Applied Soft Computing 47 (2016) 103–118

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A memory-based gravitational search algorithm for enhancing minimum variance distortionless response beamforming

Soodabeh Darzi[a,*], Tiong Sieh Kiong[b], Mohammad Tariqul Islam[c], Hassan Rezai Soleymanpour[d], Salehin Kibria[c]

[a] Center for Space Science, Universiti Kebangsaan Malaysia, Malaysia
[b] Center of System and Machine Intelligence, College of Engineering, Universiti Tenaga Nasional, Malaysia
[c] Department of Electrical, Electronic & Systems Engineering, Universiti Kebangsaan Malaysia, Malaysia
[d] Department of Electrical Engineering, Semnan University, Semnan, Iran

ARTICLE INFO

Article history:
Received 27 May 2014
Received in revised form 26 May 2016
Accepted 30 May 2016
Available online 3 June 2016

Keywords:
Gravitational search algorithm
Minimum variance distortionless response
Adaptive beamforming
Particle Swarm Optimization
Heuristic algorithm
Artificial intelligence

ABSTRACT

This paper introduces a memory-based version of gravitational search algorithm (MBGSA) to improve the beamforming performance by preventing loss of optimal trajectory. The conventional gravitational search algorithm (GSA) is a memory-less heuristic optimization algorithm based on Newton's laws of gravitation. Therefore, the positions of agents only depend on the optimal solutions of previous iteration. In GSA, there is always a chance to lose optimal trajectory because of not utilizing the best solution from previous iterations of the optimization process. This drawback reduces the performance of GSA when dealing with complicated optimization problems. However, the MBGSA uses the overall best solution of the agents from previous iterations in the calculation of agents' positions. Consequently, the agents try to improve their positions by always searching around overall best solutions. The performance of the MBGSA is evaluated by solving fourteen standard benchmark optimization problems and the results are compared with GSA and modified GSA (MGSA). It is also applied to adaptive beamforming problems to improve the weight vectors computed by Minimum Variance Distortionless Response (MVDR) algorithm as a real world optimization problem. The proposed algorithm demonstrates high performance of convergence compared to GSA and Particle Swarm Optimization (PSO).

1. Introduction

Since the concept of adaptive arrays technique comes into usage in aerospace and military applications via the employment of electronically steered antennas, adaptive beamforming has drawn significant attention in various applications. They include active jammer rejection, long-range surveillance radar, multi-beam antennas and direction finding for space communications and many other areas because of its attractive features such as high data transmission rate, strong multipath fading resolution and high co-channel interference resistance [1].

One of the famous adaptive array beamforming techniques, known as Minimum-Variance Distortionless Response (MVDR), was created by Capon in 1969 [2]. The advantage of this technique is it requires only the knowledge of the Direction of Arrival (DOA) to maximize the Signal to Noise Ratio (*SNR*) [3] through enhancement of the directivity in one direction only [4]. According to the characteristics of this technique, weights computed by MVDR are not able to form the null towards the interference source satisfactorily. Solving this problem through conventional empirical approach is very difficult, time consuming, and sometimes, in real-time application, is unmanageable. Therefore, some approaches have proposed the use of mathematical models to improve the robustness of the MVDR beamforming. However, the tradeoff between MVDR performance and computational complexity still exists [4–6]. Consequently, many meta-heuristics and exploratory methods get best results for these types of difficulties. Recently, the employment of meta-heuristics algorithm has been growing instead of exhaustive and exact procedures in applications. Approaches such as Artificial Immune System (AIS) [6], Particle Swarm Optimization (PSO) [7,8], Artificial Bee Colony (ABC) [9,10], Genetic Algorithms (GA) [11,12], Ant Colony Optimization (ACO) [13–15], Tabu Search (TS) [9,16,17], Differential Evolution (DE) [18] and Clonal Selection (CS) [19,20] have been used to improve various aspects of antenna systems.

Recently, gravitational search algorithm (GSA) was presented as a heuristic optimization algorithm inspired by Newtonian laws

* Corresponding author.
E-mail address: sdtutn@yahoo.com (S. Darzi).

http://dx.doi.org/10.1016/j.asoc.2016.05.045
1568-4946/

of gravitation. It was shown to outperform similar algorithms like PSO and GA for common benchmark functions [21]. Since the inception of GSA, a large number of researchers have introduced various modified and improved gravitational search algorithms based on the idea of memory, social information of PSO and novel strategies to define the agents' search pattern along with other stochastic strategies [22–27]. Adaptive velocity constraint in GSA [27] uses an adaptive maximum velocity constraint, which aims to control the global exploration ability of the original algorithm while increasing its convergence rate. Establishing a limitation on maximum velocity increases the stability of most algorithms and ensures convergence. However, it is detrimental to the exploration ability of an algorithm in the early phase. Thus adaptive velocity constraints were introduced in literature to ensure stability while not sacrificing early exploration.

To enhance particle memory ability in GSA, the tactic of storing previous local and global optimum solution from PSO was introduced to GSA in literature [22]. It is simply a combination of PSO with the standard GSA, in which each particle can remember its own local optimum solution (*Pbest*) and global optimum solution (*Gbest*). During the velocity update calculation of GSA, the memory of local and global best is incorporated by adding a randomly weighted vector towards each best location, just like PSO [22]. However, the presented method has some drawbacks that reduce efficiency; for example, introducing some new adjustable parameters to the algorithm, which complicate the algorithm further.

In this study, we propose a memory based version of GSA (MBGSA) to deal with complicated optimization problems like the MVDR. MBGSA is introduced and applied to maximize the Signal to interference-noise ratio (*SINR*) in MVDR beamforming technique. The proposed MBGSA utilizes the overall best solution of populations in calculation of agent positions unlike the GSA, which uses only the best solution of previous iteration. Thus, when a swarm clusters together at local optimum, the overall personal best ('*pbest*') values of the agents recorded from the earlier exploration helps to determine the acceleration parameter, which subsequently prevent the agents from locking inside the local optimum and further enable the agents to follow the gradient of entire fitness function. Thus, MBGSA is far less likely to stagnate in local optimum in the search space and prevents the loss of optimal search trajectory, hence improves the convergence performance of the proposed algorithm. MBGSA has been applied in beamforming problem with uniform linear antenna arrays at 0.5λ spacing between adjacent elements and radiating at a frequency of 2.3 GHz in this study. Fig. 1 illustrates the framework utilized to develop the proposed algorithm in this paper.

The rest of this paper is organized as follows: Section 2 introduces the brief evaluation of GSA. The proposed MBGSA is presented in Section 3. Section 4 introduces the basics of adaptive beamforming and the conventional MVDR. The benchmark functions utilized to verify the performance of the proposed algorithm are presented in Section 5. In Sections 6, results of optimizing the benchmark functions with the proposed algorithm are compared with results for GSA and its variants from literature. Simulation results of different interferences and comparison of conventional MVDR with MVDR-PSO, MVDR-GSA and MVDR-MBGSA are also reported in this section. Finally, Section 7 concludes this investigation.

2. Gravitational search algorithm (GSA)

Gravitational search algorithm (GSA) is one of the recent search algorithms for heuristic population based on the mass interactions and law of gravity [21]. This approach is employed as an artificial world of masses following the gravitation and Newtonian laws of motion. All the GSA agents are employed as objects that will be evaluated through their masses. These objects collectively move toward the objects with heavier masses iteratively using Eq. (1) due to higher gravitational force. Eq. (2) is evaluated to calculate the velocity of an object, and then it is used in Eq. (1). Thus, best solutions of the problems are represented by the heavier masses. The new position and velocity of *i*-th agent along dimension *d* will be improved based on the following equations:

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{1}$$

$$v_i^d(t+1) = \text{rand}_i \times v_i^d(t) + a_i^d(t) \tag{2}$$

where v_i^d is velocity of *i*-th agent in dimension *d*, rand_i employs a random variable to offer a randomized characteristic for the search space, $x_i^d(t)$ is position of *i*-th agent in dimension *d*, at iteration number *t*. The acceleration of agent *i* in dimension *d*, a_i^d, is calculated as below in Eq. (3):

$$a_i^d(t) = \sum_{j \in kbest, j \neq i} \text{rand}_j G(t) \frac{M_j(t)}{R_{i,j}(t) + \varepsilon} \left(x_j^d(t) - x_i^d(t) \right) \tag{3}$$

where the acceleration of *i*-th agent in dimension *d* is a_i^d, *kbest* is a function that is able to control the performance of GSA. The role of this *kbest* is to control the exploitation and exploration of GSA technique. In order to avoid local optimum stagnation, the algorithm must use the exploration at beginning. However, in later iterations, exploration must fade out and exploitation must increase. To improve the performance of GSA, by controlling exploration and exploitation, only the certain number of agents will attract the other particles and this number of agents is determined by *kbest*. *kbest* is initialized as the number equal to the total number of agents and linearly decreased to one at the end of the optimization process. Thus, at the beginning, all agents apply the force and, at the end, only one agent applies force to the others [21]. The Euclidean distance between the *i*th and *j*th agent in Eq. (3), $R_{i,j}(t)$, is evaluated according to Eq. (4). The gravitational constant at time *t* is $G(t)$ according to Eq. (5); rand_j is random value; M_j is mass of *j*-th agent that is presented in Eq. (6); ε is a small zero offset constant.

$$R_{i,j}(t) = \| X_i(t), X_j(t) \|_2 \tag{4}$$

where X_i and X_j are the locations of agents *i* and *j* respectively and the $R_{i,j}(t)$ is the Euclidean distance between them. This study employs *R* as a replacement of R^2 in Eq. (3) due to the recommendations offered in Ref. [21]. The $G(t)$, in Eq. (3), is an time-varying function that is set to G_0 at the starting and will decline exponentially to control the convergence rate of the algorithm as shown below in Eq. (5):

$$G(t) = G_0 \times \exp\left(-\beta \times {}^{t}\!/_{t_{\max}} \right) \tag{5}$$

where *t* is the current iteration, β is a gradient constant value and $t_{\max}$ is the maximum iteration number. Moreover, the normalized mass of agents that is mentioned in Eq. (3) can be obtained by Eq. (6). Eq. (7) is used to determine an objects Netwonian mass based on the fitness of the object at the position $x_i(t)$ from Eq. (1), or $\text{fit}_i(t)$, as mentioned initially.

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{6}$$

In which,

$$m_i(t) = \left\{ \begin{array}{ll} \dfrac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} & \text{if best } (t) \neq \text{worst} \\ 1 & \text{otherwise} \end{array} \right\} \tag{7}$$