Engineering Applications of Artificial Intelligence 27 (2014) 57–69

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A new genetic algorithm for solving optimization problems

Saber M. Elsayed *, Ruhul A. Sarker, Daryl L. Essam

School of Engineering and Information Technology, University of New South Wales at Canberra, Australia

ARTICLE INFO

Article history:
Received 22 November 2012
Received in revised form
2 July 2013
Accepted 23 September 2013
Available online 26 October 2013

Keywords:
Numerical optimization
Genetic algorithms
Multi-parent crossover
Constrained optimization

ABSTRACT

Over the last two decades, many different genetic algorithms (GAs) have been introduced for solving optimization problems. Due to the variability of the characteristics in different optimization problems, none of these algorithms has shown consistent performance over a range of real world problems. The success of any GA depends on the design of its search operators, as well as their appropriate integration. In this paper, we propose a GA with a new multi-parent crossover. In addition, we propose a diversity operator to be used instead of mutation and also maintain an archive of good solutions. Although the purpose of the proposed algorithm is to cover a wider range of problems, it may not be the best algorithm for all types of problems. To judge the performance of the algorithm, we have solved aset of constrained optimization benchmark problems, as well as 14 well-known engineering optimization problems. The experimental analysis showed that the algorithm converges quickly to the optimal solution and thus exhibits a superior performance in comparison to other algorithms that also solved those problems.

1. Introduction

Optimization is at the heart of many real-world problem solving processes. However, finding the optimal solution for such problems is often tedious, especially in the presence of non-linearity, high dimensionality, and multi-modality. Over the last few decades, evolutionary algorithms (EAs) have shown tremendous success in solving complex optimization problems. Although the EA family contains a number of different algorithms, the genetic algorithm (GA) is the most popular and widely used in practice (Goldberg, 1989).

GA uses crossover and mutation as the main search operators. Although two-parent crossover is widely used in practice, a good number of multi-parent (more than two parents) crossovers, such as unimodal distribution crossover (UNDX) (Ono et al., 2003), simplex crossover (SPX) (Tsutsui et al., 1999), parent centric crossover (PCX) (Deb et al., 2002) and triangular crossover (TC) (Elfeky et al., 2008), have also been employedfor solving optimization problems. UNDX uses multiple parents and creates offspring solutions around the center of mass of these parents, while a small probability is assigned to solutions away from the center of mass. Although it has shown excellent performance for highly epistasis problems (Ono et al., 2003), it may fail to generate offspring in some cases such as when the population size is too small relative to the search space. UNDX also has difficulty in finding an optimal solution near the boundaries of the search space (Ono et al., 2003). Simplex crossover (SPX) is a multi-parent recombination operator for real-coded GAs. The SPX operator assigns a uniform probability distribution for creating offspring within a restricted search space around the region marked by the parents. SPX uses the property of a simplex in the search space. It works well on functions having multimodality and/or epistasis with a medium number of parents, such as three parents on a low dimension function and four parents on a high dimension function (Tsutsui et al., 1999). However, it fails on functions that consist of tightly linked sub-functions (Tsutsui et al., 1999). PCX allows a large probability of creating a solution near each parent, rather than near the centre of the parents, in which for each offspring one parent is selected and a difference vector is calculated between the parent and the chosen μ parents. PCX applies a self adaptive approach where a new solution vector keeps moving towards the optimum (Deb et al., 2002). Note that when PCX is applied with GA, it takes longer time in comparison to other crossover operators, and it finds difficulty in solving separable multimodal problems (Rönkkönen, 2009). TC uses three parents for constrained problems, where two parents must be feasible and one infeasible, to generate a set of three offspring, where each offspring is generated as a linear combination of those three parents. TC works well where the optimal solution lies on the boundary of the feasible region of problems that have a single bounded feasible region in the continuous domain (Elfeky et al., 2008).

In this research, we have consider the fact that, for a better performance of GA, the distribution of offspring should neither be

* Corresponding author. Tel.: +61 2 626 88594; fax: +61 2 626 88276.
E-mail addresses: s.elsayed@adfa.edu.au (S.M. Elsayed), r.sarker@adfa.edu.au (R.A. Sarker), d.essam@adfa.edu.au (D.L. Essam).

0952-1976/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.engappai.2013.09.013

extremely narrow nor extremely wide in comparison to their parents. If the generated offspring are distributed narrower than their parents, they may lose diversity and converge prematurely. In contrast, if the offspring are distributed extremely widely, they may be too diverse and take too long in converging to optimality (Ono et al., 2003). So it should appropriately generate offspring that satisfies a balance between exploration and exploitation. Considering these facts, in this paper, we have introduced a new three-parent crossover with GA that generates three new offspring. Two of these offspring are to help exploitation while the third offspring is for promoting exploration. A diversity operator is also used in the algorithm that helps to escape from local optima thus avoids premature convergence. The algorithm has been tested by solving the CEC2006 benchmark problems, as well as a variety of complex real world problems, such as those concerning frequency-modulated sound waves, catalyst blend optimal control, transmission network planning, transmission pricing, antenna design, static and dynamic despatching, and spacecraft trajectory optimization. These problems are very hard in the sense that they have strong nonlinearities with many local minima. We have compared our results with other state-of-the-art algorithms and show thereby the superior performance of the proposed algorithm. It is worthy to mention here that this research study is different than (Elsayed et al., 2011b), in which a small set of test problems was used. In addition to that different constraint handling techniques plus a local search were used. Furthermore, no analysis study was provided. This paper is also different from Elsayed et al. (2011a), in which a different constraint handling method was used as well as no deep analysis was provided.

This paper is organized as follows: after the introduction, Section 2 presents the proposed crossover, the diversity operator, the design of our algorithm and its similarities and differences to other EAs. Section 3 describes the used problems. The experimental results, and the analysis of those results, are presented in Section 4. Finally, the conclusions are given in Section 5.

2. GA with multi-parent crossover (GA-MPC)

In this section, firstly, we describe the proposed crossover and the diversity operator. The proposed algorithm is then presented, as well as the constraint handling technique that is used in this research.

To begin with, let us define the mathematical model for a constrained optimization problem (COP)

$$\min f(\vec{X})$$

Subject to

$$\begin{aligned} &g_k(\vec{X}) \le 0, && k = 1, 2, \ldots, K, \\ &h_e(\vec{X}) = 0, && e = 1, 2, \ldots, E, \\ &\underline{L}_j \le x_j \le \overline{U}_j, && j = 1, 2, \ldots, D, \end{aligned} \tag{1}$$

where $\vec{X} \in \Re^D$, D is the number of decision variables, $f(\vec{X})$ the objective function, $g_k(\vec{X})$ the kth inequality constraints, $h_e(\vec{X})$ the eth equality constraint, and each x_j has a lower limit $\underline{L}_j$ and an upper limit $\overline{U}_j$.

2.1. The proposed crossover

In this research, we deal with real-valued encoding. We propose a multi-parent crossover (MPC) with the following steps:

(1) Based on a selection rule, store the individuals that will be used for crossover into a selection pool.
(2) Any duplication in the selected three individuals is removed by replacing the unwanted individual with a random individual from the selection pool.
(3) Rank these three individuals from the best ($\vec{x}_1$) to the worst ($\vec{x}_3$), based on their fitness functions and/or constraint violations.
(4) Generate a random number β that follows a normal distribution with mean value μ and standard deviation σ.
(5) Generate three offspring (o_i)

$$\vec{o}_1 = \vec{x}_1 + \beta \times (\vec{x}_2 - \vec{x}_3) \tag{2}$$

$$\vec{o}_2 = \vec{x}_2 + \beta \times (\vec{x}_3 - \vec{x}_1) \tag{3}$$

$$\vec{o}_3 = \vec{x}_3 + \beta \times (\vec{x}_1 - \vec{x}_2) \tag{4}$$

where $f(\vec{x}_1) \le f(\vec{x}_2) \le f(\vec{x}_3)$.

The idea behind MPC comes from the heuristic crossover (Wright, 1991), in which one offspring ($\vec{y}$) is generated from a given pair of parents ($\vec{x}_1, \vec{x}_2$), such that $\vec{y} = (\vec{x}_1) + r\,and$ $(\vec{x}_1 - \vec{x}_2)$, where $r\,and$ is a random number between 0 and 1, and $f(\vec{x}_1) \le f(\vec{x}_2)$. However, we use three parents instead of two, similarly to the multi-parent crossovers TC, PCX, SPX, and UNDX, as discussed earlier. In our case, the difference vectors in the above equations are not in the same order. The order in Eq. (3) is set differently from the same in Eqs. (2) and (4). In fact, Eqs. (2) and (4) are designed to move toward better fitness while Eq. (3) is to diversify the population.

2.2. A diversity operator

In this paper, we also propose a diversity operator to further diversify the generated offspring. While applying the diversity operator, two points should be considered: the proportion of population undergoing such operation (based on the diversity probability), and the scale of contribution of the operator (based on the information used to generate the final offspring).

In implementing the proposed diversity operator, we use information from the generated offspring o_i and from the archive pool (A) that contains the best m individuals in the current generation. The final offspring $o_i = \{o_i^1, o_i^2, \ldots, o_i^j, \ldots o_i^D\}$ is generated based on the diversity operator as of the following pseudocode:

For $i \in \{1, 2, \ldots PS\}$s
  For $j \in \{1, 2, \ldots D\}$
  If $rand(0, 1) < p$
  Select a random individual from the archive pool (A_{arch}).
  $o_i^j = x_{arch}^j$
  End
  End
End

where p is a diversity probability, o_i the ith offspring, PS the population size and $arch \in \{1, 2, \ldots, m\}$. Note that to generate the final o_i^j, we may use different values for $arch$, this is because we do not want to use information from a single vector as that may lead to becoming stuck in a local minima.

2.3. GA-MPC

In GA-MPC, first an initial population is generated randomly, with size PS. Then an archive pool is filled with the best m individuals (based on their constraint violations and/or fitness

Table 1
GA with multi-parent crossover (GA-MPC).

STEP 1: In generation $t = 0$, generate an initial random population of size *PS*. The variables in each individual (i) must be within the range as shown below:
$x_i^j = \underline{L} + rand \times (\overline{U}_j - \underline{L})$
where *rand* is a random number $\in [0,1]$.
STEP 2: Sort all individuals based on their constraint violations and/or objective function, and save the best m individuals in the archive pool (A).
STEP 3: Apply a tournament selection with size *TC* (randomly 2 or 3), and fill the selection pool (contains all individuals that win in the tournament process). The selection pool size should be 3*PS*.
STEP 4: For each three consecutive individuals, If $u \in [0,1] < cr$
Rank these three individuals from $f(x_i) \le f(x_{i+1}) \le f(x_{i+2})$
If one of the selected individuals is the same as another, then replace one of them with a random individual from the selection pool.
Calculate $\beta = N(\mu, \sigma)$, where $\mu = 0.7$, and $\sigma = 0.1$.
Generate three offspring (o_i):
$o_1 = x_1 + \beta \times (x_2 - x_3)$
$o_2 = x_2 + \beta \times (x_3 - x_1)$
$o_3 = x_3 + \beta \times (x_1 - x_2)$
STEP 5: For each o_i^j, generate a random number $u \in [0,1]$. If $u \in [0,1] < p$, then $o_i^j = x_{arch}^j$, where $arch \in [1, m]$.
STEP 6: If there is any duplicate individual, then
$x_i^j = x_i^j + N(0.5u, 0.25rand)$, where $rand \in [0,1]$.
STEP 7: Stop if the termination criterion is met; **else** go to **STEP 2**, and set $t = t + 1$.

function). Then a tournament selection procedure with size *tc* takes place, from which the best individual is chosen and saved in the selection pool. For the crossover operation, with a crossover rate (*cr*), for each three consecutive individuals in the selection pool, three offspring are generated as described before.After we generate all offspring, we apply a diversity operator, with a probability p, to escape from any local minima and to visit better regions in the search space. After that we merge the individuals from the archive pool with all of the offspring, and the best *PS* individuals are selected as a new population for the next generation. Also, to ensure more diversity, if any individual in the population is exactly same to another one, then one of them is shifted within the boundary with $N(0.5u, 0.25u)$, where $u \in [0,1]$. Table 1 shows the steps of GA-MPC.

2.4. Constraint handling

Constrained problems are converted to unconstrained ones, in which the objective function is penalized as

$$f(\overrightarrow{X}) = f(\overrightarrow{X}) + Penalty \tag{5}$$

The penalty value, which represents the constraints, here is equal to the sum of the individual violation of all of the violated constraints. The penalty part is handled as follows:

$$Penalty = \max(0, \sum_{k=1}^{K} \max\{0, g_k(\overrightarrow{x})\} + \sum_{e=1}^{E} \max(0, |h_e(\overrightarrow{x})| - \varepsilon)) - tol) \tag{6}$$

where *tol* is updated according the following equation:

$$tol = \begin{cases} initial\_tol, & if\, t \le n1 \\ initial\_tol - initial\_tol \times \left(\frac{t-n1}{n2}\right), & if\, t > n1 \end{cases} \tag{7}$$

where *initial_tol* is equal to the penalty value for the top θ-th individual and $\theta = (0.20 \times PS)$ t is the generation number, $n1$ is equal to 15% of the maximum generations ($T_{\max}$), $n_2$is equal to 35% of the T_{max}. This means that after 50% of T_{max}, *tol* is equal to zero. The idea is quite similar to the ε-Constraint method (Takahama and Sakai, 2009).

2.5. Similarity and differences to other EAs

Although we are using the concept of multi-parent crossover, the design is different from other GA crossovers, in which our proposed crossover neither uses a mean-centric probability distribution, such as UNDX and SPX, nor uses aparent-centric approach, such as PCX. In addition, we are not using a linear combination of parent vectors to generate the offspring as in TC. As described earlier, it is also different from the heuristic crossover.

In our case, three offspring are generated from three parents as opposed to one offspring from a pair of parents in heuristic crossover.

One may find the proposed crossover is somewhat similar to the mutation strategy of differential evolution (DE) (Storn and Price, 1995). In DE, for each parent vector from the current population (target vector), a mutant vector (donor vector) is obtained. The simplest form of generating the donor vector is that three distinct parents are selected, and the difference of two of the parents multiplied by the amplification factor is added to the third vector. Note that all of the three parents must be different from the target vector. Finally an offspring is formed by combining the donor with the target vector. The resulting child of that combination is called trial vector. So the similarity is that both GA-MPC and DE use a differential operation. The differences are:

- For each parent in DE, only one offspring is generated, while in GA-MPC for each three distinct parents, three offspring are generated.
- A recombination methodology between the target and the donor is used to generate the trial vector, while in GA-MPC a diversity operator with a probability p is used between each generated offspring and different individuals within the archive pool.
- In DE, the parents used to generate the donor vector are selected randomly, must be different to each other and different to the target vector. In GA-MPC, the selection of the parents for crossover operation is based on a tournament, and they must be different to each other.
- In DE, the better of the two vectors (the target and the trial vector) is selected for the next generation. In GA-MPC, the best *PS* individuals from the mix of the archive pool individuals and all generated offspring are selected for the next generation.

3. Problems description

In this section, we describe the 24 well-known constrained benchmark problems, and a number of engineering optimization problems, that we have used to judge the performance of the proposed algorithm.

Table 2
Details of the 24 test problems, where D is the number of decision variables, $|F|/|S|$ the estimated ratio between the feasible region and the search space, LI the number of linear inequality constraints, NI the number of nonlinear inequality constraints, LE the number of linear equality constraints, NE the number of nonlinear equality constraints, and a the number of active constraints.

Prob.	D	Obj. Fun.	$\|F\|/\|S\|$ (%)	LI	NI	LE	NE	a	Optimal
g01	13	Quadratic	0.0000	9	0	0	0	6	−15.000000000
g02	20	Nonlinear	99.9971	0	2	0	0	1	−0.8036191042
g03	10	Polynomial	0.0000	0	0	0	1	1	−1.0005001000
g04	5	Quadratic	52.1230	0	6	0	0	2	−30665.53867178
g05	4	Cubic	0.0000	2	0	0	3	3	5126.4967140071
g06	2	Cubic	0.0066	0	2	0	0	2	−6961.813875580
g07	10	Quadratic	0.0003	3	5	0	0	6	24.3062090681
g08	2	Nonlinear	0.8560	0	2	0	0	0	−0.0958250415
g09	7	Polynomial	0.5121	0	4	0	0	2	680.6300573745
g10	8	Linear	0.0010	3	3	0	0	6	7049.2480205286
g11	2	Quadratic	0.0000	0	0	0	1	1	0.7499000000
g12	3	Quadratic	4.7713	0	1	0	0	0	−1.0000000000
g13	5	Nonlinear	0.0000	0	0	0	3	3	0.0539415140
g14	10	Nonlinear	0.0000	0	0	3	0	3	−47.7648884595
g15	3	Quadratic	0.0000	0	0	1	1	2	961.7150222899
g16	5	Nonlinear	0.0204	4	34	0	0	4	−1.9051552586
g17	6	Nonlinear	0.0000	0	0	0	4	4	8853.5396748064
g18	9	Quadratic	0.0000	0	13	0	0	6	−0.8660254038
g19	15	Nonlinear	33.4761	0	5	0	0	0	32.6555929502
g20	24	Linear	0.0000	0	6	2	12	16	0.2049794002
g21	7	Linear	0.0000	0	1	0	15	6	193.7245100700
g22	22	Linear	0.0000	0	1	8	11	19	236.4309755040
g23	9	Linear	0.0000	0	2	3	1	6	−400.0551000000
g24	2	Linear	79.6556	0	2	0	0	2	−5.5080132716

3.1. Benchmark problems

These benchmark problems have been taken from the 2006 IEEE Congress on Evolutionary Computation (CEC'2006) (Liang et al., 2005) competition on single objective constrained problems and their details are shown in Table 2.

3.2. Engineering optimization problems

As we mentioned earlier, a variety of engineering optimization problems are considered in this research. These problems vary in terms of mathematical properties, presence and absence of functional constraints, number of variables, static or dynamic in nature, and modality. For constrained problems, the number of constraints and type of constraints also vary. These problems are briefly discussed below.

3.2.1. P01: Parameter estimation for frequency-modulated (FM) sound waves

The problem is to specify six parameters, $a_1,\ \omega_1, a_2, \omega_2, a_3, \omega_3$ of a sound modulation model represented by the following equation:

$$y(t) = a_1 \sin(\omega_1 t\theta + a_2 \sin(\omega_2 t\theta + a_3 \sin(\omega_3 t\theta))) \tag{8}$$

where $\theta = (2 \times \pi/100)$. The fitness function ($f$) is defined as

$$f = \sum_{t=0}^{100} (y(t) - y_0(t))^2) \tag{9}$$

$$y_0(t) = 1.0 \sin(5t\theta - 1.5 \sin(4.8t\theta + 2 \sin(4.9t\theta))) \tag{10}$$

Each parameter is in the range $[-6.4, 6.35]$. This problem is a highly complex multimodal one having strong epistasis, with minimum value $f^* = 0$. This problem has been previously studied by a GA (Herrera and Lozano, 2000).

3.2.2. P02: Lennard-Jones potential problem

This is a potential energy minimization problem that involves the minimization of molecular potential energy associated with a pure Lennard-Jones (LJ) cluster (Hoare, 2007; Moloi and Ali, 2005). This problem is a multi-modal optimization problem comprised of an exponential number of local minima (Hoare, 2007). Most of the global minima have structures based upon the Mackay icosahedrons and can be seen in the Cambridge Cluster Database (http://wwwwales.ch.cam.ac.uk/CCD.html). An algorithm can be tested over this function for its capability to conform its molecular structure, where the atoms are organized in such a way that the molecule has minimum energy. The LJ potential energy between two atoms separated by distance r is given by

$$LJ(r) = \frac{1}{r^{12}} - \frac{2}{r^6} \tag{11}$$

The total potential energy f_{LJ} of a cluster of n_p atoms is defined by

$$f_{LJ}(X_1, \ldots, X_{n_p}) = \sum_{i<j} LJ(||X_i - X_j||) \tag{12}$$

where $X_1 \in R_3$ represents the coordinates of the ith atom and the norm is the Euclidean distance.

3.2.3. P03: the bifunctional catalyst blend optimal control problem

This is a hard multimodal optimal control problem which has many local optima, as many as as 300 local optima have been reported in (Esposito and Floudas, 2000). The proposed problem is a chemical process which converts methylcyclopentane to benzene in a tubular reactor. The catalyst contains a hydrogenation component and an omerization component. The objective is to determine the mixture of the two along the length of the reactor which maximizes the concentration of the desired product in a given reaction scheme.

3.2.4. P04: optimal control of a non-linear stirred tank reactor

This is a model of a nonlinear continuous stirred tank reactor which involves two different local minima. This problem is a multimodal optimal control problem (Ali et al., 1997). The optimization objective is to determine a suitable value of u so that the performance index

$$J = \int_{t_0}^{t_f} (x_1^2 + x_2^2 + 0.1u^2)dt \tag{13}$$

is minimized and where the initial conditions are $x(0) = [0.09 \quad 0.09]^T$. Though the problem is unconstrained, the initial guess for $u(t)$ lies in [0.0 5.0], where $t \in [t_0, t_f]$, $t_0 = 0$ and $t_f = 0.72$. In this research, the integration involved in evaluation is performed using a sub-function *ode*45 that is available in MATLAB with relative tolerance set to 0.1.

3.2.5. P05 Tersoff potential function minimization problem

Tersoff potential governs the interaction of silicon atoms with a strong covalent bonding. Tersoff has given two parameterizations of silicon and these are called Si(B) and Si(C), and the two sets of parameter values respectively for Si(B) and Si(C) are tabulated in Table 2 (Ali and Törn, 2000).

Although the Tersoff potential problem is $N \times 3$ dimensional in 3-dimentional space, the number of dimensions to be evaluated can be decreased in light of the fact that it depends on the relative position of each atom instead of its actual Cartesian coordinates.

This problem is difficult, because the presence of a large number of local minimizers, even for a system with a small number of atoms, creates numerous regions of attraction for local searches (Ali and Törn, 2000). In this research we consider a 10 atoms problem, so the number of decision variables is 30.

3.2.6. P06: spread spectrum radar polly phase code design

In radar systems with pulse compression, the choice of the appropriate waveform is a key point. The problem under consideration is modeled as a min–max nonlinear non-convex optimization problem with continuous variables and numerous local optima, and in which the objective function is piecewise smooth (Dukic and Dobrosavljevic, 1990). The problem can be considered as an *NP*-hard problem. The objective is to minimize the module of the biggest among the samples of the so called auto-correlation function which is related to the complex envelope of the compressed radar pulse at the optimal receiver output, while the variables represent symmetrized phase differences.

3.2.7. P07: transmission network expansion planning (TNEP)

The electrical power transmission network expansion planning problem of consists in finding the optimal plan for an electrical system expansion, that is, it must specify the transmission lines and/or transformers that should be constructed so that the system can operate in an adequate way during a specified planning horizon (de J Silva et al., 2005). This problem can be considered difficult because of its nonconvex nature.

The mathematical model for the transmission network expansion planning problem without security constraints presents the following compact structure:

$$\min v = \sum_{(i,j)=\Omega} c_{ij} n_{ij} \tag{14}$$

Subject to

$$\begin{aligned} & Sf + g = d \\ & f_{i,j} - \gamma_{i,j}(n_{ij}^0 + n_{ij})(\theta_i - \theta_j) = 0 \\ & |f_{i,j}| \le (n_{ij}^0 + n_{ij})\overline{f}_{i,j} \\ & 0 \le g \le \overline{g} \\ & 0 \le n_{ij} \le \overline{n}_{ij} \end{aligned} \tag{15}$$

where n_{ij} are integers, and $f_{i,j}$ and θ_j are unbounded, $(i,j) \in \Omega$. Respectively, c_{ij}, $\gamma_{i,j}$, n_{ij}, n_{ij}^0, $f_{i,j}$ and $\overline{f}_{i,j}$ are the cost of a circuit that can be added to a right-of-way $i-j$, the susceptance of that circuit, the number of circuits added in a right-of-way $i-j$, the number of circuits in the base case, the total power flow and the corresponding maximum power flow by circuit in a right-of-way $i-j$. The variable v is the investment, S is the branch-node incidence transposed matrix of the power system, f is a vector with elements$f_{i,j}$, g is a vector with elements g_k (generation in bus k) whose maximum value is g, d is the demand vector,$n_{ij}$is the maximum number of circuits that can be added in a right-of way $i-j$, y_i is the phase angle in bus j, and O is the set of all rights-of-way.

3.2.8. P08: large scale transmission pricing problem

The changing environments surrounding the electricity industry have resulted in the unbundled services provided by electric utilities. Power wheeling is one of the most prevailing unbundled services which can be provided by the electric utilities. Pricing of transmission services plays a crucial role in determining whether providing transmission services is economically beneficial to both the wheelers and the customers (Galiana et al., 2003).

Many factors influence the decision about the scheme of transmission pricing to be adopted. In (Christie et al., 2000), authors proposed the principle of equivalent bilateral exchanges (EBE). That principle stated that since a solved optimal power flow meets the laws of Kirchhoff without violating any line flow or generation limit, each generation injection flows without impediment toward all of the demands, while each demand is fed by all of the injected generators.

Following the EBE principle above, an equivalent bilateral power exchange between the generation at bus i and the demand at bus j is defined by

$$GD_{ij} = \frac{P_{gi} P_{dj}}{P_d^{sys}} \tag{16}$$

where, P_d^{sys} is the total load in the system. The net flow in the line can be expressed in terms of equivalent bilateral power exchanges as

$$pf_k = \sum_i \sum_j |\gamma_{ij}^k| GD_{ij} \tag{17}$$

The objective is to solve the following equations

$$\begin{aligned} \min F(GD_{ij}) = \\ \sum_i & \left[\frac{\left[\sum_j GD_{ij} \left[\sum_j \gamma_{ij}^k \left[FC^k / \sum_i \sum_j |GD_{ij}\gamma_{ij}^k| + \sum_i \sum_j |BT_{ij}\gamma_{ij}^k|\right]\right]\right]}{P_{gi} - P'_{gi}} \right] \\ & + \sum_j \left[\frac{\left[\sum_i GD_{ij} \left[\sum_k \gamma_{ij}^k \left[FC^k / \sum_i \sum_j |GD_{ij}\gamma_{ij}^k| + \sum_i \sum_j |BT_{ij}\gamma_{ij}^k|\right]\right]\right]}{P_{dj} - P'_{dj}} \right] \end{aligned}$$

Subject to

$$\begin{aligned} & \sum_i GD_{ij} = P_{gi} - P'_{gi} \\ & \sum_i GD_{ij} = P_{dj} - P'_{dj} \end{aligned} \tag{18}$$

3.2.9. P09: circular antenna array design problem

Antenna arrays have been widely used in different applications, including radar, sonar, biomedicine, communications, and imaging (Mandal et al., 2011). Antenna arrays may be linear, two-dimensional, circular or spherical in element arrangement. A very popular type of antenna arrays is the circular array which has several advantages over other schemes. The problems are difficult due to the complicated structure and overall array configurations (Mandal et al., 2011).

Let us consider N antenna elements spaced on a circle of radius r in the x-y plane. The objective function is taken as

$$\begin{aligned} F = & |AR(\varphi_{sll}, \vec{I}, \vec{\beta}, \varphi_0)| / |AR(\varphi_{max}, \vec{I}, \vec{\beta}, \varphi_0)| + \frac{1}{DIR(\varphi_0, \vec{I}, \vec{\beta})} \\ & + |\varphi_0 - \varphi_{des}| + \sum_{k=1}^{num} |AR(\varphi_k, \varphi_0, \vec{I}, \vec{\beta}, \varphi_0)| \end{aligned} \tag{19}$$

where AR is the array factor for a conventional linear array. The first component attempts to suppress the side-lobes. φ_{sll} is the angle at which the maximum side-lobe level is attained. The second component attempts to maximize directivity of the array pattern. Nowadays directivity has become a very useful figure of merit for comparing array patterns. The third component strives to drive the maxima of the array pattern close to the desired maxima φ_{des}, DIR is the directivity for the radiation pattern The fourth component penalizes the objective function if sufficient null control is not achieved, num is the number of null control directions and φ_k specifies the k-th null control direction. In this research we use the following instantiation of the design problems: number of elements in circular array$=12$, $x1=$any string within bounds, null$=[50,120]$ in radians (no null control), $\varphi_{des}=180$ and distance$=0.5$.

3.2.10. P10: dynamic economic dispatch (DED) problem

Dynamic economic dispatch (DED) is one of the main functions of power generation operation and control. It determines the optimal settings of generator units with predicted load demand over a certain period of time. The objective is to operate an electric power system most economically while the system is operating within its security limits. DED is a complicated problem with nonlinear constraints.

It is a dynamic optimization problem taking into account the constraints imposed on system operation by generator ramping rate limits. The DED is not only the most accurate formulation of the economic dispatch problem but also the most difficult to solve because of its large dimensionality (Attaviriyanupap et al., 2002).

Normally, the DED problem can be formulated as follows. The objective function is

$$f=\sum_{t=1}^{n}\sum_{i=1}^{N}F_i(P_{it})+\lambda_1\left(\sum_{t=1}^{n}\sum_{i=1}^{N}P_{it}-P_{Dt}\right)^2+\lambda_r\left(\sum_{t=1}^{n}\sum_{i=1}^{N}P_{it}-P_{rlim}\right)^2 \quad (20)$$

where λ_1 and λ_r are penalty parameters, n is the number of hours, N is the number of units. The penalty factors regulate the objective function such that the algorithm gives a higher cost value rather than directly judging the solutions as infeasible. The penalty term reflects the violation of the equality constraint and assigns a high cost penalty function. The P_{rlim} is defined by

$$P_{rlim}=\begin{cases}P_{i(t-1)}-DR_i, & \text{if } P_{it}<P_{i(t-1)}-DR_i\\ P_{i(t-1)}+UR_i, & \text{if } P_{it}>P_{i(t-1)}+UR_i\\ P_{it}, & \text{otherwise}\end{cases} \quad (21)$$

In this research we solve two instances, with N equal 5 and 10, respectively.

3.2.11. P11: static economic load dispatch (ELD)

The economic dispatch problem concerns the question of how to distribute a power load over the units that are in service so that the total fuel cost is at a minimum.

The objective function corresponding to the production cost can be approximated as a quadratic function of the active power outputs from the generating units (Panigrahi et al., 2008). It can be represented as follows:

$$F=\sum_{i=1}^{N_G}f_i(P_i) \quad (22)$$

where $f_i(P_i)=a_iP_i^2+b_iP_i+c_i,\quad i=1,2,\ldots,N_G$ is the expression for the cost function corresponding to the i-th generating unit and a_i, b_i and c_i are its cost coefficients. P_i is the real power output (MW) of the ith generator corresponding to time period t. N_G is the number of on-line generating units to be dispatched. The cost function for a unit with a valve point loading effect is calculated by using

$$f_i(P_i)=a_iP_i^2+b_iP_i+c_i+|e_i\ \sin(f_i(P_i^{min}-P_i))| \quad (23)$$

where e_i and f_i are the cost coefficients corresponding to the valve point loading effect.

This problem is subjected to different constraints depending upon assumptions and practical implications, as shown below

(1) Power balance constraints or demand constraints: this constraint is based on the principle of equilibrium between total system generation ($\sum_{i=1}^{N_G}(P_i)$) and total system loads (P_D) and losses (P_L). That is,

$$\sum_{i=1}^{N_G}(P_i)=P_D+P_L \quad (24)$$

where P_L is obtained using B-coefficients, given by

$$P_L=\sum_{i=1}^{N_G}\sum_{j=1}^{N_G}P_iB_{ij}P_j+\sum_{i=1}^{N_G}B_0P_i+B_{00} \quad (25)$$

(2) The generator constraints: the output power of each generating unit has a lower and upper bound so that it lies in between these bounds. This constraint is represented by a pair of inequality constraints as follows:

$$P_i^{min}\le P_i\le P_i^{max} \quad (26)$$

where P_i^{min} and P_i^{max} are the lower and upper bounds for power outputs of the ith generating unit.

(3) The ramp rate limits: this assumption concerns the effect whereby adjustments of the power output are instantaneous. However, under practical circumstances, ramp rate limits restrict the operating range of all the on-line units for adjusting the generator operation between two operating periods. The generation may increase or decrease with corresponding upper and downward ramp rate limits. So units are constrained due to these ramp rate limits, such as: $P_i-P_i^{t-1}\le UR_i$, if power generation increases, and $P_i^{t-1}-P_i\le DR_i$ if power generation decreases, where P_i^{t-1} is the power generation of unit i in the previous hour and UR_i and DR_i are the upper and lower ramp rate limits respectively. The inclusion of ramp rate limits modifies the generator operation constraints as follows:

$$max(P_i^{min},\ UR_i-P_i)\ \le P_i\le min(P_i^{max},P_i^{t-1}-DR_i) \quad (27)$$

(4) Prohibited operating zone: the generating units may have certain ranges where operation is restricted on the grounds of physical limitations of machine components or instability, e.g. due to steam valve or vibration in shaft bearings. Consequently, discontinuities are produced in cost curves corresponding to the prohibited operating zones. Hence, there is a desire to avoid operation in these zones in order to economize production. Symbolically, for a generating unit i,

$$\underline{P}^{pz}\le P_i\le \overline{P}^{pz} \quad (28)$$

where $\overline{P}^{pz}$ and $\underline{P}^{pz}$ are the lower and upper limits of a given prohibited zone for unit i.

In this research we consider 5 cases. Case 1: 6-generating units of an IEEE 30-bus system. Case 2: 13-generating units. Case 3: 15-generating units. Case 4: 40-generating units, and case 5: 140-generating units.

3.2.12. P12: short-term hydrothermal scheduling problem

Short-term hydrothermal scheduling is one of the constrained power system optimization problems, and has complex and nonlinear characteristics with various types of constraints and nonlinear relationships of problem variables that make the problem of finding the global optimum difficult using any standard optimization methods and programming techniques (Hota et al., 2009). The main requirement of this problem is to generate the optimal amount of generated power for the hydro and thermal units in the system, so that they meet the load demands in the scheduling horizon of 1 day or a few days, and satisfy different constraints on the hydraulic and power system network.

The objective function (F) is to minimize the fuel cost of running the thermal system to meet the demand in scheduling horizon. The objective function is expressed mathematically as

$$\min F=\sum_{j=1}^{m}f_j(P_{Tj}) \quad (29)$$

where f_j is the cost function corresponding to the equivalent thermal unit's power generation P_{Tj} at the i-th interval. M is the total number of intervals considered for the short term schedule.

The cost function f_j can be written as

$$f_j(P_{Tj}) = a_j P_{Tj}^2 + b_j P_{Tj} + c_j + |e_j \ \sin(f_i(P_{Tj}^{\min} - P_{Tj}))| \tag{30}$$

Subject to the following various system constraints

(1) Demand constraints

$$P_{Tj} + \sum_{i=1}^{n} P_{Hij} = P_{Dj} + P_{Lossj}, for\ j = 1, 2, \ldots, m \tag{31}$$

where P_{Hij} represents the power generated by the i-th hydro unit at the j-th interval, P_{Dj} and P_{Lossj} are the power demand and power loss at the j-th interval, respectively.

(2) Thermal generator constraints

$$P_T^{min} \le P_{Tj} \le P_T^{max} \tag{32}$$

(3) Hydro generator constraint

$$P_{Hi}^{min} \le P_{Hij} \le P_{Hi}^{max} \tag{33}$$

(4) Reservoir capacity constraint

$$V_i^{min} \le V_{ij} \le V_i^{max} \tag{34}$$

(5) The water discharge constraint

$$q_i^{min} \le q_{ij} \le q_i^{max} \tag{35}$$

(6) Hydraulic continuity constraint

$$V_{i(j+1)} = V_{ij} + \sum_{u=1}^{R_u} [q_{u(j-\tau)} + S_{u(j-\tau)}] - q_{i(j+1)} - S_{i(j+1)} + r_{i(j+1)}, \quad for\ j = 1, 2, \ \ldots, \ m \tag{36}$$

where τ is the water delay time between reservoir i and its upstream point u at interval j and R_u is the set of upstream units directly above hydroplant i.

(7) The hydro power generation

$$P_{Hij} = c_{1i} V_{ij}^2 + c_{2i} q_{ij}^2 + c_{3i}(V_{ij} q_{ij}) + c_{4i}(V_{ij}) + c_{5i}(q_{ij}) + c_{6i} \tag{37}$$

where c_{1i}, c_{2i}, c_{3i}, c_{4i}, c_{5i}, and c_{6i} are the coefficients.

In this research, three instances are considered: instance 1: quadratic cost curve without prohibited discharge zones; instance 2: quadratic cost with prohibited discharge zones; and instance 3: systems with the valve point loading effect and with prohibited operating hydro discharge zones.

3.2.13. P13 Messenger-full: spacecraft trajectory optimization problem

This trajectory optimization problem represents a rendezvous mission to Mercury modeled as an MGA-1DSM problem. It includes the final resonant fly-bys, and seeks to minimize the on-board propellant consumption. Messenger is on its way to Mercury. It will become the first spacecraft to ever orbit around the planet. Its path in the solar system to reach its final destination included a long planetary tour: Earth – Earth – Venus – Venus – Mercury – Mercury – Mercury – Mercury (Biscani et al., 2010). The Messenger-full problem is complex with an unknown optimal solution. For more details and code, the reader is referred to http://www.esa.int/gsp/ACT/inf/op/globopt/MessengerFull.html.

3.2.14. P14 Cassini 2: spacecraft trajectory optimization problem

This problem represents the interplanetary trajectory of the spacecraft Cassini. The objective function represents the sum of all DV required to reach Saturn using an Earth – Venus, Venus – Earth, Jupiter – Saturn fly-by sequence with deep space manoeuvres. In this model for the Cassini trajectory: deep space manoeuvres are allowed between each one of the planets. For more details the readers are referred to: http://www.esa.int/gsp/ACT/inf/op/globopt/edvdvdedjds.htm. The Cassini 2 problem is complex with an unknown optimal solution.

4. Experimental results and analysis

In this section, we discuss the computational results, and analyze the performance of our algorithm.

The performance of GA-MPC has been compared with the state of the art algorithms. The algorithm has been coded using visual Matlab, and has been run on a PC with a 3 Core 2 Duo processor, 3.5 G RAM, and windows XP. The detailed results are shown in Table A1.

4.1. GA-MPC Results based on the CEC2006 Problems

We set the parameters: $PS=90$, $\beta=N$ (0.7, 0.1) for all test problems, and $cr=100\%$. If the offspring are generated using Eq. (2) the diversity probability (p) is set to 0.05, and if the offspring are generated using Eq. (3) and $p=0.1$ (the justification is shown in a later section). The rational of using different values for p is provided later in this section. The tournament size is selected randomly as 2 or 3. For the number of individuals that are selected to fill the archive pool $m=50\%$ of PS, 25 independent runs were performed for each test problem, the stopping criterion was to run for up to 240,000 fitness evaluations (FEs), and the equality constraints are handled as: $|h_c(\vec{x})| - \varepsilon \le 0, for\ c = 1, \ldots, E$, where E is the number of equality constraints and $\varepsilon=0.0001$.

Firstly, we wish to compare the proposed algorithm with other GAs. To do this, we selected SBX and PCX, which both of them use a non-uniform mutation (NU) with probability 0.1. The SBX and PCX parameters are set to: PS = 100, $cr=0.95$, tournament size=2 or 3 randomly, index parameter $\eta=3$, $\delta_\xi = \delta_\eta = 0.01$, for the non-uniform mutation, the mutation probability is 0.1 and, $b=5$. The detailed results (best, median, average, worst, and standard deviation ($St.\ d$)) are shown in Appendix A.

To start with, all algorithms could not solve g20 and 22, and thus these two problems are discarded from our analysis. Based on the obtained results, we found that the feasibility ratio (number of feasible runs/25), for GA-MPC was 100% while SBX-NU was 89% and PCX-NU was 83%. In regard to the best values, we found that GA-MPC was better than both SBX-NU and PCX-NU for 16 and 15 test problems, respectively, while GA-MPC was able to obtain the same results for 6 and 7 test problems, respectively. In regard to the average results, GA-MPC was superior to both SBX-NU and PCX-NU for 17 test problems each, while GA-MPC was able to obtain the same results for 5 test problems each – these 5 problems are quite simple (Fig. 1).

As an example, Fig. 2 shows the convergence patterns for GA-MPC, SBX-NU and PCX-NU. From Fig. 2, although SBX-NU is able to converge faster in the early few generations, GA-MPC is able to converge to the optimal solution, while the other two algorithms cannot.

It is also possible, however, to study the difference between any two stochastic algorithms in a more meaningful way. To this end, we have performed statistical significance testing. We have chosen a non-parametric test, Wilcoxon Signed Rank Test (Corder and Foreman, 2009) that allows us to judge the difference between paired scores when it cannot make the assumptions required by the paired-samples t test, such as that the population should be normally distributed. The results regarding the best and average fitness values are presented in Table 3. As a null hypothesis, it is

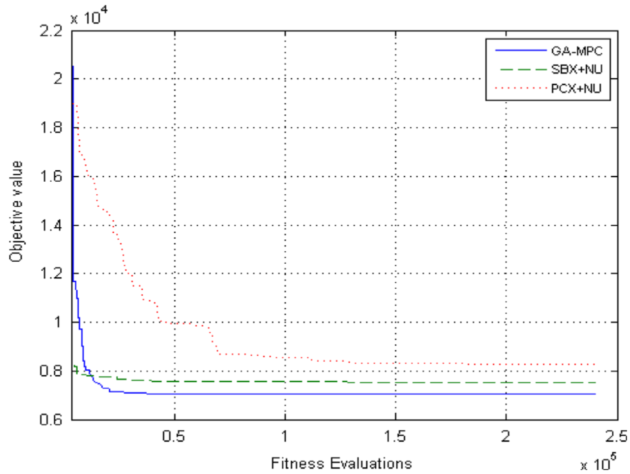

Fig. 1. Convergence pattern for GA-MPC, SBX-NU and PCX-NU for g10.

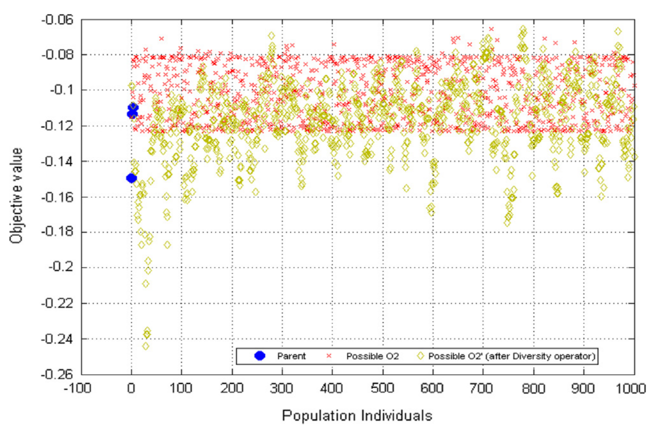

Fig. 2. 1000 possible offspring for the second parent, for g02.

Table 3
Wilcoxon sign rank test results for GA-MPC against SBX-NUPCX-NU, APF-GA, MDE and ECHTEP-2.

Algorithms	Criteria	Decision
GA-MPC – to – SBX-NU	Best fitness	+
	Average fitness	+
GA-MPC – to – PCX-NU	Best fitness	+
	Average fitness	+
GA-MPC – to – APF-GA	Best fitness	+
	Average fitness	+
GA-MPC – to – MDE	Best fitness	≈
	Average fitness	≈
GA-MPC – to – ECHTEP-2	Best fitness	≈
	Average fitness	+

assumed that there is no significant difference between the best and/or mean values of two samples. Whereas the alternative hypothesis is that there is a significant difference in the best and/or mean fitness values of the two samples, with 5% significance level. Based on the test results/rankings, we assign one of three signs (+, −, and ≈) for the comparison of any two algorithms (shown in the last column), where the "+" sign means the first algorithm is significantly better than the second, the "−"sign means that the first algorithm is significantly worse, and the "≈" sign means that there is no significant difference between the two algorithms. From Table 3, GA-MPC is clearly superior to both SBX-NU and PCX-NU in regard to the best and the average solutions obtained.

To continue this analysis, we have compared GA-MPC with the state-of-the-art algorithms such as APF-GA (adaptive penalty formulation with GA by (Tessema and Yen, 2009)), MDE (modified differntial evolution by (Mezura-Montes et al., 2006)) and ECHT-EP2 (ensemble of constraint handling techniques based on an EP algorithm by (Mallipeddi and Suganthan, 2010)). We must mention here that GA-MPC and ECHT-EP2 use 240,000 FEs, while both APF-GA and MDE use 500,000 function evaluations (FEs).

Based on the best results, GA-MPC, APF-GA, and MDE and ECHT-EP2 reached the optimal solutions for 22, 17, 20 and 19 problems, respectively. As of the average results, GA-MPC was better than APF, MDE and ECHT-EP2 for 7, 3 and 7 test problems, respectively, while GA-MPC was inferior to MDE and APF-GA for only one test problem each.

Based on the statistical test, as shown in Table 3, GA-MPC was better than APF-GA in regard to the best and average results, GA-MPC was superior to ECHTEP-2 in regard to the average results, while there was no significant difference between GA-MPC and MDE.

4.1.1. *Further performance analysis*

As of the results discussed in the previous section, GA-MPC performs very well. In this section, we have designed three set of experiments and analyzed their results to support the reasons for its superior performance.

4.2. *Experiment 1*

In the first generation ($t=1$), from only 3 parents, 1000 offspring have been generated using each Eqs. (2), (3) and (4). Here, Eq. (2) is based on parent 1 (P1), Eq. (3) is based on parent 2 (P2), and so on. For each equation, the percentage of individuals that are better than their parent has been calculated (i.e. how many offspring generated by Eq. (2), were better than P1, based on section 3.4, and so on). The same calculations has been done after applying the diversity operator. From the results, we found the following:

(1) For the test problems with non-separable objective functions (i.e. g02), 12% of the generated offspring, due to Eq. (2), are much better than P1. After the diversity operator (with $p=0.1$), the percentage was 31.1%. Based on Eq. (3), 0.35% of the generated offpsring were much better than P2, while after the diversity operator (with $p=0.1$), the percentage was 36.9% (which is a 26.5% better result than P1). Based on Eq. (4), 34% of the generated offspring was much better than P3, while after the diversity operator (with $p=0.1$), the percentage was 85.4% (which is a 27.3% better result than P1). This is summarized in Table 4.
(2) For the test problems with separable objective functions (i.e. g10), using a diversity probablity equal to 0.1, the summary results are presented in Table 5.

Based on points *a* and *b*, we concluded that for the offspring that were generated by Eq. (2), it was not best to use a high

Table 4
Percentage of the offspring those are better than their parent. The results are for the test problems with non-separable objective functions.

Parent	Percentage of the offspring which are better than their parent	
	Before diversity (%)	After diversity (%)
P1	12	34
P2	0.35	36.9
P3	34	85.4

diversity probability ($p=0.1$) for test problems with a separable objective function, while $p=0.1$ was good for the test problems with non-separable objective functions. For those offspring that were generated by Eqs. (3) and (4), $p=0.1$ was a good choice.

From point *a*, applying diversity operator on the offspring that were generated by Eq. (3), led to a high improvement of the results. The reason for this is that visiting different search areas may lead to obtain better results after applying the diversity operator, in which case getting information from good results and mixing this information with that information of bad solutions may lead to better results. Fig. 2 shows such performance.

Table 5
Percentage of the offspring which are better than their parent. The results are for the test problems with separable objective functions.

Parent	Percentage of the offspring which are better than their parent	
	Before diversity (%)	After diversity (%)
P1	43.3	25.3
P2	67.4	80.2
P3	84.9	93.9

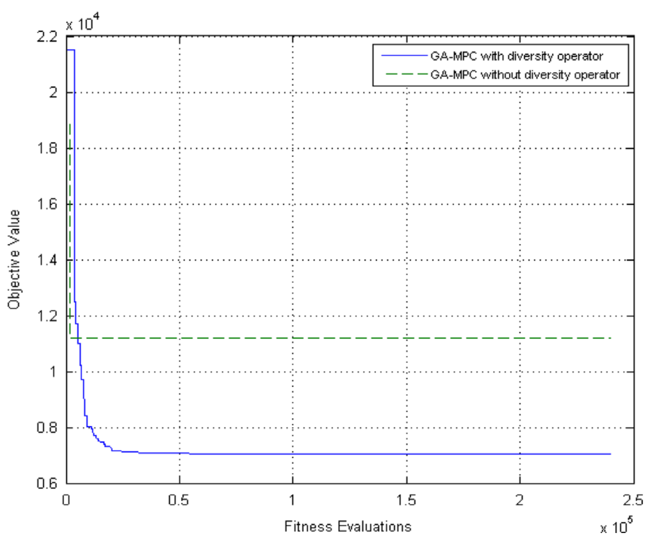

Fig. 3. Effect of the diversity operator on the quality of results for g10.

4.3. Experiment 2

In this experiment, we study the effect of the diversity operator on the performance of GA-MPC. Although without using the diversity operator the algorithm converged fast in the early generations, this convergence was to local optima, as it is shown in Fig. 3.

It was also important to analyze the effect of the diversity probability. We ran GA-MPC with different values: (a) $p=0.05$ for the offspring that were generated by Eq. (2), while $p=0.1$ was used for the offspring that were generated by Eqs. (3) and (4) (the version used), (b) $p=0.1$, (c) $p=0.2$, (d) $p=0.3$, (e) $p=0.5$ and (f) $p=1$. We found that option (a) was the best, see Fig. 4. Note that although the use of a high value of p for those test problems with a non-separable objective function might lead to a faster convergence in the beginning of the evolution process, it then becomes stuck in local optima later on.

Consequently, it is important to measure the benefit of how the diversity operator encourages a reasonable diversity. To do this, we calculated the diversity in each generation of GA-MPC with and without the diversity operator mechanism. Here we cacluated the diversity (div) as follows:

$$div = \sqrt{\frac{1}{PS-1}\sum_{i=1}^{PS}(\bar{f}-f_i)} \tag{38}$$

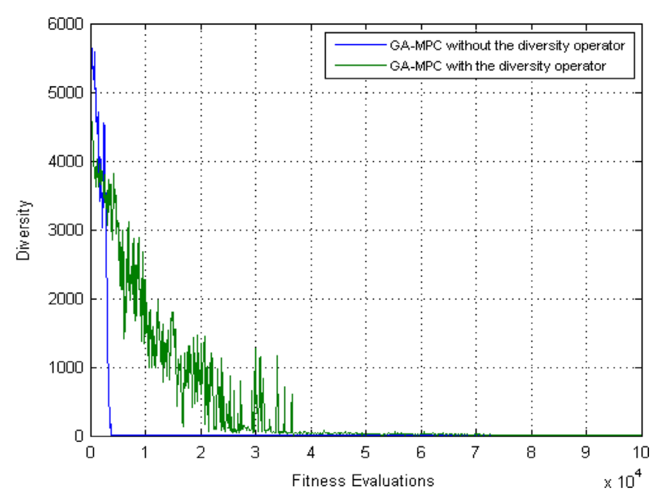

Fig. 5. Diversity value in each generation, for g10, for both GA-MPC with and without the diversity opeator.

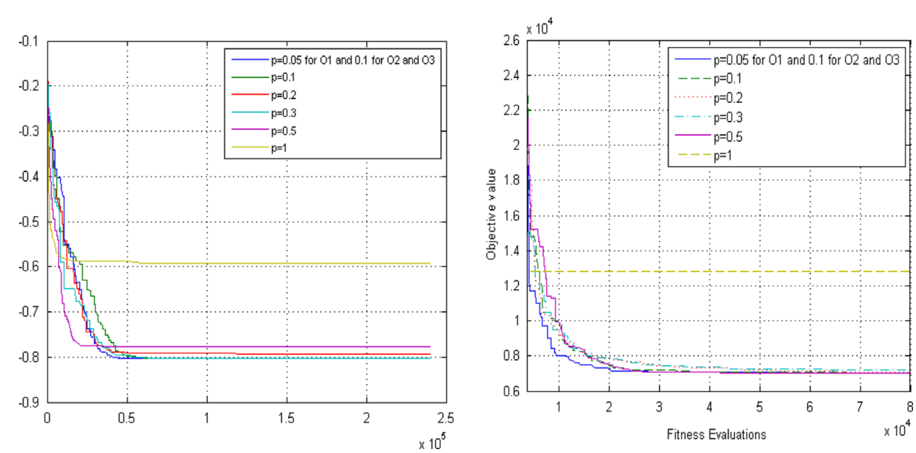

Fig. 4. Effect of changing the diversity probablity on the quality of results for g02 on the left and g10 on the right.

where f_i is the fitness value of individual i, $\bar{f}$ is the mean fitness value, i.e. $\bar{f} = \sum_{i=1}^{PS}(f_i)/PS$. We found that appling the diversity mechanism leads to better diversity, as seen in Fig. 5.

4.4. Experiment 3

In this experiment, we calculated the distance between the individuals and the known best solution (distance1) in the initial population, after 200 generations (distance2) and after 400 generations (distance3). We found that distance2 was 96% close to the optimal solution, while distance3 was more than 99.6% nearer to the golobal optimum. This can show how fast the GA-MPC is. Fig. 6 shows these distances for both g02 and g10. The reasons for such fast convergence are: (1) in Eqs. (2) and (4), the solutions are moved toward better results, (2) instead of using the global best in each generation to guide the search, different bests are used (i.e. for each 3 generated offspring, we might use a different P1 and P3, and hence the solutions might move toward different regions), (3) visiting diverse areas, due to Eq. (3), could help to obtain good solutions quickly after mixing with good solutions.

4.4.1. Solving real world optimization problems

The detailed results (best, median, average, worst, and standard deviation (*St. d*)) are shown in Appendix B. We must mention here that for the problems that have discrete variables, those discrete variables were handled as continuous ones, and then rounded off.

The performance of GA-MPC was compared against two other DE algorithms, Adaptive Differential Evolution Algorithm (ADE) (Asafuddoula et al., 2011) and another algorithm that uses different DE strategies, ensemble DE algorithm (EPSDE) (Mallipeddi and Suganthan, 2011). All algorithms have the same stopping criteria, 150,000 fitness function evaluations. The detailed results are shown in Table B1.

Based on the best fitness values obtained, GA-MPC was able to obtain better results than IDE and EPSDE for 14 and 17 problems, respectively. GA-MPC was able to obtain the same results as IDE and EPSDE for 7 and 2 problems, respectively, while GA-MPC was inferior to both IDE and EPSDE for 1 problem each. In regard to the average fitness values, GA-MPC was superior to IDE and EPSDE for 19 problems each. GA-MPC was able to obtain the same mean results as both IDE and EPSDE for only 1 problem each, while GA-MPC was inferior to IDE and EPSDE for 2 problems each.

Finally, we have statistically analyzed the performance of GA-MPC against those two algorithms, as shown in Table 6. From that table, it is clear that GA-MPC is superior to both IDE and EPSDE in regard to both the best and average fitness results.

5. Conclusions

During the last few decades, many genetic algorithm variants were introduced. However, GAs was most probably inferior to other EAs, such as DE. In this paper, we showed that the efficiency of GA could be improved by adopting a new crossover operator with a diversity operator.

The aim of the proposed crossover was to maintain both the intensification and exploration schemes. The intensification scheme can be seen from the behavior of the directed vector difference, as it is oriented to the best solutions (Eqs. 1 and 3). On the contrary, the vector difference in Eq. (2) is directed to a worse candidate. This is to maintain the diversity. The reason for using three parents, instead of one or two, is to force the algorithm to visit different sub-regions and to hence explore the landscape (increasing, not necessarily decreasing, the search space). It is worth mentioning here that the directed vector difference feature resembles the concept of gradient.

In the proposed algorithm, an initial population was generated randomly and the best m individuals were stored in an archive pool. A tournament selection with size 2 or 3 was initiated to select the best points to act as parents. Then the proposed crossover and its diversity operator were used to generate new offspring. In the proposed crossover, each three consecutive parents were able to generate three offspring. These offspring were then diversified with archive pool individuals. The generated offspring were then merged with the archive pool individual, and the best *PS* individuals were used as a new population for the next generation.

The proposed algorithm was analyzed by solving constrained optimization problems. Its performance was superior to two other GAs, as well as state-of-the-art- algorithms. Furthermore, the proposed algorithm was assessed by solving several real world test problems gathered from different fields. In doing so, the algorithm showed a superior performance in comparison with other algorithm algorithms.

For future work, we intend to conduct further theoretical analysis of our proposed algorithm.

Table 6
The Wilcoxon sign rank test results for GA-MPC against IDE and EPSDE.

Algorithms	Criteria	Decision
GA-MPC – to – IDE	Best fitness	+
	Average fitness	+
GA-MPC – to – EPSDE	Best fitness	+
	Average fitness	+

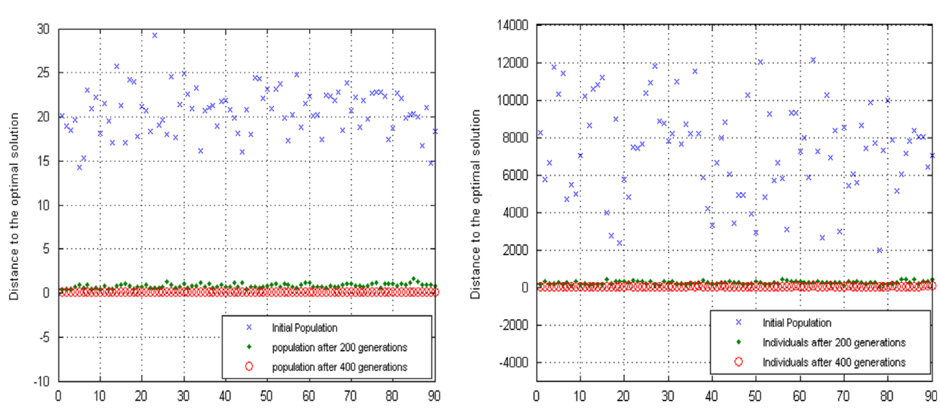

Fig. 6. Euclidean distance to the optimal solution for g02 on the left and g10 on the right.

Appendix A

Table A1
Function values achieved by GA-MPC, SBX-NU, PCX-NU,APF-GA, MDE AND ECHT-EP2 FOR 24 CEC2006 TEST PROBLE.

P	Criteria	GA-MPC	SBX-NU	PCX-NU	APF-GA	MDE	ECHT-EP2
	FEs	240,000	240,000	240,000	500,000	500,000	240,000
g01	best	**−15.0000**	**−15.000000**	−14.993200	**−15.0000**	**−15.0000**	**−15.0000**
	Average	**−15.0000**	**−15.000000**	−14.657900	**−15.0000**	**−15.0000**	**−15.0000**
	St. d	**0.00E+00**	1.10000E-05	7.49397E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**
g02	best	**−0.8036191**	−0.803513	−0.803611	−0.803601	**−0.8036191**	**−0.8036191**
	Average	−0.802921	−0.784869	−0.776714	−0.803518	−0.78616	−0.7998220
	St. d	2.4150E-03	1.27050E-02	2.11340E-02	1.00E-04	1.26E-02	6.29E-03
g03	best	**−1.0005**	**−1.0005**	**−1.0005**	**−1.001**	**−1.0005**	**−1.0005**
	Average	**−1.0005**	**−1.0005**	−1.0003	**−1.001**	**−1.0005**	**−1.0005**
	St. d	**0.00E+00**	1.14378E-05	1.57000E-04	**0.00E+00**	**0.00E+00**	**0.00E+00**
g04	best	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**
	Average	**−30665.539**	−30665.53130	−30657.28650	**−30665.539**	**−30665.539**	**−30665.539**
	St. d	**0.00E+00**	1.28820E-02	4.12575E+01	1.00E-04	**0.00E+00**	**0.00E+00**
g05	best	**5126.497**	5127.226	–	**5126.497**	**5126.497**	**5126.497**
	Average	**5126.497**	5256.734	–	5127.5423	**5126.497**	**5126.497**
	St. d	**0.00E+00**	2.09863E+02	–	1.4324E+00	**0.00E+00**	**0.00E+00**
g06	best	**−6961.814**	−6961.81378	−6961.8132	**−6961.814**	**−6961.814**	**−6961.814**
	Average	**−6961.814**	−6961.8133	−6961.80789	**−6961.814**	**−6961.814**	**−6961.814**
	St. d	**0.00E+00**	4.90000E-04	3.51000E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**
g07	best	**24.3062**	24.49210	24.37413	**24.3062**	**24.3062**	**24.3062**
	Average	**24.3062**	25.73469	24.78642	**24.3062**	**24.3062**	24.3063
	St. d	**0.00E+00**	7.86300E-01	2.98340E-01	**0.00E+00**	**0.00E+00**	3.19E-05
g08	best	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**
	Average	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**
	St. d	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.0E+00**
g09	best	**680.630**	680.63460	680.63226	**680.630**	**680.630**	**680.630**
	Average	**680.630**	680.67630	680.65105	**680.630**	**680.630**	**680.630**
	St. d	**0.00E+00**	2.92600E-02	1.01400E-02	**0.00E+00**	**0.00E+00**	2.61E-08
g10	best	**7049.24802**	7054.37289	7054.37289	**7049.24802**	**7049.24802**	7049.2483
	Average	**7049.24802**	7369.17372	7369.17372	7077.6821	**7049.24802**	7049.2490
	St. d	**0.00E+00**	1.84426E+02	1.84426E+02	5.1240E+01	**0.00E+00**	6.60E−04
g11	best	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**0.7499**
	Average	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**0.7499**
	St. d	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
g12	best	**−1.0000**	**−1.0000**	**−1.0000**	**−1.0000**	**−1.0000**	**−1.0000**
	Average	**−1.0000**	**−1.0000**	**−1.0000**	**−1.0000**	**−1.0000**	**−1.0000**
	St. d	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
g13	best	**0.053942**	0.06203	0.05404	**0.053942**	**0.053942**	**0.053942**
	Average	**0.053942**	0.61255	0.05730	**0.053942**	**0.053942**	**0.053942**
	St. d	**0.00E+00**	1.74509E+00	2.43000E-03	**0.00E+00**	**0.00E+00**	1.00E−12
g14	best	**−47.764888**	−47.14100	−46.80460	−47.76479	−47.764887	**−47.7649**
	Average	**−47.764888**	−44.44460	−44.82550	−47.76479	−47.764874	−47.7648
	St. d	**0.00E+00**	1.20213E+00	1.00682E+00	1.00E-04	1.400E-05	2.72E−05
g15	best	**961.71502**	961.71503	961.71512	**961.71502**	**961.71502**	**961.71502**
	Average	**961.71502**	961.71997	961.78555	**961.71502**	**961.71502**	**961.71502**
	St. d	**0.00E+00**	6.63000E-03	1.04580E-01	**0.00E+00**	**0.00E+00**	2.01E-13
g16	best	**−1.905155**	−1.90515	**−1.905155**	**−1.905155**	**−1.905155**	**−1.905155**
	Average	**−1.905155**	−1.90489	**−1.905155**	**−1.905155**	**−1.905155**	**−1.905155**
	St. d	**0.00E+00**	2.67000E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.12E-10
g17	best	**8853.5397**	8872.8656	8857.4660	8853.5398	**8853.5397**	**8853.5397**
	Average	**8853.5397**	8947.0512	8933.5017	8888.4876	**8853.5397**	**8853.5397**
	St. d	**0.00E+00**	5.20132E+01	1.854724E+01	29.0347	**0.00E+00**	2.13E−08
g18	best	**−0.866025**	−0.841788	−0.866018	**−0.866025**	**−0.866025**	**−0.866025**
	Average	**−0.866025**	−0.627814	−0.863268	−0.865925	**−0.866025**	**−0.866025**
	St. d	**0.00E+00**	8.69593E-02	3.63400E-03	0000	**0.00E+00**	1.00E-09
g19	best	**32.655593**	55.206828	1992.967419	**32.655593**	32.655693	32.6591
	Average	**32.655593**	87.577232	3894.752822	**32.655593**	33.34125	32.6623
	St. d	**0.00E+00**	1.62568E+01	1.17627E+03	**0.00E+00**	8.475E-01	3.4E−03
g21	best	**193.72451**	193.730749	–	196.63301	**193.72451**	**193.7246**
	Average	**193.72451**	310.419985	–	199.51581	**193.72451**	193.7438
	St. d	**0.00E+00**	2.01064E+02	–	2.3565E+00	**0.00E+00**	1.65E-02
g23	best	**−400.0551**	30.855109	–	−399.7624	**−400.0551**	−398.9731
	Average	−400.023589	418.637784	–	−394.7627	**−400.0551**	−373.2178
	St. d	6.3463E-02	3.64226E+02	-	3.8656E+00	**0.00E+00**	3.37E+01
g24	best	**−5.508013**	−5.506830	**−5.508013**	**−5.508013**	**−5.508013**	**−5.508013**
	Average	**−5.508013**	−5.471240	**−5.508013**	**−5.508013**	**−5.508013**	**−5.508013**
	St. d	**0.00E+00**	2.87260E-02	1.0000E-06	**0.00E+00**	**0.00E+00**	1.8E-15

Appendix B

Table B1
The function values achieved by GA-MPC, ADE and EPSDE over 25 runs and 150,000 fitness function evaluations.

Problems	Algorithm	Best	Median	Mean	Worst	St. d.
P01	GA-MPC	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**	**0.000000E+00**
	ADE	**0.000000E+00**	**0.000000E+00**	3.852582E+00	1.702078E+01	5.690016E+00
	EPSDE	**0.000000E+00**	**0.000000E+00**	1.78E+00	1.17E+01	4.18E+00
P02	GA-MPC	**−2.842253E+01**	**−2.747974E+01**	**−2.770069E+01**	**−2.711301E+01**	**4.673052E-01**
	ADE	**−2.842253E+01**	−2.719010E+01	−2.678273E+01	−1.820789E+01	1.974757E+00
	EPSDE	−2.14E+01	−1.83E+01	−1.83E+01	−1.47E+01	1.65E+00
P03	GA-MPC	**1.151489E-05**	**1.151489E-05**	**1.151489E-05**	**1.151489E-05**	**0.000000E+00**
	ADE	**1.151489E-05**	**1.151489E-05**	**1.151489E-05**	**1.151489E-05**	3.804327E-19
	EPSDE	**1.151489E-05**	**1.151489E-05**	**1.151489E-05**	**1.151489E-05**	1.52E-19
P04	GA-MPC	**1.3770762E+01**	**1.3770762E+01**	**1.3815430E+01**	**1.4329113E+01**	**1.5460045E-01**
	ADE	1.432911E+01	2.081992E+01	1.831217E+01	2.153960E+01	2.915478E+00
	EPSDE	1.38E+01	1.43E+01	1.67E+01	2.11E+01	3.26E+00
P05.1	GA-MPC	**−3.684537E+01**	**−3.500981E+01**	**−3.503883E+01**	**−3.410760E+01**	**8.329248E-01**
	ADE	**−3.684537E+01**	−3.410766E+01	−3.381816E+01	−3.148397E+01	1.434399E+00
	EPSDE	−3.20E+01	−2.91E+01	−2.90E+01	−2.48E+01	1.84E+00
P05.2	GA-MPC	**−2.916612E+01**	**−2.742977E+01**	**−2.748811E+01**	**−2.125851E+01**	**1.782137E+00**
	ADE	**−2.916612E+01**	**−2.742977E+01**	−2.583055E+01	−1.951175E+01	2.991160E+00
	EPSDE	−1.99E+01	−1.80E+01	−1.70E+01	−1.14E+01	2.68E+00
P06	GA-MPC	**5.000000E-01**	7.580619E-01	7.484090E-01	9.334272E-01	1.249139E-01
	ADE	**5.000000E-01**	**5.000000E-01**	**5.000000E-01**	**5.000000E-01**	**0.000000E+00**
	EPSDE	1.28E+00	1.42E+00	1.42E+00	1.56E+00	7.38E-02
P07	GA-MPC	**2.200000E+02**	**2.200000E+02**	**2.200000E+02**	**2.200000E+02**	**0.000000E+00**
	ADE	**2.200000E+02**	**2.200000E+02**	**2.200000E+02**	**2.200000E+02**	**0.000000E+00**
	EPSDE	**2.200000E+02**	**2.200000E+02**	**2.200000E+02**	**2.200000E+02**	**0.000000E+00**
P08	GA-MPC	7.999523E+01	2.026846E+02	2.131053E+02	3.199780E+02	6.723456E+01
	ADE	**7.585491E+00**	**4.730413E+01**	**1.930413E+01**	**1.159561E+02**	**5.690016E+00**
	EPSDE	7.845E+02	2.368E+03	2.529E+03	5.725E+03	1.328E+03
P09	GA-MPC	**−2.1842539E+01**	**−2.1644450E+01**	**−2.1702249E+01**	**−2.1475684E+01**	**1.1634659E-01**
	ADE	**−2.180845E+01**	−2.148024E+01	−2.095834E+0	−1.497656E+01	1.895431E+00
	EPSDE	−2.18E+01	−1.38E+01	−1.56E+01	−1.10E+01	3.79E+00
P10.1	GA-MPC	**4.730791E+04**	**4.939572E+04**	**4.944331E+04**	**5.127534E+04**	**1.165556E+03**
	ADE	5.017060E+04	5.282237E+04	5.418152E+04	7.127143E+04	4.866964E+03
	EPSDE	5.11E+04	5.21E+04	5.22E+04	5.35E+04	7.24E+02
P10.2	GA-MPC	**1.054575E+06**	**1.062514E+06**	**1.061684E+06**	**1.064775E+06**	**2.589416E+03**
	ADE	1.078526E+06	1.086166E+06	1.086684E+06	1.094758E+06	4.455655E+03
	EPSDE	1.06E+06	1.07E+06	1.07E+06	1.08E+06	2.13E+03
P11.1	GA-MPC	**1.544419E+04**	**1.544419E+04**	**1.544419E+04**	**1.544419E+04**	**1.751115E-07**
	ADE	1.544538E+04	1.547957E+04	1.548207E+04	1.556846E+04	3.047366E+01
	EPSDE	**1.54E+04**	**1.54E+04**	1.55E+04	1.55E+04	1.55E+01
P11.2	GA-MPC	1.802651E+04	1.811060E+04	1.812838E+04	1.820303E+04	4.871860E+01
	ADE	1.822472E+04	1.856815E+04	1.855026E+04	1.878500E+04	1.415147E+02
	EPSDE	1.81E+04	1.81E+04	1.81E+04	1.82E+04	4.39E+01
P11.3	GA-MPC	3.269824E+04	**3.270866E+04**	**3.270875E+04**	**3.272170E+04**	6.959762E+00
	ADE	3.274420E+04	3.285861E+04	3.285940E+04	3.305473E+04	6.841365E+01
	EPSDE	**3.26E+04**	3.27E+04	3.27E+04	3.29E+04	3.59E+01
P11.4	GA-MPC	**1.2169315E+05**	**1.2188634E+05**	**1.2190596E+05**	**1.2223143E+05**	**1.5532313E+02**
	ADE	1.240690E+05	1.252703E+05	1.255323E+05	1.274304E+05	8.362557E+02
	EPSDE	1.28E+05	1.30E+05	1.31E+05	1.38E+05	2.47E+03
P11.5	GA-MPC	**1.6947982E+06**	**1.7770371E+06**	**1.7813343E+06**	**1.9625046E+06**	**6.8395166E+04**
	ADE	1.890671E+06	1.924213E+06	1.925098E+06	1.982961E+06	2.342904E+04
	EPSDE	1.91E+06	1.93E+06	1.92E+06	1.95E+06	1.18E+04
P12.1	GA-MPC	**9.238452E+05**	**9.251644E+05**	**9.249974E+05**	**9.271881E+05**	**8.141665E+02**
	ADE	9.248215E+05	9.299678E+05	9.306026E+05	9.428696E+05	3.502770E+03
	EPSDE	9.39E+05	9.43E+05	9.43E+05	9.48E+05	2.63E+03
P12.2	GA-MPC	**9.270085E+05**	**9.306422E+05**	**9.309053E+05**	**9.344096E+05**	**1.986568E+03**
	ADE	9.283227E+05	9.346668E+05	9.929643E+05	1.784064E+06	1.964245E+05
	EPSDE	9.40E+05	9.77E+05	9.90E+05	1.07E+06	4.14E+04
P12.3	GA-MPC	**9.2369711E+05**	**9.2389917E+05**	**9.2392531E+05**	**9.2415670E+05**	**1.2880822E+02**
	ADE	9.271454E+05	9.307335E+05	9.305288E+05	9.397024E+05	3.063813E+03
	EPSDE	9.39E+05	9.44E+05	9.43E+05	9.48E+05	2.63E+03
P13	GA-MPC	**7.0955595E+00**	**1.3725975E+01**	**1.2818165E+01**	**1.6924893E+01**	**3.2413428E+00**
	ADE	1.239837E+01	1.388321E+01	2.119566E+01	2.236330E+01	5.906079E+00
	EPSDE	1.66E+01	1.85E+01	1.88E+01	2.05E+01	1.67E+00
P14	GA-MPC	**8.398688E+00**	**8.620312E+00**	**9.359342E+00**	**1.081018E+01**	**9.454327E-01**
	ADE	8.621241E+00	1.279869E+01	1.253713E+01	1.640074E+01	2.466385E+00
	EPSDE	8.78E+00	1.44E+01	1.39E+01	2.04E+01	4.08E+00

References

Ali, M.M., Storey, C., Törn, A., 1997. Application of stochastic global optimization algorithms to practical problems. Journal of Optimization Theory and Applications 95, 545–563.

Ali, M.M., Törn, A., 2000. Optimization of carbon and silicon cluster geometry for Tersoff potential using differential evolution. In: Pardalos, A.F.A.M. (Ed.), Optimization in Computational Chemistry and Molecular Biology: Local and Global Approaches. Kluwer Academic Publisher, pp. 287–300.

Asafuddoula, M., Ray, T., Sarker, R., 2011. An adaptive differential evolution algorithm and its performance on real world optimization problems. IEEE Congress on Evolutionary Computation, 1057–1062.

Attaviriyanupap, P., Kita, H., Tanaka, E., Hasegawa, J., 2002. A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function. IEEE Power Engineering Review 22, 77.

Biscani, F., Izzo, D., Yam, C., 2010. A global optimisation toolbox for massively parallel engineering optimisation. In: Proceedings of the International Conference on Astrodynamics Tools and Techniques (ICATT).

Christie, R.D., Wollenberg, B.F., Wangensteen, I., 2000. Transmission management in the deregulated environment. Proceedings of the IEEE 88, 170–195.

Corder, G.W., Foreman, D.I., 2009. Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach. John Wiley, Hoboken, NJ.

de J Silva, I., Rider, M.J., Romero, R., Garcia, A.V., Murari, C.A., 2005. Transmission network expansion planning with security constraints. IEE Proceedings: Generation, Transmission and Distribution 152, 828–836.

Deb, K., Anand, A., Joshi, D., 2002. A computationally efficient evolutionary algorithm for real-parameter optimization. Evolutionary Computation 10, 371–395.

Dukic, M.L., Dobrosavljevic, Z.S., 1990. A method of a spread-spectrum radar polyphase code design. IEEE Journal on Selected Areas in Communications 8, 743–749.

Elfeky, E., Sarker, R., Essam, D., 2008. Analyzing the simple ranking and selection process for constrained evolutionary optimization. Computer Science and Technology 23, 19–34.

Elsayed, S.M., Sarker, R.A., Essam, D.L., 2011a. GA with a new multi-parent crossover for constrained optimization. IEEE Congress on Evolutionary Computation, 857–864.

Elsayed, S.M., Sarker, R.A., Essam, D.L., 2011b. Improved genetic algorithm for constrained optimization. In: Proceedings of the 2011 International Conference on Computer Engineering & Systems (ICCES), pp. 111–115.

Esposito, W.R., Floudas, C.A., 2000. Deterministic global optimization in nonlinear optimal control problems. Journal of Global Optimization 17, 97–126.

Galiana, F.D., Conejo, A.J., Gil, H.A., 2003. Transmission network cost allocation based on equivalent bilateral exchanges. IEEE Transactions on Power Systems 18, 1425–1431.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, MA.

Herrera, F., Lozano, M., 2000. Gradual distributed real-coded genetic algorithms. IEEE Transactions on Evolutionary Computation 4, 43–63.

Hoare, M.R., 2007. Structure and Dynamics of Simple Microclusters, Advances in Chemical Physics. John Wiley & Sons, Inc., pp. 49–135.

Hota, P.K., Barisal, A.K., Chakrabarti, R., 2009. An improved PSO technique for short-term optimal hydrothermal scheduling. Electric Power Systems Research 79, 1047–1053.

Liang, J.J., Runarsson, T.P., Mezura-Montes, E., Clerc, M., Suganthan, P.N., Coello Coello, C.A., Deb, K., 2005. Problem Definitions and Evaluation Criteria for the CEC 2006 Special Session on Constrained Real-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore.

Mallipeddi, R., Suganthan, P.N., 2010. Ensemble of constraint handling techniques. IEEE Transactions on Evolutionary Computation 14, 561–579.

Mallipeddi, R., Suganthan, P.N., 2011. Ensemble differential evolution algorithm for CEC2011 problems. IEEE Congress on Evolutionary Computation, 1557–1564.

Mandal, A., Das, A.K., Mukherjee, P., Das, S., Suganthan, P.N., 2011. Modified differential evolution with local search algorithm for real world optimization. IEEE Congress on Evolutionary Computation, 1565–1572.

Mezura-Montes, E., Velazquez-Reyes, J., Coello Coello, C.A., 2006. Modified differential evolution for constrained optimization. IEEE Congress on Evolutionary Computation, 25–32.

Moloi, N.P., Ali, M.M., 2005. An iterative global optimization algorithm for potential energy minimization. Computational Optimization and Applications 30, 119–132.

Ono, I., Kita, H., Kobayashi, S., 2003. A Real-Coded Genetic Algorithm Using the Unimodal Normal Distribution Crossover, Advances in Evolutionary Computing. Springer-Verlag, New York, Inc., pp. 213–237.

Panigrahi, B.K., Ravikumar Pandi, V., Das, S., 2008. Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. Energy Conversion and Management 49, 1407–1415.

Rönkkönen, J., 2009. Continuous Multimodal Global Optimization with Differential Evolution-Based Methods. Lappeenranta University of Technology, Lappeenranta, Finland. ISBN:978-952-214-851-3.

Storn, R., Price, K., 1995. Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. Technical Report, International Computer Science Institute.

Takahama, T., Sakai, S., 2009. Solving difficult constrained optimization problems by the ε constrained differential evolution with gradient-based mutation. In: Mezura-Montes, E. (Ed.), Constraint-Handling in Evolutionary Optimization. Springer, Berlin/Heidelberg, pp. 51–72.

Tessema, B., Yen, G.G., 2009. An adaptive penalty formulation for constrained evolutionary optimization. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 39, 565–578.

Tsutsui, S., Yamamura, M., Higuchi, T., 1999. Multi-parent recombination with simplex crossover in real coded genetic algorithms. In: Proceedings of the Genetic Evolutionary Computation Conference, pp. 657–664.

Wright, A.H., 1991. Genetic Algorithms for Real Parameter Optimization. Morgan Kaufmann, San Mateo.