A binary variant of gravitational search algorithm and its application to windfarm layout optimization problem

Susheel Kumar Joshi · Jagdish Chand Bansal

Received: date / Accepted: date

Abstract In the binary search space, GSA framework encounters the shortcomings of stagnation, diversity loss, premature convergence and high time complexity. To address these issues, a novel binary variant of GSA called ‘A novel neighbourhood archives embedded gravitational constant in GSA for binary search space (BNAGGSA)’ is proposed in this paper. In BNAGGSA, the novel fitness-distance based social interaction strategy produces a self-adaptive step size mechanism through which the agent moves towards the optimal direction with the optimal step size, as per its current search requirement. The performance of the proposed algorithm is compared with the two binary variants of GSA over 23 well-known benchmark test problems. The experimental results and statistical analyses prove the supremacy of BNAGGSA over the compared algorithms. Furthermore, to check the applicability of the proposed algorithm in solving real-world applications, a windfarm layout optimization problem is considered. Two case studies with two different wind data sets of two different wind sites is considered for experiments.

Keywords Neighbourhood archive · Binary Gravitational Search Algorithm (GSA) · Gravitational Constant · Meta-heuristics · Windfarm layout optimization problem

1 Introduction

In recent years, meta-heuristic algorithms has been emerged extensively for solving complex and rigid optimization problems of both continuous and binary search space. Transfer function is one of the basic prerequisite for converting the mechanism of an algorithm from continuous search space to binary one. Under this conversion, the search functionalities of an algorithm remain preserve. Therefore the shortcomings of an algorithm in continuous search space effect its performance in binary search space, very badly. Gravitational search algorithm (GSA) [51] is a very popular and
robust optimizer for both continuous and binary search space. Rashedi et al. [32] introduced the first version of binary gravitational search algorithm (BGSA) by using the V shaped transfer function. Due to the flaws of its continuous counterpart, BGSA also suffers from the issues of stagnation occurrence and the slow convergence rate. To overcome these issues and develop more efficient and robust BGSA variants, several studies have been conducted so far. In this context, Mirjalili et al. [24] proposed a self adaptive parameter scheme for a hybrid binary model associated with particle swarm optimization (PSO) and GSA. Yuan et al. [35] proposed a local mutation strategy in BGSA for unit commitment (UC) problem. In [30], a novel transfer function is introduced to improve the efficiency of BGSA for feature subset selection. In similar study, Chakraborti et al. [7] proposed a self-adaptive weight strategy in BGSA for feature selection. In another study [6], a variant of BGSA is proposed by incorporating a mutation information scheme for feature selection. Khanesar et al. [21] proposed a XOR attached GSA variant for binary search space. Somu et al. [33] introduced a newton-raphson inspired BGSA for QoS value prediction and service ranking prediction techniques. Guha et al. [17] improved exploration ability of BGSA by introducing a clustering approach for initial population distribution for feature selection problem. Han et al. [18] proposed a recursive BGSA for two-phase feature selection method for cancer classification. Thakur et al. [34] solved a multi-criteria scheduling problem by proposing a quantum-inspired BGSA.

Windfarm layout optimization problem is one of the most popular optimization problem in the field of renewable energy. The objective of this problem is to find the optimal layout of wind turbines in a windfarm for which the total power output and windfarm efficiency both should be maximized. So far, several studies have been conducted to optimize different objectives of the windfarm through different meta-heuristic algorithms. To minimize the cost per unit power output of a windfarm, Mosetti et al. [25] proposed a binary coded genetic algorithm (GA). Grady et al. [10] improved the results of the same approach by more simulations. To improve the total power output of the windfarm, Mittal [25] introduced the fine grid spacing in windfarm. Emami et al. [12] optimized both cost and power of the windfarm using weighted aggregate method of multi-objective optimization. Marmidis et al. [23] introduced Monte Carlo approximation approach in the single objective platform to study a simple wind condition. Pookpunt et al. [28] used the binary variant of particle swarm optimization to solve the windfarm layout optimization problem. To solve the same problem, [5] proposed a variant of differential evolution using linear population size reduction approach. Ant colony optimization (ACO) algorithm [13] is also used to solve the windfarm layout optimization problem. Literatures [26] [13] studied a basic single objection framework in which the cost function of a windfarm is optimized subject to the number of turbines. In [8], the authors used the Greedy algorithm to optimize the layout. In a similar study, [9] used the same greedy approach for multiple hub heights of wind turbines. In [15], evolutionary algorithm is used to optimize the positions of wind turbines for a windfarm to obtain the minimum cost model. To optimize the wind farm layout and wind turbine size, [11] proposed an advanced modeling system using a multi-level extended pattern search algorithm. In another study, Bansal et al. [1] used the biogeography based optimization (BBO) for solving the windfarm layout problem. Biswas et al. [3] used differential evolution to perform the case studies for maximizing the windfarm efficiency with different rotor diameters and hub heights. To optimize the windfarm layout, a lot of studies has been proposed by the researchers in the multi-objective framework also. In this context, Chen et al. [10] used Multi-objective genetic algorithm (MOGA) by utilizing one Pareto solution in the optimization process. In another study, [14] optimized the windfarm layout and the number of turbines both simultaneously by using multi-objective random search algorithm.
In a recent study [4], authors proposed an approach to solve the windfarm layout optimization problem using an advanced multi-objective evolutionary algorithm based on decomposition (MOEA/D) [36,37]. In this approach, a multi-objective optimization problem is decomposed into several scalar optimization subproblems which further optimized simultaneously. Due to its ability to optimize each subproblem using information from its neighboring subproblems, MOEA/D have less computational complexity than multi-objective genetic local search (MOGLS) and non-dominated sorting genetic algorithm-II (NSGA-II). In [4], authors used MOEA/D to solve windfarm layout optimization problem for maximizing the two objectives, power output and windfarm efficiency simultaneously. With this setup, several Pareto optimal solutions can be obtained by a single run of MOEA/D. The wind condition in this study considers variable wind speed and direction for two different sites [4].

This paper proposes a novel BGSA variant which is tested over different search domains of uni-modal, multi-modal and multi-modal with fix dimensions landscapes. Furthermore, the multi-objective windfarm layout optimization problem described in [4] is effectively solved by the proposed variant under the priori approach. The contribution of this paper can be summarized as follows:

- To improve the search objectives, a novel social interaction scheme is proposed. For this, two neighborhood archives are introduced for each candidate solution regarding its current position (F archive) and its distance (D archive) from the most promising regions of the landscape. These archives provide the most appropriate neighbours to each agent in terms of its current search requirements.
- A parameter-free, fitness-distance ratio based gravitational constant is proposed for a better trade-off between exploration and exploitation of the algorithm.
- The binary variant integrated with the above mentioned novelties under the priori approach outperforms the posteriori approach of MOEA/D for the considered multi-objective windfarm layout problem.

Rest of the paper is organized as follows. Section 2 briefly describes the frameworks of GSA. In Section 3, a detailed introduction of the proposed is given for continuous search space. Section 4 describes the frameworks of proposed BNAGGSA and evaluates its performance over binary search space. The considered windfarm layout problem is described in Section 5. In section 6 proposed BNAGGSA algorithm is applied to solve the considered windfarm layout optimization problem along with the experimental settings and simulation results. Finally, Section 7 concludes the paper.

### 2 Basic Gravitational Search Algorithm

In the recent years, Gravitational Search Algorithm (GSA) [31] has been widely used meta-heuristic in different domains of science, engineering and research. It cleverly mimics the law of gravity and mass interaction and produces a robust search mechanism as follows:

Initially, a swarm of \( N \) particles is randomly generated within the predefined search space \( S \) having \( n \) dimensions. Each particle \( (X_i, i = 1, 2, ..., N) \) of the generated swarm is defined as:

\[
X_i = (x_{i1}^1, ..., x_{id}^d, ..., x_{in}^n), \quad \forall i = 1, 2, ..., N
\]  

After obtaining the fitness values \( (fit_i(t), i = 1, 2, ..., N) \) at a specific iteration \( t \), the mass \( M_i(t) \) of each \( X_i \) is calculated as:
Here best(t) and Worst(t) are the best and worst fitness of the current swarm, respectively.

The acceleration $a_d^i(t)$ of $X_i$ in $d^{th}$ dimension is defined as:

$$a_d^i(t) = \frac{F_d^i(t)}{M_i(t)}$$ (4)

Here, $F_d^i(t)$ is the total force acting on $X_i$ by a set named $K_{best}$ archive having $K$ best fit particles of the current swarm. $F_d^i(t)$ is calculated as:

$$F_d^i(t) = \sum_{j \in K_{best}, j \neq i} \text{rand}_j \times F_{ij}^d(t)$$ (5)

Here, the $K_{best}$ set is solely responsible for the social interaction in GSA framework. Its size reduces from $K$ to 1 iteratively. $\text{rand}_j$ is a uniform random number from the interval $[0, 1]$. At a specific iteration $t$, $F_{ij}^d(t)$ is the force between $i^{th}$ and $j^{th}$ particles in the $d^{th}$ dimension and calculated as:

$$F_{ij}^d(t) = G(t)\frac{M_i(t)M_j(t)}{R_{ij} + \epsilon}(x_i^d(t) - x_j^d(t))$$ (6)

Here, $G(t)$, a scaling factor called gravitational constant is an exponential deceasing function defined as:

$$G(t) = G_0 e^{-\alpha t}$$ (7)

$G_0$ and $\alpha$ are fixed values set to 100 and 20, respectively. $T$ is the maximum number of iterations. $R_{ij}(t)$ is distance between particles $X_i$ and $X_j$. $\epsilon$ is a very small positive value. Now, $a_d^i(t)$, the acceleration $a_d^i(t)$ of $X_i$ in $d^{th}$ dimension is defined as:

$$a_d^i(t) = \sum_{j \in K_{best}, j \neq i} \text{rand}_j G(t)\frac{M_i(t)M_j(t)}{R_{ij} + \epsilon}(x_i^d(t) - x_j^d(t)),$$ (8)

$d = 1, 2, ..., n$ and $i = 1, 2, ..., N$.

The acceleration $a_d^i(t)$ helps to update the position of $X_i$ through the following bootstrapping equations:

$$v_i^d(t + 1) = rand_i \times v_i^d(t) + a_i^d(t)$$ (9)

$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1)$$ (10)

Here, $rand_i$ is uniformly selected random number from the interval $[0, 1]$. $v_i^d(t)$ and $x_i^d(t)$ are the velocity and position of $X_i$ in $d^{th}$ dimension, respectively.
3 The proposed GSA variant for continuous search space

In the basic GSA framework, $K_{best}$ set or archive is the neighbourhood structure through which the particles interact. This structure provides the same $K$ sub-optimal regions of the fitness landscape to each particle for a significant search towards optimality. However, this structure ignores the other valuable informations associated with the current position of the particle. This demerit of $K_{best}$ archive makes GSA search mechanism vulnerable in terms of diversity control. The gravitational constant $G$, on the other hand, is another entity of GSA model which entirely responsible for scaling the step sizes of its particles. Although $G$ works significantly well due to its exponential decreasing property, it also does not consider the particle’s individual requirements and scales the next moves of all the particles in the same fashion. This shortcoming produces inappropriate step sizes which further causes either stagnation or sometimes bypassing the true optima.

To overcome these mentioned issues of $K_{best}$ archive and $G$, authors have proposed a novel variant of GSA, namely ‘A novel neighbourhood archives embedded gravitational constant in GSA (NAGGSA)’ [20]. In NAGGSA, first, two novel neighbourhood archives are proposed for each particle to consider its other informative characteristics as per its current position. Secondly, a novel scaling mechanism is proposed which scales the next move of a particle according to its current search requirements. To check its performance in the continuous search space, authors have done all the necessary experiments and analyses through which it proves its efficiency in the continuous search space. The detailed description about the mechanism of proposed NAGGSA can be found in [20].

4 GSA and NAGGSA in the binary search space

In the binary search space, particle updates itself through switching between 0 and 1 values. In this study, the mentioned switching is done by a well-known sigmoid function which defines a least probability of particle’s movement for small ($V_i$) while for a large ($V_i$), it provides a high probability for particle’s movement [32]. The sigmoid function is defined as:

$$S(V_i) = |tanh(V_i)|$$

(11)

After calculating $S(V_i)$, particle’s movement is decided by the following rules:

$$if \ \ rand < S(V_i) \Rightarrow X_i^{d}(t+1) = complement(X_i^{d}(t))$$

$$else \ X_i^{d}(t+1) = X_i^{d}(t)$$

(12)

Here, rand is the normally distributed random number between 0 and 1. In the binary search space, the distance between two particles (considered as strings) is measured by the hamming distance which count the minimum number of substitutions required to coincide both the particles (both strings). Additionally, for a better convergence, the particles velocity is bounded in $[-6, 6]$. Both GSA and NAGGSA follow the above procedure to convert their self into BGSA and BNAGGSA, respectively. To understand the overall search mechanism of the proposed BNAGGSA graphically, the first particle $X_1$ of the swarm is considered through its different attributes. Figures 1, 2 and 3 illustrate these graphical analyses for $f_1$ (unimodal function), $f_8$ (multimodal function) and $f_{23}$ (multimodal function with fixed dimension) (refer section 4.1). Each mentioned figures have four subfigures from (a) to (d). Subfigure (a) presents the number of neighbours $X_1$ gets by
Fig. 1 Graphical analyses of first particle $X_1$ through its different attributes over $f_1$ (uni-modal function $f_1$)

the proposed archives (either $F$ or $D$) all over the search. Each subfigure (a) also indicates that a particle can have 2 to 5 neighbours in each iteration. Subfigure (b) and Subgraph (c) illustrate the behaviour of the mean distance and the proposed $FDG_{1, \text{Neigh}}(t)$ of $X_1$ from its assigned neighbours, respectively. For a better comparison, gravitational constant $G$ of BGSA (mentioned in section 4.2) is also considered in the Subfigure (c). Subfigure (d) is just a magnified version of subfigure (c).

4.1 Result and Discussion

4.1.1 Test problems under consideration

In order to validate the performance of the proposed binary variant BNAGGSA, a set of well known continuous benchmark problems is considered and listed in Table 1. In the Table 1 $C$ represents the topological characteristics of the test problem through which the test problems are categorized into three groups: unimodal problem ($U$), multimodal problem ($M$) and multimodal problem with fix dimension ($MFD$). $m$ is the dimension of the test problem, $f_{\text{min}}$ is the optimal value of the test problem and $S$ is the subset of $\mathbb{R}^m$. As the continuous benchmark problems (refer Table 1) are considered to evaluate the performance of the proposed binary variant. It is required to have a strategy through which the binary representation can convert to real values. For this, first, a swarm
having $N$ particles is generated in such a way that its each particle can represent as a vector of $m$ bit strings ($m$ is the dimension of the test problems of Table 1). Each bit have length $L$ which is set to 15 here. In this study, the following simple strategy is used to convert each bit string into an integer:

$$X_{integer} = \sum_{i=0}^{L} (X_i \times 2^i)$$  \hspace{1cm} (13)

Here, $X_i$ is the $i^{th}$ bit of $X$ in its binary format. The obtained integer is further converted into its corresponding real value within the continuous interval $[a, b]$ as follows:

$$X_{Real} = a + X_{integer} \times \frac{(b-a)}{2^L}$$  \hspace{1cm} (14)
Fig. 3 Graphical analyses of first particle $X_1$ through its different attributes over $f_{23}$ (Multimodal test problem with fixed dimension)

Table 1: Test problems under consideration

| Test problems $f_i(x)$ | $S$ | $f_{\text{min}}$ | $C$ |
|------------------------|-----|-----------------|-----|
| $f_1(x) = \sum_{i=1}^{n} x_i^2$ | $[-100, 100]^m$ | 0 | $U$ |
| $f_2(x) = \sum_{i=1}^{n} |x_i^2| + \prod_{i=1}^{n} |x_i|$ | $[-10, 10]^m$ | 0 | $U$ |
| $f_3(x) = \sum_{i=1}^{n} \left( \sum_{j=1}^{i-1} x_j \right)^2$ | $[-100, 100]^m$ | 0 | $U$ |
| $f_4(x) = \max_i \{ |x_i|, 1 \leq i \leq n \}$ | $[-100, 100]^m$ | 0 | $U$ |
| $f_5(x) = \sum_{i=1}^{n-1} \left[ 100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$ | $[-30, 30]^m$ | 0 | $U$ |
| $f_6(x) = \sum_{i=1}^{n} (|x_i + 0.5|)^2$ | $[-100, 100]^m$ | 0 | $U$ |
| $f_7(x) = \sum_{i=1}^{n} i x_i^2 + \text{random}[0, 1]$ | [-1.28, 1.28]$^m$ | 0 | $U$ |
| $f_8(x) = \sum_{i=1}^{n} -x_i \sin \left( \sqrt{|x_i|} \right)$ | $[-500, 500]^m$ | 418.9829x$m$ | $M$ |
| $f_9(x) = \sum_{i=1}^{n} \left[ x_i^2 - 10 \cos(2 \pi x_i) + 10 \right]$ | $[-5.12, 5.12]^m$ | 0 | $M$ |
| $f_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos \left( 2 \pi x_i \right) \right) + 20 + e$ | $[-32, 32]^m$ | 0 | $M$ |
| $f_{11}(x) = \frac{1}{-600, 600} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{m} \right) + 1$ | $[-600, 600]^m$ | 0 | $M$ |
4.2 Experimental setting

In order to validate the effectiveness and robustness of the proposed algorithm, BNAGGSA is compared with BGSA [22], a binary version of FVGGSA [2] and a binary version of PTGSA [19] with the following experiment setting:

- The number of simulations/run =30,
- Swarm size=50,
- For the binary search space, the dimension of $f_1$ to $f_{13}$ in Table 3 is $m \times L$. Here, $m$ and $L$ are set to 5 and 15 respectively. Therefore, all the binary version of the considered uni-modal ($f_1$-$f_7$) and multi-modal ($f_8$-$f_{13}$) problems have 75 ($5 \times 15$) dimensional search space. On the other hand, the binary version of multi-modal with fixed dimensional problems ($f_{14}$-$f_{23}$) have different dimensions from the range $[2 \times 15, 6 \times 15]$. For $f_{14}$, $f_{16}$, $f_{17}$ and $f_{18}$, the dimension is 30 ($2 \times 15$). For $f_{15}$, $f_{21}$, $f_{22}$ and $f_{23}$, the dimension is 60 ($4 \times 15$). $f_{19}$ has 45 ($3 \times 15$) dimensional search space while $f_{20}$ is the the maximum dimensional problem having 90 ($6 \times 15$) dimension.
- Maximum number of iteration =500
Parameters for the algorithms BGSA are same as GSA except the gravitational constant $G$. In BGSA, $G$ is linear in nature defined as:

$$G(t) = G_0 \times (1 - \frac{t}{T})$$  \hspace{1cm} (15)

Here, $t$ and $T$ have their usual meaning. The parameters of the proposed variant and other considered algorithms are listed in Table 2.

The routine which converts binary representation into real, discussed in Section 4.1.1, is used for all considered algorithms.

4.3 Results and statistical analysis

The experimental results of best-so-far solution over 30 independent runs under 30 different random seeds are summarized in Table 3. The bold entries show the best results. Table 3 lists the three metrics of best-so-far solution: average best-so-far solution ($\text{ABSF}$), standard deviation ($\text{STDV}$) of best solution and the best obtained solution over 30 runs ($\text{best}$). The best-so-far solution is the solution of the last iteration of an individual run. For all unimodal test problems ($f_1$-$f_7$), the proposed BNAGGSA outperforms other BGSA variants with a large margin. It proves the fast convergence ability of the proposed variant. Out of six multimodal problems ($f_8$-$f_{13}$), all six $\text{ABSF}$ and $\text{STDV}$ values of the proposed BNAGGSA proves its supreme stagnation avoidance mechanism over others. Since the multimodal problems have multiple local optima around the global optimum in which the swarm can stagnate into the non-optimal basin. While in BNAGGSA, the proposed fitness distance ratio based gravitational constant provides a dynamic mechanism as per the search necessity which successfully enables the population to track more promising regions. Out of 10 multimodal problems with fix dimension ($f_{14}$-$f_{23}$), BNAGGSA outperforms others in terms of $\text{ABSF}$ on 9 problems ($f_{15}$, $f_{16}$, $f_{17}$, $f_{18}$, $f_{19}$, $f_{20}$, $f_{21}$, $f_{22}$, and $f_{23}$). Among all metrics of comparison, BNAGGSA proves its supremacy on five test problems ($f_{15}$, $f_{16}$, $f_{18}$, $f_{19}$, and $f_{22}$). Figure 4 illustrates the convergence behaviors of BNAGGSA and BGSA over unimodal ($f_1$), multimodal ($f_8$) and multimodal with fixed dimension ($f_{23}$) function. It is clear from Figure 4 that BNAGGSA outperforms BGSA in terms of exploitation ability due to its fastest convergence rate. Further, to compare the algorithms based on $\text{ABSF}$ over all 23 test problems, boxplot analysis has been carried out.

| Algorithm   | Parameters                                      |
|-------------|-------------------------------------------------|
| BGSA        | independent of $\alpha$, $G_0=100$              |
| BFVGGSA     | $\alpha=10$, independent of a fixed $G_0$      |
| BPTGSA      | $G_0=100$, $\alpha \in [5, 70]$                |
| BNAGGSA     | independent of $\alpha$ and $G_0$ both, $\delta=10^{-2}$, $\gamma=10^{-5}$ |

In Figure 5, the boxplot of BNAGGSA have a less interquartile range and median as compared to BGSA, BFVGGSA and BPTGSA, which further implies that BNAGGSA is more efficient over other considered algorithms. To examine the significant difference among the results of the considered algorithms, Friedman test is used. This non-parametric statistical test is performed pairwise at 1%
level of significance with the null hypothesis, ‘There is no significant difference between the results obtained by the considered pair’. In this study, for pairwise comparisons, the adjusted p-values are reported. These p-values are achieved by the post-hoc test procedure, namely ‘bonferroni’. This procedure is implemented in R programming language [29]. Table 4 presents the p-values for the pairwise comparison of BNAGGSA and other considered GSA binary variants. From Table 4, the following observations are made:

- For the pair BFVGGSA-BNAGGSA, all the p-values are less than 0.01 which implies that there is a significant difference between BFVGGSA-BNAGGSA over all the considered test problems.
- Except $f_{14}$ and $f_{18}$, all the p-values are less than 0.01 for the pair BGS-BAVGGSA. It means, for 21 problems ($f_1$, $f_2$, $f_3$, $f_4$, $f_5$, $f_6$, $f_7$, $f_8$, $f_9$, $f_{10}$, $f_{11}$, $f_{12}$, $f_{13}$, $f_{15}$, $f_{16}$, $f_{17}$, $f_{18}$, $f_{20}$, $f_{21}$, $f_{22}$ and $f_{23}$), there is a significant difference between BPTGSA-BNAGGSA. For $f_{14}$ and $f_{18}$, the p-values are greater than 0.01, which indicates that BPTGSA performs similar to the BPTGSA and vice versa.
- Except $f_3$, $f_5$, $f_{12}$, $f_{14}$, $f_{19}$ and $f_{22}$, all the p-values are less than 0.01 for the pair BPTGSA-BNAGGSA. It means, for 17 problems ($f_1$, $f_2$, $f_3$, $f_4$, $f_5$, $f_6$, $f_7$, $f_8$, $f_9$, $f_{10}$, $f_{11}$, $f_{13}$, $f_{15}$, $f_{16}$, $f_{17}$, $f_{18}$, $f_{20}$, $f_{21}$ and $f_{23}$), there is a significant difference between BPTGSA-BNAGGSA. For $f_3$, $f_5$, $f_{12}$, $f_{14}$, $f_{19}$ and $f_{22}$, the p-values are greater than 0.01, which indicates that BPTGSA performs similar to the BPTGSA and vice versa.
- Except $f_{13}$ and $f_{23}$, all the p-values are less than 0.01 for the pair BPTGSA-BNAGGSA. It means, for 21 problems ($f_1$, $f_2$, $f_3$, $f_4$, $f_5$, $f_6$, $f_7$, $f_8$, $f_9$, $f_{10}$, $f_{11}$, $f_{12}$, $f_{14}$, $f_{15}$, $f_{16}$, $f_{17}$, $f_{18}$, $f_{19}$, $f_{20}$, $f_{21}$ and $f_{22}$), there is a significant difference between BPTGSA-BNAGGSA. For $f_{13}$ and $f_{23}$, the p-values are greater than 0.01, which indicates that BPTGSA performs similar to the BPTGSA and vice versa.
- Except $f_6$, $f_7$ and $f_{14}$, all the p-values are less than 0.01 for the pair BFVGGSA-BPTGSA. It means, for 20 problems ($f_1$, $f_2$, $f_3$, $f_4$, $f_5$, $f_6$, $f_9$, $f_{10}$, $f_{11}$, $f_{12}$, $f_{13}$, $f_{15}$, $f_{16}$, $f_{17}$, $f_{18}$, $f_{19}$, $f_{20}$, $f_{21}$, $f_{22}$ and $f_{23}$), there is a significant difference between BFVGGSA-BPTGSA. For $f_6$, $f_7$ and $f_{14}$, the p-values are greater than 0.01, which indicates that BFVGGSA performs similar to the BPTGSA and vice versa.
- Except $f_5$, $f_8$, $f_{15}$, $f_{20}$, $f_{22}$ and $f_{23}$, all the p-values are less than 0.01 for the pair BPTGSA-BNAGGSA. It means, for 16 problems ($f_1$, $f_2$, $f_4$, $f_6$, $f_7$, $f_8$, $f_{10}$, $f_{11}$, $f_{12}$, $f_{13}$, $f_{14}$, $f_{15}$, $f_{18}$, $f_{19}$, $f_{20}$ and $f_{21}$), there is a significant difference between BPTGSA-BNAGGSA. For $f_5$, $f_8$, $f_{15}$, $f_{16}$, $f_{20}$, $f_{22}$ and $f_{23}$, the p-values are greater than 0.01, which indicates that BNAGGSA performs similar to the BPTGSA and vice versa.

Based on the above multiple comparative analyses, the proposed BNAGGSA is an overall better algorithm than other considered binary algorithms.

| Test Problem | metrics | BGSA   | BFVGGSA | BPTGSA   | BNAGGSA   |
|--------------|---------|--------|---------|----------|-----------|
| $f_1$        | ABSF    | 3.086914619  | 0.067298611 | 3.103842338 | 0.000352661 |
|              | STDV    | 8.206405594  | 0.163922163 | 3.526555167 | 0.0017413  |
|              | Best    | 0.015050173  | 0             | 0.08687377 | 0         |
| $f_2$        | ABSF    | 0.07792155   | 0.018473307 | 0.68175555  | 0.003205808 |
|              | STDV    | 0.047807411  | 0.016113692 | 0.421837265 | 0.00161176 |
|              | Best    | 0.007324219  | 0.001220703 | 0.042724609 | 0         |

Table 3: Performance of considered algorithms
| Function | ABSF  | STDV  | Best    |
|----------|-------|-------|---------|
| f3       | 70.47568758 | 94.49648857 | 148.2791752 | 0.000362595 |
|          | 118.817758 | 117.0452651 | 161.8985926 | 7.38406E-05 |
|          | 0.092685223 | 0.025033951 | 4.797875881 | 0.000298023 |
| f4       | 1.855875651 | 0.534667969 | 6.274007161 | 0.030924479 |
|          | 1.428978817 | 0.6907666982 | 7.254831838 | 0.042137349 |
|          | 0.390625 | 0.012207031 | 0.366210938 | 0.012207031 |
| f5       | 177.6295231 | 866.4641931 | 11352.02261 | 23.95170429 |
|          | 572.0141956 | 1892.320518 | 33935.24801 | 79.6930348 |
|          | 3.516137163 | 0.581174682 | 3.899042624 | 1.04698025 |
| f6       | 4.4 | 0.1 | 4.5 | 0 |
|          | 11.42978565 | 0.3 | 10.16120072 | 0 |
|          | 0 | 0 | 0 | 0 |
| f7       | 0.010974215 | 0.002879371 | 0.016518962 | 0.003187232 |
|          | 0.006606426 | 0.002419516 | 0.014461618 | 0.001758782 |
|          | 0.001944507 | 0.00093562 | 0.002346032 | 0.000877792 |
| f8       | -5390.732827 | -5224.073913 | -5310.475213 | -5589.778059 |
|          | 251.699878 | 250.758352 | 238.8942127 | 136.6408955 |
|          | -5806.389527 | -5636.886076 | -5804.293339 | -6037.440855 |
| f9       | 8.13412877 | 4.40403452 | 5.97501163 | 2.728963629 |
|          | 3.665399353 | 1.803598251 | 2.269528528 | 1.502279429 |
|          | 3.241000492 | 0.00332237 | 2.450572396 | 0.00023249 |
| f10      | 4.865229489 | 0.262288777 | 0.438199742 | 0.00950132 |
|          | 1.754725389 | 0.619683163 | 0.737461356 | 0.004531467 |
|          | 2.397526251 | 0.007150226 | 0.007150226 | 8.88178E-16 |
| f11      | 0.437071984 | 0.086784975 | 0.9515136 | 0.047908036 |
|          | 0.257320344 | 0.057438425 | 0.44655815 | 0.02308727 |
|          | 0.068297739 | 0.017723767 | 0.257408952 | 0.011917058 |
Table 3 Continued:

| Test problem | metrics | BGSA | BFVGGSA | BPTGSA | BNAGGSA |
|--------------|---------|------|---------|--------|---------|
| f₁₂          | ABSF    | 1.801869755 | 5.751197856 | 1592.485357 | 0.89575387 |
|              | STDV    | 1.875365125 | 12.73916346 | 7684.463846 | 0.985715835 |
|              | Best    | 0.002877555 | 0.000383423 | 0.141754486 | 3.88349E-06 |
| f₁₃          | ABSF    | 9.151127748 | 8.09012069 | 24728.1539 | 0.285255839 |
|              | STDV    | 45.84203549 | 37.69028395 | 12626.0999 | 0.120808877 |
|              | Best    | 0.039214301 | **0.004561255** | 0.141504394 | 0.101244856 |
| f₁₄          | ABSF    | **1.02513438** | 1.0327565 | 1.23962393 | 3.88349E-06 |
|              | STDV    | **0.064307845** | 0.07774098 | 0.37739433 | 0.118627136 |
|              | Best    | 0.998003842 | **0.998003838** | 0.998003838 | 0.998003838 |
| f₁₅          | ABSF    | 0.00015092 | 0.000287968 | 0.000562143 | 0.001448739 |
|              | STDV    | 0.000233295 | 0.000364887 | 0.000511368 | 0.000212638 |
|              | Best    | 0.998003842 | **0.998003838** | 0.998003838 | 0.998003838 |
| f₁₆          | ABSF    | -1.03965569 | -1.024444516 | -1.0253336 | -1.031627651 |
|              | STDV    | -0.031627919 | -0.031627919 | -0.031627284 | -1.031627919 |
|              | Best    | 0.398881909 | 0.39955722 | 0.401005955 | 0.397892311 |
| f₁₇          | ABSF    | 0.001541094 | 0.002879668 | 0.003562143 | 0.001448739 |
|              | STDV    | 0.000233295 | 0.000978373 | 0.001572025 | 0.00000000 |
|              | Best    | 0.998003842 | **0.998003838** | 0.998003838 | 0.998003838 |
| f₁₈          | ABSF    | 3.00001448 | 3.071615699 | 3.102829593 | 3.00001448 |
|              | STDV    | 3.612E-06 | 0.11115908 | 0.151152665 | 2.51263E-06 |
|              | Best    | 3 | **3** | 3 | **3** |

Table 4: p-values for pairwise comparison of considered algorithms

| TP  | - | BGSA | BFVGGSA | BPTGSA |
|-----|---|------|---------|--------|
| f₁  | BFVGGSA | < 2E − 16 | – | – |
|     | BPTGSA | < 2E − 16 | < 2E − 16 | – |
|     | BNAGGSA | 7E − 12 | < 2E − 16 | < 2E − 16 |
|     | BFVGGSA | < 2E − 16 | – | – |
| f₂  | BPTGSA | < 2E − 16 | < 2E − 16 | – |
|     | BNAGGSA | < 2E − 16 | < 2E − 16 | < 2E − 16 |
Table 4 Continued:

| TP | BGSA | BFVVGS | BPTGSA |
|----|------|--------|--------|
| $f_3$ | BFVVGS | $< 2E - 16$ | – | – |
| $f_4$ | BPTGSA | $< 1$ | $< 2E - 16$ | – |
| $f_5$ | BNAGGSA | $1.6E - 11$ | $< 2E - 16$ | $1.6E - 11$ |
| $f_6$ | BFVVGS | $< 2E - 16$ | – | – |
| $f_7$ | BPTGSA | $< 2E - 16$ | $< 2E - 16$ | – |
| $f_8$ | BNAGGSA | $5.7E - 08$ | $< 2E - 16$ | $< 2E - 16$ |
| $f_9$ | BFVVGS | $< 2E - 16$ | – | – |
| $f_{10}$ | BPTGSA | $< 2E - 16$ | $< 2E - 16$ | – |
| $f_{11}$ | BNAGGSA | $< 2E - 16$ | $< 2E - 16$ | $1.2E - 07$ |
| $f_{12}$ | BFVVGS | $< 2E - 16$ | – | – |
| $f_{13}$ | BPTGSA | $< 2E - 16$ | $< 2E - 16$ | – |
| $f_{14}$ | BNAGGSA | $0.00056$ | $< 1E - 08$ | – |
| $f_{15}$ | BFVVGS | $< 2E - 16$ | $< 2E - 16$ | – |
| $f_{16}$ | BPTGSA | $< 2E - 16$ | $< 2E - 16$ | – |
| $f_{17}$ | BNAGGSA | $< 2E - 16$ | $< 2E - 16$ | $9.8E - 08$ |
| $f_{18}$ | BFVVGS | $< 2E - 16$ | $8E - 16$ | – |
| $f_{19}$ | BPTGSA | $< 2E - 16$ | $< 2E - 16$ | $0.00057$ |
Table 4 Continued:

| TP  | BGSA     | BFVGGSA | BPTGSA | BNAGGSA |
|-----|----------|---------|--------|---------|
| f₁⁹ | BFVGGSA  | < 2E − 16 | −    | −       |
|     | BPTGSA   | < 1     | < 2E − 16 | −       |
|     | BNAGGSA  | 0.00046 | < 2E − 16 | 0.00282 |
| f₂₀ | BFVGGSA  | < 2E − 16 | −    | −       |
|     | BPTGSA   | < 2E − 16 | 4.6E − 06 | −       |
|     | BNAGGSA  | < 2E − 16 | 0.216 | 0.012   |
| f₂₁ | BFVGGSA  | < 2E − 16 | −    | −       |
|     | BPTGSA   | 1.7E − 10 | 6.8E − 10 | −       |
|     | BNAGGSA  | 0.00029 | < 2E − 16 | 0.00701 |
| f₂₂ | BFVGGSA  | < 2E − 16 | −    | −       |
|     | BPTGSA   | 9.7E − 08 | < 2E − 16 | −       |
|     | BNAGGSA  | 7.7E − 05 | < 2E − 16 | 0.68   |
| f₂₃ | BFVGGSA  | 1.5E − 09 | −    | −       |
|     | BPTGSA   | 0.29   | 1.2E − 13 | −       |
|     | BNAGGSA  | 0.54   | 4.8E − 13 | 1       |

5 Windfarm Layout Optimization Problem

Wind energy is one of the fast growing and most sustainable resource of the renewable energy in which the wind power is extracted by the turbine through rotation of its blades. The installation of the turbines in the windfarm is the most important entity as the wake loss induced by upstream turbines affects the output of downstream turbines, which further reduces the total power output of the windfarm [4]. Therefore, to maximize the total output of the windfarm the optimal pattern of wind turbines is required subject to the constraints related to the position of the turbines, rotor radius and farm radius. In [4], a wind model with real data sets is optimized under the decomposition based multi-objective environment. In this study, the practical applicability of the proposed variant BNAGGSA is proven to optimize the wind model of [4] under the single objective environment, in the binary search space.

5.1 Mathematical models of the considered windfarm and its numerical data

The windfarm layout design depends upon its minimum wake loss strategy. Wake is the only significant constraint which effects the wind model’s objectives more drastically. Wake exists when wind stream interacts with turbines. This interaction not only reduces the speed of the wind but also increases the turbulence intensity of the attached turbines. Two more issues with the existed wake are its movement towards downstream direction and its diverging nature. Under the wake effects, turbines which are located in downstream regions do not perform well. In the past years, several wake models have been developed to reduce these wake effects on the performance of their respective windfarms. In this study, Jensen wake decay model [4] is taken to find the wind velocity inside the wake region. Let us consider kᵗʰ turbine under the wake region of the nᵗʰ turbine (refer Figure 6(a)). According to the Jensen wake decay model, the wind speed for the kᵗʰ turbine is calculated as:
Fig. 4 Convergence graphs

\[ u_k = u_0 k \left[ 1 - \frac{2a}{1 + a m \sigma_m^2} \right] \]  \hspace{1cm} (16)

\[ a = \frac{1 - \sqrt{1 - C_T}}{2} \]  \hspace{1cm} (17)
Fig. 5 Boxplot analysis of average best so far (ABSF)

\[ r_m = \frac{1 - a}{1 - 2a} \]  
\[ \alpha_m = \frac{0.5}{ln\left(\frac{h_m}{z_0}\right)} \]

where,
- \( u_{ok} \) is the local wind speed at \( k^{th} \) turbine without considering the wake effect,
- \( x_{mk} \) is the distance between \( m^{th} \) and \( k^{th} \) turbine,
- \( r_m \) is the radius of \( m^{th} \) turbine rotor,
- \( r_{m1} \) is the downstream rotor radius of \( m^{th} \) turbine,
- \( h_m \) is the hub height of the \( m^{th} \) turbine,
- \( \alpha_m \) is the entrainment constant pertaining to \( m^{th} \) turbine,
- \( a \) is the axial induction factor,
- \( C_T \) is the thrust coefficient of the wind turbine rotor,
- \( z_0 \) is the surface roughness of the windfarm.

The wake region of a linear wake model is conical with the wake influence radius defined as:
\[ r_{wm} = \alpha_m x_{mk} + r_{m1} \]  

Since the velocity of wind is different for different heights therefore the local wind speed at \( k^{th} \) turbine depends upon its hub height defined as:
\[ u_{0k} = u_{ref} \frac{\log(h_k/z_0)}{\log(h_{ref}/z_0)} \]

where, \( u_{ref} \) is the wind speed at the reference height \( h_{ref} \). If the \( i^{th} \) turbine is inside the area of multiple wake flows, than the wind speed at that turbine can be calculated as:
\[ u_i = u_{0i} \left[ 1 - \frac{1}{N_i} \sum_{j=1}^{N_i} A_{ij} \left( 1 - \frac{u_{ij}}{u_{0j}} \right)^2 \right] \]
where, $u_{0i}$ and $u_{0j}$ are the local wind speeds at $i^{th}$ and $j^{th}$ turbines, respectively without considering the wake effect; $u_{ij}$ is the wind velocity at $i^{th}$ turbine under the influence of $j^{th}$ turbine; $N_t$ is the number of turbines affecting the $i^{th}$ turbine with wake effects; $r_i$ is the rotor radius of $i^{th}$ turbine; $A_{ij}$ is the overlapped rotor area of $i^{th}$ turbine under wake influence radius ($r_{wj}$) of $j^{th}$ turbine (Figure 6(b)). Under the condition of full wake effect, $A_{ij} = \pi \times r^2_i$ [1].

Fig. 6 Graphical representation of Jensen wake decay model [4]

5.1.1 Turbine power output model

The output power (in kW) from the turbine is a function of wind speed ($u$) and expressed as:

$$P(u) = \begin{cases} 
0, & \text{for } u < u_c \\
60 \times (u - u_c), & \text{for } u_c \leq u \leq u_c + \frac{25}{19} \\
250 \times (u - u_c - 1), & \text{for } u_c + \frac{25}{19} \leq u \leq 13 \\
2000, & \text{for } 13 \leq u \leq u_f \\
0, & \text{for } u > u_f
\end{cases}$$

(23)

where, $u_c$ and $u_f$ are the cut-in and cut-out wind speeds of the turbine, respectively.

5.1.2 Wind condition model

In [4], the authors considered two type of wind data sets of sites in Midland (site 1) and Corpus Christi (site 2), Texas. The wind probability distribution diagrams for site 1 and 2 are presented in Figure 7(a) and Figure 7(b), respectively. The direction of the each wind data is discretized into 16 different segments of $22.5^\circ$ each. The north direction set as $0^\circ$ and an increment of $22.5^\circ$ in clockwise direction is considered to divide $360^\circ$ in 16 segments [4]. Wind speed is discretized into these 16 segments in such a way that each segment have the probability of occurrence of each discrete wind speed. The summation of all the probabilities equals 1. For the both case studies, four discrete wind speeds are considered as $u_{ref_0} < 4m/s$, $u_{ref_1} < 8.2m/s$, $u_{ref_2} < 10.8m/s$ and $u_{ref_3} = 14.4m/s$ at hub height= 60m [4]. In this study, both the mentioned cases are reconsidered to optimize their respective objectives more accurately.
5.1.3 Objective function under consideration

In this study, two different objectives are maximized in the single objective platform through the weighted aggregate method \cite{22} written as:

$$F = w_1 \times F_1 + w_2 \times F_2$$  \hspace{1cm} (24)

Where, $F_1$ and $F_2$ are the efficiency and the total power output of the windfarm calculated as follows:

$$F_1 = \frac{\sum_{k=0}^{360} \sum_{i=1}^{N_T} f_k P_i(u_i)}{\sum_{k=0}^{360} \sum_{i=1}^{N_T} f_k P_{i,max}(\bar{u}_i)}$$  \hspace{1cm} (25)

Total power output of the windfarm is given by,

$$F_2 = \sum_{k=0}^{360} \sum_{i=1}^{N_T} f_k P_i(u_i)$$  \hspace{1cm} (26)

where, $N_T$ is the total number of turbines, $P_{i,max}$ is the maximum power output from $i^{th}$ turbine with the wind speed $\bar{u}_i$ under no wake effect, $P_i$ is the actual power output from $i^{th}$ turbine as a function of wind speed $u_i$ considering wake effect it experiences from upstream turbine(s). The probability of occurrence of each wind speed from each direction is defined by factor $f_k$ and $\sum_{k=0}^{360} f_k = 1$. $w_1$ and $w_2$ are weights to the objective functions $F_1$ and $F_2$ respectively and set to 0.5.
5.1.4 Windfarm selection and turbine data

In this study, a rectangular windfarm of $6000m \times 2000m$ is divided into 100 equal cells of size $300m \times 400m$. At the center of each cell, a turbine can be placed. The objective of this study is to maximize the total power output of this windfarm having $N_t$ number of turbines in which each turbine have 100 possible locations to be placed. The turbine and other relevant data used for calculation are listed in Table 5.

| Parameter                     | Value                        |
|-------------------------------|------------------------------|
| Turbine model                 | Vestas V-80                  |
| Rated Power, $P_{rated}$      | 2000 kW                      |
| Rotor diameter, $D$           | 80m                          |
| Thrust coefficient, $C_T$     | 0.8                          |
| Hub height, (IEC IA)          | 60m, 67m or 78m              |
| Cut-in speed, $u_c$           | 4m/s                         |
| Cut-out speed, $u_f$          | 25m/s                        |
| Surface roughness of windfarm, $z_0$ | 0.3m                      |
| Reference height, $h_{ref}$   | 60m                          |

6 Applicability of the proposed variant BNAGGSA for wind farm layout optimization problem

GSA is a well established optimizer to solve the real world optimization problems in the various fields of engineering science, like power engineering, pattern recognition, image processing, classification, communication engineering, control engineering, civil engineering, computer and software engineering, mechanical engineering, water industry etc. As per the authors’ best knowledge, no application work of GSA framework has been done so far in the field of renewable energy. In this study, GSA framework is used in the binary search space for optimizing the wind farm layout to obtain the maximum power output.

If $N$ is the swarm size than there are $N$ candidate solutions $X_1, X_2, \ldots, X_N$ for the considered windfarm layout optimization problem. Since each turbine has 100 possible locations to be placed therefore each candidate solution ($X_k$ for $k = 1, 2, \ldots, N$) is 100-dimensional vector where each dimension represents the status of that location through the binary values 0 or 1. 0 indicates that the location is empty while 1 indicates the presence of the turbine. Thus the $k^{th}$ candidate solution is encoded as binary vector $X_k = (x_1, x_2, \ldots, x_{100})$ where $x_j, \forall j = 1 : 100$ is either 0 or 1.

In this study, we optimize different windfarm layout for different number of turbines ($N_T$) by considering it as a constraint optimization problem whose objection function $F$ (refer equation (24)) is re-defined using the penalty function approach as follows:

$$F(X_i) = \begin{cases} F(X_i), & \text{if sum of ones in } X_i = N_T \\ P, & \text{otherwise} \end{cases} \quad (27)$$

Here $P$ is the penalty due to the constraints violation. Since this is a maximization problem therefore $P$ is assigned very small value set to $10^{-10}$. The different values of $N_T$ are 10, 20, 30, 40, 50 and 60 taken under consideration.
6.1 Results and discussion

In this section, the proposed binary variant is tested over two different wind data sets of two different wind sites (site 1 and 2) for the considered windfarm. All the required information about both sites are given in section 5. In both cases, the heights of all turbines are fixed. The wind deficit for a turbine under the wake effect is calculated by equations (16) to (22). The power output and the objective functions are calculated using equations (23) to (26).

In order to validate the effectiveness and robustness of proposed algorithm, BNAGGSA is compared with binary GSA [32], a binary version of FVGGSA [2], binary version of PTGSA [19] along with MOEA/D [4]. The binary version of FVGGSA (BFGGSA) and PTGSA (BPTGSA) are followed the same settings as binary GSA (refer section 4). For all the comparison, the following experimental setting is adopted:

- Swarm size =500,
- Maximum number of generations/iterations =500,
- Total number of runs/simulations =10,
- The other parameters of BGSA, BNAGGSA, BFVGGSA, BPTGSA and MOEA/D are considered from their original resources while the result of MOEA/D is reproduced from [4].
- The parameters of the consider windfarm and its turbines are already mentioned in section 5.

The results of optimized layouts of site 1 and site 2 with different numbers of turbines ($N_T$) along with corresponding power output, efficiency and capacity factor are listed in Table 6 and Table 7, respectively. Capacity factor is a ratio between the amount of energy delivered annually and the maximum amount of energy can be delivered yearly. It is clear from the results that the proposed BNAGGSA outperforms in every metric of comparison with BGSA, BFVGGSA and BPTGSA for both sites. For the optimal layout of wind site 1, MOED/D is performed superior for 10 and 20 number of turbines. For the optimal layout of 30, 40, 50 and 60 number of turbines, the proposed variant proves its supremacy with MOED/D in each metric of comparison. For the wind site 2, MOED/D performs better (in each metric) than the proposed BNAGGSA for the layout of 10 and 20 turbines. For 30 turbines layout, MOED/D performs better than BNAGGSA only in power output and capacity factor while BNAGGSA is superior in terms of efficiency. For 40, 50 and 60 turbines layout, the proposed BNAGGSA is superior than MOED/D in all the metrics of comparison.

Fig. 8 and Fig. 9 indicate the optimal layouts of $N_T$ turbines obtained by BNAGGSA for wind conditions at site 1 (case 1) and site 2 (case 2), respectively.

| $N_T$ | metrics          | BGSA   | BFVGGSA | BPTGSA | MOEA/D     | BNAGGSA     |
|-------|------------------|--------|---------|---------|------------|-------------|
| 10    | Power output (MW)| 5.1683 | 5.1683  | 5.1683  | 5.21       | 5.179361    |
|       | Efficiency       | 98.67% | 98.67%  | 98.67%  | 99.5%      | 98.89%      |
|       | Capacity factor  | 25.84% | 25.84%  | 25.84%  | 26.1%      | 25.9%       |
| 20    | Power output (MW)| 9.9969 | 9.9969  | 10.0179 | 10.18      | 10.1559     |
|       | Efficiency       | 95.43% | 95.43%  | 95.7%   | 97.1%      | 96.94%      |
|       | Capacity factor  | 24.99% | 24.99%  | 25.03   | 25.5%      | 25.39%      |
| 30    | Power output (MW)| 14.55  | 14.55   | 14.54   | 14.78      | 14.85176    |
|       | Efficiency       | 92.61% | 92.61%  | 92.55   | 94.1%      | 94.51%      |
|       | Capacity factor  | 24.25% | 24.25%  | 24.24%  | 24.6%      | 24.75%      |
| 40    | Power output (MW)| 18.88  | 18.90   | 18.87   | 19.16      | 19.22366    |

Table 6: Optimized layouts results for site 1
Table 6 Continued:

|       | Efficiency | Capacity factor | Power output (MW) |
|-------|------------|-----------------|-------------------|
| 50    | 90.09%     | 23.59%          | 22.97             |
|       | 90.21%     | 23.63%          | 23.06             |
|       | 90.06%     | 23.59%          | 22.93             |
|       | 91.4%      | 23.9%           | 23.40             |
|       | 91.75%     | 24.02%          | 23.44308          |
| 60    | 87.70%     | 22.97%          | 26.96             |
|       | 88.04%     | 23.06%          | 26.98             |
|       | 87.55%     | 22.93%          | 26.90             |
|       | 89.4%      | 23.4%           | 27.38             |
|       | 89.51%     | 23.44%          | 27.41758          |

Table 7: Optimized layouts results for site 2

| N<sub>T</sub> | metrics | BGSA       | BFVGGSA     | BPTGSA      | MOEA/D     | BNAGGSA     |
|--------------|---------|------------|-------------|-------------|------------|-------------|
| 10           | Power output (MW) | 7.926149   | 7.918980523 | 7.932965   | 7.98       | 7.949118403 |
|              | Efficiency | 98.81%     | 98.72%      | 98.89%      | 99.5%      | 99.09%      |
|              | Capacity factor | 39.63%     | 39.59%      | 39.67%      | 39.9%      | 39.75%      |
| 20           | Power output (MW) | 15.39293   | 15.37645    | 15.37053   | 15.61      | 15.50229859 |
|              | Efficiency | 95.95%     | 95.85%      | 95.81%      | 97.3%      | 96.63%      |
|              | Capacity factor | 38.48%     | 38.44%      | 38.43%      | 39.0%      | 38.76%      |
| 40           | Power output (MW) | 29.16514   | 29.35264    | 29.19196   | 30.09      | 30.11622022 |
|              | Efficiency | 90.89%     | 91.48%      | 90.98%      | 93.8%      | 93.86%      |
|              | Capacity factor | 36.46%     | 36.69%      | 36.49%      | 37.6%      | 37.65%      |
| 50           | Power output (MW) | 35.59742   | 35.65921    | 35.5257    | 36.50      | 36.51936845 |
|              | Efficiency | 88.76%     | 88.90%      | 88.58%      | 91.0%      | 91.05%      |
|              | Capacity factor | 35.59%     | 35.66%      | 35.53%      | 36.5%      | 36.52%      |
| 60           | Power output (MW) | 41.61174   | 41.83223    | 41.59735   | 42.47      | 42.49998996 |
|              | Efficiency | 86.46%     | 86.92%      | 86.43%      | 88.2%      | 88.30%      |
|              | Capacity factor | 34.68%     | 34.86%      | 34.67%      | 35.4%      | 35.42%      |

Therefore, the proposed BNAGGSA proves its applicability from all the considered GSA variants in the binary search space. Additionally, Despite using the single objective framework, the proposed BNAGGSA proves its supremacy over multi objective framework based MOEA/D for 30, 40, 50 and 60 turbines layout optimization of the considered windfarm in both mentioned wind sites.

7 Conclusion

This paper proposes a novel binary GSA variant named ‘A novel neighbourhood archives embedded gravitational constant in GSA for binary search space (BNAGGSA)’. In BNAGGSA, first two social interaction schemes are proposed for more diversified search mechanism in GSA under binary search space. Then, a novel fitness-distance ratio based gravitational constant is proposed to provide a self adaptive scaling to each particle as per its current search requirements. The performance of the BNAGGSA is compared with the two variants of BGSA on 23 popular benchmark test problems.
Fig. 8 Optimal layouts obtained by BNAGGSA for $N_T$ number of turbines of case 1 (site 1)
Fig. 9 Optimal layouts obtained by BNAGGSA for $N_T$ number of turbines of case 2 (site 2)
The results show that BNAGGSA outperformed BGSA’s variants in finding optimal solutions with better convergence speed. Additionally, a windfarm layout optimization problem with two different wind data sets (two case studies) has been solved by BNAGGSA. BNAGGSA obtained the most efficient layouts for any selected number of turbines within a specified range. Numerical experiments conclude that the BNAGGSA is able to find the better optimal placement of wind turbines (for more than 30 turbines) for both case studies compare than some recent binary variants of GSA as well as MOEA/D. Thus, the proposed BNAGGSA is recommended as an efficient solver for windfarm layout optimization problem.

Declaration

Conflicts of interest/Competing interests: All the authors declare that they has no conflict of interest.

Availability of data and material: Not applicable.

Code availability: Not applicable.

References

1. Jagdish Chand Bansal and Pushpa Farswan. Wind farm layout using biogeography based optimization. Renewable energy, 107:386-402, 2017
2. Jagdish Chand Bansal, Susheel Kumar Joshi, and Atulya K. Nagar. Fitness varying gravitational constant in gsa. Applied Intelligence, 2018.
3. Partha P Biswas, PN Suganthan, and Gehan AJ Amaratunga. Optimization of wind turbine rotor diameters and hub heights in a windfarm using differential evolution algorithm. In Proceedings of Sixth International Conference on Soft Computing for Problem Solving, pages 131-141. Springer, 2017.
4. Partha P Biswas, PN Suganthan, and Gehan AJ Amaratunga. Decomposition based multi-objective evolutionary algorithm for windfarm layout optimization. Renewable energy, 115:326-337, 2018
5. Partha P Biswas, Ponmuthurai N Suganthan, and Gehan AJ Amaratunga. Optimal placement of wind turbines in a windfarm using l-shade algorithm. In 2017 IEEE Congress on Evolutionary Computation (CEC), pages 83-88. IEEE, 2017.
6. Hamid Bostani and Mansour Sheikhan. Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems. Soft computing, 21(9):2307-2324, 2017
7. Tapabrata Chakraborti and Amitava Chatterjee. A novel binary adaptive weight gsa based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns. Engineering Applications of Artificial Intelligence, 33:80-90, 2014
8. K Chen, MX Song, ZY He, and X Zhang. Wind turbine positioning optimization of wind farm using greedy algorithm. Journal of Renewable and Sustainable Energy, 5(2):023128, 2013.
9. K Chen, MX Song, X Zhang, and SF Wang. Wind turbine layout optimization with multiple hub height wind turbines using greedy algorithm. Renewable Energy, 96:676-686, 2016.
10. Ying Chen, Hua Li, Bang He, Pengcheng Wang, and Kai Jin. Multi-objective genetic algorithm based innovative wind farm layout optimization method. Energy Conversion and Management, 105:1318-1327, 2015.
11. Bryony DuPont, Jonathan Cagan, and Patrick Moriarty. An advanced modeling system for optimization of wind farm layout and wind turbine sizing using a multi-level extended pattern search algorithm. Energy, 106:802-814, 2016.
12. Alireza Emami and Pirooz Noghreh. New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. Renewable Energy. 35(7):1559-1564, 2010.
13. Yusuf Eroglu and Serap Ulusam Seckiner. Design of wind farm layout using ant colony algorithm. Renewable Energy. 44:53-62, 2012.
14. Ju Feng, Wen Zhong Shen, and Chang Xu. Multi-objective random search algorithm for simultaneously optimizing wind farm layout and number of turbines. In Journal of Physics: Conference Series, volume 753, page 032011. IOP Publishing, 2016.
15. Javier Serrano Gonzalez, Angel G Gonzalez Rodriguez, Jos’e Castro Mora, Jes’us Riquelme Santos, and Manuel Burgos Payan. Optimization of wind farm turbines layout using an evolutive algorithm. Renewable energy, 35(8):1671-1681, 2010.
16. SA Grady, MY Hussaini, and Makola M Abdullah. Placement of wind turbines using genetic algorithms. Renewable energy, 30(2):259-270, 2005.
17. Ritam Guha, Manosij Ghosh, Akash Chakrabarti, Ram Sarkar, and Seyedali Mirjalili. Introducing clustering based population in binary gravitational search algorithm for feature selection. Applied Soft Computing, 93:106341, 2020.
18. Xiaohong Han, Dengao Li, Ping Liu, and Li Wang. Feature selection by recursive binary gravitational search algorithm optimization for cancer classification. Soft Computing, 24(6):4407-4425, 2020.
19. Susheel Kumar Joshi and Jagdish Chand Bansal. Parameter tuning for meta-heuristics. Knowledge-Based Systems, page 105094, 2019.
20. Susheel Kumar Joshi, Anshul Gopal, Shitu Singh, Atulya K Nagar, and Jagdish Chand Bansal. A novel neighborhood archives embedded gravitational constant in gsa. Soft Computing, pages 1-17, 2021.
21. Mojtaba Ahmadieh Khanesar and David Branson. Xor binary gravitational search algorithm. In 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pages 3269-3274, IEEE, 2019.
22. R Timothy Marler and Jasbir S Arora. The weighted sum method for multi-objective optimization: new insights. Structural and multidisciplinary optimization, 41(6):853-862, 2010.
23. Grigoris Marmidis, Stavros Lazarou, and Eleftheria Pyrgioti. Optimal placement of wind turbines in a wind park using monte carlo simulation. Renewable energy, 33(7):1455-1460, 2008.
24. Seyedali Mirjalili, Gai-Ge Wang, and Leandro dos S Coelho. Binary optimization using hybrid particle swarm optimization and gravitational search algorithm. Neural Computing and Applications, 25(6):1423-1435, 2014.
25. Anshul Mittal. Optimization of the layout of large wind farms using a genetic algorithm. PhD thesis, Case Western Reserve University, 2010.
26. GPCDB Mosetti, Carlo Poloni, and B Diviacco. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. Journal of Wind Engineering and Industrial Aerodynamics, 51(1):105-116, 1994.
27. Thorsten Pohlert. The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR), 2014. Rpackage.
28. Sittichoke Pookpunt and Weerakorn Ongsakul. Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. Renewable Energy, 55:266-276, 2013.
29. R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2018.
30. Esmae Rashidi and Hossein Nezamabadi-pour. Feature subset selection using improved binary gravitational search algorithm. Journal of Intelligent & Fuzzy Systems, 26(3):1211-1221, 2014.
31. Esmae Rashidi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. Information sciences, 179(13):2232-2248, 2009.
32. Esmae Rashidi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Bgsa: binary gravitational search algorithm. Natural Computing, 9(3):727-745, 2010.
33. Niveditha Somu, Gauthama Raman MR, Akshya Kaveri, Kannan Krithivasan, Shankar Sriman VS, et al. Ibgss: An improved binary gravitational search algorithm based search strategy for qos and ranking prediction in cloud environments. Applied Soft Computing, 88:105945, 2020.
34. Abhijeet Singh Thakur, Tarun Biswas, and Pratyay Kuila. Binary quantum-inspired gravitational search algorithm-based multi-criteria scheduling for multi-processor computing systems. The Journal of Supercomputing, 77(1):796-817, 2021.
35. Xiaohui Yuan, Bin Ji, Shuangquan Zhang, Hao Tian, and Yanhong Hou. A new approach for unit commitment problem via binary gravitational search algorithm. Applied Soft Computing, 22:249-260, 2014.
36. Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on evolutionary computation, 11(6):712-731, 2007.
37. Qingfu Zhang, Wudong Liu, and Hui Li. The performance of a new version of moea/d on cec09 unconstrained mop test instances. In 2009 IEEE congress on evolutionary computation, pages 203-208. IEEE, 2009.