Mass-Dispersed Gravitational Search Algorithm for Gene Regulatory Network Model Parameter Identification

Mohsen Davarynejad[1], Zary Forghany[2], and Jan van den Berg[1]

[1] Faculty of Technology, Policy and Management,
Delft University of Technology, the Netherlands
{m.davarynejad,j.vandenberg}@tudelft.nl
[2] Department of Molecular Cell Biology,
Leiden University Medical Center (LUMC), the Netherlands,
z.forghany@lumc.nl

Abstract. The interaction mechanisms at the molecular level that govern essential processes inside the cell are conventionally modeled by nonlinear dynamic systems of coupled differential equations. Our implementation adopts an S-system to capture the dynamics of the gene regulatory network (GRN) of interest. To identify a solution to inverse problem of GRN parameter identification the gravitational search algorithm (GSA) is adopted here. Contributions made in the present paper are twofold. Firstly the bias of GSA toward the center of the search space is reported. Secondly motivated by observed center-seeking (CS) bias of GSA, mass-dispersed gravitational search algorithm (mdGSA) is proposed here. Simulation results on a set of well-studied mathematical benchmark problems and two gene regulatory networks confirms that the proposed mdGSA is superior to the standard GSA, mainly duo to its reduced CS bias.

Keywords: Gravitational search algorithm, Center-seeking bias, Mass-dispersed gravitational search algorithm, Gene regulatory network model identification.

1 Introduction

Many diseases are the result of polygenic and pleiotropic effects controlled by multiple genes. Genome-wide interaction analysis (GWIA), by providing insight into the biological and biochemical pathways of disease is a natural evolution of single locus study.

The level of activation and inhibition of genes are governed by factors within a cellular environment and outside of the cell. The activation and inhibition relationship between genes are integrated by gene regulatory networks (GRNs), forming an organizational level in the cell with complex dynamics [4]. Mathematical modeling of GRNs provides a powerful tool not only to better understand such a complex system but also to develop new hypotheses on underlying mechanisms. Model parameter identification is a challenging optimization problem [21] with the objective function representing data misfit in a given norm. In this study, S-systems [22], a set of non-linear differential equations of a special form belonging to the power-law formalism are adopted as model. To estimate the model parameters and to capture the dynamics in gene expression data, Tominaga et al. [22] used standard evolutionary algorithms (EAs). Evolutionary computation is becoming a popular approach for solving S-system parameter identification [3, 10, 11, 15], mainly due to the multimodality and strong non-linear parameter-dependencies in the problem.

To the best of our knowledge, this is the first attempt to adopt the gravitational search algorithm (GSA) for S-system parameter optimization for GRN. This study provides a conceptual analysis of the search behavior of the GSA. The analysis suggests a center-seeking (CS) bias in the search process of the GSA confirmed by a setting up a test similar to one proposed by Angeline [1] on a set of several widely used numerical benchmark problems with various optimization characteristics. To partially dilute the search bias of GSA, a solution inspired from the Simulated Big Bounce [6] algorithm is proposed.

The remainder of this paper is organized as follows. A short introduction to GRNs and to S-systems is provided in Section 2. The population based model for S-system parameter identification is presented in Section 3. A brief tour of GSA followed by a conceptual analysis of its CS behavior are presented in Section 3.1. Motivated by the observed CS bias of the standard GSA, Section 3.2 presents the mdGSA, a solution borrowed from the Simulated big bounce (SBB) algorithm [6]. The experimental setup adopted to check the CS bias of GSA and mdGSA and their suitability for gene network parameter identification are presented in Section 4. The final section draws conclusions and considers implications for future research.

2 Gene regulatory Networks

GRNs in the cell are a complex dynamic network of interactions between the products of genes (mRNAs) and the proteins they produce, some of which in return act as regulators of the expression of other genes (or even their own gene) in the production of mRNA. While low cost methods to monitor gene expression through microarrays exist, we still know little about the complex interactions of these cellular components. Usually, sets of ordinary differential equations (ODEs) are used as mathematical models for these systems [23,24], however they suffer from many assumptions critical to the equations themselves. S-system approaches, on the other hand, use time-independent variables to model these processes. Assuming the concentration of N proteins, mRNAs, or small molecules at time t is given by $y_1^t, y_2^t, \ldots, y_i^t, \ldots, y_N^t$, S-systems model the temporal evolution of the ith component at time t by power-law functions of the form (1).

$$\frac{dy_i^t}{dt} = \alpha_i \left(\prod_{j=1}^{N} y_j^{g_{ij}} \right) - \beta_i \left(\prod_{j=1}^{N} y_j^{h_{ij}} \right) , \tag{1}$$

where N represents the number of genes involved in the GRN. The first term represents all factors that promote the expression of component i, y_i, whereas the second term represents all factors that inhibit its expression. In a biochemical engineering context, the non-negative parameters α_i , β_i are called rate constants, and real-valued exponents g_{ij} (G matrix, $[G]$) and h_{ij} (H matrix, $[H]$) are, respectively, referred to as kinetic order for synthesis and kinetic order for degradation.

$\mathbf{x} = \{\alpha, \beta, [G], [H]\}$ are the parameters that define the S-system. The total number of parameters in the S-system is $2N(N + 1)$, meaning the number of parameters increases quadratically and can quickly become very large. The parameter estimation task is to determine model parameters such that the dynamic profiles fit the observation.

3 Population based S-systems model parameter identification

Let us define the search space as the following:

$$E = \bigotimes_{d=1}^{D} [\mathrm{L}_x^d, \mathrm{U}_x^d], \tag{2}$$

with the objective of locating $\mathbf{x}^* \in E$, where $f(\mathbf{x}^*)$ is the extremum of a function $f(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$ and where L_x^d and U_x^d are respectively the lower and upper bound of the search domain at dimension d.

To guide the population in the search space, some measure of discrimination is needed. The most commonly used quality assessment criterion is the mean quadratic discrepancy between the observed expression pattern y_i^t and the model output $\hat{y}_i^t$ [16].

$$f = \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\frac{\hat{y}_i^t - y_i^t}{y_i^t} \right)^2 , \tag{3}$$

where T is the number of time points.

Having chosen an appropriate fitness function, the next section outlines the GSA.

3.1 A Brief Tour of the Gravitational Search Algorithm

Gravitational search algorithm (GSA) [18] is a relatively new technique that has been empirically shown to perform well on many function optimization problems [2, 8, 9, 12–14, 17, 19, 20]. GSA inspires from the evolution of complex structures in the universe. In its original version, GSA scatters particles in a feasible region of the search space where they interact with each other under Newtons gravitational force and move in the search area seeking optimal design variable. GSA shares features with several other competing schemes. Just like many of them, GSA has a way of sharing information between solutions. In contrast to EAs where solutions *die* at the end of each generation, in GSA, solutions survive through the course of the optimization process. This provides a substantial source of information for the population when searching the global optimum.

In GSA, just like many other population based optimization techniques, to guide the population in the search space E, some measure of discrimination is needed, referred here to as a fitness of each candidate solution $\mathbf{x}_i$. Each candidate solution is a particle with a mass M_i inversely proportional to its fitness $f(\mathbf{x}_i)$. A good solution is analogous to a particle with high mass and a poor solution represents a particle with a small mass. A particle with high mass resist change more than those with low mass and tend to have higher impact on other particles, thereby sharing their features with low quality solutions. The attractive gravitational force governs the movement of the particles in the search space. The search begins by an attractive force with a strength and direction as a function of the mass of particle itself, mass of other particles and its relative distance to the other particles. The force is applied to static particles of one under which their position in next time step changes and they gain a velocity. The quantity of the resultant force is determined by Newtons gravitational law. A solution with a higher mass exerts a stronger force compared to that of small mass. The kinetic energy stored in particles is a form of memory giving them the possibility to steer their movement under the influence of their memory and external forces. The sum of the force field and the particle's kinetic energy induced from its velocity and mass is the total force acting on them and together with its current position $\mathbf{x}_i(t)$ determines its next position $\mathbf{x}_i(t+1)$ in the search space.

GSA's basic steps in pseudo-code are shown in Algorithm (1). In original GSA [18], the mass of particles considering its quality is assignment as follows:

$$M_i = \frac{m_i}{\sum_{j=1}^{S} m_j}, i = 1, 2, \ldots, S \tag{4}$$

Algorithm 1 Pseudo code of gravitational search algorithm (GSA)

```
Input: Search space $E$, fitness function $f$, $S$, $G_0$, $\alpha$
Initialize particle's location, $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_S)^T$
while $t < MaxIteration$ do
    Fitness calculation
    Update $M_i$, $\forall i = 1, \ldots, S$                  ▷ According to (4) and (5)
    Update G                                         ▷ According to (8)
    Update attractive force $F_i^d$, $\forall i = 1, \ldots, S$
    Update $\mathbf{v}_i$, $\forall i = 1, \ldots, S$                  ▷ According to (9)
    Update $\mathbf{x}_i$, $\forall i = 1, \ldots, S$                  ▷ According to (10)
    $t{+}{+}$                                        ▷ $t$ is the number of iterations
end while
Output: $\mathbf{x}^*$ and $f(\mathbf{x}^*)$
```

where

$$m_i = \frac{f(\mathbf{x}_i) - \min_{j\in\{1,\ldots,S\}} f(\mathbf{x}_j)}{\max_{j\in\{1,\ldots,S\}} f(\mathbf{x}_j) - \min_{j\in\{1,\ldots,S\}} f(\mathbf{x}_j)}, \tag{5}$$

and S is the number of particles. The resultant gravitational force acting on particle i in direction d is determined using Equation (6).

$$F_i^d = \sum_{j\in Kbest} r_j F_{ij}^d, \tag{6}$$

where $Kbest$ is a set of k particles with the highest mass, $r_j \sim \mathcal{U}(0,1)$ and F_{ij}^d is gravitational force exerted by particle j on particle i. To provide a better exploration in the early iterations, $|Kbest|$ is set at S in the beginning; however the exploration must be decreased gradually. Therefore choosing a decremented function for $|Kbest|$ increases the exploitation of the algorithm when the number of iterations rises.

The force exerted by particle j acting on particle i is defined as:

$$F_{ij}^d = G\frac{M_i \times M_j}{R_{ij} + \varepsilon}\left(x_j^d - x_i^d\right) \tag{7}$$

where R_{ij} is Euclidian distance between particles i and j. and G, the gravitational constant initialized at G_0 is determined using Equation (8) as

$$G = G_0 e^{-\alpha\frac{t}{T}} \tag{8}$$

where α is algorithmic tuning parameter.

The equations of motion of every particle is described using (9) and (10) as

$$\mathbf{v}_i(t+1) = \mathbf{R} \times \mathbf{v}_i(t) + \frac{\mathbf{F}_i}{M_i}.\Delta t, \tag{9}$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1).\Delta t, \tag{10}$$

where $\Delta t = 1$, $\mathbf{R} \sim \mathcal{U}(0,1)$ is an array of size D corresponding to each element in vector $\mathbf{v}_i$.

To overcome limitations observed in performance of GSA, the next Section presents a modification of mass assignment procedure of standard GSA.

3.2 mdGSA, a mass-dispersed GSA

The CS bias of the standard GSA is a serious barrier to its use as an optimization tool when recognizing the fact that usually no real-world optimization problem has its optimal solution at exact center of the search space. A more intense discrimination of solutions may be a partial solution to this problem. Inspired from the Simulated big bounce algorithm [6], a Mass bounded to the range of $[L_M, U_M]$ is assigned to every particle considering the fitness of each particle. g, the function that maps the fitness to the Mass $g : \mathbb{R} \to \mathbb{R},\ f(\mathbf{x}_i) \mapsto g\left(f\left(\mathbf{x}_i\right)\right),\ \forall \mathbf{x}_i \in \mathbf{x}$ can be any monotonically nondecreasing (and possibly time varying) function in principle with real values defined on a the set of fitness of particle $\mathbf{x}_i$ whose value is non-negative for $f(\mathbf{x}_i)$. We take g as a linear time-invariant strictly increasing function as following [6]:

$$M_i = g\left(f\left(\mathbf{x}_i\right)\right) = L_M + \left(U_M - L_M\right) \frac{f(\mathbf{x}_i) - \max\limits_{j\in\{1,\ldots,S\}} f(\mathbf{x}_j)}{\min\limits_{j\in\{1,\ldots,S\}} f(\mathbf{x}_j) - \max\limits_{j\in\{1,\ldots,S\}} f(\mathbf{x}_j)}. \tag{11}$$

4 Experimental Setup and Results

When comparing the effectiveness of different optimization methods, a standard performance measure is the objective value a certain algorithm can reach within a certain predefined number of function evaluations. This is based on the assumption that the dominating factor in measuring computational effort is fitness evaluation, which is usually valid for complex optimization tasks of interest for real-world problems [5, 7]. This, in the experiments, is modeled as if the maximum computational resource budget available to devote a task is limited. This is equivalent as if the maximum time budget for which the best solution has to be delivered is limited.

4.1 Parameter Settings

In all the experiments adopted in this study, the population size is set at 50. The total number of fitness evaluation for mathematical optimization problems is set at 100,000 and for GRN model parameter estimation is set at 200,000. The used GSA parameters presented in [18] is as follows: G_0 is set at 100, α is set at 20, $Kbest$ is set at number of particles, S, and is linearly decreased to 1 at the last iteration. For the mdGSA the common setting are the same as GSA settings and the upper and lower bound of mass are set at 1, .01 respectively.

As the studied optimization techniques are stochastic in nature, for a given function of a given dimension 30 independent runs where executed for each experimental setup with average best-of-run and standard deviation of results being reported along with the results of Wilcoxon Rank Sum test for statistical test of significance.

4.2 Standard optimization problems

To asses the performance of the GSA, a set of four standard optimization problems, each with distinguishing characteristics posing different difficulties to the optimization algorithm, were selected. The problems selected are benchmark when comparing different optimization algorithms and are taken from prior studies [1]. This set of optimization problems, in their original from, are crafted to have the optima at or near the center of search space.

Some population-based optimization techniques suffer from a notable search bias. They tend to perform best when the optimum is located at or very near to

Table 1. Test problems. Adopted from [1].

Function	Search Space-1 (SS-1)	Search Space-2 (SS-2)
Sphere (F1)	$[-80,100]^D$	$[-20,100]^D$
Rosenbrock (F2)	$[-480,600]^D$	$[-120, 600]^D$
Rastrigin (F3)	$[-40, 50]^D$	$[-10,50]^D$
Griewank (F4)	$[-4.0,5.0]^D$	$[-1.0,5.0]^D$

the center of the search space. When developing a new optimization algorithm this makes the comparison unreliable. So in this study, to move the optimal solutions from the center, the search space is cut out from one side, 10% of the whole search space in the case of search space-1 (SS-1) and 40% in the case of search space-2 (SS-2). The test beds and their associated search spaces are listed in Table 1.

Table 2 compares the GSA and mass-dispersed GSA (mdGSA) algorithms on the set of four adopted mathematical benchmark problems under two different search space and when the dimension D is set at 50. Against all expectations, the performance of the GSA is found to deteriorate as a result of shrinking the search space. That may be explained by search bias. As result of shrinking the search space, the optimal solution moves from the center of the search space. When an algorithm has a center-seeking bias, this move of the location of the optimal solution deteriorates its performance.

In GSA, a change in number of particles changes the mass assigned to them as a result of increase in the denumerator of Equation (4). This increase in denumerator smoothes out the difference between the mass of particles, making them relatively equally the same in exerting attractive force and equally resistant to change in their position as a result of applied gravitational force. The swarm may be considered as one object with uniform mass distribution. Under the Newtonian gravitational force this brings the particles closer to the center resulting in an increase in density of swarm. As a result, they accelerate faster towards the center. This may explain the observed center-seeking behavior of standard GSA on a set of standard optimization problems.

To measure center-seeking bias (CSB) of an algorithm, one needs a benchmark and a criteria. A benchmark is when the search space is shrunk by S_L% of the original search space, and the criteria is deterioration of the fitness value when the search space is shrunk by S_U%. Considering the case of GSA on F4 test function as an example, a measure for CSB is $\text{CSB}^{S_L,S_U}(\text{GSA}, \text{F4}) = (411.27 - 7.42)/(S_U - S_L) = 13.46$, when the S_L and S_U are set at 10 and 40 receptively. A preferred algorithm is one with smallest CSB, among other criterions.

The observed search bias of GSA was indeed a motivation to borrow the mass assignment mechanism of the Simulated big bounce (SBB) algorithm [6] and propose the mdGSA to partially resolve the search bias. In all the four test beds, the $\text{CSB}^{10,40}$ of mdGSA is lower than that of GSA (Table 2).

Table 2. Statistical results of 30 runs obtained by GSA and mdGSA. Mean: Mean of the Best Values, StdDev: Standard Deviation of the Best Values, CSB_{10}^{40}: center-seeking bias when S_L and S_U are set at 10 and 40 respectively.

Function		GSA		$\mathrm{CSB}^{10,40}$	mdGSA		$\mathrm{CSB}^{10,40}$
		SS-1	SS-2		SS-1	SS-2	
Sphere	Mean	4.41E-17	4.23E-17	~ 0	5.35E-11	5.37E-11	~ 0
	StdDev	(1.21E-17)	(9.59E-18)		(4.70E-12)	(3.85E-12)	
Rastrigin	Mean	27.79	110.83	2.76	36.31	100.62	2.14
	StdDev	(5.43)	(12.57)		(7.68)	(11.23)	
Rosenbrock	Mean	70.53	208.69	4.60	44.45	76.18	1.06
	StdDev	(39.39)	(134.71)		(0.257)	(66.65)	
Griewank	Mean	7.42	411.27	13.46	8.21E-4	1.31E-3	1.63E-5
	StdDev	(2.28)	(17.81)		(3.13E-3)	4.25E-3	

4.3 GRN model parameter identification

To assess the performance of the methodologies studied here, two gene regulatory networks, NET1 and NET2, each consist of a network of two genes generated by the parameters given in Table 3 were adopted.

The gene expression levels of the networks are plotted in Figure 1 each consist of 50 time course of expression level per gene. The search space for α_i and β_i is limited to [20, 0.0] and for g_{ij} and h_{ij} to $[-4.0,\ 4.0]$.

The fitness transitions for different methodologies for NET1 and NET2 are plotted in a logarithmic scale in Figure 2. The Figures are average of 30 independent runs. In both NET1 and NET2, both the GSA and mdGSA start with a sharp fitness decrease in the beginning. GSA becomes almost stagnate after a short number of fitness evaluation. The mdGSA in both cases had much better progression compared to the GSA.

Preliminary analysis showed that neither the GSA nor the mdGSA produced normally distributed results under all settings. Consequently the GSA

Table 3. S-System Parameters for Network Model NET1 and NET2 adopted for model validation [22].

GRN	i	α_i	β_i	g_{i1}	g_{i2}	h_{i1}	g_{i2}
NET1	1	3	3	0	2.5	-1	0
	2	3	3	-2.5	0	0	2
NET2	1	3	3	0	-2.5	.1	0
	2	3	3	2.5	0	0	.1

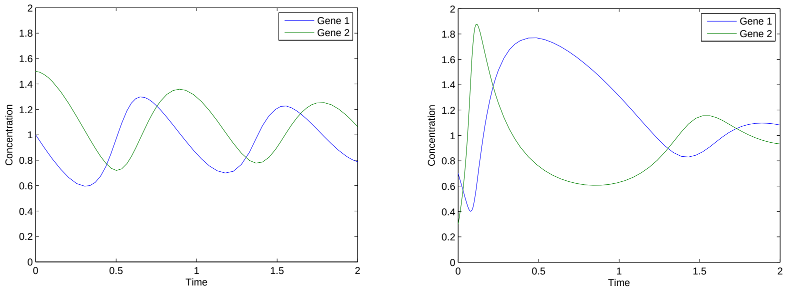

Fig. 1. Target time dynamics of first and second gene networks, NET1 and NET2.

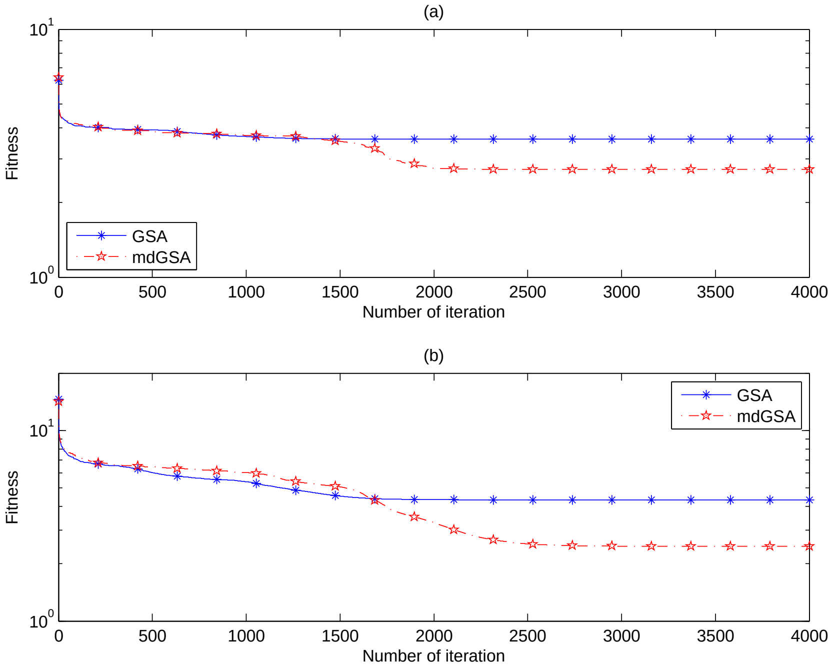

Fig. 2. Performance comparison of the GSA and mdGSA. (a) on NET1, (b) on NET2.

Table 4. A Wilcoxon Rank Sum test of the fitness of last generation for NET1 and NET2 (30 runs) obtained by GSA and mdGSA. Mean: Mean of the Best Values, Std-Dev: Standard Deviation of the Best Values.

		Simulation results		
GRN		Mean	StdDev	p-Value
NET1	GSA	3.60	0.37	2.14E-5
	mdGSA	2.39	1.30	-
NET2	GSA	4.32	1.61	3.32E-6
	mdGSA	2.43	0.87	-

and mdGSA was compared using the nonparametric Wilcoxon Rank Sum test to determine which one finds the lowest fitness values. The test, in contrast to t-test is solely based on the order in which the observations from the two samples fall. As presented in Table 4, the results of the proposed mdGSA are better than that of GSA when the standard cut-off for considering a p-value for a statistically significant difference is set at $p < 0.05$.

5 Conclusions and Future Work

In this paper, the GSA and its variant proposed in this paper, mdGSA, are employed for estimating genetic networks using S-system formalism. The performance of the mdGSA method, which enhances the global searching capability of GSA and alleviates its center-seeking bias, was verified using four standard benchmark problems and two networks. The experiments showed that the proposed method is capable to identify model parameters.

The center-seeking bias is a serious barrier to any optimization algorithm when recognizing the fact that no real-world optimization problem has its optimal solution at exact center of the search space. As part of our future work, we are interested in in-depth understanding the observed CS bias of the GSA.

Finally, we wish to apply the GSA and its variants algorithms to actual biological gene network and to conduct a comprehensive comparison against other popular population-based optimization algorithms.

References

1. P.J. Angeline. Using selection to improve particle swarm optimization. In *International Conference on Evolutionary Computation*, pages 84–89, 1998.
2. A. Chatterjee, G.K. Mahanti, and N.N. Pathak. Comparative performance of gravitational search algorithm and modified particle swarm optimization algorithm for synthesis of thinned scanned concentric ring array antenna. *Progress In Electromagnetics Research B*, 25:331–348, 2010.
3. A.R. Chowdhury and M. Chetty. An improved method to infer gene regulatory network using s-system. In *Congress of Evolutionary Computation, (CEC'11)*, pages 1012–1019, 2011.

4. A. Crombach and P. Hogeweg. Evolution of evolvability in gene regulatory networks. *PLoS computational biology*, 4(7):e1000112, 2008.
5. M. Davarynejad, C.W. Ahn, J.L.M. Vrancken, J. van den Berg, and C.A. Coello Coello. Evolutionary hidden information detection by granulation-based fitness approximation. *Applied Soft Computing*, 10(3):719–729, 2010.
6. M. Davarynejad and J. van den Berg. Simulated big bounce: a continuous space global optimizer. Technical report, Faculty of technology policy and management, Delft University of Technology, The Netherlands, 2012.
7. M. Davarynejad, J. Vrancken, J. van den Berg, and C.A. Coello Coello. A Fitness Granulation Approach for Large-Scale Structural Design Optimization. In Raymond Chiong, Thomas Weise, and Zbigniew Michalewicz, editors, *Variants of Evolutionary Algorithms for Real-World Applications*, pages 245–280. Springer-Verlag, Berlin, 2011.
8. S. Duman, U. Güvenç, Y. Sönmez, and N. Yörükeren. Optimal power flow using gravitational search algorithm. *Energy Conversion and Management*, 59:86–95, 2012.
9. S. Duman, U. Güvenç, and N. Yörükeren. Gravitational search algorithm for economic dispatch with valve-point effects. *International Review of Electrical Engineering (IREE)*, 5(6):2890–2895, 2010.
10. Z. Forghany, M. Davarynejad, and B.E. Snaar-Jagalska. Gene regulatory network model identification using artificial bee colony and swarm intelligence. In *IEEE Conference on Evolutionary Computation (CEC'12)*, pages 949–954, 2012.
11. S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita. Dynamic modeling of genetic networks using genetic algorithm and s-system. *Bioinformatics*, 19(5):643–650, 2003.
12. C. Li and J. Zhou. Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm. *Energy Conversion and Management*, 52(1):374–381, 2011.
13. C. Li, J. Zhou, J. Xiao, and H. Xiao. Parameters identification of chaotic system by chaotic gravitational search algorithm. *Chaos, Solitons & Fractals*, 45(4):539–547, 2012.
14. C. Lopez-Molina, H. Bustince, J. Fernandez, P. Couto, and B. De Baets. A gravitational approach to edge detection based on triangular norms. *Pattern Recognition*, 43(11):3730–3741, 2010.
15. T. Nakayama, S. Seno, Y. Takenaka, and H. Matsuda. Inference of s-system models of gene regulatory networks using immune algorithm. *Journal of Bioinformatics and Computational Biology*, 9:75–86, 2011.
16. N. Noman and H. Iba. Inference of gene regulatory networks using s-system and differential evolution. In *Genetic and Evolutionary Computation Conference, Washington, DC*, pages 439–446, 2005.
17. R.E. Precup, R.C. David, E. Petriu, S. Preitl, and A. Paul. Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity. *Soft Computing in Industrial Applications*, pages 141–150, 2011.
18. E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi. Gsa: a gravitational search algorithm. *Information Sciences*, 179(13):2232–2248, 2009.
19. E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi. Filter modeling using gravitational search algorithm. *Engineering Applications of Artificial Intelligence*, 24(1):117–122, 2011.
20. B. Shaw, V. Mukherjee, and 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*, 35(1):21–33, 2012.

21. J. Sun, J.M. Garibaldi, and C. Hodgman. Parameter estimation using metaheuristics in systems biology: A comprehensive review. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(1):185–202, 2012.
22. D. Tominaga, M. Okamoto, Y. Maki, S. Watanabe, and Y. Eguchi. Nonlinear numerical optimization technique based on a genetic algorithm for inverse problems: Towards the inference of genetic networks. In *German Conference on Bioinformatics Computer Science and Biology*, pages 127–140, 1999.
23. K.Y. Tsai and F.S. Wang. Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics*, 21(7):1180, 2005.
24. M. Vilela, I.C. Chou, S. Vinga, A.T. Vasconcelos, E.O. Voit, and J.S. Almeida. Parameter optimization in s-system models. *BMC systems biology*, 16(2):35, 2008.