

Adaptive Hooke-Jeeves-evolutionary algorithm for linear equality constrained problems



Nazir Ahmad Chaudhry ¹, Muhammad Saeed ², Javaid Ali ², Muhammad Farhan Tabassum ^{2,*}, Muhammad Luqman ²

¹Department of Mathematics, Faculty of Engineering, Lahore Leads University, Lahore, 54000, Pakistan

²Department of Mathematics, University of Management and Technology, Lahore, 54000, Pakistan

ARTICLE INFO

Article history:

Received 3 March 2018

Received in revised form

20 June 2018

Accepted 2 July 2018

Keywords:

Derivative free methods

Hooke-Jeeves method

Genetic algorithm

Hybrid method

ABSTRACT

This paper proposes a novel hybrid algorithm called Genetic Algorithm based Simplex Adaptive Hooke and Jeeves (GA-SAHJ) method for solving equality constrained non-linear optimization problems. The proposed hybrid technique uses Genetic Algorithm (GA) as the global optimizer and a modified Hooke and Jeeves method for further refinements of the current solution within the landscape of a feasible region. The convergence proof of the modified approach is also provided. The effectiveness of the proposed GA-SAHJ method is demonstrated by applying it on six test instances each involving at least one equality constraint. The results witness that the proposed hybrid approach is capable of producing highly accurate and fully feasible solutions of the considered problems.

© 2018 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

The general optimization problem can be written as follow:

$$\left. \begin{array}{l} \text{minimize} \quad f(\underline{x}) \\ \text{subject to} \quad h_j(\underline{x}) = 0, \quad (j = 1, 2, \dots, J) \\ \quad \quad \quad g_k(\underline{x}) \leq 0, \quad (k = 1, 2, \dots, K) \end{array} \right\} \text{(A)}$$

where $f(\underline{x})$, $h_j(\underline{x})$ and $g_k(\underline{x})$ are functions of the design vector: $\underline{x} = (x_1, x_2, \dots, x_d)^T$ and $\underline{x} \in \mathbb{R}^d$.

Here each component x_i of \underline{x} are called design or decision variables, and they can be real continuous, discrete, or a mix of these two. The functions $f(\underline{x})$ is called the objective function or simply cost functions. The space spanned by the decision variables is called the design space or search space, whereas the space formed by the objective function values is called the solution space or response space. The equalities for $h_j(\underline{x})$ and inequalities for $g_k(\underline{x})$ are called constraints (Koziel and Yang, 2011).

It has been reported that the approaches like gradient based methods, linear programming and sequential quadratic programming perform poorly due to non-availability of derivatives, discontinuities

and multiple local minima (Miettinen et al., 2003). The derivative free methods, Hooke-Jeeves, Nelder-Mead Simplex and multidirectional search methods were used to obtain the optimal truss designs (Saeed et al., 2016). Tabassum et al. (2015) optimized the design of oxygen production system by using the Hooke-Jeeves, Nelder-Mead Simplex method (Tabassum et al., 2015) and the same problem by differential evolution along with 2-parameter exponential penalty function approach that gave slightly better results (Tabassum et al. 2016). There exist some examples on which the Nelder-Mead Simplex Algorithm method failed to find optimal solutions. Ali et al. (2017) proposed a simplex volume based novel strategy for rescuing the method from stagnations or complete failures. There are examples where these methods have successfully been applied to the problems involving variables up to few hundreds. In recent decades, DFMs have successfully been applied in multiple areas of engineering design, scheduling, bio-systems, molecular biology, neural networks, decision making and Image processing problems. Some well-known DFMs can be found in (Conn et al., 2009; Belegundu and Chandrupatla, 2011; Deb, 2004; Rao, 1996). These methods highly depend on provided initial guess to start and aim at local search. Moreover their performances depend strongly on the dimensions of optimization problems. Generally derivative free algorithms are designed for unconstrained optimization; therefore, they cannot be applied to constrained problems directly. The application of

* Corresponding Author.

Email Address: farhanuet12@gmail.com (M. F. Tabassum)

<https://doi.org/10.21833/ijaas.2018.08.015>

2313-626X/© 2018 The Authors. Published by IASE.

This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

DFMs to an unconstrained problem (A) is possible only if an equivalent unconstrained is formed. Among the several methods of constraint handling approaches, penalties functions are the most common approaches in the evolutionary algorithms to handle almost all of the types of constraints (equality and inequality). This approach embeds constraints into the objective function and designs an equivalent unconstrained problem. The embedded part is comprised of positive multiples of degrees of constraint violations act as penalization to the algorithm for exploring any infeasible solution (Bertsekas, 1982; Joines and Houck, 1994; Michalewicz and Janikow, 1991).

Penalty functions also inherit some drawbacks, such as non-optimal tuning of penalty parameters and raised complexity, high non-linearity, non-convexity or non-differentiability of resulting objective function. For example the main drawback of both of annealing penalty (Kirkpatrick et al., 1983) and static penalty function is a high number of parameters which require careful fine tune for better results (Michalewicz, 1995). In dynamic penalties the quality of the obtained solution is very sensitive to changes in its fixed constants and there are no clear guidelines regarding the sensitivity of the approach to different values of constants (Smith and Coit, 1997). The drawback of adaptive penalties is the issue of choosing the generational gap that provides reasonable information to guide the search (Rasheed, 1998). In the practical applications of co-evolutionary penalties in EAs, there arise necessities of four additional highly sensitive parameters (Coello, 1999; 2000), making it difficult to use. Death penalty is confined by not explaining and exploiting any kind of information from the infeasible points that can be produced by the EA to supervise and lead the search (Coit and Smith, 1996). In In genetic algorithms a 2-paramer exponential penalty function approach was used for handling equality and inequality constraints which provided comparable results on various well known constraint optimization problems (Chaudhry et al., 2009). Boudjehem et al. (2011) gave the idea to reduce the dimension of the optimization problem to a mono-dimensional that equipped new algorithm with the ability to determine a narrow space around the global optimum by reducing the number of evaluations.

Genetic algorithm (GA) and Hooke and Jeeves (HJ) method both are not free from encountering similar drawbacks when applied to the penalized objective function, especially when linear equality constraint is accompanied. Linear equality constraint constitutes narrow feasible region and hence appears to be resistant to global explorations of GA and local explorations of the moves of HJ method which can be noticed from Fig. 1. In this way GA and HJ may fail without making a single improvement in the objective function value.

To overcome the shortcomings of HJM and GA on equality constrained problems their hybrid approach is needed to be constructed. Hybridization of

different algorithms by uniting their strengths is an effective approach in meta-heuristics, especially in evolutionary algorithms. The examples of such hybridization can be found in (Miettinen et al., 2003; Bertsekas, 1982; Coit and Smith, 1996; Hooke and Jeeves, 1961; Coelho and Mariani, 2009). Pandian (2010) proposed a meta-hetaeristic approach by hybridizing simulated annealing and genetic algorithm to search for the best feasible solutions to the decision variables for solving a nonlinear objective function in industrial management problems.

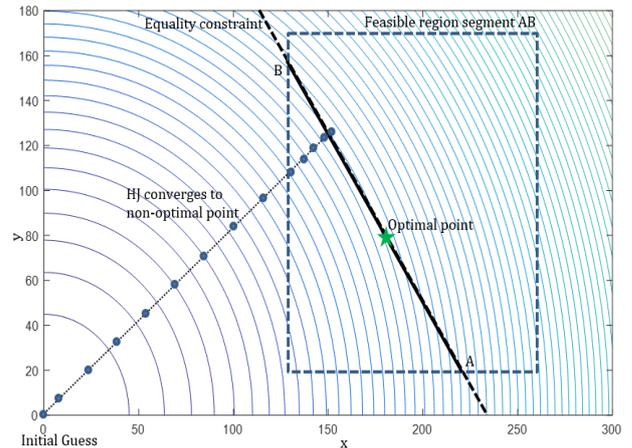


Fig. 1: Stagnation of HJM to a non-optimal point

In this paper we modify HJM by utilizing advantages of equality constraints rather than eliminating it. The modified HJM is named as Simplex Adaptive HJM (SAHJ) method. The proposed SAHJM is further hybridized with GA for global search and the resulting method is named as Genetic Algorithm based Simplex Adaptive Hooke-Jeeves (GA-SAHJ) method.

Rest of the paper is organized as follows. Section 2 exhibits the basic concepts, definitions and the construction of SAHJ algorithm, in Section 3 the convergence proof of SAHJ for unconstrained optimization problems is presented; Section 4 consists of flowchart of the proposed hybrid GA-SAHJ algorithm and numerical examples are presented in Section 5. In the last section, some concluding remarks are also presented.

2. Simplex adaptive Hooke and Jeeves method (SAHJ)

The proposed modification concerns with the constrained problems involving a linear equality constraint of the form:

$$g(\underline{x}) = \underline{u} \cdot \underline{x} - b = 0 \tag{1}$$

Such a constraint defines hyper-plane with a normal \underline{u} in the search space. For each of such hyper planes one or more members of the basis of search space become redundant. Therefore, the effective dimensions of the feasible region determined by (1) become lesser than those of the optimization

problem. We require a set of linearly independent vectors which span the feasible region and lie in planes parallel to (1). Once an initial guess lying in the feasible region is found then the proposed scheme guarantees that no infeasible point is explored. Following assumptions are important for practicability of the scheme.

Assumption 2.1: The hyper-plane (1) does not pass through the origin i.e.; $b \neq 0$.

Assumption 2.2: The normal \underline{u} is such that $u_i \neq 0 : 1 \leq i \leq n$.

The assumption 2.1 is quite possible because otherwise a parallel hyper-plane with arbitrary non-zero b can be used. The proposed modified algorithm requires bases of the hyper-plane through origin and parallel to (1). We will refer such bases as parallel bases. The assumption 2.2 is concerned with the case when any $u_i = 0$ which will be addressed in coming sections. The equality constraint (1) can be expressed as:

$$\sum_{i=1}^n u_i x_i = b$$

Assumptions 1 and 2 guarantee the existence of distinct points $P^{(i)} = (x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \dots, x_n^{(i)})$, $1 \leq i \leq n$ such that:

$$x_j^{(i)} = \begin{cases} \frac{b}{u_j} & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases}$$

These points along with the origin form a non-degenerate simplex in R^n . The face opposite to the origin is a part of the hyper-plane (1). Replacing the origin by centroid of its opposite face we get a degenerate simplex in R^n but it is a non-degenerate n -simplex in hyper-plane (1) whose active dimensions are $(n-1)$.

Definition 2.1: By inactive search direction we mean any direction \underline{d} such that

$$\underline{u} \cdot \underline{d} \neq 0$$

Inactiveness is based on the fact that such a direction does not help in reducing the efforts of exploration of the points in the feasible region of the problem involving (1).

Definition 2.2: The Constrained Minimum. A point \underline{x}^* is said to be a stationary point of $f(\underline{x})$ subject to $g(\underline{x}) = \underline{u} \cdot \underline{x} - b = 0$ if there is a non-zero scalar κ^* such that $\nabla f(\underline{x}^*) = \kappa^* \nabla g(\underline{x}^*)$, where κ^* is called Lagrange multiplier (Joines and Houck, 1994).

3. SAHJ algorithm

The following steps interpret the exploitation of the sub-simplex for determination of search

directions which form the active spanning set of the hyper-plane (1).

Step 1: Choose the HJ parameter:

$$\underline{\Delta} = [\Delta_1, \Delta_2, \Delta_3, \dots, \Delta_n], 0 < \alpha < 1, \lambda > 1,$$

termination parameter: ξ , Initial guess: $\underline{x}^{(0)}$.

Set $m = 1$.

Step 2: (a) Translate $\underline{x}^{(0)}$ to the hyper-plane by

$$\underline{z} = \underline{x}^{(0)} - \left[\left(\frac{\underline{u} \cdot \underline{x}^{(0)} - b}{\|\underline{u}\|^2} \right) \underline{u} \right]$$

(b) Set $\underline{x}_b = \underline{z}$.

(c) Find the vertices of sub-simplex and compute its centroid $G = \frac{1}{n} \sum_{i=1}^n P^{(i)}$

(d) Compute $\underline{v}_i = G - P^{(i)}$, $1 \leq i \leq n$ and construct n possible parallel bases of $(n - 1)$ distinct \underline{v}_i .

Step 3: Choose m^{th} parallel basis and apply Gram-Schmidt process to get

$$\underline{d}_i : 1 \leq i \leq (n - 1).$$

Step 4: Is $\|\underline{\Delta}\| < \xi$? If yes, Terminate. Else Go to Step 5.

Step 5: Perform an exploratory move on \underline{x}_b to get \underline{x} .

If $f(\underline{x}) < f(\underline{x}_b)$, set $\underline{x}_e = \underline{x}$ and Go to step 6; else set $\underline{\Delta} = \underline{\Delta} \cdot \lambda$. Set $m = \text{mod}(m, n) + 1$ and go to Step 3.

Step 6: Find $\underline{x}_p = \underline{x}_b + \lambda (\underline{x}_e - \underline{x}_b)$ and apply another exploratory move with \underline{x}_p as the base point. Suppose \underline{x}_E is the result. Go to step 7.

Step 7: If $f(\underline{x}_E) < f(\underline{x}_e)$, set $\underline{x}_b = \underline{x}_e$, $\underline{x}_e = \underline{x}_E$ and Go to step 6; else set $\underline{x}_b = \underline{x}_e$, $\underline{\Delta} = \alpha \underline{\Delta}$ and Go to step 4.

The proposed SAHJ differs from HJ method in following three aspects.

- (i) The step length reduction parameter α is chosen randomly in the range $[0.5, 1)$.
- (ii) The search directions may not be along coordinate axes and are fewer than the dimensions.
- (iii) The set of search directions is updated after every failed exploratory move at step 5.

4. Convergence of SAHJ method

Before establishing main convergence results of the algorithm we state and prove two important results which will be helpful in proving convergence of the proposed algorithm.

Claim 4.1: The search directions generated by SAHJ at step 2 are all active.

Proof: According to the definition (1), we need to show that each \underline{v}_i is orthogonal to \underline{u} .

$$\begin{aligned} \underline{u} \cdot \underline{v}_i &= \underline{u} \cdot (G - P^{(i)}) \\ &= \underline{u} \cdot \left(\frac{1}{n} \sum_{j=1}^n P^{(j)} - P^{(i)} \right) \\ &= \frac{1}{n} \sum_{j=1}^n \underline{u} \cdot (P^{(j)} - P^{(i)}) \\ &= \frac{1}{n} \sum_{j=1}^n ((\underline{u} \cdot P^{(j)} - b) - (\underline{u} \cdot P^{(i)} - b)) = 0 \end{aligned}$$

Which shows that every \underline{v}_i lies in some hyper-plane parallel to (1).

Claim 4.2: The algorithm SAHJ can generate the points lying only in the hyper-plane (1).

Proof: The step 2 (a), 2(b) ensure that SAHJ starts with a feasible initial guess. Consider the initial guess be \underline{x}_b . Claim 3.1 guarantees that step 2 (d) generate only active search directions. On contrary suppose that there is a step length Δ_k and a direction \underline{d}_k such that the point $\underline{x}_b + \Delta_k \underline{d}_k$ generated by exploratory move does not lie in the feasible region of (7), i.e.,

$$\begin{aligned} \underline{u} \cdot (\underline{x}_b + \Delta_k \underline{d}_k) - b &\neq 0 \\ \Rightarrow \underline{u} \cdot \underline{d}_k &\neq 0 \end{aligned}$$

That is contradiction to the activeness of \underline{d}_k . Now we step forward to our main convergence results with following two assumptions.

Assumption 4.1: The set $A = \{ \underline{x} : f(\underline{x}) = r; r \in R \} \cap \{ \underline{x} : g(\underline{x}) = 0 \}$ is compact and the function $f(\underline{x})$ is continuously differentiable on A.

Assumption 4.2: The provided initial guess lies within the box constraints and there is a unique stationary point in a finite neighborhood of the point

$$\underline{z} = \underline{x}^{(0)} - \left[(\underline{u}, \underline{x}^{(0)}) - b \right] \frac{\underline{u}}{\|\underline{u}\|^2}$$

Main result: Under the assumptions 2.1, 2.2, 3.1 and 3.2 the algorithm SAHJ converges to a stationary point of $f(\underline{x})$ subject to $g(\underline{x}) = 0$ and $x_i^L < x_i < x_i^U$; $x_i^L, x_i, x_i^U \in R$.

Proof: The initial guess provided to the algorithm lies within the box constraints. Step 2 (a) translates the initial guess to the feasible region. By claim 3.1 the algorithm generates only active search directions and claim 3.2 ensures that the algorithm can generate only feasible points. Algorithm generates a set of $(n - 1)$ linearly independent active search direction. This set can be extended to a basis of R^n by including only one element, namely $\underline{d}_n = \frac{\underline{u}}{\|\underline{u}\|}$. Every exploratory move along \underline{u} with non-zero step length generates an infeasible point. Using penalized objective function such a point results in an

unsuccessful exploratory move. Suppose at any iteration k_o the current base point be $\underline{x}_b^{(k_o)}$.

The proof is dependent on the successful and unsuccessful exploratory moves. Suppose that the exploratory moves are unsuccessful for iterations $k_o + 1, k_o + 2, k_o + 3, \dots$. Then for any $j > k_o$, the current base point $\underline{x}_b^{(j)}$ is the same as $\underline{x}_b^{(k_o)}$ and is such that:

$$\begin{aligned} \min_{1 \leq i \leq n} \left\{ f \left(\underline{x}_b^{(j-1)} \pm \alpha^{(j-1)} \Delta_i \underline{d}_i \right) \right\} &\geq f \left(\underline{x}_b^{(j-1)} \right) \\ \min_{1 \leq i \leq n} \left\{ f \left(\underline{x}_b^{(k_o)} \pm \alpha^{(j-1)} \Delta_i \underline{d}_i \right) \right\} &\geq f \left(\underline{x}_b^{(k_o)} \right) \text{ for each } j \end{aligned}$$

first consider

$$\begin{aligned} f \left(\underline{x}_b^{(k_o)} + \alpha^{(j-1)} \Delta_i \underline{d}_i \right) - f \left(\underline{x}_b^{(k_o)} \right) &\geq 0 \text{ for each } i \\ \frac{f \left(\underline{x}_b^{(k_o)} + \alpha^{(j-1)} \Delta_i \underline{d}_i \right) - f \left(\underline{x}_b^{(k_o)} \right)}{\alpha^{(j-1)} \Delta_i} &\geq 0 \end{aligned}$$

as $j \rightarrow \infty$ then $\alpha^{(j-1)} \rightarrow 0$ this implies that

$$\left(\nabla f \left(\underline{x}_b^{(k_o)} \right) \right)^T \cdot \underline{d}_i \geq 0 \tag{2}$$

similarly

$$f \left(\underline{x}_b^{(k_o)} - \alpha^{(j-1)} \Delta_i \underline{d}_i \right) - f \left(\underline{x}_b^{(k_o)} \right) \geq 0$$

implies that

$$\left(\nabla f \left(\underline{x}_b^{(k_o)} \right) \right)^T \cdot \underline{d}_i \leq 0 \tag{3}$$

from (2) and (3)

$$\left(\nabla f \left(\underline{x}_b^{(k_o)} \right) \right)^T \cdot \underline{d}_i = 0, \text{ for } i = 1, 2, 3, \dots, (n - 1)$$

Now we consider that $\underline{x}_b^{(k_o)}$ is not a stationary point. Then after a finite number k_1 of iterations, the exploratory move must be successful otherwise a reduction in step length parameter occurs. Continuing the process we get a strictly decreasing sequence $\{ \underline{x}_b^{(k_m)} \}$ such that

$$f \left(\underline{x}_b^{(k_1)} \right) > f \left(\underline{x}_b^{(k_2)} \right) > f \left(\underline{x}_b^{(k_3)} \right) \dots$$

Since $f(\underline{x})$ is bounded below so due to monotonicity $f \left(\underline{x}_b^{(k_m)} \right)$ converges to its lower bound. The compactness of feasible region implies that there is a feasible point \underline{x}^* such that

$$\lim_{m \rightarrow \infty} \underline{x}_b^{(k_m)} = \underline{x}^*$$

and

$$\lim_{m \rightarrow \infty} f \left(\underline{x}_b^{(k_m)} \right) = f \left(\underline{x}^* \right)$$

Considering the unsuccessful exploratory moves with base point \underline{x}^* and by the virtue of (2) and (3) we arrive at:

$$(\nabla f(\underline{x}^*))^T \cdot \underline{d}_i = 0 \text{ for } 1 \leq i \leq (n - 1)$$

Which shows that $(\nabla f(\underline{x}^*))$ is orthogonal to every active direction. Hence there is a non-zero constant κ^* such that

$$\begin{aligned} (\nabla f(\underline{x}^*)) &= \kappa^* \underline{u} \\ \Rightarrow \nabla f(\underline{x}^*) &= \kappa^* \nabla g(\underline{x}^*) \end{aligned}$$

This proves the existence of constrained stationary point of the problem.

We have shown that in case of all failed exploratory moves the algorithm converges to a stationary point and successful exploratory moves provide descent change which leads to existence of a stationary point.

4.1. Extension to penalized objective function

The algorithm SAHJ can converge to a constrained minimum point of the said problem when the assumption 2.2 is not fulfilled but the search directions are constructed in a special way. For illustration consider $u_i = 0$ for some $i \in \{1, 2, 3, \dots, n\}$ then the hyper-plane (1) is parallel to i^{th} coordinate axis. In this case step 2 (d) can generate only $(n - 1)$ search directions. The i^{th} search direction is calculated as:

$$\underline{v}_i = G + \underline{e}^{(i)},$$

provided $\underline{e}^{(i)}$ is the unit vector along i^{th} coordinate axis. Along with such extensions the above convergence results are also valid.

The unconstrained penalized objective function used here is of the form (Bertsekas, 1982; Runarsson and Yao, 2000):

$$\mathcal{F}(\underline{x}) = f(\underline{x}) + \mathcal{R} \varphi(\underline{x}) \quad (4)$$

where, $\varphi(\underline{x})$ is the penalty function and \mathcal{R} is the penalty factor. For every feasible solution $\mathcal{F}(\underline{x})$ coincides with $f(\underline{x})$ and for every infeasible solution $\mathcal{F}(\underline{x}) > f(\underline{x})$.

The convergence for penalized function $\mathcal{F}(\underline{x})$ can also be proved. There is coincidence between $\nabla \mathcal{F}(\underline{x})$ and $\nabla f(\underline{x})$ for explorations along all the active search directions. While using penalized function the exploration is made along the members of a basis of the search space R^n which can be obtained by including a unit vector \hat{u} along \underline{u} in the parallel basis of the hyper-plane through origin. The resulting linear basis and the counter parts of its members form a maximal positive basis for the search space. In this way a finite set $\{\underline{B}_m; 1 \leq m \leq n\}$ of positive bases is formed and exploited between steps 3 to step 5. The proof of existence of a stationary point for penalized function $\mathcal{F}(\underline{x})$ is based on its differentiability and use of finite set of maximal positive bases. The proof of convergence of

SAHJ on $\mathcal{F}(\underline{x})$ coincides with the one convergence proof of directional direct search method (Conn et al., 2009).

5. Genetic algorithm based simplex adaptive Hooke and Jeeves algorithm (GA-SAHJ)

Genetic Algorithm (GA) is a well-practiced evolutionary global search algorithm. We associate GA with SAHJ to design global search method which ultimately explores the local search to converge to the global solution. The resulting combined approach is named as Genetic Algorithm based Simplex Adaptive Hook and Jeeves (GA-SAHJ) method for further use. The flow chart of the hybrid algorithm is given in Fig. 2.

In the following subsection we present numerical results for evaluating the efficiency of the proposed algorithm on selected test cases.

6. Numerical performances

For comparisons of performances of GA-SAHJ and GA-HJ we considered some test problems from the literature. For both of the methods the penalized objective functions given by (10) were constructed by using approach in (Michalewicz and Schoenauer, 1996) for inequality constraints and a square penalty function given in (Deb, 2004) for equality constraint. For numerical implementations of GA-HJ and GA-SAHJ methods, step length used were $\Delta_i=1$, the step length reduction parameter $\alpha = 0.5$ for test problems 1-6 and two termination criteria (i) $\xi = 10^{-20}$ and (ii) number of function evaluations becomes larger than $10000 \times$ (the number of variables).

GA is implemented through optimization toolbox in MATLAB (Chipperfield et al., 1994) with parameters: Population size = 50, crossover fraction = 0.8, mutation rate = 0.01, elite count = 2. Remaining parameters used are same as the default settings. Six test problems of constrained minimization were taken from (Hock and Schittkowski, 1981; Himmelblau, 1972). GA is run for 20 to 30 generations and the obtained final point was retained. 200 iterations of both of SAHJ and HJ were recorded with retained point as initial guess. The Figs. 3-8 show the convergence of three approaches GA-SAHJ, GA-HJ and GA.

Test Problem 6.1: Minimize

$$f(\underline{x}) = (x_1 + x_2)^2 + (x_2 + x_3)^4$$

subject to

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= -1 \\ -10 \leq x_j &\leq 10; \quad j = 1,2,3. \end{aligned}$$

The problem has global minimum at

$$\underline{x}^* = (0.5, -0.5, 0.5)$$

with

$$f(\underline{x}^*) = 0.$$

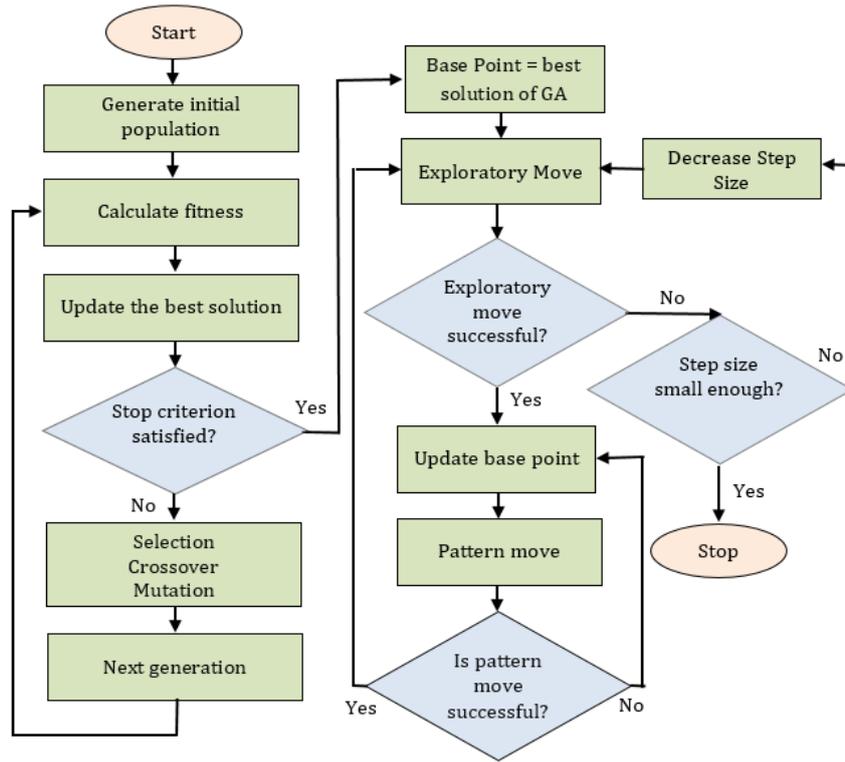


Fig. 2: Genetic algorithm based simplex adaptive Hooke and Jeeves algorithm

Test Problem 6.2: Minimize

$$f(\underline{x}) = (x_1 + 3x_2 + x_3)^2 + 4(x_1 + x_2)^2$$

subject to

$$\begin{aligned} x_1^3 - 6x_2 - 4x_3 &\geq -3 \\ x_1 + x_2 + x_3 &= 1 \\ 0 \leq x_1, x_2, x_3 &\leq 10 \end{aligned}$$

the constrained minimum of the problem is

$$f(\underline{x}^*) = 1 \text{ at } \underline{x}^* = (0, 0, 1)$$

Test Problem 6.3: Minimize

$$f(\underline{x}) = 2 - x_1x_2x_3$$

subject to

$$\begin{aligned} x_1 + 2x_2 + 2x_3 - x_4 &= 0 \\ 0 \leq x_1, x_2, x_3 &\leq 1, 0 \leq x_4 \leq 2 \end{aligned}$$

the optimal point is

$$\underline{x}^* = \left(\frac{2}{3}, \frac{1}{3}, \frac{1}{3}, 2\right)$$

with

$$f(\underline{x}^*) = \frac{52}{27}.$$

Test Problem 6.4: Minimize

$$f(\underline{x}) = (x_1 - 1)^2 + (x_2 - x_3)^2 + (x_4 - x_5)^2$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 + x_5 &= 5 \\ x_3 - 2(x_4 + x_5) + 3 &= 0 \\ -5 \leq x_1, x_2, x_3, x_4, x_5 &\leq 5 \end{aligned}$$

the minimal point of the problem is

$$\underline{x}^* = (1, 1, 1, 1, 1)$$

with

$$f(\underline{x}^*) = 0.$$

Test Problem 6.5: Minimize

$$f(\underline{x}) = (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 + 4x_4 - 7 &= 0 \\ x_3 + 5x_5 - 6 &= 0 \\ 10^{-6} \leq x_1, x_2, x_3 &\leq 10 \end{aligned}$$

the optimal solution is

$$\underline{x}^* = (1, 1, 1, 1, 1)$$

having minimum value

$$f(\underline{x}^*) = 0.$$

Test Problem 6.6: Minimize

$$f(\underline{x}) = -32.174 \left\{ 255 \log \left(\frac{x_1 + x_2 + x_3 + 0.03}{0.9x_1 + x_2 + x_3 + 0.03} \right) + \right.$$

$$280 \log \left(\frac{x_2 + x_3 + 0.03}{0.07x_2 + x_3 + 0.03} \right) + 290 \log \left(\frac{-x_3 + 0.03}{0.13x_3 + 0.03} \right)$$

subject to

$$x_1 + x_2 + x_3 - 1 = 0$$

$$0 \leq x_1, x_2, x_3 \leq 1$$

The best solution of the problem is

$$\underline{x}^* = (0.6178126908, 0.328202223, 0.05398508606)$$

with

$$f(\underline{x}^*) = -26272.51448.$$

Table 1 is based on 50 independent runs of GA-SAHJ and GA-HJ. Since GA solutions fall far away up to 2500 generations so those solutions are not included.

Table 1: Statistics for 50 independent runs for above test cases

Test Problem	Best		Mean		Worst	
	GA-SAHJ	GA-HJ	GA-SAHJ	GA-HJ	GA-SAHJ	GA-HJ
6.1	2.237×10^{-39}	3.074×10^{-8}	1.388×10^{-11}	0.0086663	1.39×10^{-10}	0.018513
6.2	1.0000	1.0000	1.0000	1.616	1.0000	3.9348
6.3	1.9261	1.932	1.9345	1.9474	2.000	2.000
6.4	9.728×10^{-6}	0.0070858	0.00086657	1.9374	0.0018652	5.8951
6.5	5.412×10^{-14}	8.156×10^{-7}	6.77×10^{-5}	8.155×10^{-5}	0.0010517	0.00045528
6.6	-26272.5145	-26272.5007	-26272.5145	-26250.0561	-26272.5145	-26176.4998

It can be observed from Figs. 3-8 that in all cases GA-SAHJ has found the minimum points much faster than GA-HJ. Moreover the GA-HJ method stagnates at some feasible point while its following iterations fail in producing further improvements in the objective function values, especially in test problems 6.3, 6.5, and 6.6.

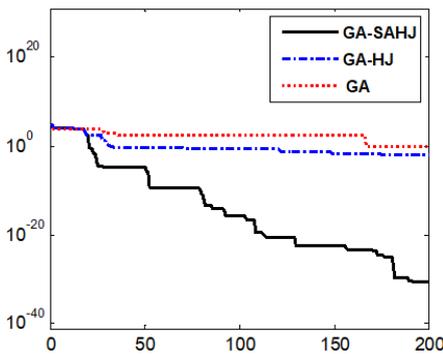


Fig. 3: Convergence of GA-SAHJ, GA-HJ and GA on test problem 6.1

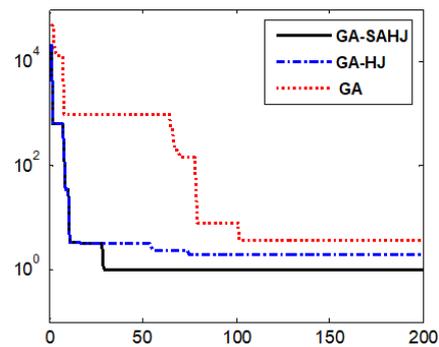


Fig. 4: Convergence of GA-SAHJ, GA-HJ and GA on test problem 6.2

7. Conclusion

This paper presented a convergent hybrid optimization method namely Genetic Algorithm based Simplex Adaptive Hooke and Jeeves Algorithm (GA-SAHJ) to efficiently solve the equality constrained optimization problems. The proposed algorithm guarantees to return a stationary point of the equality constrained problems. The effectiveness

of the GA-SAHJ has been demonstrated through six benchmark problems taken from the literature.

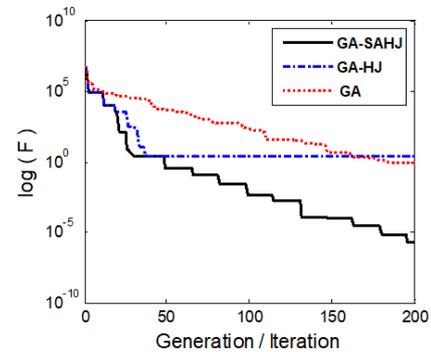


Fig. 5: Convergence of GA-SAHJ, GA-HJ and GA on test problem 6.3

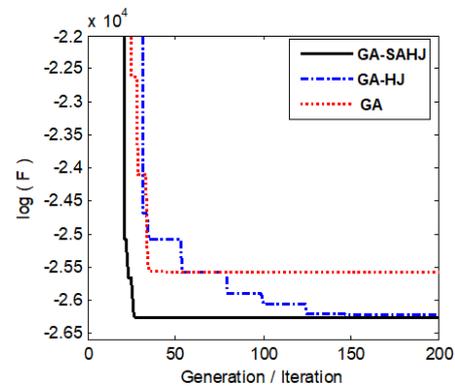


Fig. 6: Convergence of GA-SAHJ, GA-HJ and GA on test problem 6.4

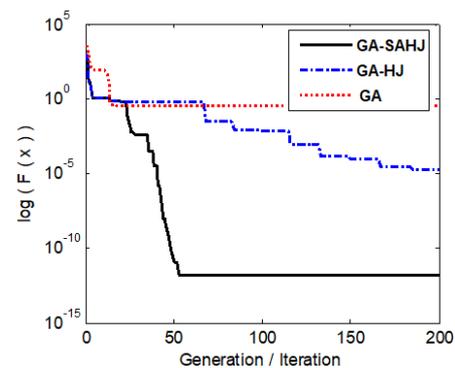


Fig. 7: Convergence of GA-SAHJ, GA-HJ and GA on test problem 6.5

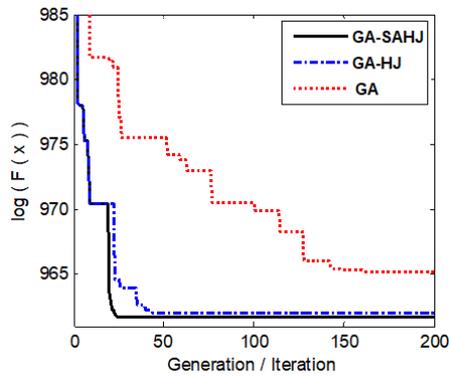


Fig. 8: Convergence of GA-SAHJ, GA-HJ and GA on test problem 6.6

The convergence curves in Figs. 3-8 show the superior convergence speed of GA-SAHJ over GA and GA-HJ. Moreover, the statistical results in Table 1 witness that best and mean fitness values produced by GA-SAHJ are highly better than those of GA-HJ under the similar implementation conditions. These results emphasize the capabilities of the GA-SAHJ for such problems and encourage its further implementation to other fields as well as engineering problems. As a future direction, it is intended to hybridize the proposed SAHJ with modern metaheuristics for solving economic dispatch problem with valve point loading effect resembling with the considered test instances. Additionally, suitable linearization schemes can be used to solve nonlinear equality constrained problems with the proposed scheme.

References

Ali J, Saeed M, Chaudhry NA, Tabassum MF, and Luqman M (2017). Low cost efficient remedial strategy for stagnated Nelder Mead simplex method. *Pakistan Journal of Science*, 69(1): 119-126.

Belegundu AD and Chandrupatla TR (2011). *Optimization concepts and applications in engineering*. Cambridge University Press, Cambridge, UK.

Bertsekas DP (1982). *Constrained optimization and Lagrange multiplier methods*. Massachusetts Institute of Technology, Athena Scientific Belmont, Massachusetts Academic Press, Cambridge, USA.

Boudjehem D, Boudjehem B, and Boukaache A (2011). Reducing dimension in global optimization. *International Journal of Computational Methods*, 8(3): 535-544.

Chaudhry NA, Ahmad MO, and Ali J (2009). Constraint handling in genetic algorithms by a 2-parameter exponential penalty function approach. *Pakistan Journal of Science*, 61(3): 122-129.

Chipperfield A, Fleming P, Pohlheim H, and Fonseca C (1994). *Genetic algorithm toolbox for use with MATLAB (version 1.2)*. University of Sheffield, Sheffield, UK.

Coelho LDS and Mariani VC (2009). An improved harmony search algorithm for power economic load dispatch. *Energy Conversion and Management*, 50(10): 2522-2526.

Coello CAC (1999). The use of a multiobjective optimization technique to handle constraints. In the *Second International Symposium on Artificial Intelligence (Adaptive Systems)*. Institute of Cybernetics, Mathematics and Physics, Ministry of Science Technology and Environment La Habana, Cuba: 251-256.

Coello CAC (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2): 113-127.

Coit DW and Smith AE (1996). Penalty guided genetic search for reliability design optimization. *Computers and Industrial Engineering*, 30(4): 895-904.

Conn AR, Scheinberg K, and Vicente LN (2009). Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points. *SIAM Journal on Optimization*, 20(1): 387-415.

Deb K (2004). *Optimization for engineering design, algorithms and examples*. Prentice Hall of India, Private limited, New Delhi, India.

Himmelblau DM (1972). *Applied nonlinear programming*. McGraw Hill Book-Company, New York, USA.

Hock W and Schittkowski K (1981). *Test examples for nonlinear programming codes*. Springer-Verlag, Berlin Heidelberg, Germany.

Hooke R and Jeeves TA (1961). Direct search solution of numerical and statistical problems. *Journal of the ACM (JACM)*, 8(2): 212-229.

Joines JA and Houck CR (1994). On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. In the *First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, IEEE, Orlando, USA: 579-584*.

Kirkpatrick S, Gelatt CD, and Vecchi MP (1983). Optimization by simulated annealing. *Science*, 220(4598): 671-680.

Koziel S and Yang XS (2011). *Computational optimization, methods and algorithms*. Springer, Berlin, Germany.

Michalewicz Z (1995). Genetic algorithms, numerical optimization, and constraints. In the *6th International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, USA: 195: 151-158*.

Michalewicz Z and Janikow CZ (1991). Handling constraints in genetic algorithms. In the *4th International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, USA: 151-157*.

Michalewicz Z and Schoenauer M (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1): 1-32.

Miettinen K, Mäkelä MM, and Toivanen J (2003). Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms. *Journal of Global Optimization*, 27(4): 427-446.

Pandian V (2010). Hybrid simulated annealing and genetic algorithms for industrial production management problems. *International Journal of Computational Methods*, 7(2): 279-297.

Rao SS (1996). *Engineering optimization: Theory and practice*. New Age International, New Delhi, India.

Rasheed K (1998). An adaptive penalty approach for constrained genetic-algorithm optimization. In the *3rd Annual Genetic Programming Conference, Morgan Kaufmann, San Francisco, USA: 584-590*.

Runarsson TP and Yao X (2000). Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4(3): 284-294.

Saeed S, Chaudhry NA, Tabassum MF, and Rafiq M (2016). Optimal design of 16 bar truss structure by pattern search methods. *Pakistan Journal of Science*, 68(1): 74-81.

Smith AE and Coit DW (1997). Constraint handling techniques – penalty functions. In: Bäck T, Fogel DB, and Michalewicz Z (Eds.), *Handbook of Evolutionary Computation*. CRC Press, Boca Raton, Florida, USA.

Tabassum MF, Saeed M, Chaudhry NA, Ali J, and Sana A (2016). Modeling and optimizing oxygen production system using differential evolution. *Science International*, 28(1): 1-6.

Tabassum MF, Saeed M, Sana A, and Ahmad N (2015). Solution of 7 bar tress model using derivative free methods. *Proceedings of the Pakistan Academy of Sciences*, 52(3): 265-271.