20	6.769E+04±8.763E+03	G030-A20-B10-N40	6.053E+04±5.932E+03	G030-A20-B10-N60	6.467E+04±6.975E+03
G030-A20-B*li*-N20	6.246E+04±8.416E+03	G030-A20-B*li*-N40	6.493E+04±4.951E+03	G030-A20-B*li*-N60	5.893E+04±7.172E+03
G100-A03-B03-N20	9.854E+01±2.946E+02	G100-A03-B03-N40	2.649E+04±4.135E+03	G100-A03-B03-N60	3.762E+04±5.470E+03
G100-A03-B10-N20	3.632E+04±6.418E+03	G100-A03-B10-N40	5.029E+04±3.961E+03	G100-A03-B10-N60	5.364E+04±4.566E+03
G100-A03-B*li*-N20	8.641E-01±2.032E-01	G100-A03-B*li*-N40	1.492E+04±3.767E+03	G100-A03-B*li*-N60	2.715E+04±4.191E+03
G100-A07-B03-N20	2.985E+04±4.338E+03	G100-A07-B03-N40	4.042E+04±6.438E+03	G100-A07-B03-N60	4.825E+04±5.336E+03
G100-A07-B10-N20	4.277E+04±5.287E+03	G100-A07-B10-N40	5.440E+04±5.298E+03	G100-A07-B10-N60	5.336E+04±4.257E+03
G100-A07-B*li*-N20	2.163E+04±1.781E+03	G100-A07-B*li*-N40	4.090E+04±4.118E+03	G100-A07-B*li*-N60	4.831E+04±4.420E+03
G100-A20-B03-N20	5.156E+04±9.801E+03	G100-A20-B03-N40	5.609E+04±7.152E+03	G100-A20-B03-N60	5.525E+04±3.953E+03
G100-A20-B10-N20	5.981E+04±6.791E+03	G100-A20-B10-N40	5.628E+04±6.578E+03	G100-A20-B10-N60	6.264E+04±5.568E+03
G100-A20-B*li*-N20	5.046E+04±6.598E+03	G100-A20-B*li*-N40	5.480E+04±3.895E+03	G100-A20-B*li*-N60	5.985E+04±6.493E+03
G300-A03-B03-N20	1.093E+00±3.260E-01	G300-A03-B03-N40	8.605E-01±2.653E-01	G300-A03-B03-N60	7.023E-01±2.632E-01
G300-A03-B10-N20	8.947E+00±1.929E+00	G300-A03-B10-N40	1.067E+04±3.871E+03	G300-A03-B10-N60	2.572E+04±4.680E+03
G300-A03-B*li*-N20	3.221E+00±8.570E-01	G300-A03-B*li*-N40	1.391E+00±1.876E+00	G300-A03-B*li*-N60	2.617E+00±6.230E+00
G300-A07-B03-N20	1.520E-02±5.989E-03	G300-A07-B03-N40	3.149E+03±2.807E+03	G300-A07-B03-N60	2.088E+04±4.495E+03
G300-A07-B10-N20	5.469E+00±6.025E+00	G300-A07-B10-N40	2.295E+04±3.613E+03	G300-A07-B10-N60	3.866E+04±4.003E+03
G300-A07-B*li*-N20	5.747E-02±2.628E-02	G300-A07-B*li*-N40	1.292E-02±3.368E-03	G300-A07-B*li*-N60	1.541E+04±2.404E+03
G300-A20-B03-N20	1.668E+04±6.771E+03	G300-A20-B03-N40	4.044E+04±5.679E+03	G300-A20-B03-N60	4.751E+04±2.916E+03
G300-A20-B10-N20	2.635E+04±3.069E+03	G300-A20-B10-N40	4.529E+04±3.731E+03	G300-A20-B10-N60	4.902E+04±4.951E+03
G300-A20-B*li*-N20	1.645E+04±2.557E+03	G300-A20-B*li*-N40	3.790E+04±7.217E+03	G300-A20-B*li*-N60	4.767E+04±5.483E+03

Table 3: *Means and S.D. of the fitness values obtained for function F5 Ackley.*

Algorithm	Mean±S.D.	Algorithm	Mean±S.D.	Algorithm	Mean±S.D.
G030-A03-B03-N20	19.31±1.09	G030-A03-B03-N40	8.17±9.55	G030-A03-B03-N60	0.32±0.08
G030-A03-B10-N20	1.43±0.24	G030-A03-B10-N40	14.44±3.03	G030-A03-B10-N60	18.57±0.67
G030-A03-Bli-N20	12.20±8.60	G030-A03-B*li*-N40	17.55±5.75	G030-A03-B*li*-N60	5.19±7.27
G030-A07-B03-N20	15.71±7.84	G030-A07-B03-N40	5.80±4.20	G030-A07-B03-N60	17.77±0.54
G030-A07-B10-N20	1.38±0.87	G030-A07-B10-N40	18.27±0.62	G030-A07-B10-N60	19.32±0.56
G030-A07-Bli-N20	18.64±2.24	G030-A07-B*li*-N40	0.56±1.21	G030-A07-B*li*-N60	16.44±0.90
G030-A20-B03-N20	17.70±0.95	G030-A20-B03-N40	19.66±0.27	G030-A20-B03-N60	19.81±0.38
G030-A20-B10-N20	18.27±0.83	G030-A20-B10-N40	20.05±0.16	G030-A20-B10-N60	20.13±0.18
G030-A20-Bli-N20	17.23±0.92	G030-A20-B*li*-N40	19.60±0.24	G030-A20-B*li*-N60	19.93±0.19
G100-A03-B03-N20	15.96±7.60	G100-A03-B03-N40	12.46±9.46	G100-A03-B03-N60	7.91±8.85
G100-A03-B10-N20	20.31±0.15	G100-A03-B10-N40	18.41±5.56	G100-A03-B10-N60	8.91±9.15
G100-A03-Bli-N20	14.61±8.25	G100-A03-B*li*-N40	10.56±9.07	G100-A03-B*li*-N60	5.25±7.46
G100-A07-B03-N20	15.19±7.60	G100-A07-B03-N40	13.44±8.82	G100-A07-B03-N60	11.39±8.23
G100-A07-B10-N20	20.24±0.08	G100-A07-B10-N40	12.68±9.08	G100-A07-B10-N60	0.71±0.73
G100-A07-Bli-N20	13.52±8.77	G100-A07-B*li*-N40	12.76±7.68	G100-A07-B*li*-N60	14.61±6.29
G100-A20-B03-N20	19.22±0.93	G100-A20-B03-N40	9.19±8.68	G100-A20-B03-N60	14.89±0.95
G100-A20-B10-N20	19.83±0.12	G100-A20-B10-N40	1.90±0.49	G100-A20-B10-N60	16.48±0.59
G100-A20-Bli-N20	18.09±4.54	G100-A20-B*li*-N40	6.36±4.94	G100-A20-B*li*-N60	14.10±0.89
G300-A03-B03-N20	20.26±0.10	G300-A03-B03-N40	14.68±8.55	G300-A03-B03-N60	12.91±9.17
G300-A03-B10-N20	20.33±0.14	G300-A03-B10-N40	18.38±5.40	G300-A03-B10-N60	19.79±0.55
G300-A03-Bli-N20	16.95±7.11	G300-A03-B*li*-N40	10.90±8.58	G300-A03-B*li*-N60	15.78±7.38
G300-A07-B03-N20	15.83±7.85	G300-A07-B03-N40	17.41±5.88	G300-A07-B03-N60	11.80±8.86
G300-A07-B10-N20	20.34±0.12	G300-A07-B10-N40	16.08±6.86	G300-A07-B10-N60	15.32±7.23
G300-A07-Bli-N20	12.38±8.28	G300-A07-B*li*-N40	11.67±8.85	G300-A07-B*li*-N60	10.59±6.97
G300-A20-B03-N20	16.39±6.22	G300-A20-B03-N40	18.55±1.93	G300-A20-B03-N60	18.72±1.30
G300-A20-B10-N20	19.35±1.55	G300-A20-B10-N40	16.81±6.03	G300-A20-B10-N60	19.34±1.17
G300-A20-Bli-N20	17.25 ±5.04	G300-A20-B*li*-N40	18.66±1.21	G300-A20-B*li*-N60	19.24±0.66

Table 4: *Performance rankings for each function and function group. Only the top 5 ranks in each group (in the last three columns) are shown. .*

Algorithm	F1	F2	F3	F4	F5	F6	Mean rank for unimodal	Mean rank for multimodal	Mean rank for all 6 functions
G100-A07-B03-N20	27	11	15	5	35	37	4	35	6
G300-A07-B03-N20	2	19	9	12	39	38	**1**	39	2
G300-A07-Bli-N20	3	21	11	30	22	32	5	16	3
G030-A07-Bli-N40	74	6	20	22	2	9	22	**1**	9
G100-A07-Bli-N40	35	12	1	18	25	19	6	8	**1**
G100-A20-B03-N40	57	38	61	35	14	22	51	3	30
G100-A20-Bli-N40	53	70	74	58	10	4	79	2	46
G300-A07-B03-N40	14	13	12	13	49	43	2	54	12
G300-A07-Bli-N40	1	18	47	10	19	25	8	8	4
G030-A03-Bli-N60	52	23	56	16	7	31	32	5	22
G100-A20-Bli-N60	67	62	54	56	29	7	73	3	48
G300-A07-B03-N60	20	14	3	20	20	47	3	27	5

4.2 Experiment I: Results and Discussion

It can be observed from Table 4 that while G300-A07-B03-N20 outperforms in unimodal function set, its performance is disappointing for multimodal functions with a mean rank of 39. Similarly while G030-A07-Bli-N40 outperforms in multimodal function set, it is ranked 22 for unimodal functions. This demonstrates that parameter setting of original GSA highly depends on characteristics of the functions. The best performer for all six functions tested on average becomes G100-A07-Bli-N40, with relatively good ranks of 6 and 8 for unimodal and multimodal sets respectively.

By analyzing values of parameters for these top performers, we may summarize that $G^0 = 100$, $\alpha = 7$ and a linearly decreasing K is recommended for solving a problem with unknown characteristics. For unimodal functions, one should increase the G^0 and reduce NP for faster convergence. And for multimodal functions, G^0 should be reduced to avoid premature convergence.

4.3 Experiment II: Setup

This experiment will improve GSA with the selected top parameter cases from experiment I, which includes G300-A07-B03-N20, G030-A07-Bli-N40, and G100-A07-Bli-N40. The proposed changes are as explained in section 3. The performance will be evaluated using the first 15 benchmark functions in the special session of CEC 2013 [18]. This benchmark suit has different characteristics often found in real-world problems and consists of 5 unimodal functions (C1 - C5) and 10 multimodal functions (C6 - C15). All functions have their optimum shifted randomly; and many of which are rotated and non-separable or asymmetric. Detailed descriptions can be reached from [18]. The test will be conducted on 30 dimensions. Since the parameter CR controls the probability of running the differential mutation, three different values of CR in {0.15, 0.5, 0.85} will be evaluated for the proposed algorithm. Thus the following 15 variants will be tested for each function.

GSA-NP: original GSA with NP = 20, 40 and 60

$$\left.\begin{array}{l}\text{G300-A07-B03-N20}\\ \text{G030-A07-B}li\text{-N40}\\ \text{G100-A07-B}li\text{-N40}\backslash\end{array}\right\}\text{from Experiment I}$$

G300-A07-B03-N20 with CR = 0.15, 0.5 and 0.85
G030-A07-Bli-N40 with CR = 0.15, 0.5 and 0.85
G100-A07-Bli-N40 with CR = 0.15, 0.5 and 0.85.

Each variant will be executed for 30 independent runs. Each run will terminate at the maximum number of objective function calls equal to 200,000. The mean and s.d. of the final fitness values obtained from all runs will be computed and tabulated in Tables 5 to 7. Like in experiment I, Wilcoxon's signed rank test is conducted to draw a statistically comparison between the two variants at 0.05 significant level in order to rank all variants in the same test function.

4.4 Experiment II: Results and Discussion

The ranks of each algorithm are averaged for all 15 functions and grouped by modality as shown in Table 8. A lower average rank presents a better performance. It's obvious from this table that the original GSA is defeated by all proposed variants. G300-A07-B03-N20-C0.85 outperforms in unimodal functions while G300-A07-B03-N20-C0.15 outperforms in the cases of multimodal functions and for all 15 tested functions. Both winner variants are with $G^0 = 300$, $\alpha = 7$, $\beta = 3$ and $NP = 20$, but with different CR values. A higher value of G0 offers enough convergence speed in the early iterations. The exponential decay of gravitational coefficient G with $\alpha = 7$ and the exponential decay of number K with $\beta = 3$ together shift the exploration to exploitation in a right pace.

Difference values of the parameter CR are recommended for different function characteristics. A high value of CR, like 0.85, is recommended for unimodal functions to take advantage of fast convergence of the proposed differential mutation. In contrast, a lower value of CR, e.g. 0.15, outperforms for the cases of multimodal or unknown functions which prefer gravitational search operator to the differential mutation.

Another observation of the Table 8 is that the variants with the proposed differential mutation operator outperform the variants without in all 3 cases. For example, G030-A07-Bli-N40 has an average rank of 12.1, while its DMGSA variants with CR equals to 0.15, 0.5 and 0.85 have average ranks of 5.0, 4.3 and 3.9, respectively. Similar outcome is for G300-A07-B03-N20 and G100-A07-Bli-N40. This confirms the effectiveness of the proposed mutation operator, regardless of the inherent parameters of GSA.

Fig 5 illustrates average convergence graphs of some CEC2013 functions for comparing the convergence of tested variants. The convergence graphs of GSA-N20 and G300-A07-B03-N20 can sometimes fluctuate as clearly seen in functions C14 and C12. This is because the replacement policy in original GSA is not greedy as in the proposed DMGSA. However the GSA configuration of $G^0 = 300$, $\alpha = 7$, $\beta = 3$ and $NP = 20$ considerably outperforms the original GSA (with $G^0 = 100$, $\alpha = 20$ with a linearly decreasing K) in all cases. Furthermore, the proposed differential mutation successfully assists in improving the performance in most cases.

5. APPLICATION FOR RECONSTRUCTING GENE REGULATORY NETWORK

This section describes an application of DMGSA for reconstructing gene regulatory network (GRN). GRN plays a very important role in cellular metabolism during development of living organism. With current increasing number of DNA and mRNA sequences are becoming available, reconstructing gene networks from expression profile data can help biologists investigate complex interactions among genes of an organism. In computational biology, reconstructing GRN employs mathematical model for analysis of gene expression data. The reconstructing process requires selecting the network model and fitting structural parameters of the network to the available data [20]. The most popular and well-researched model is S-system model, which is an ordinal differential equation model characterized by non-linear power law functions [21]. The model is given by

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} X_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} X_j^{h_{i,j}} \tag{8}$$

where N is the number of genes in the system and X_i is the expression level of the gene i. The first term represents factors that promote the expression X_i while the second term represents factors that inhibit the expression. Parameters α_i and β_i are non-negative rate constants, whereas $g_{i,j}$ and $h_{i,j}$ are kinetic orders that reflect the intensity of interactions from gene j to i in the synthesis and degradation processes, respectively. Reconstructing GRN of N genes with S-system model with (8) can be considered as an optimization problem with $2 \times N(N+1)$ parameters of $\{\alpha_i, \beta_i, g_{i,j}, h_{i,j}\}$ that must be estimated simultaneously [22][23]. This means that the number of parameters increasing quadratically; for example, there are 60 parameters to be optimized for a small five-gene GRN.

5.1 Algorithms and Parameter Setting

In this section, the proposed DMGSA, original GSA and OABCDE are employed to optimize a small-scale five-gene GRN problem in order to compare the performance. OABCDE is a hybrid algorithm of artificial bee colony optimization algorithm and DE with an opposition-based learning to enhance exploration and exploitation simultaneously [24]. Details and parameter settings of OABCDE can be reached from [24]. For the case of DMGSA, we choose the optimal combination of $G^0 = 300$, $\alpha = 7$ and $\beta = 3$, while CR are tested for both 0.15 and 0.85; thus they are named for brevity as DMGSA-15 and DMGSA-85, respectively.

By using these optimization algorithms, some measure is required to guide the population in the search space and hence is the fitness function. This work uses the mean quadratic discrepancy between the model output X_{cal} and observed expression X_{act} in (9), which is the most common quality assessment criterion [23],

$$f = \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\frac{X_{cal,i,t} - X_{act,i,t}}{X_{act,i,t}} \right)^2 \tag{9}$$

The GRN problem in this work is a hypothetical network generated by the parameters given in Table 9. The gene expression levels of the networks are plotted on the left hand side of Fig 6, each consists of 50 time course of expression. The search space for α_i and β_i, is [0.0, 15.0] and for $g_{i,j}$ and $h_{i,j}$ to [-3.0, 3.0]. Delta time is 0.01. The initial value (T_0) of 5 genes are [0.7, 0.12, 0.14, 0.16, 0.18]. A structure skeletalizing is applied to reduce the computational burden in a similar manner by [22][25]. If the absolute value of a parameter is less than a threshold value δ, then the parameter is reset to 0. In this experiment, we use δ = 0.001.

Since the optimization problem in this section is a hypothetical five-gene GRN, each agent or individual of the population is encoded as [$\alpha_1, \ldots, \alpha_5, \beta_1, \ldots, \beta_5, g_{1,1}, \ldots, g_{5,5}, h_{1,1}, \ldots, h_{5,5}$] with 60 dimensions. Each algorithm runs by using a population size of 40 and 80 to investigate the difference. In each case, the algorithm runs for 30 independent runs as the algorithms are stochastic. Maximum number of objective function calls for each run is set to 400,000.

5.2 Results and Discussion

Table 10 shows basic statistical results of the final fitness values from running each algorithm; sorted

Table 5: *Means and s.d. of fitness values obtained from 30 independent runs of unimodal functions (C1-C5) in CEC2013.*

Algorithm	C11		C12		C13		C14		C15	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
GSA-N20	7.849E+04	9.937E+03	2.726E+09	1.221E+09	4.945E+22	1.895E+23	1.005E+07	1.735E+07	7.259E+04	3.188E+04
G300-A07-B03-N20	1.794E-02	5.983E-03	2.930E+05	9.419E+04	2.036E+05	2.357E+05	7.102E+04	2.134E+04	5.793E-03	1.738E-03
G300-A07-B03-N20-C0.15	8.894E-04	3.502E-04	2.596E+07	9.732E+06	6.716E+09	3.925E+09	4.502E+04	6.552E+03	1.313E-03	3.410E-04
G300-A07-B03-N20-C0.50	8.033E-03	1.315E-03	4.976E+07	1.649E+07	3.216E+09	3.501E+09	2.926E+04	5.694E+03	3.019E-03	9.292E-04
G300-A07-B03-N20-C0.85	1.293E-03	2.447E-04	3.251E+07	2.102E+07	2.120E+09	4.334E+09	1.654E+04	5.847E+03	6.477E-04	1.611E-04
GSA-N40	8.578E+04	1.117E+04	2.393E+09	9.139E+08	2.727E+21	6.726E+21	3.826E+06	7.008E+06	6.281E+04	1.994E+04
G030-A07-Bli-N40	8.745E+04	1.165E+04	2.601E+09	7.857E+08	2.993E+22	1.032E+23	1.742E+06	2.466E+06	5.808E+04	2.098E+04
G030-A07-Bli-N40-C0.15	9.442E-04	2.382E-04	2.829E+07	9.119E+06	1.005E+10	3.385E+09	5.783E+04	9.280E+03	2.588E-04	9.828E-05
G030-A07-Bli-N40-C0.50	6.793E-04	8.984E-05	5.736E+07	1.466E+07	5.098E+09	4.674E+09	4.389E+04	6.302E+03	1.643E-04	3.372E-05
G030-A07-Bli-N40-C0.85	3.851E-04	7.143E-05	4.799E+07	1.355E+07	7.635E+09	1.074E+10	4.067E+04	8.365E+03	1.044E-04	1.786E-05
GSA-N60	8.680E+04	1.162E+04	2.603E+09	7.795E+08	1.836E+22	7.736E+22	3.528E+06	5.308E+06	5.566E+04	2.069E+04
G100-A07-Bli-N40	6.233E+04	8.445E+03	1.968E+09	8.338E+08	1.631E+21	3.358E+21	4.083E+06	7.186E+06	4.465E+04	1.577E+04
G100-A07-Bli-N40-C0.15	2.999E-03	8.083E-04	2.704E+07	1.056E+07	8.502E+09	4.004E+09	5.337E+04	9.103E+03	1.110E-03	3.896E-04
G100-A07-Bli-N40-C0.50	2.199E-03	3.334E-04	5.422E+07	1.508E+07	5.230E+09	5.194E+09	4.476E+04	8.145E+03	7.077E-04	1.870E-04
G100-A07-Bli-N40-C0.85	1.262E-03	1.492E-04	4.725E+07	1.758E+07	3.447E+09	4.042E+09	3.758E+04	8.698E+03	5.137E-04	1.165E-04

Table 6: *Means and s.d. of fitness values obtained from 30 independent runs of multimodal functions (C6 - C10) in CEC2013 .*

Algorithm	C11		C12		C13		C14		C15	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
GSA-N20	2.084E+04	6.042E+03	1.049E+08	1.804E+08	2.141E+01	7.375E-02	5.131E+01	1.712E+00	1.431E+04	2.479E+03
G300-A07-B03-N20	2.585E+01	1.988E+01	1.290E+02	3.165E+01	2.139E+01	8.234E-02	2.349E+01	3.979E+00	3.540E-01	8.739E-02
G300-A07-B03-N20-C0.15	2.896E+01	1.873E+01	1.252E+02	3.194E+01	2.095E+01	4.230E-02	3.077E+01	1.948E+00	1.212E-01	4.885E-02
G300-A07-B03-N20-C0.50	2.522E+01	2.032E+01	1.001E+02	3.737E+01	2.097E+01	4.419E-02	3.495E+01	1.631E+00	1.622E-01	3.780E-02
G300-A07-B03-N20-C0.85	2.208E+01	1.732E+01	1.064E+02	4.642E+01	2.096E+01	4.481E-02	3.787E+01	1.439E+00	6.011E-02	1.617E-02
GSA-N40	1.930E+04	4.370E+03	3.923E+07	4.895E+07	2.134E+01	1.158E-01	5.007E+01	2.228E+00	1.393E+04	1.868E+03
G030-A07-Bli-N40	1.993E+04	5.149E+03	5.531E+07	6.248E+07	2.135E+01	7.744E-02	5.027E+01	1.082E+00	1.481E+04	2.942E+03
G030-A07-Bli-N40-C0.15	5.828E+01	1.614E+01	1.185E+02	2.282E+01	2.095E+01	5.493E-02	3.085E+01	1.059E+00	1.787E+01	4.621E+00
G030-A07-Bli-N40-C0.50	6.263E+01	3.291E+01	9.873E+01	1.814E+01	2.096E+01	4.356E-02	3.545E+01	1.012E+00	1.309E+00	3.481E-01
G030-A07-Bli-N40-C0.85	2.713E+01	2.002E+01	8.293E+01	1.862E+01	2.094E+01	6.816E-02	3.839E+01	1.681E+00	2.113E-02	1.135E-02
GSA-N60	2.001E+04	4.329E+03	2.121E+07	2.362E+07	2.135E+01	6.348E-02	4.970E+01	1.705E+00	1.184E+04	2.133E+03
G100-A07-Bli-N40	1.596E+04	3.681E+03	3.296E+07	4.481E+07	2.136E+01	8.237E-02	5.042E+01	3.041E+00	1.117E+04	1.736E+03
G100-A07-Bli-N40-C0.15	5.656E+01	1.939E+01	1.194E+02	1.985E+01	2.096E+01	5.181E-02	2.975E+01	1.101E+00	1.825E+01	7.307E+00
G100-A07-Bli-N40-C0.50	3.555E+01	3.237E+01	1.051E+02	2.464E+01	2.097E+01	6.620E-02	3.589E+01	1.380E+00	1.294E+00	1.596E+00
G100-A07-Bli-N40-C0.85	3.084E+01	2.056E+01	7.535E+01	3.007E+01	2.097E+01	5.291E-02	3.877E+01	8.891E-01	3.447E-02	7.374E-03

Table 7: *Means and s.d. of fitness values obtained from 30 independent runs of multimodal functions (F11 - F15) in CEC2013.*

Algorithm	C11		C12		C13		C14		C15	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
GSA-N20	1.447E+03	2.399E+02	1.288E+03	7.751E+01	1.303E+03	1.796E+02	1.035E+04	6.345E+02	1.037E+04	4.784E+02
G300-A07-B03-N20	1.203E+02	2.592E+01	1.811E+02	3.913E+01	4.610E+02	1.817E+02	3.603E+03	5.389E+02	3.755E+03	6.297E+02
G300-A07-B03-N20-C0.15	1.415E+00	1.969E+00	2.178E+02	1.839E+01	2.241E+02	2.037E+01	8.898E+02	1.414E+02	5.989E+03	3.204E+02
G300-A07-B03-N20-C0.50	9.708E+01	1.077E+01	2.069E+02	1.589E+01	2.173E+02	2.229E+01	4.200E+03	3.981E+02	7.154E+03	2.874E+02
G300-A07-B03-N20-C0.85	1.004E+02	3.799E+01	2.351E+02	2.186E+01	2.334E+02	2.237E+01	5.984E+03	5.029E+02	7.427E+03	1.939E+02
GSA-N40	1.371E+03	2.339E+02	1.401E+03	1.612E+02	1.336E+03	1.817E+02	1.007E+04	5.752E+02	1.005E+04	4.178E+02
G030-A07-Bli-N40	1.380E+03	2.153E+02	1.428E+03	1.968E+02	1.387E+03	1.668E+02	9.965E+03	7.450E+02	1.000E+04	3.937E+02
G030-A07-Bli-N40-C0.15	2.359E+01	2.775E+00	2.384E+02	2.074E+01	2.494E+02	1.439E+01	1.296E+03	1.709E+02	5.956E+03	3.179E+02
G030-A07-Bli-N40-C0.50	1.240E+02	1.000E+01	2.230E+02	1.293E+01	2.355E+02	1.402E+01	4.302E+03	3.245E+02	7.214E+03	3.376E+02
G030-A07-Bli-N40-C0.85	1.724E+02	1.620E+01	2.314E+02	2.853E+01	2.267E+02	1.588E+01	6.039E+03	4.845E+02	7.303E+03	3.592E+02
GSA-N60	1.299E+03	2.075E+02	1.354E+03	1.223E+02	1.306E+03	1.866E+02	9.724E+03	4.140E+02	9.799E+03	4.635E+02
G100-A07-Bli-N40	1.063E+03	1.274E+02	1.038E+03	9.709E+01	1.061E+03	1.236E+02	9.838E+03	4.916E+02	9.721E+03	4.284E+02
G100-A07-Bli-N40-C0.15	2.277E+01	1.884E+00	2.348E+02	1.936E+01	2.445E+02	1.972E+01	1.236E+03	1.601E+02	5.910E+03	3.629E+02
G100-A07-Bli-N40-C0.50	1.226E+02	1.051E+01	2.338E+02	1.889E+01	2.308E+02	1.091E+01	4.351E+03	2.879E+02	7.317E+03	3.451E+02
G100-A07-Bli-N40-C0.85	1.805E+02	2.245E+01	2.207E+02	2.149E+01	2.209E+02	1.633E+01	6.143E+03	4.069E+02	7.327E+03	2.697E+02

Table 8: *Average ranks of tested variants.*

Algorithm	Average rank of unimodal C1 - C5	Average rank of multimodal C6 - C15	Average rank of all 15 functions
GSA-N20	13.0	12.2	12.5
G300-A07-B03-N20	6.0	4.6	5.1
G300-A07-B03-N20-C0.15	4.6	**2.3**	**3.1**
G300-A07-B03-N20-C0.50	5.8	3.2	4.1
G300-A07-B03-N20-C0.85	**3.6**	4.5	4.2
GSA-N40	12.0	11.6	11.7
G030-A07-Bli-N40	12.8	11.8	12.1
G030-A07-Bli-N40-C0.15	5.0	5.0	5.0
G030-A07-Bli-N40-C0.50	4.2	4.4	4.3
G030-A07-Bli-N40-C0.85	4.0	3.8	3.9
GSA-N60	12.2	11.3	11.6
G100-A07-Bli-N40	11.2	11.0	11.1
G100-A07-Bli-N40-C0.15	6.8	4.4	5.2
G100-A07-Bli-N40-C0.50	6.0	4.0	4.7
G100-A07-Bli-N40-C0.85	3.8	3.8	3.8

ascendingly by the mean of fitness values. Non-parametric Wilcoxon signed-rank test at 0.05 significant level is used for comparison between any two algorithms. Any algorithms with the same rank are those whose statistical comparison are not significant. It can be observed that running one algorithm using different NP(=40 and 80) does not make any statistical significance. The performance of DMGSA-85 and OABCDE are not statistically different; they both outperform DMGSA-15; and original GSA is defeated. That is DMGSA-85 $\approx$ OABCDE >> DMGSA-15 >> GSA, regardless of NP. GRN problem with this high dimension is multimodal in nature, and hence a higher value of CR surpasses a low CR in DMGSA. This is consistent with the results in experiment II.

Although performance of DMGSA-85 is not statistical different with OABCDE, DMGSA-85 provides the best result (lowest fitness) at 0.065, better than OABCDE. This is meaningful in real practice which emphasizes on the optimal solution achieved more than the average results, given enough time. All codes in the research are Java 8 running with NetBeans 8.2 on 64-bit Windows 10 and i7 CPU at 3.4 GHz with 8 GB of main memory. Running time of 20 runs for DMGSA, GSA and OABCDE are approximate 25, 25 and 20 minutes respectively using only one core. The time complexity of DMGSA and OABCDE will be discussed in the next subsection.

Fig 6 illustrates time series of the tested GRN. The left hand side is the actual time series from simulation and the right hand side is the time series produced from the best run of DMGSA-0.85. We can observe only minor difference between both figures.

5.3 Running Time Complexity

From Fig 4, the time complexity of velocity and position update of the proposed DMGSA is O(D), which is the same as of the original GSA. From Fig 1, the time complexity of Step 1 and Step 2 is O(NP), Step 2 is O(D^2), Step 4 and Step 5 is O(D). Since each agent performs all these steps within a generation, the time complexity for one generation of both GSA and DMGSA is equal to O($NP \cdot D^2$).

For OABCDE [24], each generation is composed of the following 3 actions: the employed bee's exploring, the onlooker bee's dancing and the opposition-based learning. Each of them has time complexity of O($NP \cdot D$), O($NP \cdot D$) and O($NP \cdot \log(NP)$) respectively. Therefore the time complexity for one generation is equal to O($NP \cdot D$) + O($NP \cdot \log(NP)$) = O($NP \cdot D$), since $NP \approx D$. Considering the time complexities of DMGSA and OABCDE, we can see why DMGSA (and GSA) took a longer running time than OABCDE in our GRN optimization in previous subsection.

6. CONCLUSION

The gravitational coefficient G and the number (K) of other best agents for calculating the gravitational force have impacts on balancing exploration and exploitation of GSA. The first experiment in this work studies the performance effected from varying G and K. The decay rate α of 7 for G is recommended from this study. The second experiment evaluates the performance of the proposed hybrid DMGSA algorithm. The results confirm the performance achieved from the proposed differential mutation with a high value of $G^0 = 300$ and the exponential decay of K with $\beta = 3$. The proposed differential mutation operator provides a fast convergence and thus is controlled with a new parameter CR. A high value of CR (like

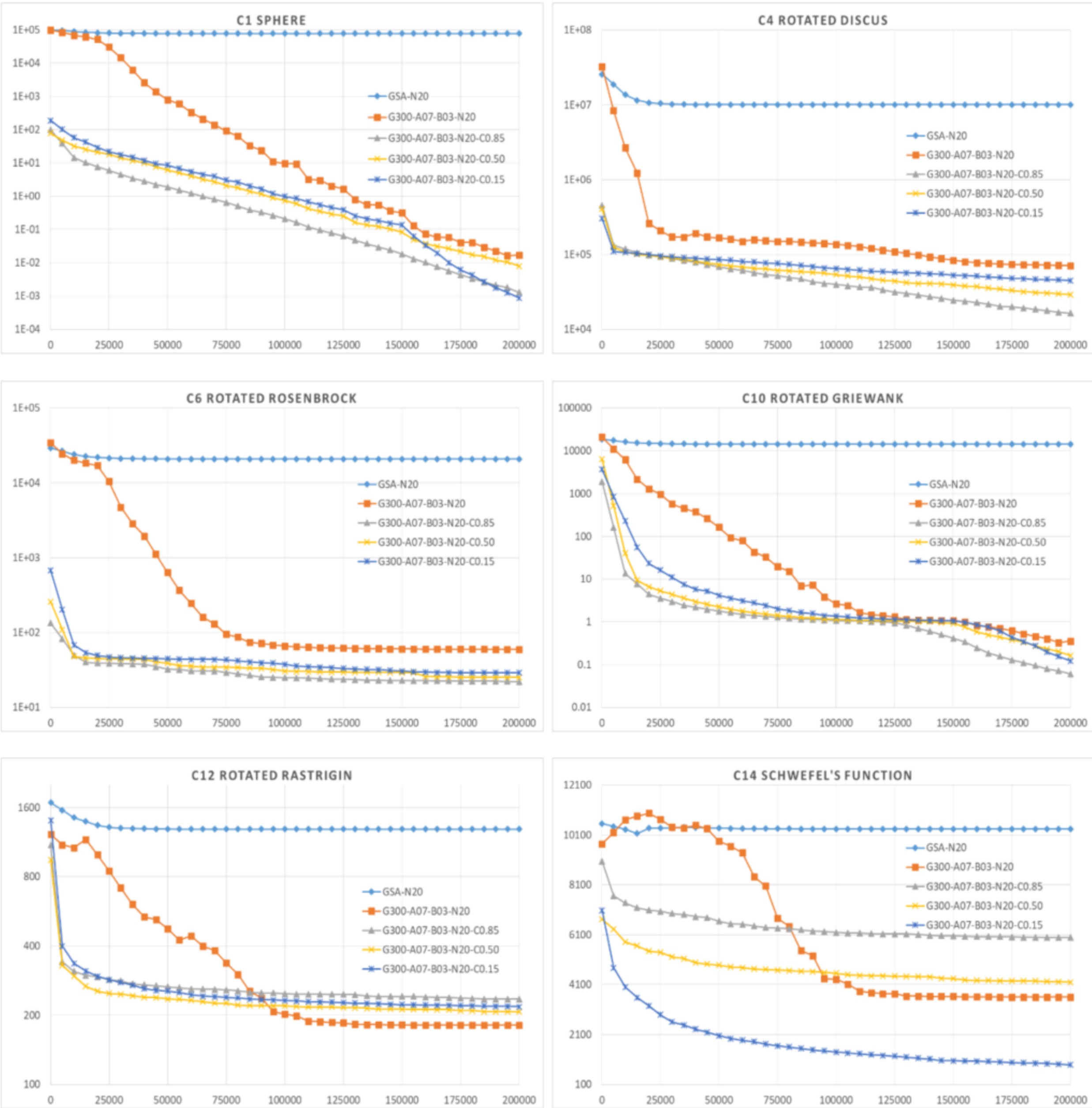

Fig.5: *Convergence graphs of some CEC2013 functions.*

Table 9: *S-System for network model.*

i	α	g_1	g_2	g_3	g_4	g_5	β	h_1	h_2	h_3	h_4	h_5
1	5			1		-1	10	2				
2	10	2					10		2			
3	10		-1				10		-1	2		
4	8			2		-1	10				2	
5	10				2		10					2

Table 10: *Statistical results of GRN optimization.*

Algorithm	NP	Mean	S.D.	Lowest	Highest	Statistical rank using Wilcoxon signed-rank test
DMGSA-85	40	0.077	0.026	0.065	0.167	1
OABCDE	40	0.085	0.008	0.071	0.100	1
OABCDE	80	0.089	0.009	0.076	0.107	1
DMGSA-85	80	0.094	0.056	0.065	0.290	1
DMGSA-15	40	0.183	0.059	0.100	0.386	5
DMGSA-15	80	0.269	0.051	0.165	0.393	5
GSA	40	0.799	0.783	0.074	2.130	7
GSA	80	0.929	0.426	0.117	1.661	7

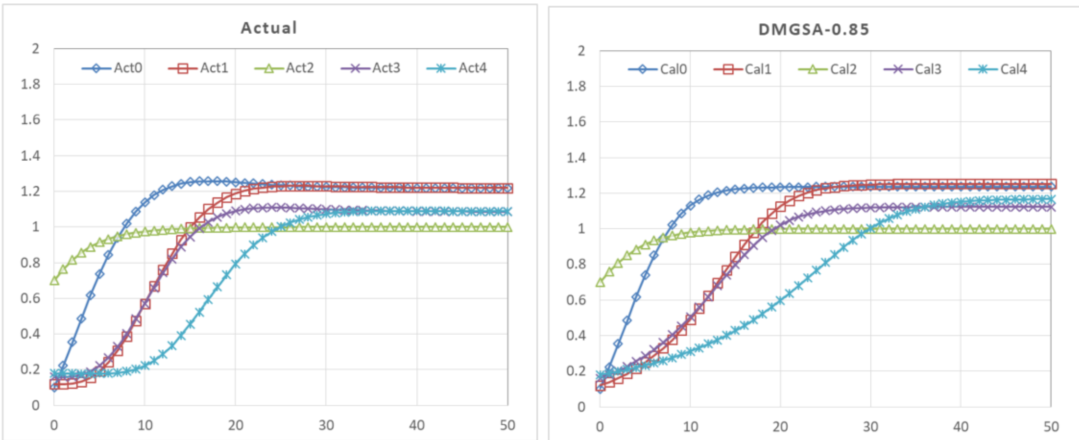

Fig.6: *Time series of GRN problem: actual (left) and reconstructed using DMGSA (right).*

0.85) is recommended for unimodal functions while a low value of CR (like 0.15) is for multimodal functions or the functions with unknown modality. Lastly DMGSA is applied to reconstruct a small-scale hypothetical GRN of 5-gene, becoming an optimization problem of 60 dimensions. The results confirm the performance of DMGSA. Possible directions for future works include testing with other differential mutation operators and applying the algorithm to more complex GRN problems and other real-world problems. In addition, it might be beneficial to compare the results with the optimization tools such as CPLEX or Gurobi.

References

[1] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, "GSA: a gravitational search algorithm," *Inf Sci*, vol.179, no.13, pp.2232-48, 2009.

[2] B. Shaw, V. Mukherjee, S.P. Ghoshal, "A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems," *International Journal of Electrical Power & Energy Systems*, vol.35, no.1, pp.21-33, 2012.

[3] M. Khatibinia, S. Khosravi, "A hybrid approach based on an improved gravitational search algorithm and orthogonal crossover for optimal shape design of concrete gravity dams," *Appl. Soft Comput. J.*, vol.16, pp.223-233, 2014.

[4] T. Ganesan, et al., "Swarm intelligence and gravitational search algorithm for multi-objective optimization of synthesis gas production," *Appl. Energy*, vol.103, pp.368-374, 2013.

[5] D.L.G. Alvarez, et al., "Comparing multiobjective swarm intelligence metaheuristics for DNA motif discovery," *Eng. Appl. Artif. Intell.*, vol.26, no.1, pp.314-326, 2013.

[6] T. Chakraborti, K. Das Sharma, A. Chatterjee, "A novel local extrema based gravitational search algorithm and its application in face recognition using one training image per class," *Eng. Appl. Artif. Intell.*, vol.34, pp.13-22, 2014.

[7] M. Yin, et al., "A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering," *Expert Syst. Appl.*, vol.38, no. 8, pp.9319-9324, 2011.

[8] X. Han, et al., "Feature subset selection by gravitational search algorithm optimization," *Inf. Sci.*, vol. 281, pp.128-146, 2014.

[9] R.-C. David, et al., "Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity," *Inf. Sci.*, vol. 247, pp. 154-173, 2013.

[10] E. Rashedi, E. Rashedi, H. Nezamabadi-pour,

"A comprehensive survey on gravitational search algorithm," *Swarm and Evolutionary Computation*, pp.1-18, 2017.

[11] G. Sun, A. Zhang, Z. Wang, Y. Yao, J. Ma, G.D. Couples, "Locally informed gravitational search algorithm," *Knowl Based Syst.*, vol.104, pp.134-44, 2016.

[12] R. Storn, K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

[13] S. Das, P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," in *IEEE Transactions on Evolutionary Computation*, vol.15, no.1, pp. 4-31, Feb. 2011.

[14] S. Sarafrazi, H. Nezamabadi-pour, S. Saryazdi, "Disruption: A new operator in gravitational search algorithm," *Scientia Iranica*, Volume 18, Issue 3, pp.539-548, 2011.

[15] M. Doraghinejad, H. Nezamabadi-pour, "Black hole: a new operator for gravitational search algorithm," *Int. J. Comput. Intell. Syst.*, vol.7, no.5, pp.809-826, 2014.

[16] X. Li, M. Yin and Z. Ma, "Hybrid differential evolution and gravitation search algorithm for unconstrained optimization," *Int' J. of the Physical Sciences*, vol.6, no.25, pp.5961-5981.

[17] Baoyong Yin, Zhaolu Guo, Zhengping Liang, Xuezhi Yue, "Improved gravitational search algorithm with crossover," *Computers & Electrical Engineering*, vol. 66, pp.505-516, 2018.

[18] J.J. Liang, B.Y. Qu, P.N. Suganthan, and A.G. Hernández-Daz, "Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization," Nanyang Technological University, Tech. Rep., 2013.

[19] J. Kennedy, R.C. Eberhart, "Particle swarm optimization," *Proc. of IEEE Int' Conf. on Neural Networks.*, Piscataway, NJ, 1995, pp. 1942-1948.

[20] W.-P. Lee, Y.-T. Hsiao, "Inferring gene regulatory networks using a hybrid GA-PSO approach with numerical constraints and network decomposition," *Information Sciences*, vol.188, pp.80-99, 2012.

[21] M.A. Savageau, *Biochemical Systems Analysis. A Study of Function and Design in Molecular Biology.* Addison-Wesley, 1976.

[22] D. Tominaga, N. Koga, and M. Okamoto, "Efficient Numerical Optimization Algorithm Based on Genetic Algorithm for Inverse Problem," *Proc. Genetic and Evolutionary Computation Conf.*, pp. 251-258, July 2000.

[23] N. Noman and H. Iba., " Inference of gene regulatory networks using s-system and differential evolution," in *Genetic and Evolutionary Computation Conference*, Washington, DC, pp.439-446, 2005.

[24] C. Worasucheep, "An opposition-based hybrid artificial bee colony with differential evolution," *2015 IEEE Congress on Evolutionary Computation (CEC)*, Sendai, pp. 2611-2618, 2015.

[25] K.-Y. Tsai, F.-S. Wang, "Evolutionary Optimization with Data Collocation for Reverse Engineering of Biological Networks," *Bioinformatics*, vol. 21, no. 7, pp.1180-1188, 2005.

Chukiat Worasucheep received the B.Eng. degree in Computer Engineering from Chulalongkorn University, Thailand in 1991, and M.Sc. degree in Computer Science from Oregon State University, USA, in 1996. He is currently an associate professor in Computer Science at Faculty of Science, King Mongkut University of Technology Thonburi, Thailand. His current research interests are automatic trading system, data analytics and computation intelligence techniques for financial applications.