10.2478/v10170-011-0014-2

Archives of Control Sciences
Volume 22(LVIII), 2012
No. 1, *pages 87–105*

A hybrid PSO approach for solving non-convex optimization problems

TIMOTHY GANESAN, PANDIAN VASANT and IRRAIVAN ELAMVAZUTHY

The aim of this paper is to propose an improved particle swarm optimization (PSO) procedure for non-convex optimization problems. This approach embeds classical methods which are the Kuhn-Tucker (KT) conditions and the Hessian matrix into the fitness function. This generates a semi-classical PSO algorithm (SPSO). The classical component improves the PSO method in terms of its capacity to search for optimal solutions in non-convex scenarios. In this work, the development and the testing of the refined the SPSO algorithm was carried out. The SPSO algorithm was tested against two engineering design problems which were; 'optimization of the design of a pressure vessel' (P1) and the 'optimization of the design of a tension/compression spring' (P2). The computational performance of the SPSO algorithm was then compared against the modified particle swarm optimization (PSO) algorithm of previous work on the same engineering problems. Comparative studies and analysis were then carried out based on the optimized results. It was observed that the SPSO provides a better minimum with a higher quality constraint satisfaction as compared to the PSO approach in the previous work.

Key words: Kuhn-Tucker conditions (KT), non-convex optimization, particle swarm optimization (PSO), semi-classical particle swarm optimization (SPSO)

1. Introduction

In recent years, many engineering design problems are classified as non-convex constrained optimization problems (see [1], [2] and [3]). Constrained optimization techniques (meta-heuristic) such as genetic algorithms [4], genetic programming [5] as well as particle swarm optimization (PSO) [6] have been developed and implemented to these problems.

Particle Swarm Optimization (PSO) is an optimization method developed based on the movement and intelligence of swarms. PSO integrates the concept of social interaction to problem solving and decision-making. PSO was developed by James Kennedy and Russell Eberhart [3] in 1995. Particle swarm is the system model or social structure

T. Ganesan is with Department of Mechanical Engineering, University Technology Petronas, Malaysia. P. Vasant, the corresponding Author, is with Fundamental and Applied Sciences Department, University Technology Petronas, e-mail: pandian_vasant@yahoo.com. I. Elamvazuthy is with Department of Electrical and Electronics Engineering, University Technology Petronas.

Received 20.01.2012.

of a basic creature which makes a group to have some objectives such as food searching and predator-prey interactions. Hence, the governing principle is that it is an important to take part with the most of the population in a group that has the same activity. Recently, PSO has been applied to various fields of power system including economic dispatch problems as well as in optimization problems in electric power systems (see Phuangpornpitak *et al* [7]).

As observed in [8], [9] and [10], classical optimization methods such as the Augmented Lagrange, Penalty and Karush-Kuhn-Tucker (KKT) techniques can be used to modify existing artificial intelligence algorithms such as neural networks. In [9] and [10], the Augmented Lagrange HNN was used to solve the economic load dispatch problem which involves only equality constraints. Similarly, in this work, the Kuhn-Tucker (KT) conditions [11] and the determinant of the Hessian matrix [12] is used to modify certain components of the PSO fitness function to provide the algorithm with the capacity to handle the nonlinearities, such as those encountered in engineering design problems. The SPSO algorithm is then used to solve two engineering design problems. These two problems are the 'optimization of the design of a pressure vessel' (P1) [13] and the 'optimization of the design of a tension/compression spring' (P2) [14]. The results are then compared against the results obtained with the modified PSO algorithm in [15].

This paper is organized as follows: Section 2 presents the problem description for P1 and P2, Section 3 discusses the development of the SPSO approach. The analysis and computational results are included in Section 4. In Section 5, some discussions on the computational experiments are provided. Section 6 presents the concluding remarks and recommendations for future research work.

2. Description of the engineering design application data

Two engineering design problems are considered in this work the 'optimization of the design of a pressure vessel' (P1) [13] and the 'optimization of the design of a tension/compression spring' (P2) [14].

The problem P1 is a pressure vessel design optimization problem. The optimization of the design of a compressed air storage tank that has a working static pressure of 3,000 psi and a minimum volume of 750 ft^3 is considered. This cylindrical vessel is capped (closed) at both ends by hemispherical heads. The tank is constructed of rolled steel plates and its shell is made in two halves. These two halves are joined by the longitudinal welds to form the cylinder. The objective is to minimize the total cost (including the cost of the materials that forms the welding [13]). The design variables involved are the thickness x_1, the thickness of the head x_2, the inner radius x_3, and the length of the cylindrical section of the vessel x_4. The variables x_1 and x_2 are discrete values which are integer multiples of 0.0625 inches. The formal statement of problem P1 is as the following:

Minimize:

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{1}$$

subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leqslant 0 \tag{2}$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \leqslant 0 \tag{3}$$

$$g_3(\vec{x}) = -x_3^2x_4^2\pi - \frac{4}{3}x_3^3\pi + 1296000 \leqslant 0 \tag{4}$$

$$g_4(\vec{x}) = x_4 - 240 \leqslant 0 \tag{5}$$

$$x_1 \geqslant 0.0625, \quad x_2 \leqslant 6.1875, \quad x_3 \geqslant 10, \quad x_4 \leqslant 200 \tag{6}$$

where x_1, x_2, x_3 and, x_4 are the decision variables, $g_1(\vec{x})$, $g_2(\vec{x})$, $g_3(\vec{x})$ and $g_4(\vec{x})$ are the constraints and $f(\vec{x})$ is the objective function.

The second problem considered in this work is the optimization of the design of a tension/compression spring (P2). The aim of this problem is to minimize the weight of a tension/compression spring, subject to certain constraints. The constraints considered are the minimum deflection, shear stress, surge frequency; limits on the outside diameter and on design variables. There are three design variables (or decision variables) in this problem which is the wire diameter x_1, the mean coil diameter x_2, and the number of active coils x_3. The formal representation of problem P2 is as the following:

Minimize:

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2 \tag{7}$$

subject to

$$g_1(\vec{x}) = \left(1 - \frac{x_2^3x_3}{7178x_1^4}\right) \leqslant 0 \tag{8}$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566x_1^3x_2 - x_1^4} + \frac{1}{5108x_1^2} - 1 \leqslant 0 \tag{9}$$

$$g_3(\vec{x}) = \left(1 - \frac{140.45x_1}{x_2^2x_3}\right) \leqslant 0 \tag{10}$$

$$g_4(\vec{x}) = \left(\frac{x_1 + x_2}{1.5} - 1\right) \leqslant 0 \tag{11}$$

$$0.05 \leqslant x_1 \leqslant 2, \quad 0.25 \leqslant x_2 \leqslant 1.3, \quad 2 \leqslant x_3 \leqslant 15 \tag{12}$$

where $x1$, x_2 and x_3 are the decision variables, $g_1(\vec{x})$, $g_2(\vec{x})$, $g_3(\vec{x})$ and $g_4(\vec{x})$ are the constraints and $f(\vec{x})$ is the objective function.

3. Semi-classical particle swarm optimization

The PSO algorithm was initially developed in 1995 by Kennedy and Eberhart [3]. This technique originates from two different ideas. The first idea was based on the observation of swarming or flocking habits of certain types of animals (for instance; birds, bees and ants). The second concept was mainly related to the study of evolutionary computation. The PSO algorithm works by searching the search space for candidate solutions and evaluating them to some fitness function with respect to the associated criterion. The candidate solutions are analogous to particles in motion (swarming) through the fitness landscape in search for the optimal solution. Initially the PSO algorithm chooses some candidate solutions (candidate solutions can be randomly chosen or be set with some *a priori* knowledge). Then each particle's position and velocity (candidate solutions) are evaluated against the fitness function. Then if the fitness function is not satisfied, then update the individual and social component with some update rule. Next update the velocity and the position of the particles. This procedure is repeated iteratively until the all candidate solutions satisfy the fitness function and thus converges into a fix position. It is important to note that the velocity and position updating rule is crucial in terms of the optimization capabilities of the PSO algorithm. The velocity of each particle in motion (swarming) is updated using the following equation.

$$v_i(n+1) = wv_i(n) + c_1r_1[\hat{x}_i(n) - x_i(n)] + c_2r_2[g(n) - x_i(n)] \tag{13}$$

where each particle is identified by the index i, $v_i(t)$ is the particle velocity and $x_i(t)$ is the particle position with respect to iteration (n). The parameters w, c_1, c_2, r_1 and r_2 are usually defined by the user. These parameters are typically constrained by the following inequalities:

$$w \in [0, 1.2],\ \ c1 \in [0, 2],\ \ c2 \in [0, 2],\ \ r1 \in [0, 1],\ \ r2 \in [0, 1]. \tag{14}$$

The term $wv_i(t)$ in equation (13) is the inertial term which keeps the particle moving in the same direction as its original direction. The inertial coefficient w serves as a dampener or an accelerator during the particles motion. The term $c_1r_1[\hat{x}_i(n) - x_i(n)]$ also known as the cognitive component functions as memory. Hence, the particle tends return to the location in the search space where the particle had a very high fitness value. The term $c_2r_2[g(n) - x_i(n)]$ known as the social component serves to move the particle to the locations where the swarm has moved in the previous iterations. After the computation of the particle velocity, the particle position is then calculated as follows:

$$x_i(n+1) = x_i(n) + v_i(n+1) \tag{15}$$

The iterations are then continued until the all candidate solutions are at their fittest positions in the fitness landscape and some stopping criterion which is set by the user is met. For more comprehensive texts on PSO methods, refer to [16], [17] and [18].

In this work, the fitness criterion described above is enhanced by using the determinant of the Hessian matrix and the KT conditions. The KT conditions are the necessary conditions for optimality. Let the multivariate optimization problem can generalized as the following:

$$\begin{aligned}
&\text{Max} \ \rightarrow \ f(x) \ \ \text{subject to}\\
&g_1(x_1,\ldots,x_n) \leqslant b_1\\
&g_2(x_1,\ldots,x_n) \leqslant b_2\\
&\vdots\\
&g_m(x_1,\ldots,x_n) \leqslant b_m
\end{aligned} \tag{16}$$

where $g_1(x_1,\ldots,x_n)$ are the constraints, $f(x)$ is the objective function, n is the maximum number of decision variables, m is the maximum number of constraints and $i,j \in N$. Then the Lagrangian of the system can be expressed as the following:

$$L(x_j,\bar{\lambda}_i) = f(x_j) - \sum_{i=1}^{m} \bar{\lambda}_i g_i(x_j) \ \text{ such that } \ \forall i \in [1,m] \ \ \textit{and} \ \ \forall j \in [1,n]. \tag{17}$$

It can be observed that the vector, x_j is the stationary point if for some $\lambda_i \in R$ the stationarity condition holds. The stationarity equation (differential of the Lagrangian) is as the following:

$$\nabla L(x_j,\bar{\lambda}_i) = \frac{\partial f(x_j)}{\partial x_j} - \sum_{i=1}^{m} \bar{\lambda}_i \frac{\partial g_i(x_j)}{\partial x_j} = 0 \ \text{ such that } \ \forall j \in [1,n]. \tag{18}$$

However, the KT conditions will only identify the stationary points but will not provide the information if the points are locally minimal or maximal. To overcome this setback, the Hessian matrix of the differential Lagrangian is employed. If the determinant of the Hessian matrix of the Lagrangian is positive definite (det$\left(H(\nabla L(\mathrm{x_j},\lambda_\mathrm{i})\right) > 0$), then the point is locally convex (minimum point) and vice versa. Therefore, the fitness criterion of the SPSO algorithm is embedded with the KT conditions [11] and the Hessian [12] of the Lagrangian to provide enhance its search accuracy for the local optimal.

3.1. Formulation for problem P1

For problem P1, the stationarity equation where $\forall i \in [1,4]$ are as the following:

$$\nabla L(x_1,\bar{\lambda}_i) = 0.6224x_3x_4 + 6.3322x_1x_4 + 39.68x_1x_2 + \bar{\lambda}_1 \tag{19}$$

$$\nabla L(x_2,\bar{\lambda}_i) = 1.7781x_3^2 + \bar{\lambda}_2 \tag{20}$$

$$\begin{aligned}
\nabla L(x_3,\bar{\lambda}_i) &= 0.6224x_1x_4 + 3.5562x_2x_3 + 19.84x_1^2 - 0.0193\bar{\lambda}_1\\
&\quad -0.00954\bar{\lambda}_2 + 2x_3x_4^2\pi\bar{\lambda}_3 + 4x_3^2\pi\bar{\lambda}_3
\end{aligned} \tag{21}$$

$$\nabla L(x_4,\bar{\lambda}_i) = 0.6224x_1x_3 + 3.1161x_1^2 + 2x_4x_3^2\pi\bar{\lambda}_3 - \bar{\lambda}_4. \tag{22}$$

For $\forall j \in [1,4]$, λ_1 and λ_2 can be calculated directly as the following:

$$\lambda_1 = -(0.6224x_3x_4 + 6.3322x_1x_4 + 39.68x_1x_2) \tag{23}$$

$$\lambda_2 = -1.7781x_3^2. \tag{24}$$

By substituting λ_2 into equation (21), λ_3 is be obtained and by substituting λ_3 into equation (22), λ_4 can be calculated.

$$\lambda_3 = \frac{-0.6224x_1x_4 - 3.5562x_2x_3 - 19.84x_1^2 + 0.0193\lambda_1 + 0.00954\lambda_2}{2\pi x_3x_4^2 + 4\pi x_3^2} \tag{25}$$

$$\lambda_4 = 0.6224x_1x_3 + 3.1161x_1^2 + 2\pi x_3^2x_4\lambda_3. \tag{26}$$

The Hessian matrix of the differential Lagrangian is represented as the following:

$$H(\nabla L(\mathrm{x_j},\lambda_\mathrm{i})) = \frac{\partial^2 L}{\partial x_i \partial x_j}. \tag{27}$$

In this work, the notation for the Hessian matrix of the differential Lagrangian is as in equation (28) and (29):

$$L_j^i = \frac{\partial^2 L}{\partial x_i \partial x_j} \tag{28}$$

$$L_j^i = \begin{pmatrix} L_1^1 & L_2^1 & L_3^1 & L_4^1 \\ L_1^2 & L_2^2 & L_3^2 & L_4^2 \\ L_1^3 & L_2^3 & L_3^3 & L_4^3 \\ L_1^4 & L_2^4 & L_3^4 & L_4^4 \end{pmatrix} \quad \text{for} \quad i,j \in [1,4]. \tag{29}$$

The expressions for the Hessian matrix entries for problem P1 is as the following:

$$\begin{aligned}
&L_1^1 = 6.3322x_4 + 39.68x_2,\ L_2^1 = 39.68x_1,\ L_3^1 = 0.6224x_4,\ L_4^1 = 0.6224x_3 + 6.3322x_1,\\
&L_1^2 = 0,\ L_2^2 = 0,\ L_3^2 = 3.5562x_3,\ L_4^2 = 0,\\
&L_1^3 = 0.6224x_4 + 39.68x_1,\ L_2^3 = 3.5562x_3,\ L_3^3 = 3.5562x_2 + 2\pi x_4^2\lambda_3 + 8\pi x_3\lambda_3,\\
&L_4^3 = 0.6224x_1 + 4\pi x_3x_4\lambda,\\
&L_1^4 = 0.6224x_3 + 6.2322x_1,\ L_2^4 = 0,\ L_3^4 = 0.6224x_1 + 4\pi x_3x_4\lambda_3,\ L_4^4 = 2\pi x_3^2\lambda_3
\end{aligned} \tag{30}$$

The determinant off the Hessian matrix of the differential Lagrangian is then solved using Laplace expansion [19].

3.2. Formulation for problem P2

For problem P1, the stationarity equations where $\forall i \in [1,3]$ are as the following:

$$\begin{aligned}\nabla L(x_1,\bar{\lambda}_i) = 2x_1x_2x_3 + 4x_1x_2 - \frac{4\bar{\lambda}_1 x_2^3 x_3}{7178x_1^5} + \frac{4\bar{\lambda}_2 x_2^2(37698x_2x_1^2 - 4x_1^3)}{(12566x_2x_1^3 - x_1^4)^2} - \\ -\frac{\bar{\lambda}_2(25132x_1x_2^2 - 3x_1^2x_2)}{(12566x_2x_1^2 - x_1^3)^2} + \frac{\bar{\lambda}_2}{2554x_1^3} + \frac{140.45\bar{\lambda}_3}{x_2^2x_3} - \frac{2}{3}\bar{\lambda}_4\end{aligned} \tag{31}$$

$$\nabla L(x_2,\bar{\lambda}_i) = x_1^2x_3 + 2x_1^2 + \frac{3\bar{\lambda}_1 x_2^2 x_3}{7178x_1^4} + \frac{\bar{\lambda}_2(4x_2x_1^4 + x_1^5)}{(12566x_2x_1^3 - x_1^4)^2} - \frac{280.9\bar{\lambda}_3 x_1}{x_2^3x_3} - \frac{2}{3}\bar{\lambda}_4 \tag{32}$$

$$\nabla L(x_3,\bar{\lambda}_i) = x_1^2x_2 + \frac{\bar{\lambda}_1 x_2^3}{7178x_1^4} - \frac{140.45\bar{\lambda}_3 x_1}{x_2^2x_3^2} \tag{33}$$

In problem P2, the distribution of the λ_i in the differential lagrangian equations (equation (27), (28)) makes λ_1 a free variable. Therefore, the values of λ_i can only be obtained iteratively by first guessing the free variable λ_1. Firstly, the equations (27) and (28) are simplified as the following:

$$\frac{\partial L}{\partial x_1} = \alpha_0 + \alpha_1\lambda_1 + \alpha_3\lambda_3 + \alpha_4\lambda_4 \tag{34}$$

$$\frac{\partial L}{\partial x_2} = \beta_0 + \beta_1\lambda_1 + \beta_3\lambda_3 + \beta_4\lambda_4 \tag{35}$$

$$\frac{\partial L}{\partial x_3} = \varphi_0 + \varphi_1\lambda_1 + \varphi_3\lambda_3 \tag{36}$$

where,

$$\begin{aligned}
&\alpha_0 = 2x_1x_2x_3 + 4x_1x_2,\ \ \alpha_1 = -\frac{4x_2^3x_3}{7178x_1^5},\ \ \alpha_3 = \frac{140.45}{x_2^2x_3},\ \ \alpha_4 = -\frac{2}{3}\\
&\alpha_2 = \frac{4x_2^2(37698x_2x_1^2 - 4x_1^3)}{(12566x_2x_1^3 - x_1^4)^2} - \frac{25132x_1x_2^2 - 3x_1^2x_2}{(12566x_2x_1^2 - x_1^3)^2} + \frac{1}{2554x_1^3}\\
&\beta_0 = x_1^2x_3 + 2x_1^2,\ \ \beta_1 = \frac{3x_2^2x_3}{7178x_1^4},\\
&\beta_2 = \frac{4x_2x_1^4 + x_1^5}{(12566x_2x_1^3 - x_1^4)^2},\ \ \beta_3 = -\frac{280.9x_1}{x_2^3x_3},
\end{aligned} \tag{37}$$

$$\beta_4 = -\frac{2}{3} \tag{38}$$

$$\varphi_0 = x_1^2 x_2, \;\; \varphi_1 = \frac{x_2^3}{7178 x_1^4}, \;\; \varphi_1 = -\frac{140.45\bar{\lambda}_3 x_1}{x_2^2 x_3^2}. \tag{39}$$

The iterative procedure begins by guessing the value of λ_1. Since $\frac{\partial L}{\partial x_3} = 0$, therefore λ_3 can be obtained as the following:

$$\lambda_3 = \frac{-\varphi_0 - \varphi_1 \lambda_1}{\varphi_2}. \tag{40}$$

Thus, the value of λ_1 is known. Next we simplify further equations (30) and (31) to isolate the λ_1 and λ_3 terms. The expression produced is as the following:

$$\frac{\partial L}{\partial x_1} = (\alpha_0 + m_0) + \alpha_2 \lambda_2 + \alpha_4 \lambda_4 \tag{41}$$

$$\frac{\partial L}{\partial x_2} = (\beta_0 + n_0) + \beta_2 \lambda_2 + \beta_4 \lambda_4 \tag{42}$$

where

$$m_0 = \alpha_1 \lambda_1 + \alpha_3 \lambda_3 \;\text{ and }\; n_0 = \beta_1 \lambda_1 + \beta_3 \lambda_3. \tag{43}$$

The expressions in equation (37) is then rearranged to express λ_2 in terms of λ_4:

$$\lambda_2 = \frac{-\alpha_4 \lambda_4 - (\alpha_0 + m_0)}{\alpha_2} \tag{44}$$

The λ_2 expression is then substituted into equation (38) to represent it only in terms of λ_4:

$$(\beta_0 + n_0) + \beta_2 \left(\frac{-\alpha_4 \lambda_4 - (\alpha_0 + m_0)}{\alpha_2} \right) + \beta_4 \lambda_4 = 0 \tag{45}$$

$$\lambda_4 = \frac{(\alpha_0 \beta_2 / \alpha_2) + (m_0 \beta_2 / \alpha_2) - \beta_0 - n_0}{\beta_4 - (\alpha_4 \beta_2 / \alpha_2)} \tag{46}$$

By substituting equation (42) into equation (40), the value of λ_2 is obtained. Thus, by initializing the value of the free variable λ_1, the values of λ_2, λ_3 and λ_4 is obtained iteratively until the stationary condition is satisfied.

The notation used in P1 is applied similarly in P2. The Hessian matrix of the differential Lagrangian is as in equation (36):

$$L_j^i = \begin{pmatrix} L_1^1 & L_2^1 & L_3^1 \\ L_1^2 & L_2^2 & L_3^2 \\ L_1^3 & L_2^3 & L_3^3 \end{pmatrix} \quad \text{for} \;\; i, j \in [1, 3]. \tag{47}$$

The expressions for the Hessian matrix entries for problem P2 is as the following:

$$
\begin{aligned}
L_1^1 &= \{2x_2x_3+4x_2\} - \left\{\frac{10\lambda_1 x_2^3 x_3}{3589x_1^6}\right\} \\
&+ \left\{\begin{array}{l} [(12566x_2x_1^3 - x_1^4)^2(4\lambda_1 x_2^2)(75396x_2x_1 - 12x_1^2) - \\ (4\lambda_1 x_2^2)(37698x_2x_1^2 - 4x_1^3)(25132x_2x_1^3 - 2x_1^4)(37698x_2x_1^2 - 4x_1^3)] \\ /(12566x_2x_1^3 - x_1^4)^4 \end{array}\right\} \\
&- \left\{\begin{array}{l} [(12566x_2x_1^2 - x_1^3)^2(25132\lambda_2 x_2^2 - 6\lambda_2 x_1 x_2) - 2(25132\lambda_2 x_1 x_2^2 - 3\lambda_2 x_2 x_1^2) \\ \times(12566x_2x_1^2 - x_1^3)(25132x_1x_2 - 3x_1^2)]/(12566x_2x_1^2 - x_1^3)^4 \end{array}\right\} \\
&- \left\{\frac{3\lambda_2}{2554x_1^4}\right\}
\end{aligned}
$$

$$
\begin{aligned}
L_2^1 &= \{2x_1x_3+4x_1\} - \left\{\frac{12\lambda_1 x_2^2 x_3}{7178x_1^5}\right\} \\
&+ \left\{\begin{array}{l} [(12566x_2x_1^3 - x_1^4)^2(\lambda_2)(16x_1^3x_2 + 5x_1^4) - \\ (2\lambda_2)(4x_1^4x_2 + x_1^5)(12566x_2x_1^3 - x_1^4)(37698x_2x_1^2 - 4x_1^3)] \\ /(12566x_2x_1^3 - x_1^4)^4 \end{array}\right\} - \left\{\frac{280.9\lambda_3}{x_3x_2^3}\right\}
\end{aligned}
$$

$$
\begin{gathered}
L_3^1 = 2x_1x_2 - \left\{\frac{4\lambda_1 x_2^3}{7178x_1^5}\right\} - \left\{\frac{140.45\lambda_3}{x_2^2x_3^2}\right\} \\
L_3^2 = x_1^2 + \left\{\frac{3\lambda_1 x_2}{7178x_1^4}\right\} + \left\{\frac{280.9\lambda_3 x_1}{x_2^3x_3^2}\right\} \\
L_1^3 = \{2x_1x_2\} - \left\{\frac{4\lambda_1 x_2^3}{7178x_1^5}\right\} - \left\{\frac{140.45\lambda_3}{x_2^3x_3^3}\right\} \qquad (48) \\
L_2^3 = x_1^2 + \left\{\frac{3\lambda_1 x_2^2}{7178x_1^4}\right\} + \left\{\frac{280.9\lambda_3 x_1}{x_2^3x_3^2}\right\} \\
L_3^3 = \left\{\frac{280.9\lambda_3 x_1}{x_2^2x_3^3}\right\}
\end{gathered}
$$

$$
\begin{aligned}
L_1^2 &= \{2x_3+4x_1\} - \left\{\frac{12\lambda_1 x_2^2 x_3}{7178x_1^5}\right\} \\
&+ \left\{\begin{array}{l} [(12566x_2x_1^3 - 4x_1^4)^2(4\lambda_2)(37698x_2^3x_1^2 - 4x_1^3x_2^2) - \\ (4\lambda_2)(37698x_1^2x_2^3 - 4x_1^3x_2^2)(25132x_1^3)(12566x_2x_1^3 - 4x_1^4)] \\ /(12566x_2x_1^3 - 4x_1^4)^4 \end{array}\right\} \\
&- \left\{\begin{array}{l} [(12566x_2x_1^2 - x_1^3)^2\lambda_2(50264x_1x_2 - 3x_1^2) - \lambda_2(50264x_1x_2 - 3x_1^2) \\ \times(25132x_1^2)(12566x_2x_1^2 - x_1^3)]/(12566x_2x_1^2 - x_1^3)^4 \end{array}\right\} - \left\{\frac{280.9\lambda_3}{x_3x_2^3}\right\}
\end{aligned}
$$

$$L_2^2=\left\{\frac{6\lambda_1 x_2 x_3}{7178x_1^4}\right\}+\left\{\begin{array}{l}[(12566x_2x_1^3-x_1^4)^2(4x_1^4\lambda_3)-\\(\lambda_3)(4x_1^4x_2+x_1^5)(25132x_1^3)(12566x_2x_1^3-x_1^4)]\\/(12566x_2x_1^3-x_1^4)^4\end{array}\right\}-\left\{\frac{842.7\lambda_3 x_1}{x_3x_2^4}\right\}$$

The determinant of the Hessian matrix in P2 is then obtained using the Laplace expansion [19].

The fitness criterion in the PSO algorithm is enhanced using the KT conditions and the Hessian matrix. The PSO algorithm runs through the search space then identifies the local minimum point with the aid of the Hessian. If these points satisfy the KT condition, then the solution is printed. This procedure is repeated until the position and the celocity of the particles in the swarm converge. The detail mechanism of the program is represented in the algorithm and the flowchart (see Fig. 1) below:

```
Step 1: Initialize number of particles, $i$ and the algorithm parameters $w$, $c_1$, $c_2$, $r_1$, $r_2$, $n_0$.
Step 2: Set initial position $x_i(n)$ and velocity $v_i(n)$.
Step 3: Compute individual and social influence.
Step 4: Compute position $x_i(n+1)$ and velocity $v_i(n+1)$ at next iteration.
Step 5: Compute $\lambda_i$.
Step 6: Evaluate fitness function:
        IF the candidate solutions are locally convex $H(\nabla L(\mathrm{x_j},\lambda_\mathrm{i}) > 0$ and the KT
        conditions holds go to step 7.
        ELSE update position $x_i$ and velocity $v_i$ and go to Step 3.
Step 7: Print solutions.
        IF the particles velocity and position have converged, then halt
        ELSE update position $x_i$ and velocity $v_i$ and go to Step 3.
```

where n is the number of iterations.

4. Computational results and analysis

The SPSO algorithm developed in this work is programmed using the C++ programming language on a personal computer (PC) with an Intel dual core processor running at 2 GHz.

4.1. Computational results for problem P1

In problem P1, the objective function $f(x)$, the design variables, number of iterations and the execution time obtained by the SPSO algorithm is compared against the modified PSO algorithm, [15]. In problem P1, a better optimal value for the objective function, $f(x)$ is achieved by SPSO algorithm as compared to the PSO [15] algorithm. In terms of constraint satisfaction, the SPSO algorithm performed very well. The PSO

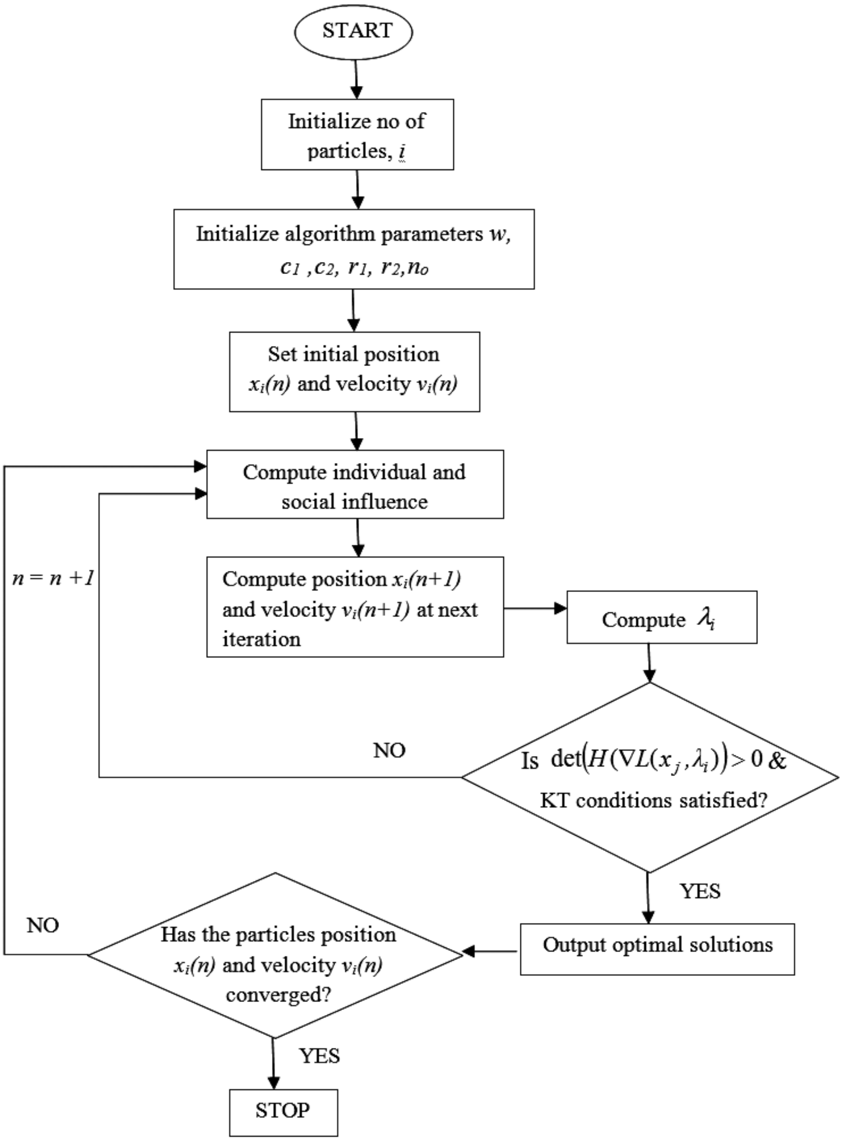
START
Initialize no of particles, i
Initialize algorithm parameters w, c_1, c_2, r_1, r_2, n_o
Set initial position $x_i(n)$ and velocity $v_i(n)$
Compute individual and social influence
Compute position $x_i(n+1)$ and velocity $v_i(n+1)$ at next iteration
Compute λ_i
Is $\det(H(\nabla L(x_j, \lambda_i)) > 0$ & KT conditions satisfied?
NO
YES
$n = n+1$
Output optimal solutions
Has the particles position $x_i(n)$ and velocity $v_i(n)$ converged?
NO
YES
STOP

Figure 1. Flowchart for the SPSO algorithm.

[15] algorithm seems to a certain degree have over-shot the $g_1(x)$ constraint which is $g_1(x) \leqslant 0$. Besides that, the SPSO and the PSO algorithm [15] have satisfied all the given constraints. The values of the lambda coefficients, $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ obtained at the

converged solution of the objective function for problem P1 is (-1176.54, -237.53, 1.68E-11, -31.6327).

The progression of the objective function, $f(x)$ with respect to the number of iterations, n for the SPSO algorithm is as in Fig. 2. In Fig. 2, the objective function, $f(x)$ obtained using the SPSO algorithm has the maximum value of 5703.96 at the 1st iteration and the minimum value of 2019.53 at the 31st iteration. Therefore, the optimal value is only reached at the 31st iteration with good constraint satisfaction. It must be noted that the progression of the objective function with respect to the iteration using the SPSO algorithm for problem P1 is stable and convergent. The progression of the determinant of the Hessian matrix with respect to the number of iterations, n for problem P1 is as in Fig. 3.

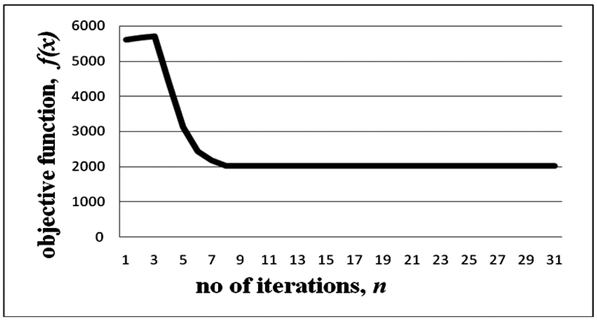

Figure 2. The objective function, $f(x)$ with respect to the number of iterations, n for the SPSO algorithm for P1.

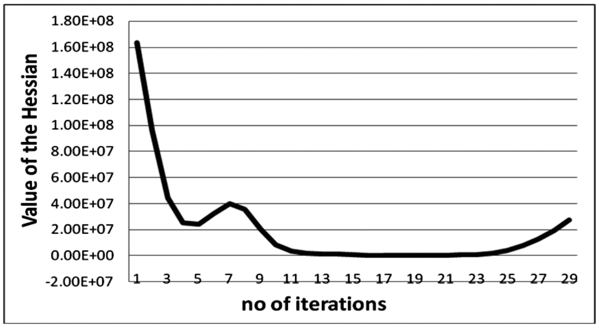

Figure 3. The value of the determinant of the Hessian matrix with respect to the number of iterations, n for the SPSO algorithm for P1.

In Fig. 3, it can be observed that the determinant of the Hessian matrix for the SPSO algorithm is positive definite from the 1st iteration until the 17th iteration. Then it takes

Table 3. Comparison of the optimization results for P1

P1	SPSO	PSO[15]
$f(x)$	2019.53	6059.71
x_1	0.959383	0.8125
x_2	4.48071	0.4375
x_3	11.5579	42.0984
x_4	75.815	176.637
No. of iterations	31	-
Execution time (secs)	0.09	-

Table 4. Comparisons of constraint satisfaction for P1

P1	SPSO	PSO[15]
$g_1(x)$	-0.736315	5.72E-09
$g_2(x)$	-4.37045	-3.59E-02
$g_3(x)$	-1.56E+15	-1.72E+08
$g_4(x)$	-164.185	-63.3634

the negative value until the 19th iteration. From then on, determinant of the Hessian matrix remains positive until convergence of the algorithm. The regions where the Hessian matrix is positive definite can be identified as regions where the associated points are locally convex. Thus this is the region where the potential local minimum to the objective function exists.

The computational results are as in Tab. 1. The performance of each algorithm in terms of constraints satisfaction, $g(x)$ is also obtained (where all values of $g(x) \leqslant 0$. The results of the constraints for problem P1 is as in Tab. 2:

4.2. Computational results for problem P2

For problem P2, the objective function $f(x)$, the design variables, number of iterations and the execution time obtained by the SPSO algorithm is compared against the modified PSO algorithm, [15].

The progression of the objective function, $f(x)$ with respect to the number of iterations, n for the KHN algorithm is as in Fig. 4. The values of the lambda coefficients, $(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ at the converged solution of the objective function for problem P1 is (0.0011, -0.00585508, 0.000491826, 0.118188).

In Fig. 4, the objective function, $f(x)$ obtained using the SPSO algorithm has the maximum value of 0.0105989 at the 1st iteration and the minimum value of 0.00820813

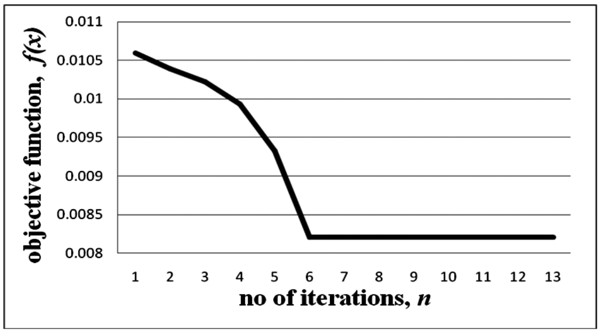

Figure 4. The objective function, $f(x)$ with respect to the number of iterations, n for the SPSO algorithm for P2.

at the 13th iteration. The optimal value is reached at the 13th iteration with good constraint satisfaction. The progression of the determinant of the Hessian matrix with respect to the number of iterations, n for problem P2 is as in Fig. 5.

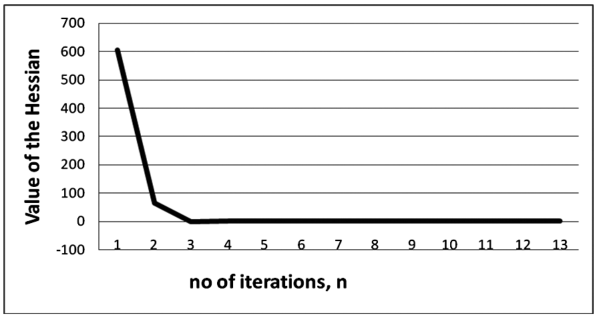

Figure 5. The value of the determinant of the Hessian matrix with respect to the number of iterations, n for the SPSO algorithm for P2.

Similar as in P1, the determinant of the Hessian matrix for the SPSO algorithm can be obtained. It can be observed to be positive definite from the 1st iteration until the 3rd iteration. From the 4th iteration onwards, the determinant of the Hessian matrix remains positive until convergence of the algorithm. Thus, except for the points obtained in the 3rd and the 4th iterations, the rest of the region with respect to the iteration has a potential local minimum to the objective function.

The computational results are as in Tab. 3. The performance of each algorithm in terms of constraints satisfaction, $g(x)$ is also obtained (where all values of $g(x) \leqslant 0$). The results of the constraints for problem P2 is as in Tab. 4.

Table 5. Comparison of the optimization results for P2

P2	SPSO	PSO[15]
$f(x)$	0.008208	0.01267
x_1	0.061122	0.05169
x_2	0.387959	0.35675
x_3	3.66314	11.2871
No. of i	13	-
Execution time (secs)	0.106	-

Table 6. Comparisons of constraint satisfaction for P2

P2	SPSO	PSO[15]
$g_1(x)$	-1.13504	-9.001053
$g_2(x)$	-0.42808	-0.134247
$g_3(x)$	-1.46E+01	-4.05379
$g_4(x)$	-0.70061	-0.727707

Table 7. Parameter setting for the SPSO algorithm for problems P1

Parameters Setting for P1	Values
Initial parameter (c_1, c_2, r_1, r_2, w)	(1.1, 1.2, 0.5, 0.5, 0.8)
Number of particles	4
Initial social influence (s_1, s_2, s_3, s_4)	(1.1, 1.05, 1.033, 1.025)
Initial personal influence (p_1, p_2, p_3, p_4)	(3, 6, 11, 18)

Table 5 and Tab. 6 shows the parameter setting for problems P1 and P2 respectively (based on heuristics) prior to the execution of the algorithm.

5. Discussion on computational experiment

The objective function, $f(x)$ obtained using the SPSO algorithm is be observed to decline in a decay fashion with respect to the iterations for problem P1. However in problem P1, as mentioned previously, the objective function decreases parabolically until convergence. Both the SPSO and the PSO [15] algorithms perform stable computation

Table 8. Parameter setting for the SPSO algorithm for problems P2

Parameters Setting for P1	Values
Initial parameter (c_1, c_2, r_1, r_2, w)	(1.1, 1.2, 0.5, 0.5, 0.02)
Number of particles	3
Initial social influence (s_1, s_2, s_3)	(1.1, 1.05, 1.03)
Initial personal influence (p_1, p_2, p_3)	(3, 6, 11)
initial guess for, λ_1	0.0001

without divergent solutions. From the results, it can be seen that the SPSO **approach performs better** in finding the optimal solution as compared to the PSO [15] approach. In terms of implementation, a pure PSO or meta-heuristic method is easier to develop as compared to the SPSO approach due to the analytical component that may include **lengthy derivations**.

From the computational results, it can be seen that the SPSO approach performs better in finding the optimal solution as compared to the PSO [15] approach. In P1 the PSO [15] algorithm is seen to have broken the constraint $g_1(x)$ to a very negligible degree. However, the SPSO algorithm abides all constraints in P1 and P2 satisfactorily (see Tab. 2 and Tab. 4). Thus, it can be said that the SPSO algorithm performs well in terms of **feasibility** for this two test problems. In problem P1 and P2, the SPSO algorithm is a **better minimizer** as compared to the PSO [15] algorithm. The PSO algorithm may have slight difficulties in handling the nonlinearities existing in the problem. However, the SPSO algorithm handles the **nonlinearities** very well. The SPSO algorithm also takes very little computational time approximately a fraction of a second (see Tab. 1 and Tab. 3).

The SPSO method outperforms the previous PSO [15] method. Due to the employment of a more accurate fitness criterion (the KT conditions and the Hessian matrix) which were derived analytically based on the problem statement, the **adaptability** of the SPSO algorithm to the problem is better as compared to the PSO [15] algorithm. This in effect minimizes the objective function further to the PSO [15] algorithm. Therefore, the analytical derivation of the fitness function improves the **robustness** of the algorithm. Hybridizing the SPSO method with algorithms like Tabu search (see [20], [21] and [22]) may provide it with a more efficient system for handling situations with multiple constraints for large scale problems and thus pave the way for obtaining a solution closer to the **global optimum**. From this work, it can also be inferred that the implementation of more analytical methods to improve the fitness function would provide a better generic algorithm for solving a wide range of problems. Due to the PSO component in the SPSO algorithm, the **stability and the convergence** of the computations are assured as seen in Fig. 2 and Fig. 4. A global solution with **better computational time** may be obtained by strengthening the analytical methods employed in the fitness function in the

SPSO algorithm. The **stopping criteria** for the SPSO algorithm is set such that, if the KT conditions are satisfied, the determinant of the Hessian matrix is positive definite and position/velocity of the particles converge; the program is then set to halt.

6. Conclusions and recommendations

In problem P1, the objective function is minimized further using the proposed SPSO algorithm (2019.53) as compared to the PSO (6059.71) algorithm used in [15]. As for problem P2, the objective function is minimized further using the proposed SPSO algorithm (0.0082081) as compared to the PSO (0.012665) algorithm [15]. In this work, a new local minimum is reached for the objective function of both test problems using the SPSO algorithm. The SPSO algorithm developed in this work is an enhancement of the PSO algorithm in terms of the capacity to handle highly non-convex problems. The KT conditions and the Hessian matrix improve the PSO algorithm in terms of its accuracy to search for optimal solutions. Further development in terms of the analytical techniques to enhance the fitness function, may improve the algorithms capacity to seek out optimal solutions in non-convex scenarios.

Nomenclature

$f(x)$	objective function,
i	index for constraint number,
j	index for variable number,
$g(x)$	constraint functions,
x_j	decision variables,
$v_i(t)$	particle velocity,
$x_i(t)$	particle position,
n	number of iterations,
w, c_1, c_2, r_1, r_2	PSO parameters,
s_1, s_2, s_3	parameters for particle social influence in PSO,
p_1, p_2, p_4	parameters for particle social influence in PSO,
b_i	invariant at the right hand side of the constraints,
$L(x_j, \lambda_i)$	the Lagrangian of the system,
λ_i	Lagrange coefficients,
L^i_j	Hessian matrix entries,
ω_i, β_i, α_i, m_0, n_0	variants introduced for derivations in P2.

References

[1] J.V. Outrata, M. Kocvara and J. Zowe: Nonsmooth approach to optimization problems with equilibrium constraints. Nonconvex optimization and its appli-

cations. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[2] D.P. BERTSEKAS: Nonlinear programming. Athena Scientific, Belmont, Massachusetts, second ed. (1999).

[3] Y. CHEN and M. FLORIAN: The nonlinear bilevel programming problem: Formulations, regularity and optimality conditions. *Optimization*, **32** (1995), 193-209.

[4] 4] J.H. HOLLAND: Adaptation in natural and artificial systems: An Introductory analysis with applications to biology. Control and Artificial Intelligence. MIT Press, USA, 1992.

[5] J.R. KOZA: Genetic programming: On the programming of computers by means of natural selection. MIT Press, USA, 1992.

[6] J. KENNEDY (1995) R. EBERHART: Particle swarm optimization. *IEEE Proceedings of the Int. Conf. on Neural Networks*, Perth, Australia, (1995).

[7] N. PHUANGPORNPITAK, W. PROMMEE, S.TIA and W. PHUANGPORNPITAK: A study of particle swarm technique for renewable energy power systems. *PEA-AIT Int. Conf. on Energy and Sustainable Development: Issues and Strategies*, Thailand, (2010), 1-7.

[8] V.N. DIEU and W. ONGSAKUL: Economic dispatch with emission and transmission constraints by augmented Lagrange Hopfield network. *Trans. in Power System Optimization (GJTO), www.pcoglobal.com/gjto.htm*, (2010), 77-83.

[9] V.N. DIEU and W. ONGSAKUL: Enhanced merit order and augmented Lagrange Hopfield network for ramp rate constrained unit commitment. *Proc. of IEEE Power System Society General Meeting*, Canada, (2006)

[10] Y. HUANG and C. YU: Improved Lagrange nonlinear programming neural networks for inequality constraints. *Proc. of Int. Joint Conf. on Neural Networks*, Orlando, Florida, USA, (2007).

[11] H.W. KUHN and A.W. TUCKER: Nonlinear programming. *Proc. of 2nd Berkeley Symp.*, Berkeley, University of California Press. (1951), 481-492.

[12] K. BINMORE and J. DAVIES: Calculus concepts and methods. Cambridge University Press, 2007.

[13] E. SANDGREN: Nonlinear integer and discrete programming in mechanical design optimization. *J. of Mechanical Deigns - T. ASME*, **112**(2), (1990), 223-229.

[14] A. BELEGUNDU: A study of mathematical programming methods for structural optimization. PhD thesis, Department of Civil Environmental Engineering, University of Iowa, Iowa, 1982.

[15] C.A. COELLO: Solving engineering optimization problems with simple constrained particle swarm optimizer. *Informatica*, **32** (2008), 319-326.

[16] Y. SHI and R. EBERHART: A modified particle swarm optimizer. *Proc. of the IEEE Int. Conf. on Evolutionary Computation*, (1998), 69-73.

[17] F. VAN DEN BERGH: An analysis of particle swarm optimizers. PhD thesis, University of Pretoria, 2001.

[18] E. ZITZLER, M. LAUMANNS and S. BLEULER: A tutorial on evolutionary multiobjective optimization. *In X. Gandibleux and others, editors, Metaheuristics for Multiobjective Optimisation*, Lecture Notes in Economics and Mathematical Systems, Springer, 2004.

[19] H.E. ROSE: Linear algebra. A pure mathematical approach. Springer, 2002, 57-60.

[20] F. GLOVER: Tabu search, Part I. *ORSA J. on Computing*, **1**(3), (1989), 190-206.

[21] J.A. BLAND and G.P. DAWSON: Tabu search and design optimization. *Computer - aided design*, **23**(3) (1991), 195-201.

[22] A. THESEN: Design and evaluation of tabu search algorithms for Mmultiprocessor scheuling. *J. of Hueristics*, **4** (1998), 141-160.