Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 139036, 21 pages
http://dx.doi.org/10.1155/2015/139036

Research Article

A Modified Hybrid Genetic Algorithm for Solving Nonlinear Optimal Control Problems

Saeed Nezhadhosein,[1] Aghileh Heydari,[1] and Reza Ghanbari[2]

[1] *Department of Applied Mathematics, Payame Noor University, Tehran 193953697, Iran*
[2] *Department of Applied Mathematics, Faculty of Mathematical Science, Ferdowsi University of Mashhad, Mashhad 9177948953, Iran*

Correspondence should be addressed to Saeed Nezhadhosein; s_nejhadhosein@pnu.ac.ir

Received 4 September 2014; Accepted 30 January 2015

Academic Editor: Alain Vande Wouwer

Here, a two-phase algorithm is proposed for solving bounded continuous-time nonlinear optimal control problems (NOCP). In each phase of the algorithm, a modified hybrid genetic algorithm (MHGA) is applied, which performs a local search on offsprings. In first phase, a random initial population of control input values in time nodes is constructed. Next, MHGA starts with this population. After phase 1, to achieve more accurate solutions, the number of time nodes is increased. The values of the associated new control inputs are estimated by Linear interpolation (LI) or Spline interpolation (SI), using the curves obtained from the phase 1. In addition, to maintain the diversity in the population, some additional individuals are added randomly. Next, in the second phase, MHGA restarts with the new population constructed by above procedure and tries to improve the obtained solutions at the end of phase 1. We implement our proposed algorithm on 20 well-known benchmark and real world problems; then the results are compared with some recently proposed algorithms. Moreover, two statistical approaches are considered for the comparison of the LI and SI methods and investigation of sensitivity analysis for the MHGA parameters.

1. Introduction

NOCPs are dynamic optimization problems with many applications in industrial processes such as airplane, robotic arm, bio-process system, biomedicine, electric power systems, and plasma physics [1].

High-quality solutions and the less required computational time are main issues for solving NOCPs. The numerical methods, direct [2] or indirect [3], usually have two main deficiencies, less accuracy and convergence to a local solution. In direct methods, the quality of solution depends on discretization resolution. Since these methods, using control parametrization, convert the continuous problem to discrete problem, they have less accuracy. However, the adaptive strategies [4, 5] can overcome these defects, but they may be trapped by a local optimal, yet. In indirect approaches, the problem, through the use of the Pontryagins minimum principle (PMP), is converted into a two-boundary value problem (TBVP) that can be solved by numerical methods such as shooting method [6]. These methods need the good initial guesses that lie within the domain of convergence. Therefore, the numerical methods, usually are not suitable for solving NOCPs, especially for large-scale and multimodal models.

Metaheuristics as the global optimization methods can overcome these problems, but they usually need more computational time, though they do not really need good initial guesses and deterministic rules. Several researchers used metaheuristics to solve optimal control problems. For instance, Michalewicz et al. [7] applied floating-point Genetic algorithms (GA) to solve discrete time optimal control problems; Yamashita and Shima [8] used the classical GAs to solve the free final time optimal control problems with terminal constraints. Abo-Hammour et al. [9] used continuous GA for solving NOCPs. Moreover, the other usages of GA for optimal control problems can be found in [6, 10]. Lopez Cruz et al. [11] applied differential evolution (DE) algorithms for solving the multimodal optimal control problems. Recently, Ghosh et al. [12] developed an ecologically inspired optimization technique, called invasive weed optimization (IWO), for solving optimal control problems. The other well-known metaheuristic algorithms used for solving NOCPs are genetic programming (GP) [13], particle swarm optimization (PSO) [14, 15], ant colony optimization (ACO) [16], and DE [17, 18].

To increase the quality of solutions and decrease the running time, hybrid methods were introduced, which used a local search in the implementation of a population-based metaheuristics, [19]. Modares and Naghibi-Sistani [20] proposed a hybrid algorithm by integrating an improved PSO with successive quadratic programming (SQP) for solving NOCPs. Recently, Sun et al. [21] proposed a hybrid improved GA, which used simplex method (SM) to perform a local search, for solving NOCPs and applied it for chemical processes.

Based on the success of the hybrid methods for solving NOCPs, mentioned above, we here use a modified hybrid genetic algorithm (MHGA), which combines GA with SQP, see [22], as a local search. SQP is an iterative algorithm for solving nonlinear programming (NLP) problems, which uses gradient information. It can moreover be used for solving NOCPs, see [23–25]. For decreasing the running time in the early generations (iterations) of MHGA, a less number of iterations for SQP was used and then, when the promising region of search space was found, we increase the number of iterations of SQP, gradually.

To perform MHGA for solving an NOCP, the time interval is uniformly divided by using a constant number of time nodes. Next, in each of these time nodes, the control variable is approximated by a scaler vector of control input values. Thus, an infinite dimensional NOCP is changed to a finite dimensional NLP. Now, we encounter two conflict situations: the quality of the global solution and the needed computational time. In other words, when the number of time nodes is increased, then we expect that the quality of the global solution is also increased, but we know that in this situation the computational time is increased dramatically. In other situation, if we consider less number of time nodes, then the computational time is decreased but we may find a poor local solution. To conquer these problems, MHGA performs in two phases. In the first phase (exploration phase), to decrease the computational time and to find a promising region of search space, MHGA uses a less number of time nodes. After phase 1, to increase the quality of solutions obtained from phase 1, the number of time nodes is increased. Using the population obtained in phase 1, the values of the new control inputs are estimated by Linear or Spline interpolations. Next, in the second phase (exploitation phase), MHGA uses the solutions constructed by the above procedure, as an initial population.

The paper is organized as follows: in Section 2, the formulation of the problem is introduced. In Section 3, the proposed MHGA is presented. In Section 4, we introduce our algorithm for solving NOCP. In Section 5, we provide 20 numerical benchmark examples, to compare the proposed algorithm with the other recently proposed algorithms. In Section 6, we consider two statistical approaches for the comparison of the LI and SI methods and investigation of sensitivity analysis of the algorithm parameters. The impact of SQP, as local search, in the proposed algorithm is surveyed in Section 7. We conclude in Section 8.

2. Formulation of Problem

The bounded continuous-time NOCP is considered as finding the control input $u(t) \in \mathbb{R}^m$, over the planning horizon $[t_0, t_f]$, which minimizes the cost functional:

$$J = \phi\left(x\left(t_f\right), t_f\right) + \int_{t_0}^{t_f} g\left(x\left(t\right), u\left(t\right), t\right) dt \tag{1}$$

subject to

$$\dot{x}\left(t\right) = f\left(x\left(t\right), u\left(t\right), t\right), \tag{2}$$

$$c\left(x, u, t\right) = 0, \tag{3}$$

$$d\left(x, u, t\right) \leq 0, \tag{4}$$

$$\psi\left(x\left(t_f\right), t_f\right) = 0, \tag{5}$$

$$x\left(t_0\right) = x_0, \tag{6}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector for the system and $x_0 \in \mathbb{R}^n$ is the initial state. The functions $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$, $g : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}$, $c : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^{n_c}$, $d : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^{n_d}$, $\psi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^{n_\psi}$, and $\phi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$ are assumed to be sufficiently smooth on appropriate open sets. Cost function (1) must be minimized subject to dynamic (2), control and state equality constraints (3) and control and state inequality constraints (4), the final state constraints (5), and the initial condition (6). A special case of the NOCPs is the linear quadratic regulator (LQR) problem where the dynamic equations are linear and the objective function is a quadratic function of x and u. The minimum time problems, tracking problem, terminal control problem and minimum energy are another special case of NOCPs.

3. Modified Hybrid Genetic Algorithm

In this section, first MHGA, as a subprocedure for the main algorithm, is introduced. To perform MHGA, the control variables are discretized. Next, NOCP is changed into a finite dimensional NLP; see [21, 26]. Now, we can imply a GA to find the global solution of the corresponding NLP. In the following, we introduce GA operators.

3.1. Underlying GA. GAs, introduced by Holland in 1975, are heuristics and probabilistic methods [27]. These algorithms start with an initial population of solutions. This population is evaluated by using genetic operators that include selection, crossover, and mutation. Here, in MHGA, the underlying GA has the following steps.

Initialization. The time interval is divided into $N_t - 1$ subintervals using time nodes $t_0, \ldots, t_{N_t-1}$ and then the control input

```
{Initialization} Input the the size of tournament, $N_{\text{tour}}$ and select $N_{\text{tour}}$
individuals from population, randomly. Let $k_i$, $i = 1, 2, \ldots, N_{\text{tour}}$ be the indexes of them.
repeat
  if $N_{\text{tour}} = 2$ then
    if $I(U^{k_1}) < I(U^{k_2})$ then
      Let $k = k_1$.
    else
      Let $k = k_2$.
    end if
  end if
Perform Tournament algorithm with size $N_{\text{tour}}/2$ with output $k_1$.
Perform Tournament algorithm with size $N_{\text{tour}}/2$ with output $k_2$.
until $k_1 \neq k_2$
Return the $k$th individual of population.
```

Algorithm 1: Tournament algorithm.

values are computed (or selected randomly). This can be done by the following stages.

(1) Let $t_j = t_0 + jh$, where $h = (t_f - t_0)/(N_t - 1)$, $j = 0, 1, \ldots, N_t - 1$, be time nodes, where t_0 and t_f are the initial and final times, respectively.

(2) The corresponding control input value at each time node t_j is an $m \times 1$ vector, u_j, which can be calculated randomly, with the following components.

$$u_{ij} = u_j(i) = u_{\text{left}}(i) + \left(u_{\text{right}}(i) - u_{\text{left}}(i)\right) \cdot r_{ij}, \qquad i = 1, 2, \ldots, m, \quad j = 0, 1, \ldots, N_t - 1, \tag{7}$$

where r_{ij} is a random number in $[0, 1]$ with a uniform distribution and $u_{\text{left}}, u_{\text{right}} \in \mathbb{R}^m$ are the lower and the upper bound vectors of control input values, which can be given by the problem's definition or the user (e.g., see the NOCPs numbers (6) and (5) in the Appendix, resp.). So, each individual of the population is an $m \times N_t$ matrix as $U = (u_{ij})_{m \times N_t} = [u_j]_{j=0}^{N_t-1}$. Next, we let $U^{(k)} = (u_{ij})_{m \times N_t}$, $k = 1, 2, \ldots, N_p$ as kth individual of the population, which N_p is the size of the population.

Evaluation. For each control input matrix, $U^{(k)}$, $k = 1, 2, \ldots, N_p$, the corresponding state variable is an $n \times N_t$ matrix, $X^{(k)}$, and it can be computed by the forth Runge-Kutta method on dynamic system (2) with the initial condition (6), approximately. Then, the performance index, $J(U^{(k)})$, is approximated by a numerical method (denoted by $\widetilde{J}$). If NOCP includes equality or inequality constraints (3) or (4), then we add some penalty terms to the corresponding fitness value of the solution. Finally, we assign $I(U^{(k)})$ to $U^{(k)}$ as the fitness value as follows:

$$\begin{aligned} I\left(U^{(k)}\right) = \widetilde{J} &+ \sum_{l=1}^{n_d} \sum_{j=0}^{N_t-1} M_{1l} \max\left\{0, d_l\left(x_j, u_j, t_j\right)\right\} \\ &+ \sum_{h=1}^{n_c} \sum_{j=0}^{N_t-1} M_{2h} c_h^2\left(x_j, u_j, t_j\right) \\ &+ \sum_{i=p}^{n_\psi} M_{3p} \psi_p^2\left(x_{N_t-1}, t_{N_t-1}\right), \end{aligned} \tag{8}$$

where $M_1 = [M_{11}, \ldots, M_{1n_d}]^T$, $M_2 = [M_{21}, \ldots, M_{2n_c}]^T$ and $M_3 = [M_{31}, \ldots, M_{3n_\psi}]^T$ are big numbers, as the penalty parameters, $c_h(\cdot,\cdot)$, $h = 1, 2, \ldots, n_c$, $d_l(\cdot,\cdot)$, $l = 1, 2, \ldots, n_d$, and $\psi_p(\cdot,\cdot)$, $p = 1, 2, \ldots, n_\psi$ are defined in (3), (4), and (5), respectively.

Selection. To select two parents, we use a tournament selection [27]. It can be applied for parallel GA. The tournament operator applies competition among the same individuals, and the best of them is selected for next generation. At first we select a specified number of individuals from population, randomly. This number is tournament selection parameter, which is denoted by N_{tour}. The tournament algorithm is given in Algorithm 1.

Crossover. When two parents $U^{(1)}$ and $U^{(2)}$ are selected, we use the following stages to construct an offspring.

(1) Select the following numbers

$$\lambda_1 \in [0, 1], \qquad \lambda_2 \in [-\lambda_{\max}, 0], \qquad \lambda_3 \in [1, 1 + \lambda_{\max}] \tag{9}$$

randomly, where $\lambda_{\max}$ is a random number in $[0, 1]$.

(2) Let

$$\text{of}^k = \lambda_k U^{(1)} + (1 - \lambda_k) U^{(2)}, \quad k = 1, 2, 3, \tag{10}$$

where λ_k, $k = 1, 2, 3$ is defined in (9). For $i = 1, \dots, m$ and $j = 1, \dots, N_t$, if $(\text{of}^k)_{ij} > u_{\text{right}}(i)$, then let $(\text{of}^k)_{ij} = u_{\text{right}}(i)$ and if $(\text{of}^k)_{ij} < u_{\text{left}}(i)$, then let $(\text{of}^k)_{ij} = u_{\text{left}}(i)$.

(3) Let $\text{of} = \text{of}^*$, where of^* is the best of^k, $k = 1, 2, 3$ constructed by (10).

Mutation. We apply a perturbation on each component of the offspring as follows:

$$(\text{of})_{ij} = (\text{of})_{ij} + r_{ij} \cdot \alpha, \quad i = 1, 2, \dots, m, \; j = 1, 2, \dots, N_t, \tag{11}$$

where r_{ij} is selected randomly in $\{-1, 1\}$ and α is a random number, with a uniform distribution in $[0, 1]$. If $(\text{of})_{ij} > u_{\text{right}}(i)$, then let $(\text{of})_{ij} = u_{\text{right}}(i)$ and if $(\text{of})_{ij} < u_{\text{left}}(i)$, then let $(\text{of})_{ij} = u_{\text{left}}(i)$.

Replacement. Here, in the underling GA, we use a traditional replacement strategy. The replacement is done, if the new offspring has two properties: first, it is better than the worst person in the population, $I(\text{of}) < \max_{1 \le i \le N_p} I(U^{(i)})$, second, it is not very similar to a person in the population; that is, for each $i = 1, \dots, N_p$, at least one of the following conditions is satisfied:

$$\begin{gathered} \left| I(\text{of}) - I\left(U^{(i)}\right) \right| > \acute{\varepsilon}, \\ \left\| \text{of} - U^{(i)} \right\| > \acute{\varepsilon}, \end{gathered} \tag{12}$$

where $\acute{\varepsilon}$ is the machine epsilon.

Termination Conditions. Underlying GA is terminated when at least one of the following conditions is occurred.

(1) The maximum number of generations, N_g, is reached.

(2) Over a specified number of generations, N_i, we do not have any improvement (the best individual is not changed) or the two-norm, or error, of final state constraints will reach a small number as the desired precision, ε; that is,

$$\varphi_f = \|\psi\|_2 < \varepsilon, \tag{13}$$

where $\psi = [\psi_1, \psi_2, \dots, \psi_{n_\psi}]^T$ is the vector of final state constraints, defined in (5).

3.2. SQP.

The fitness value in (8) can be viewed as a nonlinear objective function with the decision variable as $u = [u_0, u_1, \dots, u_{N_t - 1}]$. This cost function with upper and lower bounds of input signals construct a finite dimensional NLP problem as following

$$\min \quad I(u) = I\left(u_0, u_1, \dots, u_{N_t - 1}\right) \tag{14}$$

$$\text{s.t.} \quad u_{\text{left}} \le u_j \le u_{\text{right}}, \quad j = 0, 1, \dots, N_t - 1 \tag{15}$$

SQP algorithm [22, 28] is performed on the NLP (14)-(15), using $\overline{u}^0 = \text{of}$ and constructed in (11), as the initial solution when the maximum number of iteration is *sqpmaxiter*.

SQP, is an effective and iterative algorithm for the numerical solution of the constrained NLP problem. This technique is based on finding a solution to the system of nonlinear equations that arise from the first-order necessary conditions for an extremum of the NLP problem. Using an initial solution of NLP, $\overline{u}^k$, $k = 0, 1, \dots$, a sequence of solutions as $\overline{u}^{k+1} = \overline{u}^k + d^k$ is constructed, which d^k is the optimal solution of the constructed quadratic programming (QP) that approximates NLP in the iteration k based on $\overline{u}^k$, as the search direction in the line search procedure. For the NLP (15), the principal idea is the formulation of a QP subproblem based on a quadratic approximation of the Lagrangian function as $L(u, \lambda) = I(u) + \lambda^T h(u)$, where the vector λ is Lagrangian multiplier and $h(u)$ return the vector of, inequality constraints evaluated at u. The QP is obtained by linearizing the nonlinear functions as follows:

$$\min \frac{1}{2} d^T H\left(\overline{u}^k\right) d + \nabla I\left(\overline{u}^k\right)^T d \tag{16}$$

$$\nabla h\left(\overline{u}^k\right)^T d + h\left(\overline{u}^k\right) \le 0. \tag{17}$$

Similar to [26], here a finite difference approximation is applied to compute the gradient of the cost function and the constraints, with the following components:

$$\frac{\partial I}{\partial u_j} = \frac{I\left(\cdots u_j + \delta \cdots\right) - I(u)}{\delta}, \quad j = 0, 1, \dots, N_t - 1, \tag{18}$$

where δ is the double precision of machine. So, the gradient vector is $\nabla I = [\partial I / \partial u_0, \dots, \partial I / \partial u_{N_t - 1}]^T$. Also, at each major iteration a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, H, is calculated using the BFGS method [28], where λ_i, $i = 1, \dots, m$, is an estimate of the Lagrange multipliers. The general procedure for SQP, for NLP (14)-(15), is as follows.

(1) Given an initial solution $\overline{u}^0$. Let $k = 0$.

(2) Construct the QP subproblem (16)-(17), based on $\overline{u}^0$, using the approximations of the gradient and the Hessian of the the Lagrangian function.

(3) Compute the new point as $\overline{u}^{k+1} = \overline{u}^k + d^k$, where d^k is the optimal solution of the current QP.

(4) Let $k = k + 1$ and go to step (2).

Here, in MHGA, SQP is used as the local search, and we use the maximum number of iterations as the main criterion for stopping SQP. In other words, we terminate SQP when it converges either to local solution or the maximum number of SQP's iterations is reached.

3.3. MHGA.

In MHGA, GA uses a local search method to improve solutions. Here, we use SQP as a local search. Using SQP as a local search in the hybrid metaheuristic is common, for example, see [20].

MHGA can be seen as a multi start local search where initial solutions are constructed by GA. From another perspective, MHGA can be seen as a GA that the quality of

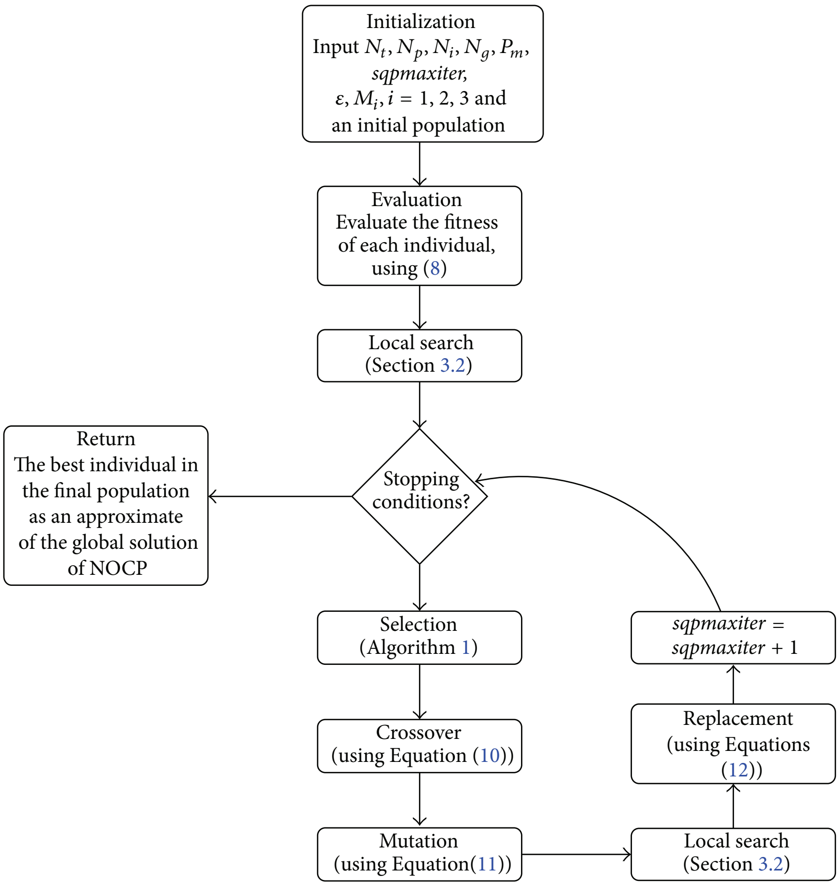

Initialization
Input N_t, N_p, N_i, N_g, P_m, *sqpmaxiter,* $\varepsilon, M_i, i = 1, 2, 3$ and an initial population
Evaluation
Evaluate the fitness of each individual, using (8)
Local search
(Section 3.2)
Stopping conditions?
Return
The best individual in the final population as an approximate of the global solution of NOCP
Selection
(Algorithm 1)
Crossover
(using Equation (10))
Mutation
(using Equation(11))
Local search
(Section 3.2)
Replacement
(using Equations (12))
sqpmaxiter = *sqpmaxiter* + 1

Figure 1: Flowchart of the MHGA algorithm.

its population is intensified by SQP. In the beginning of MHGA, a less number of iterations for SQP was used. Then, when the promising regions of search space were found by GA operators, we increase the number of iterations of SQP gradually. Using this approach, we may decrease the needed running time (in [19] the philosophy of this approach is discussed).

Finally, we give our modified MHGA, to find the global solution, by the flowchart in Figure 1.

4. Proposed Algorithm

Here, we give a new algorithm, which is a direct approach, based on MHGA, for solving NOCPs. The proposed algorithm has two main phases. In the first phase, we perform MHGA with a completely random initial population constructed by (7). In the first phase, to find the promising regions of the search space, in a less running time, we use a few numbers of time nodes. In addition, to have a faster MHGA, the size of the population in the first phase is usually less than the size of the population in the second phase.

After phase 1, to maintain the property of individuals in the last population of phase 1 and to increase the accurately of solutions, we add some additional time nodes. When the number of time nodes is increased, it is estimated that the quality of solution obtained by numerical methods (e.g., Runge-Kutta and Simpson) is increased. Thus, we increase time nodes from N_{t_1} in phase 1 to N_{t_2} in phase 2. The corresponding control input values of the new time nodes are added to individuals. To use the information of the obtained solutions from phase 1 in the construction of the initial population of phase 2, we use either Linear or Spline interpolation to estimate the value of the control inputs in the new time nodes in each individual of the last population of phase 1. Moreover, to maintain the diversity in the initial population of phase 2, we add new random individuals to the population using (7). In the second phase, MHGA starts with this population and new value of parameters. Finally, the proposed algorithm is given in Algorithm 2.

5. Numerical Experiments

In this section, to investigate the efficiency of the proposed algorithm, 20 well-known and real world NOCPs, as benchmark problems, are considered which are presented in terms of (1)–(6) in the Appendix. These NOCPs are selected with single control signal and multi control signals, which will be demonstrated in a general manner.

The numerical behaviour of algorithms can be studied from two view of points, the relative error of the performance index and the status of the final state constraints. Let J be the obtained performance index by an algorithm, φ_f, defined in

```
{Initialization} Input the desired precision, $\varepsilon$, in (13), the penalty parameters,
$M_i$, $i = 1, 2, 3$, in (8), the bounds of control input values in (7), $u_{\text{left}}$ and $u_{\text{right}}$.
{Phase 1} Perform MHGA with a random population and $N_{t_1}, N_{p_1}, N_{i_1}, N_{g_1}, P_{m_1}$ and $sqpmaxiter_1$.
{Construction of the initial population of the phase 2} Increase time nodes uniformly to $N_{t_2}$
and estimate the corresponding control input values of the new time nodes in each
individual obtained from phase 1, using either Linear or Spline interpolation.
Create $N_{p_2} - N_{p_1}$ new different individuals with $N_{t_2}$ time nodes, randomly.
{Phase 2} Perform MHGA with the constructed population and
$N_{t_2}, N_{p_2}, N_{i_2}, N_{g_2}, P_{m_2}$ and $sqpmaxiter_2$.
```

Algorithm 2: The proposed algorithm.

(13), be the error of final state constraints, and J^* be the best obtained solution among all implementations, or the exact solution (when exists). Now the relative error of J, E_J, of the algorithm can be defined as

$$E_J = \left| \frac{J - J^*}{J^*} \right|. \tag{19}$$

To more accurate study, we now define a new criterion, called factor, to compare the algorithms as follows:

$$K_\psi = E_J + \varphi_f. \tag{20}$$

Note that K_ψ shows the summation of two important errors. Thus, based on K_ψ we can study the behaviour of the algorithms on the quality and feasibility of given solutions, simultaneously.

To solve any NOCP described in the Appendix, we must know the algorithm's parameters including: MHGA's parameters; including N_t, N_p, N_i, N_g, P_m and $sqpmaxiter$, in both phases in Algorithm 2, and the problem's parameters including ε, in (13), M_i, $i = 1, 2, 3$, in (8), u_{left} and u_{right}, in (7). To estimate the best value of the algorithm's parameters, we ran the proposed algorithm with different values of parameters and then select the best. However, the sensitivity of MHGA parameters are studied in the next section. In both of phases of Algorithm 2, in MHGA, we let $sqpmaxiter_i = 4$ and $P_{m_i} = 0.8$, for $i = 1, 2$. Also, we consider $N_{g_1} = N_{g_2}$, $N_{i_1} = N_{i_2}$, $N_{p_1} = 9$ and $N_{p_2} = 12$. The other MHGA parameters are given in the associated subsection and the problem's parameters in Table 2. For each NOCP, 12 different runs were done and the best results are reported in Table 1, which the best value of each column is seen in the bold.

The reported numerical results of the proposed algorithm, for each NOCP, include the value of performance index, J, the relative error of J, E_J, defined in (19), the required computational time, Time, the norm of final state constraints, φ_f, defined in (13) and the factor, K_ψ, defined in (20).

The algorithm was implemented in Matlab R2011a environment on a Notebook with Windows 7 Ultimate, CPU 2.53 GHz and 4.00 GB RAM. Also, to implement SQP in our proposed algorithm, we used "fmincon" in Matlab when the "Algorithm" was set to "SQP". Moreover, we use composite Simpson's method [29] to approximate integrations.

Remark 1. We use the following abbreviations to show the used interpolation method in our proposed algorithm:

(1) LI: linear interpolation.

(2) SI: spline interpolation.

For comparing the numerical results of the proposed algorithm two subsections are considered, comparison with some metaheuristic algorithms, in Section 5.1, and comparison with some numerical methods, in Section 5.2. We give more details of these comparisons in the following subsections.

5.1. Comparison with Metaheuristic Algorithms.

The numerical results for the NOCPs numbers (1)–(3), in the Appendix, are compared with a continuous GA, CGA, as a metaheuristic, proposed in [9], which gave better solutions than shooting method and gradient algorithm, from the indirect methods category [2, 30], and SUMT from the direct methods category [26]. For NOCPs numbers (4) and (5), the results are compared with another metaheuristic, which is a hybrid improved PSO, called IPSO, proposed in [20].

5.1.1. VDP Problem [9].

The first NOCP in the Appendix is Van Der Pol Problem, VDP, which has two state variables and one control variable. VDP problem has a final state constraint, which is $\psi = x_1(t_f) - x_2(t_f) + 1 = 0$. The results of the proposed algorithm with the MHGA's parameters as $N_{t_1} = 31$, $N_{t_2} = 71$, $N_g = 300$ and $N_i = 200$, are reported in Table 1. From Table 1, it is obvious that the numerical results of LI and SI methods are more accurate than CGA, with less amount of K_ψ.

5.1.2. CRP Problem [9].

The second NOCP in the Appendix is Chemical Reactor Problem, CRP, which has two state variables and one control variable. The results of the proposed algorithm, with the MHGA's parameters as $N_{t_1} = 31$, $N_{t_2} = 71$, $N_g = 300$ and $N_i = 200$, are shown in the second row of Table 1. CRP problem has two final state constraints, $\psi = [x_1, x_2]^T$. Although, from Table 1, the norm of final state constraints, φ_f, for the CGA, equals $\varphi_f^* = 7.57 \times 10^{-10}$, is less than φ_f's of LI and SI methods, which equals 1.15×10^{-9} and 5.99×10^{-9}, respectively, but the performance index,

Table 1: The best numerical results in 12 different runs of the SI and LI methods for NOCPs in the Appendix.

Problem	Algorithm	J	E_J	φ_f	K_ψ	Time
VDP	CGA	1.7404	0.0912	2.67×10^{-11}	0.0912	501.28
	LI	**1.5949**	**0**	1.08×10^{-13}	**1.08×10^{-13}**	282.82
	SI	1.5950	6.27×10^{-5}	**9.99×10^{-14}**	6.27×10^{-5}	239.14
CRP	CGA	1.63×10^{-2}	0.2835	**7.57×10^{-10}**	0.2835	501.28
	LI	**1.27×10^{-2}**	**0**	1.15×10^{-9}	**1.15×10^{-9}**	171.94
	SI	**1.27×10^{-2}**	**0**	5.99×10^{-9}	5.99×10^{-9}	124.05
FFRP	CGA	83.63	1.3256	4.65×10^{-3}	1.3302	1413
	LI	**35.96**	**0**	1.22×10^{-5}	**1.22×10^{-5}**	1268.58
	SI	35.99	8.34×10^{-4}	**9.01×10^{-6}**	8.43×10^{-4}	1343.40
MSNIC	IPSO	0.1727	0.0165	—	—	126.26
	LI	**0.1699**	**0**	—	—	138.82
	SI	0.1700	5.89×10^{-4}	—	—	138.68
CSTCR	IPSO	0.1355	0.2098	—	—	31.34
	LI	**0.1120**	**0**	—	—	103.58
	SI	**0.1120**	**0**	—	—	106.73
No. 6	Bézier	−5.3898	0.0251	—	—	NR[a]
	LI	**−5.4309**	**0.0177**	—	—	138.38
	SI	**−5.4309**	**0.0177**	—	—	107.34
Number 7	HPM	0.2353	0.1677	4.20×10^{-6}	0.1677	NR
	LI	**0.2015**	**0**	2.35×10^{-9}	2.35×10^{-9}	213.92
	SI	**0.2015**	**0**	**2.82×10^{-10}**	**2.82×10^{-10}**	206.96
Number 8	SQP	6.36×10^{-6}	2.20×10^{9}	—	—	NR
	SUMT	5.15×10^{-6}	1.78×10^{9}	—	—	NR
	LI	**2.89×10^{-15}**	**0**	—	—	72.82
	SI	3.64×10^{-15}	0.2595	—	—	68.68
Number 9	SQP	1.7950	0.0873	—	—	NR
	SUMT	1.7980	0.0891	—	—	NR
	LI	**1.6509**	**0**	—	—	639.10
	SI	**1.6509**	**0**	—	—	595.0
Number 10	SQP	0.2163	0.3964	—	—	NR
	SUMT	0.1703	0.0994	—	—	NR
	LI	**0.1549**	**0**	—	—	625.70
	SI	**0.1549**	**0**	—	—	676.62
Number 11	SQP	3.25	0.1648	0	0.1648	NR
	SUMT	3.25	0.1648	0	0.1648	NR
	LI	**2.7901**	**0**	1.62×10^{-9}	1.62×10^{-9}	147.78
	SI	**2.7901**	**0**	**5.86×10^{-10}**	**5.86×10^{-10}**	219.46
Number 12	SQP	−0.2490	4.0×10^{-3}	0	4.0×10^{-3}	NR
	SUMT	−0.2490	4.0×10^{-3}	0	4.0×10^{-3}	NR
	LI	**−0.2500**	**0**	2.85×10^{-8}	2.85×10^{-8}	547.71
	SI	**−0.2500**	**0**	**3.90×10^{-10}**	**3.90×10^{-10}**	545.48
Number 13	SQP	1.68×10^{-2}	0.1748	0	0.1748	NR
	SUMT	1.67×10^{-2}	0.1678	0	0.1678	NR
	LI	**1.43×10^{-2}**	**0**	**1.18×10^{-9}**	**1.18×10^{-9}**	425.99
	SI	1.44×10^{-2}	7.0×10^{-3}	6.32×10^{-9}	7.0×10^{-3}	490.27
Number 14	SQP	3.7220	0.0961	0	0.0961	NR
	SUMT	3.7700	0.1103	0	0.1103	NR
	LI	**3.3956**	**0**	**7.86×10^{-7}**	**7.86×10^{-7}**	1041.24
	SI	3.3965	2.65×10^{-4}	2.76×10^{-6}	2.67×10^{-4}	1151.44

Table 1: Continued.

Problem	Algorithm	J	E_J	φ_f	K_ψ	Time
Number 15	SQP	1.03×10^{-3}	4.3368	0	4.3368	NR
	SUMT	9.29×10^{-4}	3.8135	0	3.8135	NR
	LI	$\mathbf{1.93 \times 10^{-4}}$	**0**	$\mathbf{8.10 \times 10^{-10}}$	$\mathbf{8.10 \times 10^{-10}}$	333.54
	SI	1.94×10^{-4}	5.20×10^{-3}	1.64×10^{-9}	5.20×10^{-3}	340.14
Number 16	SQP	2.2120	0.0753	0	0.0753	NR
	SUMT	2.2080	0.0734	0	0.0734	NR
	LI	**2.0571**	**0**	3.94×10^{-12}	3.94×10^{-12}	469.82
	SI	**2.0571**	**0**	$\mathbf{5.60 \times 10^{-13}}$	$\mathbf{5.60 \times 10^{-13}}$	704.23
Number 17	SQP	−8.8690	2.25×10^{-5}	0	2.25×10^{-5}	NR
	SUMT	−8.8690	2.25×10^{-5}	0	2.25×10^{-5}	NR
	LI	**−8.8692**	**0**	$\mathbf{1.45 \times 10^{-10}}$	$\mathbf{1.45 \times 10^{-10}}$	321.83
	SI	**−8.8692**	**0**	2.79×10^{-10}	2.79×10^{-10}	309.59
Number 18	SQP	0.0368	0.1288	—	—	NR
	SUMT	0.0386	0.1840	—	—	NR
	LI	**0.0326**	**0**	—	—	551.63
	SI	**0.0326**	**0**	—	—	606.71
Number 19	SQP	0.3439	4.9293	0	4.9293	NR
	SUMT	0.3428	4.9103	0	4.9103	NR
	LI	0.0689	0.1879	4.10×10^{-3}	0.1919	1645.09
	SI	**0.0580**	**0**	$\mathbf{4.05 \times 10^{-4}}$	$\mathbf{4.05 \times 10^{-4}}$	1787.83
Number 20	SQP	77.52	1.2554	0	1.2554	NR
	SUMT	76.83	1.2353	0	1.2353	NR
	LI	34.3716	2.32×10^{-5}	2.25×10^{-4}	2.48×10^{-4}	815.19
	SI	**34.3708**	**0**	$\mathbf{1.57 \times 10^{-4}}$	$\mathbf{1.57 \times 10^{-4}}$	804.20

[a] Not reported.

the relative error of J and the factor of the proposed algorithm is better, and so the proposed algorithm is more robust than CGA.

5.1.3. FFRP Problem [9]. The third NOCP in the Appendix is Free Floating Robot Problem, FFRP, which has six state variables and four control variables. FFRP has been solved by CGA and the proposed algorithm, with the MHGA's parameters as $N_{t_1} = 31$, $N_{t_2} = 71$, $N_g = 300$ and $N_i = 200$. This problem has six final state constraints, $\psi = [x_1 - 4, x_2, x_3 - 4, x_4, x_5, x_6]^T$. The numerical results are shown in Table 1. The values of J, E_J, φ_f and K_ψ for LI and SI methods, separately, are less than CGA. Therefore, the proposed algorithm can achieve much better quality solutions than the CGA, with reasonable computational time.

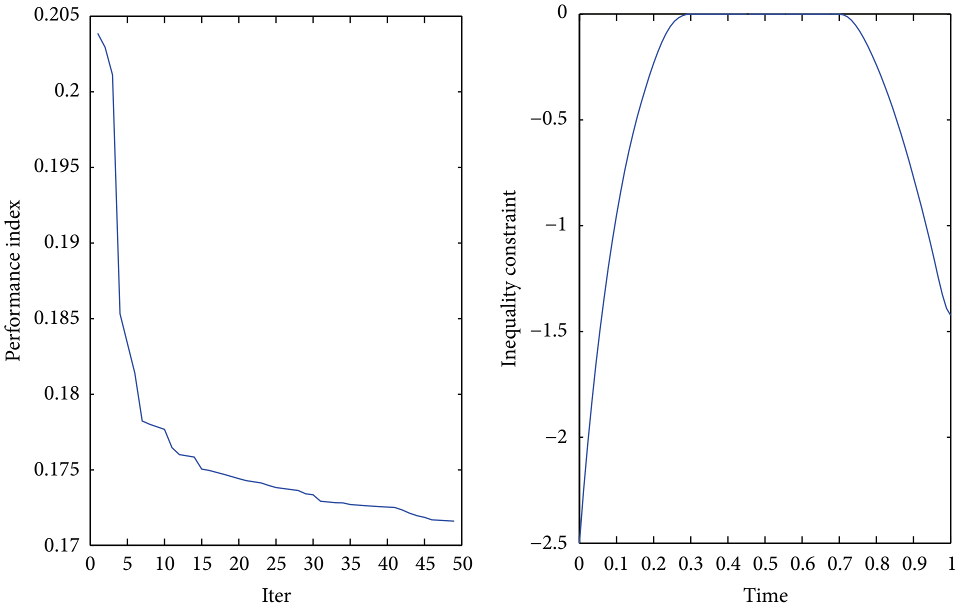
5.1.4. MSNIC Problem [20]. For the forth NOCP in the Appendix, which is a Mathematical System with Nonlinear Inequality Constraint, NSNIC, the numerical results are compared with IPSO. MSNIC contains an inequality constraint, $d(x,t) = x_2(t) + 0.5 - 8(t - 0.5)^2 \leq 0$. The problem solved by several numerical methods as [24, 31]. From [20], IPSO method could achieved more accurate results than mentioned numerical methods. Also, MSNIC can be solved by the proposed algorithm, with the MHGA's parameters as $N_{t_1} = 31$, $N_{t_2} = 91$, $N_g = 100$ and $N_i = 60$. From the forth row of Table 1, the absolute error of J, E_J, for LI and SI methods equal 0 and 5.89×10^{-4}, respectively, which are less than IPSO's, 0.0165.

Subplots (a) and (b) in Figure 2, show the graphs of the convergence rate for the performance index and the inequality constraint, respect to the number of iteration, respectively.

5.1.5. CSTCR Problem [20]. The fifth NOCP in the Appendix is a model of a nonlinear Continuous Stirred-tank Chemical Reactor, CSTCR. It has two state variables $x_1(t)$ and $x_2(t)$, as the deviation from the steady-state temperature and concentration, and one control variable $u(t)$, which represent the effect of the flow rate of cooling fluid on chemical reactor. The objective is to maintain the temperature and concentration close to steady-state values without expending large amount of control effort. Also, this is a benchmark problem in the handbook of test problems in local and global optimization [32], which is a multimodal optimal control problem [33]. It involves two different local minima. The values of the performance indexes, for these solutions, equal 0.244 and 0.133. Similarly to the MSNIC, the numerical results of the proposed algorithm, with the MHGA's parameters as $N_{t_1} = 31$, $N_{t_2} = 51$, $N_g = 100$, and $N_i = 50$, are compared with IPSO. From Table 1, the performance index, J, for LI and SI methods is equal to $J^* = 0.1120$ which is less than IPSOs 0.1355.

Table 2: The problem parameters for NOCPs in the Appendix.

Parameters	Problem number									
	1	2	3	4	5	6	7	8	9	10
u_{left}	−0.5	−1.5	−15	−20	0	−2	0	−2	−1	−20
u_{right}	2	2	10	20	5	2	1	2	1	20
M_i	10^3	10^2	70	1	—	1	—	10^2	—	10^2
ε	10^{-12}	10^{-10}	10^{-3}	—	—	—	—	—	—	10^{-9}
Parameters	Problem no.									
	11	12	13	14	15	16	17	18	19	20
u_{left}	−5	−1	−2	$-\pi$	−1	−3	−30	−1	−15	−15
u_{right}	5	1	2	π	1	3	30	1	10	10
M_i	10	10^2	—	10^2	10^2	10^2	—	10	70	10^{-3}
ε	10^{-9}	10^{-11}	10^{-11}	—	10^{-10}	10^{-10}	10^{-10}	—	10^{-3}	10^{-4}



(a)

(b)

Figure 2: Graphical results of MSNIC problem using SI method: (a) convergence rate of the performance index and (b) the inequality constraint, $d(x,t) \leq 0$, in respect to the number of iterations.

5.2. Comparison with Numerical Methods. For NOCPs numbers (6)–(20), the comparison are done with some numerical methods. Unfortunately, for these methods, usually, the final state constraints and the required computational time are not reported, which are shown with NR in Table 1, but these values are reported for both LI and SI methods in Table 1. For all NOCPs, in this section, the MHGA's parameters are considered as $N_{t_1} = 31$, $N_{t_2} = 51$, $N_g = 100$, $N_i = 50$, with the problem parameters in Table 2.

5.2.1. Compared with Bézier [34]. The NOCP number (6), in the Appendix, has exact solution, which has an inequality constraint as $d(x_1,t) = -6 - x_1(t) \leq 0$. The exact value of performance index equals $J^* = -5.5285$ [35]. This problem has been solved by a numerical method proposed in [34], called Bézier. From sixth row of Table 1, the absolute error of the LI and SI methods equal 0.0177, which is less than Bézier's, 0.0251.

Figure 3, shows the graphs of the convergence rate of the performance index, subplot (a), and inequality constraint, subplot (b), respect to the number of iteration, using SI method.

5.2.2. Compared with HPM [36]. For NOCP number (7) in the Appendix, which is a constraint nonlinear model, the numerical results of the proposed algorithm are compared with HPM, proposed in [36]. This NOCP has a final state constraint as $\psi = x - 0.5 = 0$. From [36], the norm of final state constraint for HPM equals 4.2×10^{-6}, however, this criterion for the LI and SI methods equals 2.35×10^{-9} and $\varphi_f^* = 2.82 \times 10^{-10}$, respectively. From Table 1, it is obvious that the obtained values of the performance index, the norm

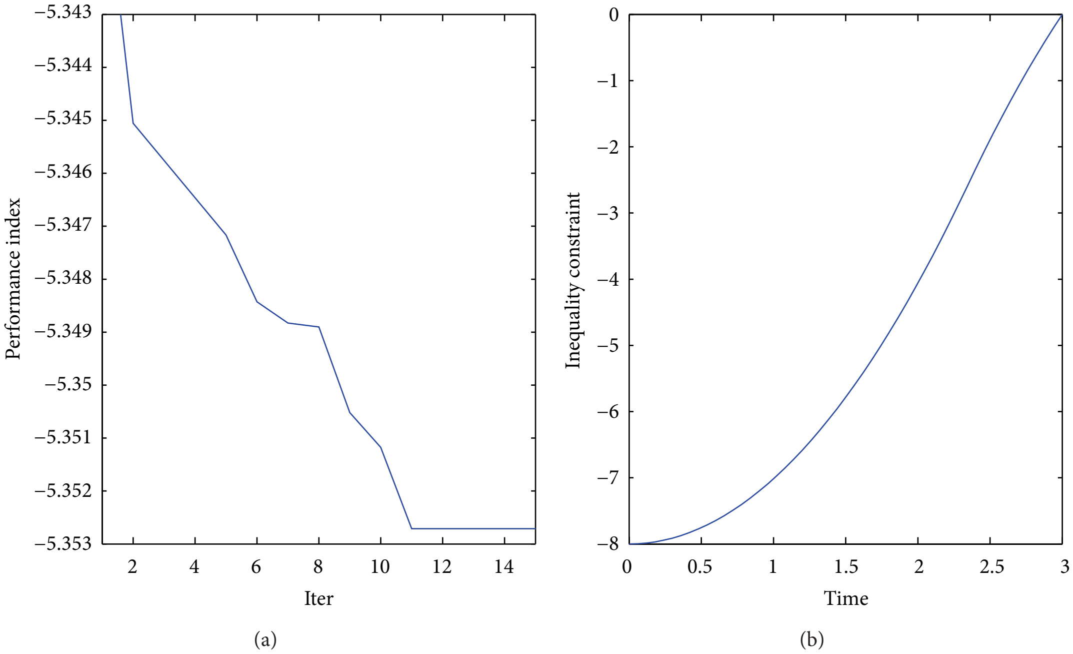


Figure 3: Graphical results for NOCP number (6) using SI method: (a) convergence rate for the performance index and (b) the inequality constraint, $d(x_1, t) \leq 0$, in respect to the number of iterations.

of final state constraint and K_ψ for the SI method are more accurate than LI than HPM methods.

5.2.3. Compared with SQP and SUMT [26]. For the NOCPs numbers (8)–(20), in the Appendix, the numerical results of LI and SI methods are compared with two numerical methods contain: SQP and SUMT, proposed in [26]. Among these NOCPs, only two problems numbers (8) and (18) are unconstrained, and the others have at least one constraint, final state constraint or inequality constraint. All of these NOCPs solved by the proposed algorithm, with the problem parameters in Table 2, and their results are summarized in Table 1. Because the final state constraints, in these methods, are not reported, we let $\varphi_f = 0$ to calculate the factor K_ψ.

Figures 4–16 show the graphical results of NOCPs numbers (8)–(20) in the Appendix, using SI method. For unconstrained NOCPs, numbers (8) and (18), only the graph of convergent rate of performance index is shown; see Figures 4 and 14. For constraint NOCPs and NOCPs numbers (11)–(17) and numbers (19)-(20), with final state constraint, the graphs of convergent rates of performance index and the error of final state constraint are shown; see Figures 7–13 and 15-16. For the constraint NOCPs with inequality constraints, NOCPs numbers (9) and (10), the graphs of convergent rate of performance index and inequality constraint are shown; see Figures 5 and 6.

Table 1 shows that the proposed algorithm, LI and SI methods, was 100 percent successful in point of views the performance index, J, and the factor, K_ψ, numerically. So, the proposed algorithm provides robust solutions with respect to the other mentioned, numerical or metaheuristic, methods. To compare J in LI and SI methods, in 35 percent of NOCPs, LI is more accurate than SI, in 10 percent SI is more accurate than LI and in 55 percent are same. In point of view the final conditions, 65 percent of NOCPs in the Appendix have final state constraints. In all of them φ_f in the proposed algorithm is improved, except CRP problem, however the factor of the proposed algorithm is the best, yet. In 54 percent of these NOCPs, LI is better performance and in 46 percent SI is better. Also, in point of the running time, Time, LI is same as SI, that is, in 50 percent of NOCPs LI and in 50 percent SI is better than another. So, the proposed algorithm could provide very suitable solutions, in a reasonable computational time. Also, for more accurate comparison of LI and SI methods a statistical approach will done in next section.

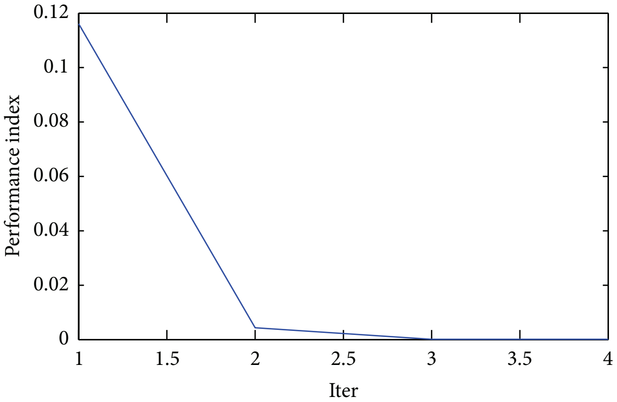
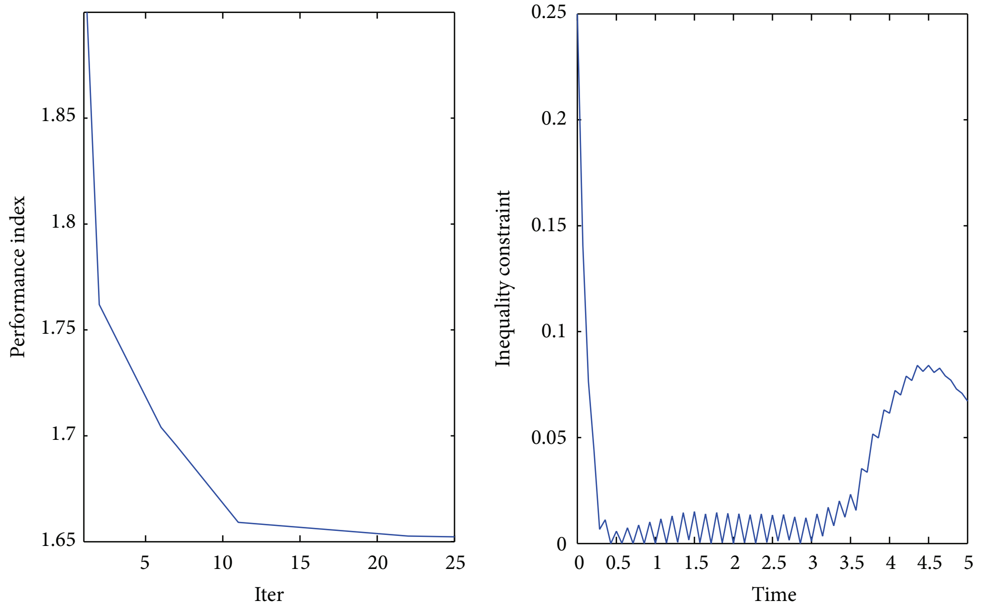


Figure 4: Convergence rate for the performance index of SI method for NOCP number (8).

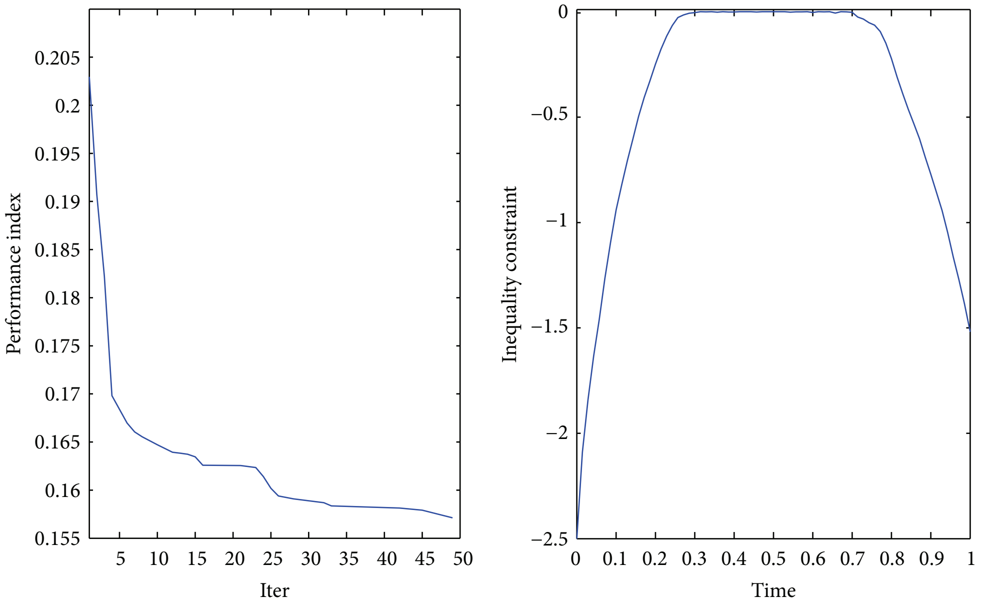
To compare with CGA, the mean of relative error of J, E_J, for CGA, LI and SI methods, in NOCPs (1)–(3), equal 0.5668, 0 and 2.98×10^{-4}, respectively. Also the mean for the error



(a)

(b)

Figure 5: (a) Convergence rate for the performance index and (b) the inequality constraint of SI method, for NOCP number (9).



(a)

(b)

Figure 6: (a) Convergence rate of the performance index and (b) the inequality constraint for SI method, for NOCP number (10).

of the final state constraints, φ_f, are 1.6×10^{-3}, 4.07×10^{-6} and 3.01×10^{-6}, respectively, and for K_ψ, these values equal 0.5883, 4.07×10^{-6} and 3.02×10^{-4}. Thus, we can say that the feasibility of the solutions given by the proposed algorithm and CGA are competitive.

From Table 1, to compare with numerical methods, SQP and SUMT, in NOCPs (8)–(20), the mean of E_J for LI, SI, SQP and SUMT equals 0.0145, 0.0209, 1.69×10^8 and 1.36×10^8, respectively. Also, the mean for the error of final state constraints, for these NOCPs, equal 3.34×10^{-4}, 4.41×10^{-5}, 0 and 0, respectively. For K_ψ, these values are 0.0213, 0.0014, 1.2263 and 1.1644. Therefore, the performance index, J, and the factor, K_ψ, for the LI and SI methods are more accurate than SQP and SUMT. So the proposed algorithm gave more better solution in comparison with the numerical methods.

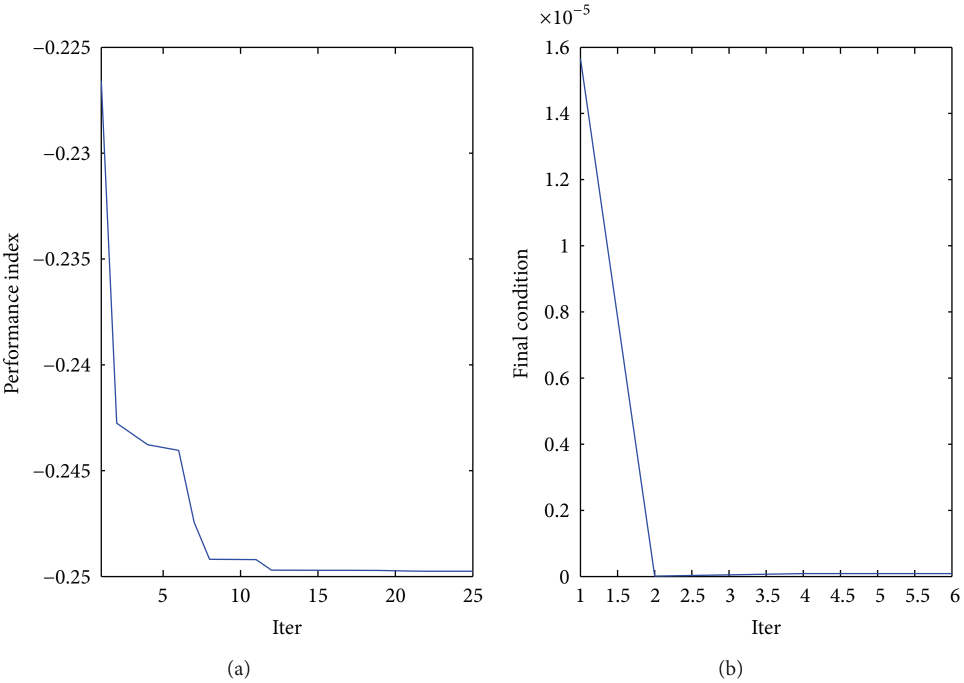


FIGURE 7: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (11).

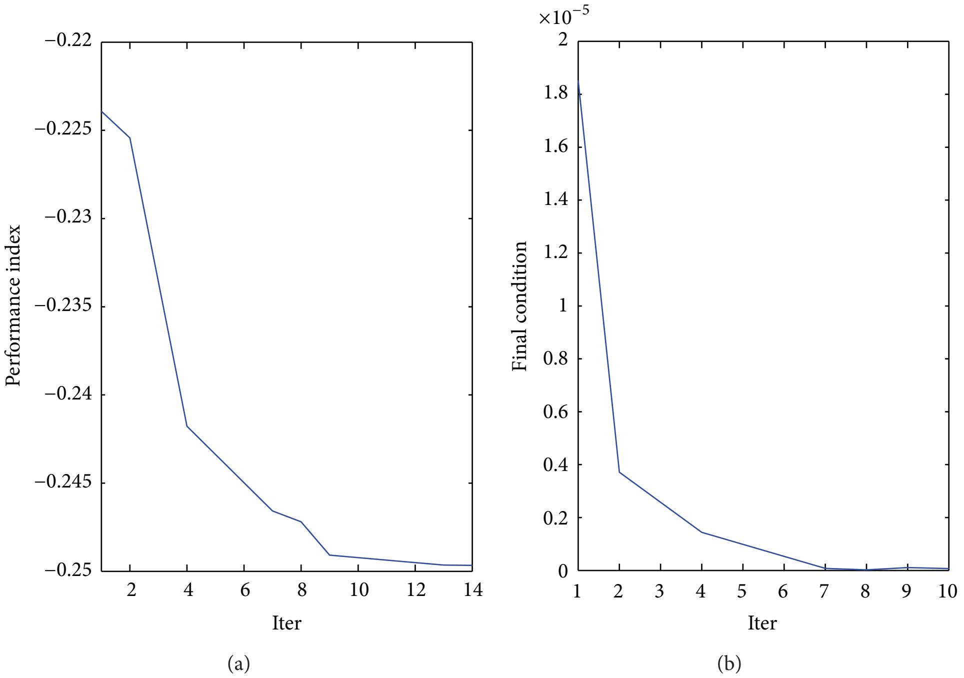


FIGURE 8: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (12).

Therefore, based on this numerical study, we can conclude that the proposed algorithm outperform well-known numerical method. Since, the algorithms were not implemented on the same PC, the computational times of them are not competitive. Therefore, we did not give the computational times in bold in Table 1.

6. Sensitivity Analysis and Comparing LI and SI

In this section, two statistical analysis, based on the one-way analysis of variance (ANOVA), used for investigating the sensitivity of MHGA parameters, and Mann-Whitney,

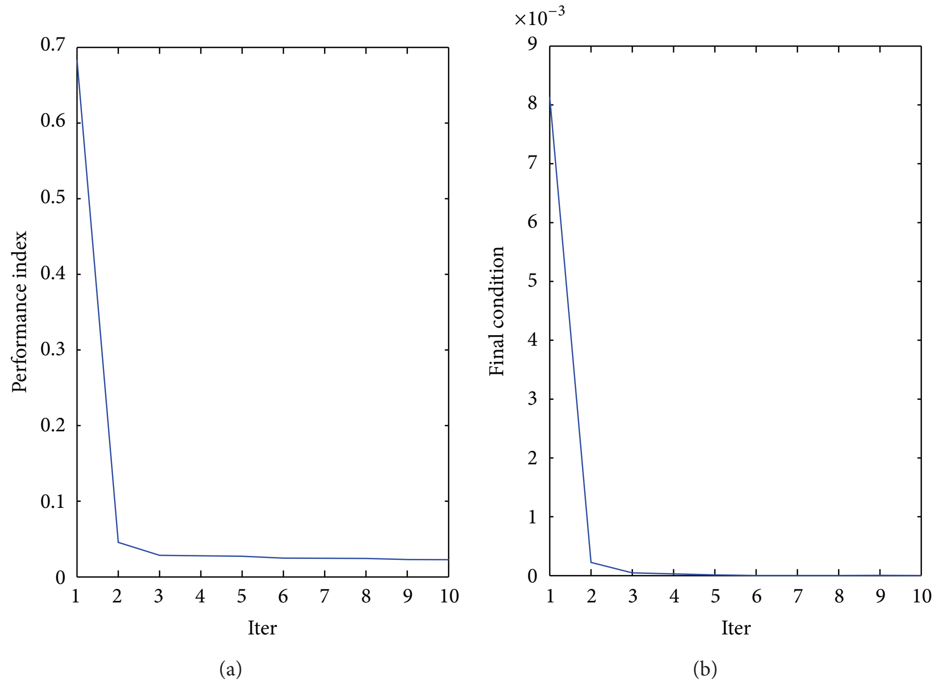


Figure 9: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (13).

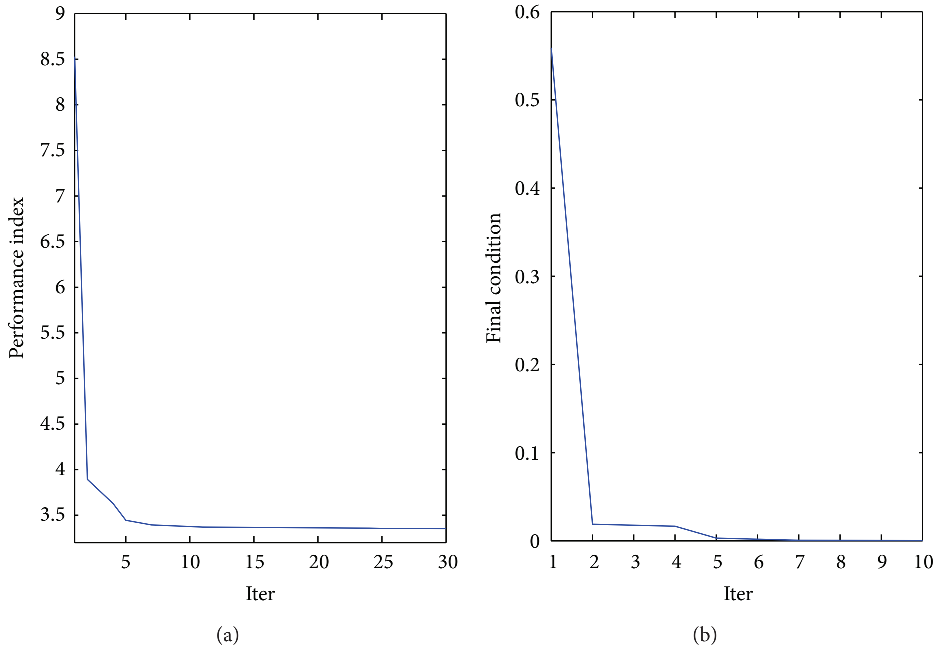


Figure 10: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (14).

applied comparing LI and SI methods, are done, by the statistical software IBM SPSS version 21.

6.1. Sensitivity Analysis. In order to survey the sensitivity of the MHGA parameters from the proposed algorithm, the VDP problem is selected, for example. The influence of these parameters are investigated for this NOCP on the dependent outputs consist of the performance index, J, the relative error of J, E_J, the required computational time, Time, the error of final state constraints, φ_f and the factor, K_ψ. The independent parameters are consist of the number of time nodes in both two phases, N_{t_1} and N_{t_2}, the size of population in both two phases, N_{p_1} and N_{p_2}, the maximum number of generations without improvement, N_i,

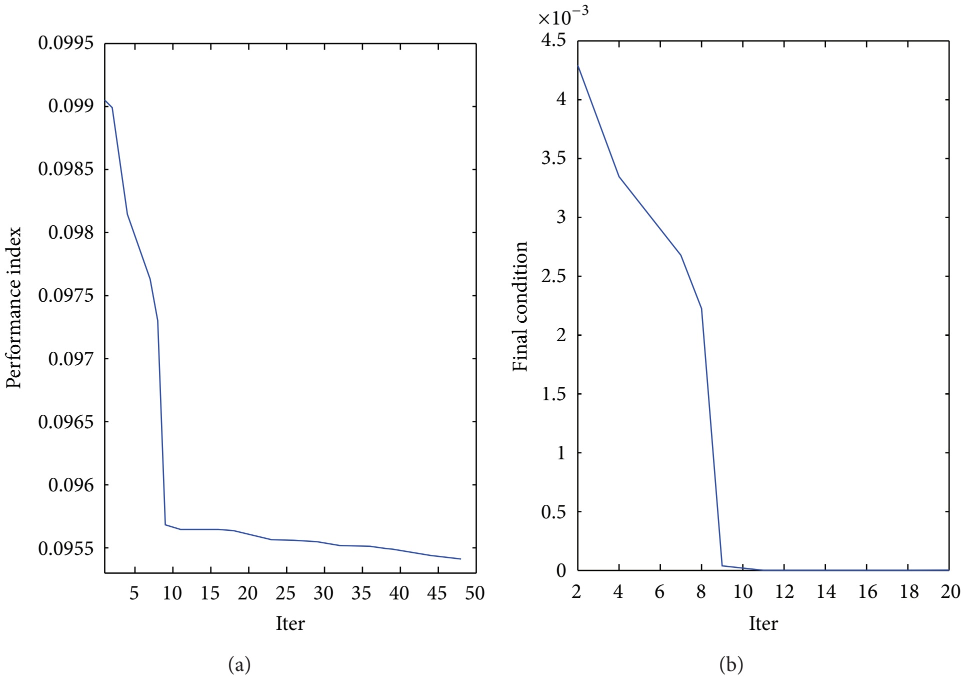


FIGURE 11: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (15).

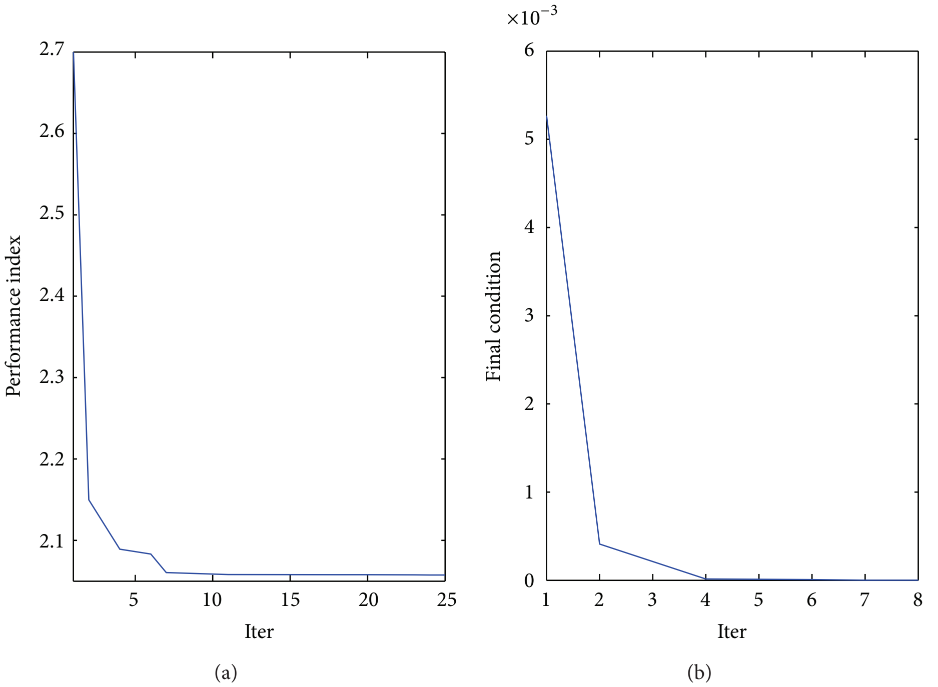


FIGURE 12: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (16).

and the maximum number of generations, N_g. Because the mutation implementation probability, P_m, has a less influence in numerical results, it is not considered. Also *sqpmaxiter* is changed in each iteration of the proposed algorithm, so it is not considered too.

At first, we selected at least four constant value for each of parameters and then in each pose, 35 different runs were made, independently. The statistical analysis is done based on ANOVA. The descriptive statistics, which contains the number of different runs (N), the mean of each

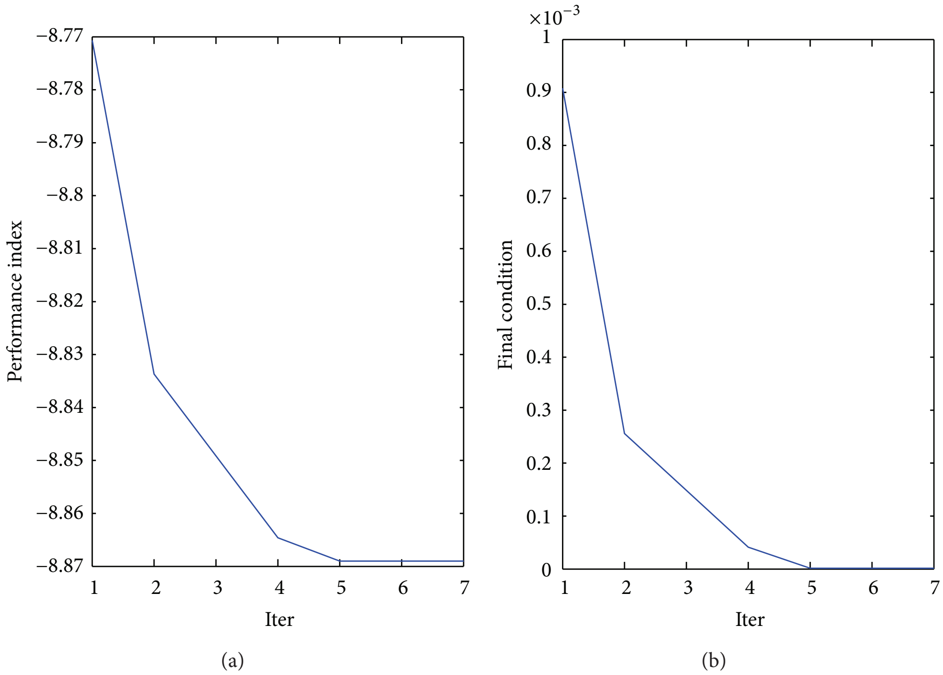


FIGURE 13: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (17).

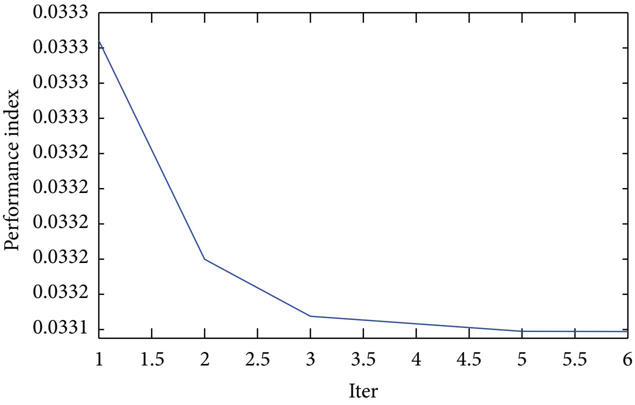


FIGURE 14: Convergence rate for the performance index of SI method for NOCP numbers (18).

output in N different runs, (Mean), standard deviation, (S.d), maximum, (Max) and minimum, (Min), of each output, could be achieved. Among of the MHGA parameters, we present the descriptive statistics of the ANOVA, only for the N_{t_1} parameter, which is reported in Table 3.

Table 4, summarized the statistical data, contain the test statistics (F) and P-values, of ANOVA tests. Sensitivity analysis for each parameter, separately, are done based on this table, as follows.

N_{t_1}: From Table 4, the significant level, or P-value, for K_ψ is equal to 0.279, which is greater than 0.05. So, from ANOVA, N_{t_1} parameter has no significant effect on the factor K_ψ. With similar analysis, this parameter has no effect on the other outputs, except required computational time, Time, because its P-value is equal to 0, which is less than 0.05, that is, this output sensitive to the parameter N_{t_1}.

N_{t_2}: From the second row of Table 4, the outputs J, E_J, K_ψ and Time are sensitive to the parameter N_{t_2}, and the only output φ_f is independent.

N_{p_1}: From the third row of Table 4, only the computational required time, Time, is sensitive, because the its P-value is equal to 0.006, which is less than 0.05, and the other outputs, contain J, E_J, φ_f, K_ψ, are independent.

N_{p_2}: From the forth row of Table 4, the output φ_f is independent respect to the parameter N_{p_2}, but other outputs, contain J, E_J, Time, K_ϕ are sensitive to this parameter.

N_g: From the fifth row of Table 4, the outputs J, E_J and K_ψ are independent to the parameter N_g, and other outputs, contain φ_f and Time are sensitive respect to this parameter.

N_i: The sensitivity analysis is similar to N_{p_1}.

From above cases, it is obvious that all parameters can be effect on the required computational time, except N_g. Moreover, intuitions shows the norm of final state constraints, φ_f is independent output with respect to all parameters, that is, any of the parameters could not effect on this output and it is not sensitive with respect to any of the MHGA parameters.

6.2. Comparison of LI and SI. To compare the efficiency of the LI and SI methods, for NOCPs in the Appendix, we used the Mann-Whitney nonparametric statistical test [37].

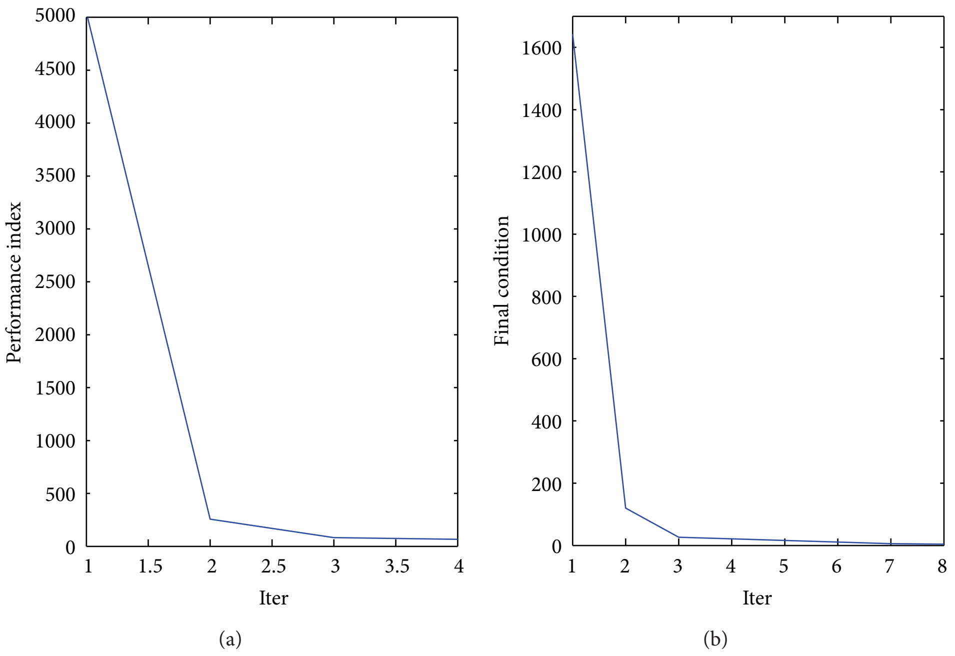


Figure 15: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (19).

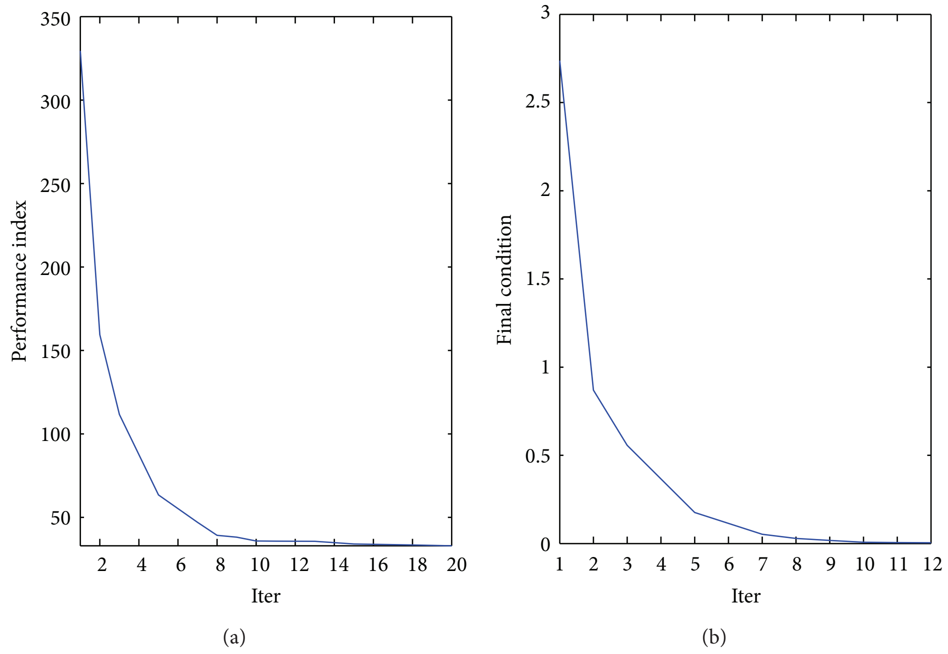


Figure 16: (a) Convergence rates of the performance index and (b) the error of final state constraint for SI method, for NOCP number (20).

Using the nonparametric statistical tests for comparing the performance of several algorithms presented in [38]. For this purpose, At first, 15 different runs, for each of NOCP were made. Table 5, shows the mean of numerical results, separately. Here, we apply the fix MHGA parameters as $N_{p_1} = 9$, $N_{p_2} = 12$, $N_{t_1} = 31$, $N_{t_2} = 91$, $N_g = 100$ and $N_i = 50$ with the previous problem parameters in Table 2. Comparison criterions contain: J, φ_f, K_ψ and Time. The results of Mann-Whitney test are shown in Tables 6–9. From Table 6, since $Z = 0 > Z_{0.01} = -2.34$, then the average relative error of the LI and SI methods are same with the significant level 0.01. In other words, with probability 99%, we can say that LI and SI methods have same effectiveness form the perspective of error for J. Similarly, the results of Tables 7–9 indicate that LI

Table 3: Descriptive statistics for the sensitivity analysis of N_{t_1}, for VDP problem.

Output	N_{t_1}	N	Mean	S.d	Min	Max
J	11	35	1.5333	0.0069	1.5286	1.5598
	21	35	1.5328	0.0097	1.5285	1.5849
	31	35	1.5329	0.0076	1.5285	1.5699
	41	35	1.5455	0.0607	1.5285	1.8449
	Total	**140**	**1.5357**	**0.0294**	**1.5285**	**1.8449**
φ_f	11	35	2.81×10^{-8}	9.52×10^{-8}	1×10^{-12}	4.52×10^{-7}
	21	35	4.67×10^{-8}	2.35×10^{-7}	1×10^{-12}	1.41×10^{-6}
	31	35	1.63×10^{-8}	4.91×10^{-8}	1×10^{-12}	2.21×10^{-7}
	41	35	8.17×10^{-8}	2.85×10^{-7}	1×10^{-12}	1.26×10^{-6}
	Total	**140**	$\mathbf{4.26 \times 10^{-8}}$	$\mathbf{1.91 \times 10^{-7}}$	$\mathbf{1 \times 10^{-12}}$	$\mathbf{1.41 \times 10^{-6}}$
Time	11	35	89.7838	27.2081	45.6614	155.4394
	21	35	109.8147	46.0264	60.4192	284.0310
	31	35	116.3591	30.0227	77.2673	179.5103
	41	35	144.8290	39.3091	81.0425	284.9514
	Total	**140**	**114.2131**	**41.0891**	**45.6615**	**284.9514**
E_J	11	35	0.0031	0.0045	0	0.0204
	21	35	0.0028	0.0064	0	0.0369
	31	35	0.0029	0.0050	0	0.0271
	41	35	0.0111	0.0397	0	0.2070
	Total	**140**	**0.0047**	**0.0192**	**0**	**0.2070**
K_ψ	11	35	0.0031	0.0045	4.75×10^{-9}	0.0204
	21	35	0.0027	0.0064	5.50×10^{-10}	0.0369
	31	35	0.0029	0.0050	1.54×10^{-13}	0.0271
	41	35	0.0111	0.0397	1.93×10^{-10}	0.2070
	Total	**140**	**0.0047**	**0.0192**	$\mathbf{1.54 \times 10^{-13}}$	**0.2070**

Table 4: Summary statistical data of ANOVA test for the parameters, $N_{t_1}, N_{t_2}, N_{p_1}, N_{p_2}, N_g, N_i$.

	Parameters	J	E_J	φ_f	Time	K_ψ
Test statistic (F)	N_{t_1}	1.294	1.296	0.624	10.79	1.296
	N_{t_2}	1105.72	3.93	0.868	59.39	3.93
	N_{p_1}	1.945	1.835	1.271	4.317	1.835
	N_{p_2}	2.740	2.478	1.011	23.02	2.478
	N_g	3.541	3.591	0.491	0.890	3.591
	N_i	0.495	0.309	0.301	5.849	1.046
P value	N_{t_1}	0.280	0.279	0.601	0	0.279
	N_{t_2}	0	0.005	0.489	0	0.005
	N_{p_1}	0.125	0.143	0.286	0.006	0.143
	N_{p_2}	0.021	0.034	0.413	0	0.034
	N_g	0.009	0.008	0.743	0.472	0.008
	N_i	0.739	0.819	0.877	0	0.386

and SI methods, from the perspective of errors for φ_f, Time, and K_ψ, have same behaviour.

7. The Impact of SQP

In this section, we investigation the impact of the SQP, on the proposed algorithm. For this purpose, we remove the SQP from Algorithm 2, as without SQP algorithm. For comparing the numerical results with the previous results, the required running time in each NOCP is considered fixed, which is the maximum of running time in 12 different runs, were done in Section 5. Also, the all of the parameters for each problem was set as same as Section 5. The numerical results of the without SQP algorithm is summarized in Table 10. By comparing the results of Tables 10 and 1, it is obvious that, for all NOCPs in the Appendix, the obtained values of the performance index and the norm of final state constraint for the proposed algorithm (Algorithm 2), are more accurate than the without SQP algorithm.

Table 5: The mean of numerical results of 15 different runs for NOCPs in the Appendix, using LI and SI methods.

Problem	LI				SI			
	J	φ_f	K_ψ	Time	J	φ_f	K_ψ	Time
VDP	1.5977	1.57×10^{-9}	1.70×10^{-3}	177.20	1.6002	6.67×10^{-8}	3.26×10^{-3}	157.57
CRP	0.0126	3.07×10^{-8}	3.07×10^{-8}	197.14	0.0119	1.61×10^{-8}	1.61×10^{-8}	195.32
FFPP	47.74	8.05×10^{-5}	0.3277	1332.80	53.51	5.92×10^{-5}	0.4868	1379.89
MSNIC	0.1702	—	—	136.78	0.1703	—	—	140.92
CSTCR	0.1120	—	—	173.85	0.1120	—	—	176.06
Number 6	−5.4307	—	—	105.41	−5.4308	—	—	103.72
Number 7	0.2015	3.26×10^{-9}	6.95×10^{-6}	208.07	0.2020	3.13×10^{-9}	6.62×10^{-6}	210.19
Number 8	6.30×10^{-15}	—	—	63.76	1.24×10^{-14}	—	—	58.37
Number 9	1.6512	—	—	592.39	1.6512	—	—	581.25
Number 10	0.1550	—	—	631.53	0.1550	—	—	655.97
Number 11	2.9701	2.70×10^{-9}	2.72×10^{-7}	228.60	2.9701	3.37×10^{-9}	8.11×10^{-7}	232.76
Number 12	−0.2499	2.01×10^{-8}	4.64×10^{-5}	547.18	−0.2499	1.15×10^{-9}	8.27×10^{-5}	527.65
Number 13	0.0147	1.15×10^{-8}	0.0252	577.97	0.0151	1.39×10^{-8}	0.0472	660.98
Number 14	3.4761	6.72×10^{-6}	0.0237	1045.85	3.4290	7.57×10^{-6}	9.50×10^{-3}	1087.67
Number 15	1.95×10^{-4}	1.11×10^{-8}	5.69×10^{-3}	363	1.94×10^{-4}	7.04×10^{-9}	4.86×10^{-3}	365
Number 16	2.0571	1.66×10^{-10}	1.66×10^{-10}	543.91	2.0571	1.15×10^{-10}	1.15×10^{-10}	544.31
Number 17	−8.8692	8.03×10^{-8}	8.03×10^{-8}	227.44	−8.8692	6.61×10^{-8}	6.61×10^{-8}	232.54
Number 18	0.0326	—	—	659.57	0.0326	—	—	708.97
Number 19	0.11588	6.55×10^{-4}	0.6828	1618.13	0.1091	8.84×10^{-4}	0.8808	1599.90
Number 20	52.42	0.0176	0.5429	841.65	44.64	4.53×10^{-4}	0.3765	868.91

Table 6: Results of Mann-Whitney test on relative errors of the pair (LI, SI) for J.

Method	Mean rank	Sum of ranks	Test statistics	Value
LI	20.50	410.0	Mann-Whitney U	200
SI	20.50	410.0	Wilcoxon W	410
—	—	—	Z	0

Table 7: Results of Mann-Whitney test on relative errors of the pair (LI, SI) for φ_f.

Method	Mean rank	Sum of ranks	Test statistics	Value
LI	13.62	177	Mann-Whitney U	83
SI	13.38	174	Wilcoxon W	174
—	—	—	Z	−0.077

Table 8: Results of Mann-Whitney test on relative errors of the pair (LI, SI) for Time.

Method	Mean rank	Sum of ranks	Test statistics	Value
LI	20.25	405.0	Mann-Whitney U	195
SI	20.75	415.0	Wilcoxon W	405
—	—	—	Z	−0.135

Table 9: Results of Mann-Whitney test on relative errors of the pair (LI, SI) for K_ψ.

Method	Mean rank	Sum of ranks	Test statistics	Value
LI	13.54	176.0	Mann-Whitney U	84
SI	13.46	175.0	Wilcoxon W	175
—	—	—	Z	−0.026

Table 10: The numerical results of the without SQP algorithm, for NOCPs in the Appendix.

	Problem no.									
	1	2	3	4	5	6	7	8	9	10
J	1.7296	0.0174	213.80	0.1702	0.1130	−5.2489	0.2390	1.69×10^{-1}	1.8606	0.2048
ϕ_f	3.30×10^{-9}	1.35×10^{-5}	1.1445	—	—	—	7.58×10^{-9}	—	—	—
	Problem no.									
	11	12	13	14	15	16	17	18	19	20
J	4.8816	−0.1273	0.0153	4.5727	9.79×10^{-4}	2.3182	−9.0882	0.0336	25.13	202.74
ϕ_f	1.67×10^{-5}	4.68×10^{-9}	0.0161	4.33×10^{-4}	5.94×10^{-3}	1.56×10^{-5}	1.64	—	17.06	2.5283

8. Conclusions

Here, a two-phase algorithm was proposed for solving bounded continuous-time NOCPs. In each phase of the algorithm, a MHGA was applied, which performed a local search on offsprings. In first phase, a random initial population of control input values in time nodes was constructed. Next, MHGA started with this population. After phase 1, to achieve more accurate solutions, the number of time nodes was increased. The values of the associated new control inputs were estimated by Linear interpolation (LI) or Spline interpolation (SI), using the curves obtained from phase 1. In addition, to maintain the diversity in the population, some additional individuals were added randomly. Next, in the second phase, MHGA restarted with the new population constructed by above procedure and tried to improve the obtained solutions at the end of phase 1. We implemented our proposed algorithm on 20 well-known benchmark and real world problems; then the results were compared with some recently proposed algorithms. Moreover, two statistical approaches were considered for the comparison of the LI and SI methods and investigation of sensitivity analysis for the MHGA parameters.

Appendix

Test Problems

The following problems are presented using notation given in (1)–(6).

(1) (VDP) [9] $g = (1/2)(x_1^2 + x_2^2 + u^2)$, $t_0 = 0$, $t_f = 5$, $f = [x_2, -x_2 + (1 - x_1^2)x_2 + u]^T$, $x_0 = [1, 0]^T$, $\psi = x_1 - x_2 + 1$.

(2) (CRP) [9] $g = (1/2)(x_1^2 + x_2^2 + 0.1u^2)$, $t_0 = 0$, $t_f = 0.78$, $f = [x_1 - 2(x_1 + 0.25) + (x_2 + 0.5)\exp(25x_1/(x_1 + 2)) - (x_1 + 0.25)u, 0.5 - x_2 - (x_2 + 0.5)\exp(25x_1/(x_1 + 2))]^T$, $x_0 = [0.05, 0]^T$, $\psi = [x_1, x_2]^T$.

(3) (FFRP) [9] $g = (1/2)(u_1^2 + u_2^2 + u_3^2 + u_4^2)$, $t_0 = 0$, $t_f = 5$, $f = [x_2, (1/10)((u_1 + u_2)\cos x_5 - (u_2 + u_4)\sin x_5), x_4, (1/10)((u_1 + u_3)\sin x_5 + (u_2 + u_4)\cos x_5), x_6, (5/12)((u_1 + u_3) - (u_2 + u_4))]^T$, $x_0 = [0, 0, 0, 0, 0, 0]^T$, $\psi = [x_1 - 4, x_2, x_3 - 4, x_4, x_5, x_6]^T$.

(4) (MSNIC) [20] $\phi = x_3$, $t_0 = 0$, $t_f = 1$, $f = [x_2, -x_2 + u, x_1^2 + x_2^2 + 0.005u^2]^T$, $d = [-(u - 20)(u + 20), x_2 + 0.5 - 8(t - 0.5)^2]^T$, $x_0 = [0, -1, 0]^T$.

(5) (CSTCR) [20] $g = x_1^2 + x_2^2 + 0.1u^2$, $t_0 = 0$, $t_f = 0.78$, $f = [-(2 + u)(x_1 + 0.25) + (x_2 + 0.5)\exp(25x_1/(x_1 + 2)), 0.5 - x_2 - (x_2 + 0.5)\exp(25x_1/(x_1 + 2))]^T$, $x_0 = [0.09, 0.09]^T$.

(6) [34] $g = 2x_1$, $t_0 = 0$, $t_f = 3$, $f = [x_2, u]^T$, $d = [-(2 + u)(2 - u), -(6 + x_1)]^T$, $x_0 = [2, 0]^T$.

(7) [36] $g = u^2$, $t_0 = 0$, $t_f = 1$, $f = (1/2)x^2 \sin x + u$, $x_0 = 0$, $\psi = x - 0.5$.

(8) [26] $g = x^2 \cos^2 u$, $t_0 = 0$, $t_f = \pi$, $f = \sin(u/2)$, $x_0 = \pi/2$.

(9) [26] $g = (1/2)(x_1^2 + x_2^2 + u^2)$, $t_0 = 0$, $t_f = 5$, $f = [x_2, -x_1 + (1 - x_1^2)x_2 + u]^T$, $d = -(x_2 + 0.25)$, $x_0 = [1, 0]^T$.

(10) [26] $g = x_1^2 + x_2^2 + 0.005u^2$, $t_0 = 0$, $t_f = 1$, $f = [x_2, -x_2 + u]^T$, $d = [-(20 + u)(20 - u), -(8(t - 0.5)(t - 0.5) - 0.5 - x_2)]^T$, $x_0 = [0, -1]^T$.

(11) [26] $g = (1/2)u^2$, $t_0 = 0$, $t_f = 2$, $f = [x_2, u]^T$, $x_0 = [1, 1]^T$, $\psi = [x_1, x_2]^T$.

(12) [26] $g = -x_2$, $t_0 = 0$, $t_f = 1$, $f = [x_2, u]^T$, $d = -(1 - u)(1 + u)$, $x_0 = [0, 0]^T$, $\psi = x_2$.

(13) [26] $g = (1/2)(x_1^2 + x_2^2 + 0.1u^2)$, $t_0 = 0$, $t_f = 0.78$, $f = [-2(x_1 + 0.25) + (x_2 + 0.5)\exp(25x_1/(x_1 + 2)) - (x_1 + 0.25)u, 0.5 - x_2 - (x_2 + 0.5)\exp(25x_1/(x_1 + 2))]^T$, $x_0 = [0.05, 0]^T$, $\psi = [x_1, x_2]^T$.

(14) [26] $g = (1/2)u^2$, $t_0 = 0$, $t_f = 10$, $f = [\cos u - x_2, \sin u]^T$, $d = -(\pi + u)(\pi - u)$, $x_0 = [3.66, -1.86]^T$, $\psi = [x_1, x_2]^T$.

(15) [26] $g = (1/2)(x_1^2 + x_2^2)$, $t_0 = 0$, $t_f = 0.78$, $f = [-2(x_1 + 0.25) + (x_2 + 0.5)\exp(25x_1/(x_1 + 2)) - (x_1 + 0.25)u, 0.5 - x_2 - (x_2 + 0.5)\exp(25x_1/(x_1 + 2))]^T$, $d = -(1 - u)(1 + u)$, $x_0 = [0.05, 0]^T$, $\psi = [x_1, x_2]^T$.

(16) [26] $\phi = x_3$, $t_0 = 0$, $t_f = 1$, $f = [x_2, u, (1/2)u^2]^T$, $d = x_1 - 1.9$, $x_0 = [0, 0, 0]^T$, $\psi = [x_1, x_2 + 1]^T$.

(17) [26] $\phi = -x_3$, $t_0 = 0$, $t_f = 5$, $f = [x_2, -2 + u/x_3, -0.01u]^T$, $d = -(30 - u)(30 + u)$, $x_0 = [10, -2, 10]^T$, $\psi = [x_1, x_2]^T$.

(18) [26] $\phi = (x_1 - 1)^2 + x_2^2 + x_3^2$, $g = (1/2)u^2$, $t_0 = 0$, $t_f = 5$, $f = [x_3 \cos u, x_3 \sin u, \sin u]^T$, $x_0 = [0, 0, 0]^T$.

(19) [26] $g = 4.5(x_3^2 + x_6^2) + 0.5(u_1^2 + u_2^2)$, $t_0 = 0$, $t_f = 1$, $f = [9x_4, 9x_5, 9x_6, 9(u_1 + 17.25x_3), 9u_2, -(9/x_2)(u_1 - 27.075x_3 + 2x_5x_6)]^T$, $x_0 = [0, 22, 0, 0, -1, 0]^T$, $\psi = [x_1 - 10, x_2 - 14, x_3, x_4 - 2.5, x_5, x_6]^T$.

(20) [26] similar to problem (3) with $\psi = [x_1 - 4, x_2, x_3 - 4, x_4, x_5 - \pi/4, x_6]^T$.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, Society for Industrial and Applied Mathematics, 2010.

[2] D. E. Kirk, *Optimal Control Theory: An Introduction*, Dover Publications, 2004.

[3] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes I. Characterization of the nominal solution," *Computers and Chemical Engineering*, vol. 27, no. 1, pp. 1–26, 2003.

[4] T. Binder, L. Blank, W. Dahmen, and W. Marquardt, "Iterative algorithms for multiscale state estimation. I. Concepts," *Journal of Optimization Theory and Applications*, vol. 111, no. 3, pp. 501–527, 2001.

[5] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt, "Dynamic optimization using adaptive control vector parameterization," *Computers and Chemical Engineering*, vol. 29, no. 8, pp. 1731–1751, 2005.

[6] Y. C. Sim, S. B. Leng, and V. Subramaniam, "A combined genetic algorithms-shooting method approach to solving optimal control problems," *International Journal of Systems Science*, vol. 31, no. 1, pp. 83–89, 2000.

[7] Z. Michalewicz, C. Z. Janikow, and J. B. Krawczyk, "A modified genetic algorithm for optimal control problems," *Computers and Mathematics with Applications*, vol. 23, no. 12, pp. 83–94, 1992.

[8] Y. Yamashita and M. Shima, "Numerical computational method using genetic algorithm for the optimal control problem with terminal constraints and free parameters," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 30, no. 4, pp. 2285–2290, 1997.

[9] Z. S. Abo-Hammour, A. G. Asasfeh, A. M. Al-Smadi, and O. M. Alsmadi, "A novel continuous genetic algorithm for the solution of optimal control problems," *Optimal Control Applications & Methods*, vol. 32, no. 4, pp. 414–432, 2011.

[10] X. H. Shi, L. M. Wan, H. P. Lee, X. W. Yang, L. M. Wang, and Y. C. Liang, "An improved genetic algorithm with variable population-size and a PSO-GA based hybrid evolutionary algorithm," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1735–1740, November 2003.

[11] I. L. Lopez Cruz, L. G. van Willigenburg, and G. van Straten, "Efficient differential evolution algorithms for multimodal optimal control problems," *Applied Soft Computing Journal*, vol. 3, no. 2, pp. 97–122, 2003.

[12] A. Ghosh, S. Das, A. Chowdhury, and R. Giri, "An ecologically inspired direct search method for solving optimal control problems with Bézier parameterization," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1195–1203, 2011.

[13] A. V. A. Kumar and P. Balasubramaniam, "Optimal control for linear system using genetic programming," *Optimal Control Applications & Methods*, vol. 30, no. 1, pp. 47–60, 2009.

[14] M. S. Arumugam and M. V. C. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 324–336, 2008.

[15] M. Senthil Arumugam, G. Ramana Murthy, and C. K. Loo, "On the optimal control of the steel annealing processes as a two-stage hybrid systems via PSO algorithms," *International Journal of Bio-Inspired Computation*, vol. 1, no. 3, pp. 198–209, 2009.

[16] J. M. van Ast, R. Babuška, and B. de Schutter, "Novel ant colony optimization approach to optimal control," *International Journal of Intelligent Computing and Cybernetics*, vol. 2, no. 3, pp. 414–434, 2009.

[17] M. H. Lee, C. Han, and K. S. Chang, "Dynamic optimization of a continuous polymer reactor using a modified differential evolution algorithm," *Industrial and Engineering Chemistry Research*, vol. 38, no. 12, pp. 4825–4831, 1999.

[18] F.-S. Wang and J.-P. Chiou, "Optimal control and optimal time location problems of differential-algebraic systems by differential evolution," *Industrial and Engineering Chemistry Research*, vol. 36, no. 12, pp. 5348–5357, 1997.

[19] S. Babaie-Kafaki, R. Ghanbari, and N. Mahdavi-Amiri, "Two effective hybrid metaheuristic algorithms for minimization of multimodal functions," *International Journal of Computer Mathematics*, vol. 88, no. 11, pp. 2415–2428, 2011.

[20] H. Modares and M.-B. Naghibi-Sistani, "Solving nonlinear optimal control problems using a hybrid IPSO-SQP algorithm," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 476–484, 2011.

[21] F. Sun, W. Du, R. Qi, F. Qian, and W. Zhong, "A hybrid improved genetic algorithm and its application in dynamic optimization problems of chemical processes," *Chinese Journal of Chemical Engineering*, vol. 21, no. 2, pp. 144–154, 2013.

[22] J. J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical Optimization: Theoretical and Practical Aspects*, Springer, London, UK, 2006.

[23] K. L. Teo, C. J. Goh, and K. H. Wong, *A Unified Computational Approach to Optimal Control Problems*, Pitman Monographs and Surveys in Pure and Applied Mathematics, Longman Scientific and Technical, 1991.

[24] C. J. Goh and K. L. Teo, "Control parametrization: a unified approach to optimal control problems with general constraints," *Automatica*, vol. 24, no. 1, pp. 3–18, 1988.

[25] B. C. Fabien, "Numerical solution of constrained optimal control problems with parameters," *Applied Mathematics and Computation*, vol. 80, no. 1, pp. 43–62, 1996.

[26] B. C. Fabien, "Some tools for the direct solution of optimal control problems," *Advances Engineering Software*, vol. 29, no. 1, pp. 45–61, 1998.

[27] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, John Wiley & Sons, New York, NY, USA, 2007.

[28] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, Berlin, Germany, 1999.

[29] K. Atkinson and W. Han, *Theoretical Numerical Analysis: A Functional Analysis Framework*, vol. 39 of *Texts in Applied Mathematics*, Springer, Dordrecht, The Netherlands, 3rd edition, 2009.

[30] A. E. Bryson, *Applied Optimal Control: Optimization, Estimation and Control*, Halsted Press, Taylor & Francis, 1975.

[31] W. Mekarapiruk and R. Luus, "Optimal control of inequality state constrained systems," *Industrial and Engineering Chemistry Research*, vol. 36, no. 5, pp. 1686–1694, 1997.

[32] C. A. Floudas and P. M. Pardalos, *Handbook of Test Problems in Local and Global Optimization*, Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[33] M. M. Ali, C. Storey, and A. Törn, "Application of stochastic global optimization algorithms to practical problems," *Journal of Optimization Theory and Applications*, vol. 95, no. 3, pp. 545–563, 1997.

[34] F. Ghomanjani, M. H. Farahi, and M. Gachpazan, "Bézier control points method to solve constrained quadratic optimal control of time varying linear systems," *Computational & Applied Mathematics*, vol. 31, no. 3, pp. 433–456, 2012.

[35] J. Vlassenbroeck, "A Chebyshev polynomial method for optimal control with state constraints," *Automatica*, vol. 24, no. 4, pp. 499–506, 1988.

[36] S. Effati and H. Saberi Nik, "Solving a class of linear and nonlinear optimal control problems by homotopy perturbation method," *IMA Journal of Mathematical Control and Information*, vol. 28, no. 4, pp. 539–553, 2011.

[37] D. C. Montgomery, *Applied Statistics and Probability for Engineers*, John Wiley & Sons, Toronto, Canada, 1996.

[38] S. García, D. Molina, M. Lozano, and H. Francisco, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization," *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization