Computers and Chemical Engineering 103 (2017) 1–11

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

A new optimization method: Electro-Search algorithm

Amir Tabari [a], Arshad Ahmad [a,b,*]

[a] Institute of Future Energy, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia
[b] Department of Chemical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia

ARTICLE INFO

Article history:
Received 30 July 2016
Received in revised form 23 January 2017
Accepted 27 January 2017
Available online 7 February 2017

Keywords:
Optimization methods
Meta-heuristics
Electro-Search algorithm
Process optimization
Acetic-acid dehydration

ABSTRACT

Natural phenomena have been the inspiration for proposing various optimization algorithms such as genetic algorithms (GA), particle swarm optimization (PSO) and simulated annealing (SA) methods. The main contribution of this study is to propose a novel optimization method, Electro-Search algorithm, based on the movement of electrons through the orbits around the nucleus of an atom. Electro-Search (ES) algorithm incorporates some physical principals such as Bohr model and Rydberg formula, adopting a three-phase scheme. In the atom spreading phase, the atoms (i.e., candidate solutions) are randomly spread all over the molecular space (i.e., search space). In the orbital transition phase, the electrons jump to larger orbits, aiming for orbits with higher energy levels (i.e., better fitness value). The atoms are then relocated towards the global optimum point in the atom relocation phase, navigated by other atoms' trajectory. Besides, the ES tuning parameters are progressively updated through successive iterations via a self-tuning approach developed, namely Orbital-Tuner method (OTM). The efficiency of ES algorithm is examined in various optimization problems and compared with other well-known optimization methods. The effectiveness and robustness of ES algorithm is then tested in achieving the optimal design of an industrial problem. The results demonstrated the superiority of the new ES algorithm over other optimization algorithms tested, and outperforms current optimization algorithms in real-life industrial optimization problems.

1. Introduction

Optimization has evolved from a subject of academic interest into a powerful tool for the economic industry (Biegler and Grossmann, 2004). This subject aims to identify the best possible set of conditions for achieving a particular goal under a given set of constraints (Floudas and Pardalos, 2009; Neumaier, 2004). Numerous algorithms have been developed to solve various engineering optimization problems over the past decades (Floudas and Gounaris, 2009). Among these, gradient search algorithms are popular for fast convergence and are effective in providing solution in the vicinity of a starting point. However, when the problem has multiple optima, the selection of the starting point is crucial for obtaining accurate results, and the optimal solution may not necessarily represent the global optimum (Edgar et al., 2002). This problem has generated much research on novel meta-heuristic algorithms for solving complex engineering problems. Meta-heuristic algorithms essentially consist of a set of integrating rules and randomness to emulate a specific natural phenomenon (Gendreau and Potvin, 2010). The common way to classify meta-heuristics is based upon trajectory methods versus population-based methods. These methods differ by the way that they deal with the candidate solutions through the search process. In other words, trajectory meta-heuristic methods apply a single solution during the search procedure, while a population of candidate solutions is exploited by population-based meta-heuristics. The main trajectory-based meta-heuristics include simulated annealing (SA) (Clerc and Kennedy, 2002), tabu search (TS) (Hedar and Fukushima, 2006), hill climbing (HC) (Vaughan et al., 2005), greedy randomized adaptive search procedures (GRASP) (Feo and Resende, 1995), iterated local search (ILS) (Lourenc et al., 2002), variable neighborhood search (VNS) (Hansen and Mladenović, 2001), and harmony search (HS) (Lee and Geem, 2005). The main population-based meta-heuristics include genetic algorithms (GA) (Goldberg, 1989), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), scatter search (SS) (Martí et al., 2006), memetic algorithms (MA) (Moscato, 1989), differential evolution (DE) (Storn and Price, 1997), ant colony optimization (ACO) (Hu et al., 2008), artificial bee colony optimization (ABCO) (Karaboga and Bahriye, 2007), path relinking (PR) (Ribeiro and Resende, 2012). More recently, the efforts in this direction resulted in developing meta-heuristics applicable in

* Corresponding author at: Institute of Future Energy, Universiti Teknologi Malaysia, 81310 Johor Bahru, Malaysia. Tel.: +60 75535877; fax: +60 75586166.
E-mail address: arshad@utm.my (A. Ahmad).

http://dx.doi.org/10.1016/j.compchemeng.2017.01.046
0098-1354/

Nomenclature

Ac	accelerator coefficient
D	relocation distance
e	electron
e_{best}	best electron position
E	energy of a photon
f_{N_i}	fitness function value of N_i
hc	speed of light
n	number of atoms
N	nucleus
N_{best}	best nucleus position
N_{new}	new nucleus
r	orbital radius
R	Rydberg constant
Re	Rydberg energy constant

Greek letters

λ	wavelength of the light

Abbreviations

ABCO	artificial bee colony optimization
CDC	conventional distillation column
DWC	dividing-wall column
EDA	estimation of distribution algorithm
ES	electro-search
GA	genetic algorithm
GRASP	greedy randomized adaptive search procedures
HAD	heterogeneous azeotropic distillation
HC	hill climbing
HS	harmony search
ILS	iterated local search
MA	memetic algorithm
OTM	Orbital-Tuner method
PGHA	PSO-GA-based hybrid algorithm
PR	path relinking
PSO	particle swarm optimization
SA	simulated annealing
SHAD	semicontinuous heterogeneous azeotropic distillation
SQP	sequential quadratic programming
TAC	total annual cost
TS	tabu search
VNS	variable neighborhood search
VPGA	Variable population-size genetic algorithm

handling constrained and unconstrained global optimization problems including simple multimembered evolution strategy (SMES) (Mezura-Montes and Coello, 2005), Molecular-Inspired Parallel Tempering (MIPT) (Ochoa et al., 2010), Bat algorithm (Yang, 2010), and Multi Leader Multi Objective Particle Swarm Optimization (MLMOPSO) (Shokrian and High, 2014).

These meta-heuristics are simpler to implement from a computational standpoint, and also easier to utilize in a specific application because they do not rely on differentiation. The optimization capabilities of these methods applied to highly multimodal problems, black-box objective functions, and problems with uncertainties have been widely examined (Glover and Kochenberger, 2003; Schneider and Kirkpatrick, 2006). Although, these algorithms overcome several shortcomings of conventional numerical methods, researchers are constantly pursuing novel heuristic algorithms inspired by nature for solving intricate real-world optimization problems (Grossmann and Biegler, 2004).

In this paper, a nature-inspired optimization algorithm denoted "Electro-Search" (ES) algorithm is proposed based on the orbital movement of the electrons around the atomic nucleus. The new features of the ES algorithm facilitate its search of global optimum point without the need for suitable initial values of the tuning parameters. After formulating the ES algorithm based on the simplified movement of electrons, its performance is evaluated against various standard benchmark test functions available in the literature. The optimal results are then compared with the results of selected algorithms, namely genetic algorithm (GA), simulated annealing (SA), variable population-size genetic algorithm (VPGA), PSO-GA-based hybrid algorithm (PGHA) (Shi et al., 2005) and sequential quadratic programming (SQP) (Boggs and Tolle, 2000), in terms of success rate, number of function evaluations, and computation time. Finally, the capability of ES, PSO, and SQP algorithms for optimizing different designs of heterogeneous azeotropic distillation (HAD) is investigated comparatively.

2. Structure of an atom

An atom is the tiniest unit of matter by which all chemical elements are characterized. Atoms are made of a nucleus and one or more electrons rotating around it. A nucleus is composed of one or more protons and normally the same number of neutrons, which constitute 99.94% of the atomic mass. Protons and electrons have positive and negative electric charges, respectively, and neutrons have no electric charge. An atom is electrically neutral when the number of protons is equal to the number of electrons. Conversely, if the number of electrons differs from the number of protons the atom can be charged either positively or negatively and is denoted as ion. Specific forces act between the atom constituents. Protons and neutrons in the nucleus are attracted to each other by the nuclear force which is stronger than electromagnetic force acting between electrons and protons.

Atomic number is the number of protons in the nucleus that basically defines an element. Isotopes of an element are the atoms with the same number of protons and different number of neutrons. The magnetic properties of an atom are influenced by the electrons. Attachment of one or more atoms by formation of chemical bonds result in new chemical compounds, i.e., molecules. It is worth noting that the most changes observed in the nature are depending upon the ability of atoms to associate and dissociate with other atoms.

2.1. Bohr model

Fig. 1 illustrates Bohr's atomic model, which consists of protons and neutrons occupying a dense central region called the nucleus,

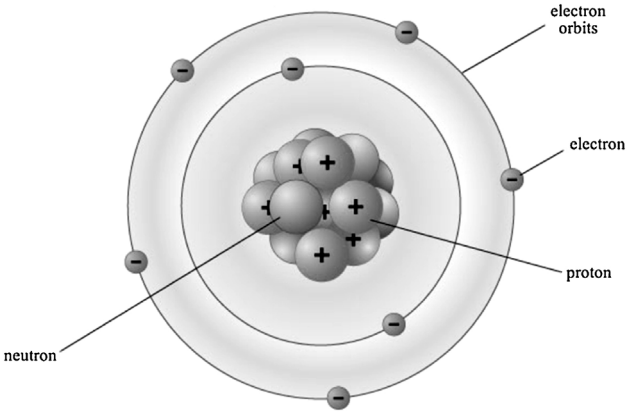

Fig. 1. The Bohr atom.

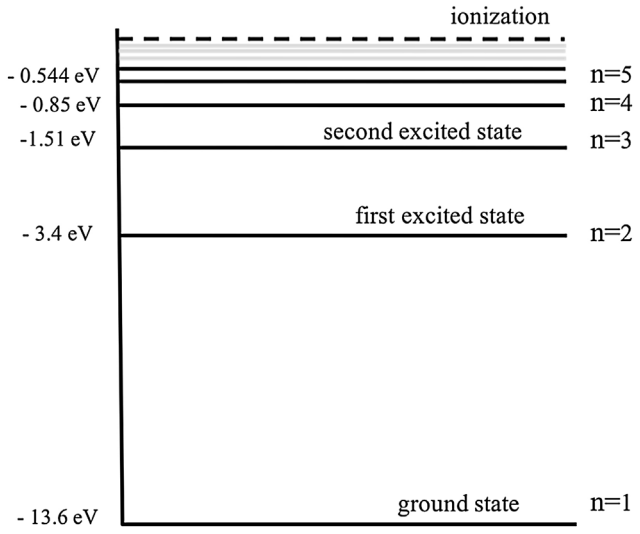

Fig. 2. Quantized energy levels in hydrogen.

and the electrons orbiting the nucleus, akin to the motion of planets around the Sun. Note that Fig. 1 is not to scale because in reality, the radius of the nucleus is approximately 100,000 times smaller than the radius of the entire atom (Bohr, 1913).

The basic feature of quantum mechanics incorporated in the Bohr Model is that the energy of the particles in the Bohr atom is restricted to a discrete set of values or "quantized" levels. This means that only certain orbits with specific radii are allowed, and orbits in between simply are not stable. Fig. 2 shows such quantized energy levels for the hydrogen atom.

These levels are labeled by an integer n that is denoted as the quantum number. The lowest energy state is generally termed the ground state. The states with increasingly higher energy compared with the ground state are called the first excited state, second excited state, and so on. For example, in the case of the hydrogen atom, an electron with energy beyond the so called "ionization potential" is no longer bound to the atom. The energy levels above this ionization potential, approximately 13.6 electron-Volt (eV) above the ground state, form a continuum.

Electrons can make transitions between the orbits allowed by quantum mechanics via absorbing or emitting the energy difference between the orbits. Fig. 3 shows an atomic excitation caused by absorption of a photon and an atomic de-excitation caused by emission of a photon (Lerner and Trigg, 2005; Parker, 1993).

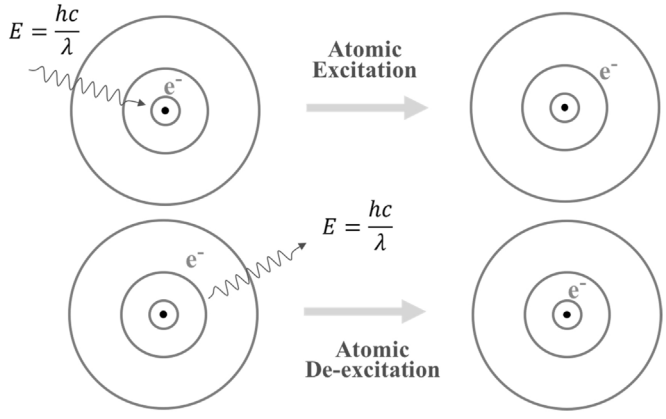

Fig. 3. Electron excitation and de-excitation by absorption and emission of light, respectively.

In each case the wavelength of the emitted or absorbed light is exactly such that the photon carries the energy difference between the two orbits. This energy may be calculated by dividing the product of the Planck constant and the speed of light (hc) by the wavelength of the light (λ). Thus, an atom can absorb or emit only certain discrete wavelengths (or equivalently, frequencies or energies).

2.2. Rydberg formula

After an electron moves from its original energy level to a higher energy level, it may return to the initial state by passing through any intermediate level by emitting a photon. The different energy levels of an atom determine the wavelengths of emitted photon. In case of hydrogen atom, the energy of the emitted photon is given by the following formula:

$$E = E_i - E_f = R_E \left(\frac{1}{n_f^2} - \frac{1}{n_i^2} \right) \quad (1)$$

where n_f and n_i are the final and initial energy level, respectively, and R_E is the Rydberg energy. Accounting for the relationship between photon energy and wavelength, $E = hc/\lambda$, the wavelength of the emitted photon is described by the Rydberg formula:

$$\frac{1}{\lambda} = R \left(\frac{1}{n_f^2} - \frac{1}{n_i^2} \right) \quad (2)$$

where the Rydberg constant (R) is R_E/hc. The Rydberg formula can be extended to multi-electron atoms by replacing "n" with "$n - b$", where "b" represents a screening effect due to the inner electron shell (Lerner and Trigg, 2005; Parker, 1993). The search for candidate solutions in an optimization problem is analogous to electrons searching for higher energy levels (i.e., better fitness function value). Such variations in orbital energy levels were described by Bohr's model and Rydberg's formula. The formulation of the new algorithm based on these analogies is presented in the following section.

3. Electro-Search algorithm

In the Electro-Search algorithm, the domain of candidate solutions is analogous to the molecular space where various atoms are positioned. The electrons orbiting the nucleus of each atom vary their orbits gradually in order to achieve molecular states characterized by the highest energy level, which is analogous to the optimum point of the objective function.

3.1. Outline of the search procedure

The Electro-Search algorithm can be divided in three phases defined below:

3.1.1. 1st phase: atom spreading

In this phase, the candidate solutions are randomly distributed over the search space. Each candidate represents an atom, composed of a nucleus around which the electrons orbit (Fig. 4). The electrons are confined into defined orbits around the nucleus and could transition between them by absorbing or emitting specific amounts of energy in the process (Bohr Model).

3.1.2. 2nd phase: orbital transition

In this phase, the electrons around each nucleus move to larger orbits, aiming for orbits with higher energy levels. This orbital transition is performed inspired by the concept of the quantized energy

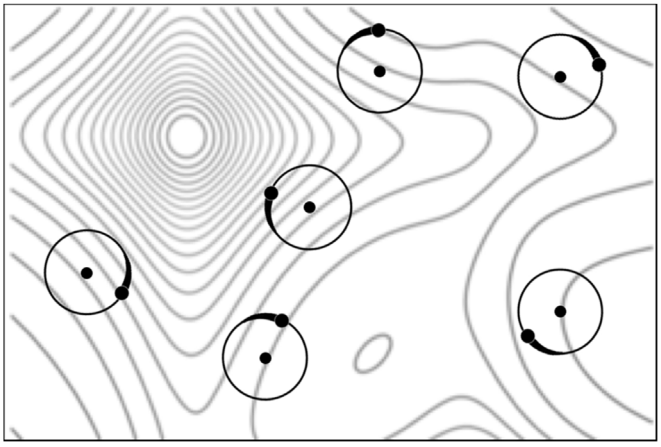

Fig. 4. Schematic diagram representing the 1st phase of Electro-Search algorithm (atom Spreading).

levels in hydrogen atom (see Fig. 2), and can be expressed for each nucleus as follows:

$$e_i = N_i + (2 \times rand - 1)\left(1 - \frac{1}{n^2}\right) \cdot r \tag{3}$$

$$n \in \{2, 3, 4, 5\}$$

$$rand \in [0, 1]$$

where N_i is the current position of the nucleus, *rand* is selected as uniform random numbers in the range [0,1], n is the energy level that defines the vicinity in which the electrons can be positioned, and r is the Orbital radius determined by D_k (Eq. (4)) except for the first iteration which is defined randomly. Once electrons occupy these new orbits, the electron with the highest energy level around each nucleus (i.e., best fitness value) is selected as the best electron (e_{best}). Accordingly, one particular electron around each nucleus is marked as the best electron which is used for relocating the corresponding atom in the next step (Fig. 5).

3.1.3. 3rd phase: nucleus relocation

In this step, the position of the new nucleus (N_{new}) is assigned based on the energy of an emitted photon, defined by the difference of energy level between the two atoms (Rydberg formula). For each

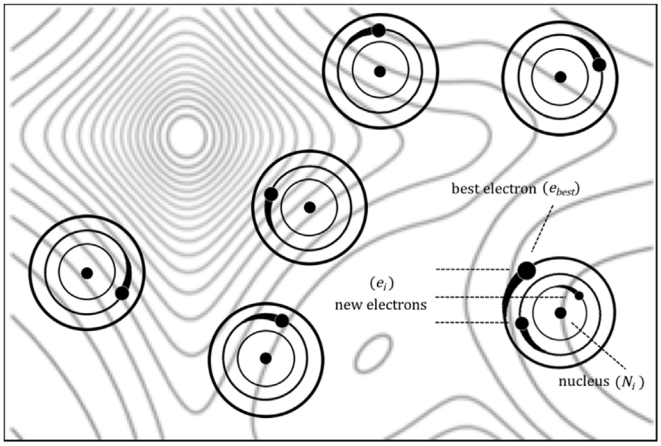

Fig. 5. Schematic diagram representing the 2nd phase of the Electro-Search algorithm (orbital transition).

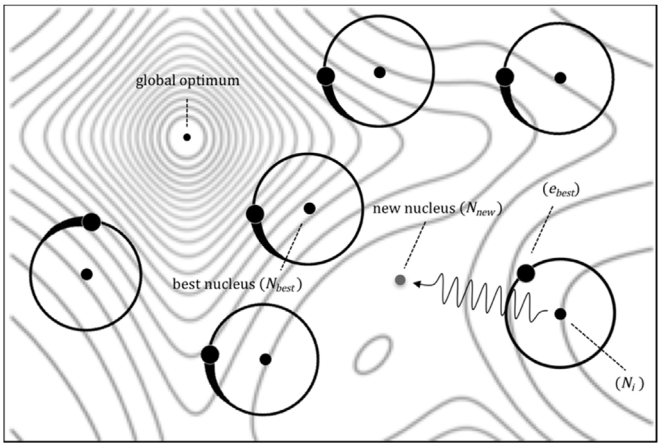

Fig. 6. Schematic diagram representing the 3rd phase of the Electro-Search algorithm (nucleus relocation).

nucleus, the standard form of the formula for the nucleus relocation can be expressed in vector notation as:

$$\vec{D}_k = \left(\vec{e}_{best} - \vec{N}_{best}\right) + Re_k \otimes \left(\frac{1}{\vec{N}_{best}^2} - \frac{1}{\vec{N}_k^2}\right) \tag{4}$$

$$\vec{N}_{new,k} = \vec{N}_k + Ac_k \times \vec{D}_k \tag{5}$$

where at each iteration k, the relocation distance $\vec{D}_k$ is computed for each nucleus based on the current best nucleus position ($\vec{N}_{best}$), the best electron around the nucleus ($\vec{e}_{best}$), and the current position of the nucleus ($\vec{N}_k$) affected by Rydberg's energy constant (Re_k). The position of the new nucleus ($\vec{N}_{new}$) is then updated using its current position and the computed relocation distance, affected by Accelerator coefficient (Ac_k). Note that the symbol $\otimes$ denotes element-by-element vector multiplication. This procedure is performed on all nuclei, leading to gradual reposition of all atoms towards the global optimum point (see Fig. 6). The convergence speed of this relocation depends on the algorithm coefficients *Re* and *Ac* which are randomly set in the first iteration. In the following iterations, a new self-tuning approach, called Orbital-Tuner method, is introduced to update these coefficients at each iteration.

3.2. Orbital-Tuner method (OTM)

As was shown in Eqs. (4) and (5), the algorithm coefficients *Re* and *Ac* are required for calculating the location of the new nuclei ($\vec{N}_{new}$). Initially, the coefficients are randomly selected. Once the first set of nuclei are positioned on their new locations by the procedure described in the previous section, the coefficients of the subsequent iterations are updated using a new method called the Orbital-Tuner method. The Orbital-Tuner method (OTM) is developed based on the cumulative normal density function (Stuckman, 1988; Tigli et al., 1994) which has been improved by calculating the center of "mass" instead of selecting two candidates and finding their center of gravity. This method requires many input variables, including the current algorithm coefficients and nucleus fitness values, but renders only one output, namely the updated algorithm coefficient (i.e., Re_{k+1} or Ac_{k+1}). The Orbital-Tuner method is described for each atom by the following scheme:

$$Re_{k+1} = Re_k + \left(Re_{best} + \sum_{i=1}^{n} \frac{Re_i / f_{N_i|Re_i}}{1/f_{N_i|Re_i}}\right)/2 \tag{6}$$

$$Ac_{k+1} = Ac_k + \left(Ac_{best} + \sum_{i=1}^{n} \frac{Ac_i / f_{N_i|Ac_i}}{1/f_{N_i|Ac_i}}\right)/2 \tag{7}$$

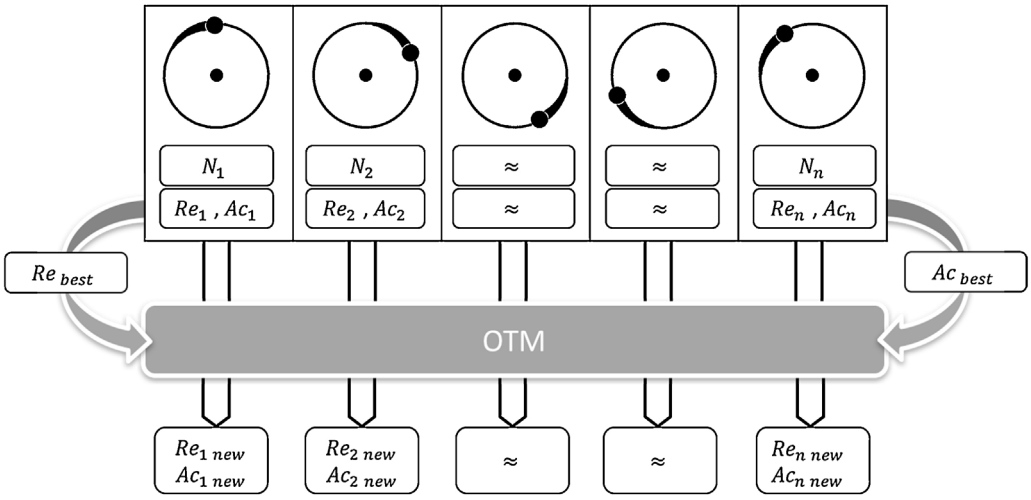

Fig. 7. Schematic diagram of Orbital-Tuner method (OTM).

where n is the number of atoms, Re_i and Ac_i are the algorithm coefficients of available nucleui, $f_{N_i|Re_i}$ and $f_{N_i|Ac_i}$ are the fitness function values of these nucleui, Re_{best} and Ac_{best} are the corresponding algorithm coefficients of the N_{best}, and Re_{k+1} and Ac_{k+1} are the values of Rydberg energy constant and Accelerator coefficient of the iteration $k+1$, respectively. Using this method, the trajectory information of all repositioned atoms are utilized to iteratively navigate all other atoms towards the global optimum as shown in Fig. 7. Because the algorithm coefficients Re and Ac are regulated via a self-tuning approach, the selection of initial values for such parameters does not affect significantly the performance of the algorithm in terms of convergence speed, function evaluation, and atoms' trajectory toward the global optimum.

3.3. Parameter selection

3.3.1. Number of atoms (n)

Typically, in a random search method, the thoroughness of the search space sampling increases with the number of atoms in the initial population. However, this number will also influence other performance criteria, such as the required algorithm iterations, function evaluations, success rate, etc. As discussed in next section, increasing the number of atoms decreases the number of algorithm iterations required to achieve the goal at the same time increasing the success rate. Despite the importance of the criteria mentioned, the number of function evaluations is considered as the main performance criterion in real-life applications. An initial population composed of few atoms results in a low success rate and requires more iterations, whereas a large population increases the success rate but also requires too many function evaluations. Table 2 shows the different evaluation criteria for optimization of benchmark test functions.

3.3.2. Orbital radius (r)

As was previously described, the electrons orbiting each nucleus could move to larger orbits. The orbital radius defines the maximum radius of orbits the electrons around each nucleus are allowed transfer to in each iteration. This parameter is determined by the relocation distance (Eq. (4)). Obviously, the relocation distance ($\bar{D}_k$), and hence the orbital radius of each nucleus, vary through the iterations as the atom approaches the optimum position. As the orbital radius is gradually shortened through the iterations, the transition zone for electrons is progressively reduced. This reduction allows the algorithm to converge towards the goal by confining the relocation of the electrons in the vicinity of the potential global optimum point.

3.3.3. Number of electrons (e)

The new electrons around each nucleus in the Electro-Search algorithm accounts for the randomness of search space exploration. The converging speed is reduced by presence of random electrons in each iteration. This expedient avoids the premature convergence to non-optimal points, as well as improves the search space exploration, especially when the initial atoms are positioned far from the optimum point. Furthermore, the gradual contraction of orbital radius through the iterative process restricts the area allowed for orbital relocation, thereby preventing new random electrons to be placed far from the nucleus.

3.4. Convergence criteria

Stopping criteria are used in evolutionary algorithms to take into account the convergence of the population through the iterations. Once the population of the candidate solutions is "stagnant", the algorithm is terminated (Engelbrecht, 2007). Different termination criteria have been employed in the optimization literature such as maximum number of iterations, finding an acceptable solution, observing no improvement through a specific number of iterations, etc. (Rangaiah, 2010; Rangaiah and Bonilla-Petriciolet, 2013). In the Electro-Search algorithm, the maximum number of iterations was selected as the termination criterion. According to mentioned three-phase procedure, the fundamental steps of Electro-Search algorithm can be represented by the flowchart shown in Fig. 8.

4. Performance evaluation

To examine the capability of the proposed Electro-Search algorithm in global optimization, several optimization experiments are carried out and the results are compared with other well-known optimization algorithms. The ES algorithm was programmed in MATLAB™ using a PC with dual-core 1.80 GHz CPU and 4 GB of Ram.

4.1. Benchmark test functions

As shown in Table 1, several distinguished benchmark functions in the literature (Clerc and Kennedy, 2002; Hedar and Fukushima, 2006) were used to evaluate the performance of ES algorithm in terms of success rate, function evaluations, and CPU time. These

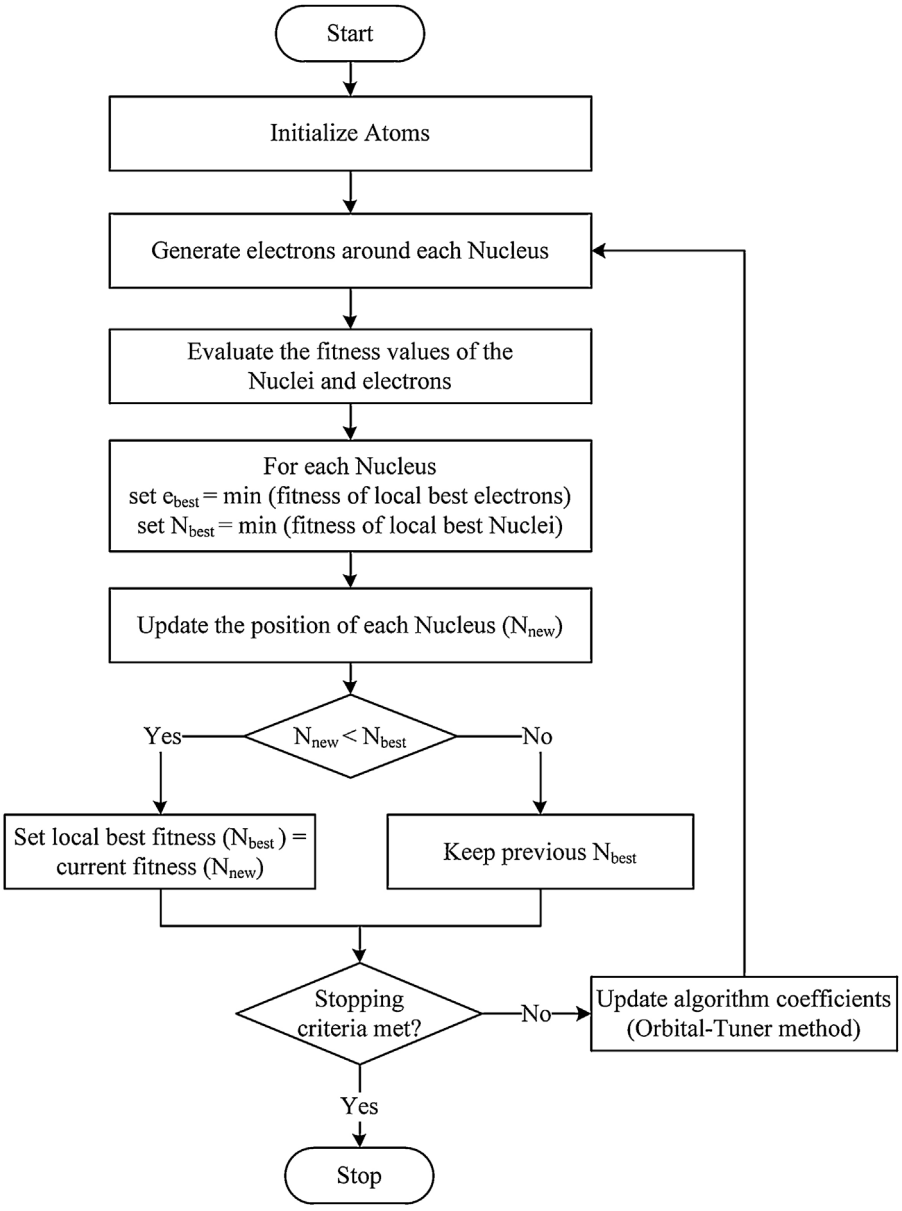

Fig. 8. Flowchart of the Electro-Search algorithm.

include diverse multimodal test functions to take into account the various complications emerged in global optimization problems.

The functions addressed, the number of dimensions (n), the acceptable range of the variable (x) and the optimum values of test functions are summarized in Table 1. Each optimization experiment was performed 30 times with random initialization of the algorithm coefficients (*Re* and *Ac*). The number of iterations to achieve the optimum, and the number of function evaluations were recorded. The success rate was calculated for each category as the percentage of optimization experiments that reached the optimum value. Tests were performed with different number of atoms (i.e., population sizes) as $N = 10$ 10, 20, 30 and 50 atoms. The atoms were allowed to be placed on any position outside the defined range through the optimization process. Results of the optimization of different benchmark test functions by the ES algorithm are shown in Table 2.

As shown in Table 2, the success rate of unity was not achieved for the $N = 10$ except for Sphere function, showing that such initial number of atoms was insufficient to find the optimum. The success rate is unity for Sphere function regardless of population size, and the lowest number of function evaluations was achieved using $N = 20$ atoms, marked as the best result for the Sphere function test. For Shubert, Schaffer, Michalewicz, Rosenbrock and Powell's functions, with $N = 20$, 30, and 50 atoms, a success rate of unity was obtained; however, the number of iterations required to reach the goal decreases and the number of function evaluations increases with the population size. The results for Rastrigin, Beale, Griewank and Schwefel's functions showed that the population size of $N = 10$ and 20 are insufficient to reach the optimum point and the lowest number of function evaluations (to obtain success rate of unity) was achieved using $N = 30$ atoms.

Based on the results shown in Table 2, the population size of $N = 10$ does not fully guarantee a successful optimization. Our results suggest that a population size in a range of N: [20–30] atoms maximizes the success rate while, at the same time, minimizes the number of function evaluations. Moreover, Fig. 9 shows that the best solution CPU time for optimizing Schwefel's function is higher than all the other functions, which is expected due to

Table 1
Benchmark test function.

Function	Formula	Dim. (n)	Range	Optimal f
Shubert	$f\left(\vec{x}\right) = \prod_{i=1}^{n}\sum_{j=1}^{5} j\cos\left[(j+1)x_i + j\right]$	2	$[-10, 10]^n$	−186.7309
Rastrigin	$f\left(\vec{x}\right) = \sum_{i=1}^{n}\left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right)$	2	$[-5.12, 5.12]^n$	0
Schaffer	$f\left(\vec{x}\right) = 0.5 - \frac{\left(\sin\sqrt{x_1^2+x_2^2}\right)^2 - 0.5}{\left(1+0.001\left(x_1^2+x_2^2\right)\right)^2}$	2	$[-100, 100]^n$	0
Beale	$f\left(\vec{x}\right) = (1.5 - x_1 + x_1x_2)^2 + \left(2.25 - x_1 + x_1x_2^2\right)^2 + \left(2.625 - x_1 + x_1x_2^3\right)^2$	2	$[-4.5, 4.5]^n$	0
Michalewicz	$f\left(\vec{x}\right) = -\sum_{i=1}^{n}\sin\left(x_i\right)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right),\ m = 10$	10	$[0, \pi]^n$	−9.66015
Sphere	$f\left(\vec{x}\right) = \sum_{i=1}^{n} x_i^2$	20	$[-100, 100]^n$	0
Rosenbrock	$f\left(\vec{x}\right) = \sum_{i=1}^{n-1}\left(100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right)$	20	$[-30, 30]^n$	0
Griewank	$f\left(\vec{x}\right) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	20	$[-600, 600]^n$	0
Powell	$f\left(\vec{x}\right) = \sum_{i=1}^{n/4}(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	24	$[-4, 5]^n$	0
Schwefel	$f\left(\vec{x}\right) = 418.9829n - \sum_{i=1}^{n} x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	26	$[-500, 500]^n$	0

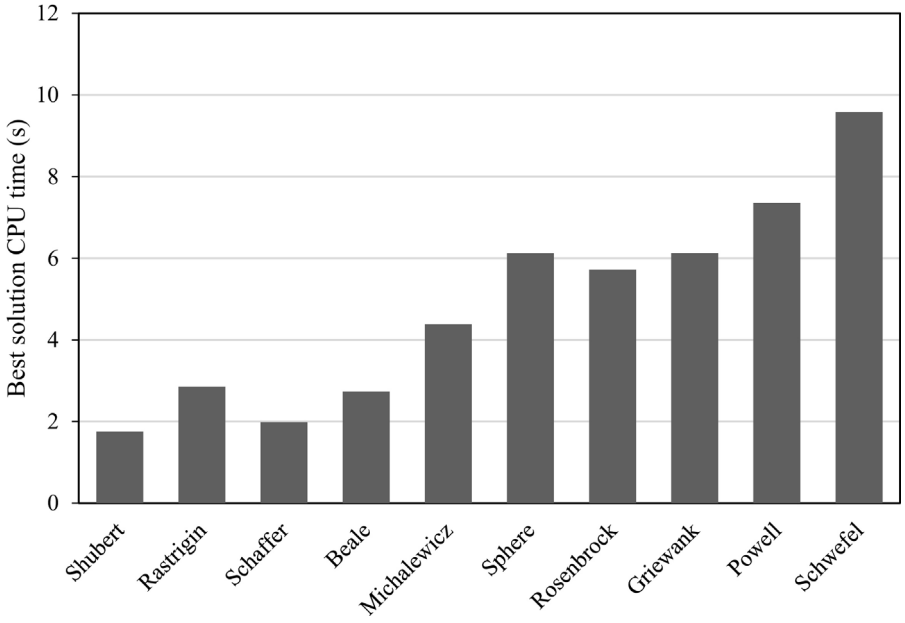

Fig. 9. Comparative results for the test functions: best solution CPU time for ES algorithm.

higher number of function evaluations involved. Among all functions, Schubert's function requires the lowest computational time due to its simplicity and lower dimensions compared with the other more complex functions.

4.2. Comparison with other algorithms

The performances of the ES algorithm in terms of success rate, function evaluations, and computation time were also compared to other optimization algorithms. The tuning parameters of the GA (e.g., population size, crossover method and probability, mutation method and probability, etc.) were obtained from Leboreiro and Acevedo (2004). The VPGA and PGHA algorithms were performed based on a study by Shi et al. (2005). The parameters reported by Patel and Rao (2010) were used to tune the PSO parameters (e.g., population size, cognitive and social parameters, maximum velocity, etc.), and the tuning parameters of SA were taken from Faber et al. (2005). Note that 30 atoms were used in operation of ES

Table 2
Optimization results of Electro-Search algorithm.

Function	Number of atoms (initial population)	Average number of algorithm iterations to achieve the goal	Success rate	Average number of function evaluations
Shubert	10	265	0.9	1660
	20	198	1	2212
	30	188	1	3188
	50	168	1	4547
Rastrigin	10	151	0.9	1680
	20	113	0.9	2520
	30	108	1	3267
	50	98	1	4905
Schaffer	10	549	0.7	7842
	20	129	1	2592
	30	126	1	3807
	50	125	1	6255
Beale	10	294	0.9	2144
	20	225	0.9	2680
	30	204	1	3863
	50	188	1	5510
Michalewicz	10	392	0.8	4322
	20	307	1	3767
	30	289	1	4763
	50	233	1	5984
Sphere	10	623	1	6237
	20	278	1	5562
	30	229	1	6885
	50	203	1	10170
Rosenbrock	10	436	0.9	4850
	20	334	1	6696
	30	319	1	9585
	50	272	1	13635
Griewank	10	585	0.8	7312
	20	252	0.9	5620
	30	210	1	6318
	50	182	1	9135
Powell	10	830	0.7	7461
	20	510	1	6335
	30	447	1	10540
	50	365	1	12898
Schwefel	10	1208	0.7	9505
	20	685	0.8	8213
	30	574	1	11540
	50	437	1	13763

algorithm. In this comparative study, each test problem was run 100 times using the same experimental conditions as described in the previous section.

Table 3 summarizes the experiment results for different optimization algorithms on defined test functions. As shown, ES, SQP and PSO algorithms achieved the topmost success rate (i.e., unity) for Shubert, Beale, Sphere, and Rosenbrock's functions. For Griewank's function, the best success rate was obtained by ES and SA (i.e., 0.95), followed by PSO and PGHA (i.e., 0.94), whereas for Schwefel's function, the success rate of 0.98 was achieved by both ES and SQP algorithms. The results also indicate that the ES algorithm achieved the best results using the least number of function evaluations compared with all other algorithms.

Furthermore, the computational time required by the ES algorithm to reach the optimum point was shorter than the other algorithms, despite ES's higher success rate. The plot in Fig. 10 visually illustrates the comparison of best solution CPU time for different test functions.

4.3. Nonlinear constrained optimization problem

A NLP problem taken from Hock and Schittkowski (1981) with five continuous variables and three nonlinear equality constraints is addressed in this section to further investigate the performance of ES algorithm on constrained optimization problems. The problem is summarized as follows:

$$Minf\,(x) = e^{x_1 x_2 x_3 x_4 x_5} \tag{7}$$

s.t.

$$h1\,(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$h2\,(x) = x_2 x_3 - 5x_4 x_5 = 0$$

$$h3\,(x) = x_1^3 + x_2^3 + 1 = 0$$

$$-2.3 \le x_i \le 2.3\ (i = 1, 2)$$

$$-3.2 \le x_i \le 3.2\ (i = 3, 4, 5)$$

The optimal solution for the problem as provided by Hock and Schittkowski (1981) is $(x_1, x_2, x_3, x_4, x_5; f) = (-1.717143, 1.595709, 1.8277247, -0.763643, -0.763645; 0.0539489)$. This problem was also studied by Yiqing et al. (2007) using Improved PSO (R-PSO) algorithm, simple multimembered evolution strategy (SMES), and stochastic ranking (SR). Other applications include the work of Shokrian and High (2014) using Multi Leader Multi Objective Particle Swarm Optimization (MLMOPSO). The results from these studies are compared with those produced from the constrained optimization using the proposed ES algorithm. To address the constraints, the penalty function method proposed by Homaifar et al. (1994) was used. The comparison is shown in Table 4.

The optimal solution of the problem, $f(x) = 0.0539489$, was obtained by the ES algorithms in 10,000 iteration numbers. Based on the results reported (Shokrian and High, 2014; Yiqing et al., 2007), the SR algorithm required 350,000 iterations, SMES with 240,000 iterations, R-PSO with 30,000 iteration of the outer loop, and MLMOPSO algorithm performed 10,000 iteration numbers. The Optimum, Best, Average and Worst values of all algorithms are provided in Table 4. The ES algorithm could reach the optimum value with considerably fewer iteration numbers than the other algorithms, showing competence for dealing with constrained NLP problems.

4.4. Optimization problem in an industrial chemical plant

To evaluate the capabilities of the ES algorithm for the optimization of industrial problems, acetic acid (HAc) dehydration process was selected to be optimized as one of the important operations in the aromatic acid production (Chien and Kuo, 2006; Gau, 2005). As shown in Fig. 11, three systems for HAc dehydration, i.e., Conventional distillation column (CDC), Dividing-wall column (DWC), and Semicontinuous heterogeneous azeotropic distillation (SHAD), are optimized in this section using the procedure detailed by Wang et al. (2008) and Tabari and Ahmad (2015).

The performance of the ES algorithm was also evaluated against the PSO and SQP algorithms. These algorithms were selected due to their better performances in optimization of benchmark test functions compared to all other algorithms (see Table 3). The maximum number of function evaluations was set to 300 as termination criteria and each test problem was run 30 times. sPSO and SQP were run using the MATLAB Optimization Toolbox. The tuning parameters of the PSO were obtained from Patel and Rao (2010). Process optimization with SQP requires approximating the derivative with respect

Table 3
Comparison of different optimization algorithms.

Function	Average number of function evaluations Success rate						
	GA	VPGA	PSO	SA	SQP	PGHA	ES
Shubert	2847.4 0.97	2741.9 0.98	2651.5 1	2818.2 0.99	2522.9 1	2804.1 0.99	2398.6 1
Rastrigin	3901.6 0.9	3756.7 0.8	3633 1	3861.3 0.95	3457 0.95	3842.2 0.92	3286.1 1
Schaffer	2939.4 0.9	2829.2 0.9	2736.6 0.95	2908.1 0.94	2603.3 0.98	2898.5 0.94	2475.3 0.98
Beale	2923.3 0.99	2814.9 1	2722.4 1	2893.4 0.99	2597.9 1	2879 1	2592.6 1
Michalewicz	4422.8 0.98	4258.6 1	4118.4 0.98	4377.2 0.99	3918.8 0.98	4355.5 1	3725.2 0.99
Sphere	6576.5 0.9	6332.6 0.75	6123.5 1	6508.8 0.9	5826.7 1	6475.9 0.95	5538.9 1
Rosenbrock	5503.8 0.85	5299.6 0.88	5125.5 1	5447.5 0.95	4891.1 1	5420.3 0.96	4880.4 1
Griewank	7446.2 0.8	7170.3 0.85	6934 0.94	7369.8 0.95	6597.6 0.93	7333 0.94	6271.8 0.95
Powell	7410.1 0.95	7135.3 0.97	6899.7 0.99	7333.9 0.98	6565.4 0.98	7296.8 0.97	6240.5 1
Schwefel	10988.5 0.95	10580.7 0.96	10233.2 0.97	10876 0.97	9765.1 0.98	10821.8 0.97	9743.3 0.98

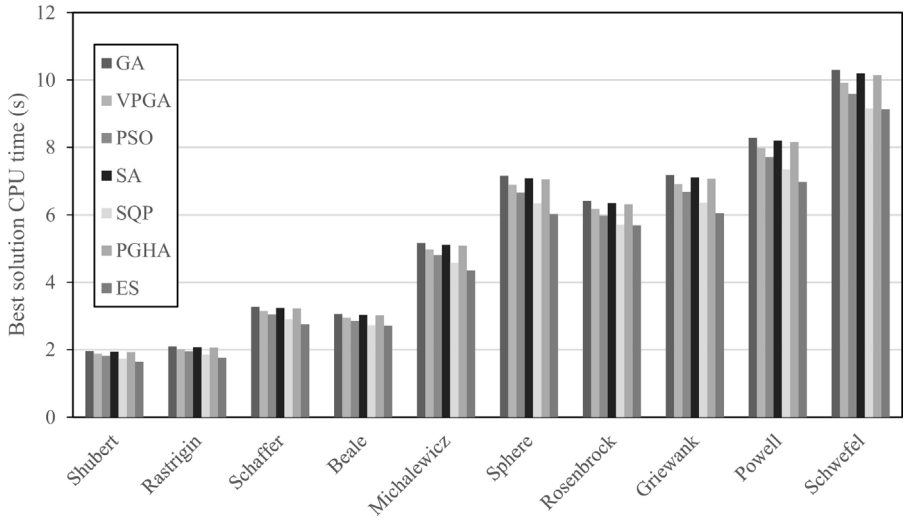

Fig. 10. Comparative results for the test functions: best solution CPU time for GA, VPGA, PSO, SA, SQP, PGHA and ES algorithms.

Table 4
Comparative results for NLP problem: optimum values achieved by different algorithms.

Solution	Optimum	SR	SEMS	R-PSO	MLMOPSO	ES
Best	0.0539489	0.053957	0.053986	0.0539498	0.0539498	0.0539498
Average	0.0539489	0.057006	0.166385	0.0539801	0.0540200	0.0558861
Worst	0.0539489	0.216915	0.468294	0.0540528	0.0583200	0.0563455

to the variable which was carried out by calculating the central finite differences. This procedure results in a linear approximation by three function evaluations for each component of the derivative (Cha and Mayne, 1989). All three distillation processes were rigorously simulated in AspenHysys™ and then optimized by three different optimization algorithms in order to obtain the optimal design variables for each system. The optimization problem (i.e., cost calculation, constraints, and penalty function) was formulated based on the procedure presented by Tabari and Ahmad (2015).

Fig. 12 displays the comparative results for the three processes after applying the optimal design variables in terms of total annual cost (TAC) after the termination criteria (i.e. the maximum

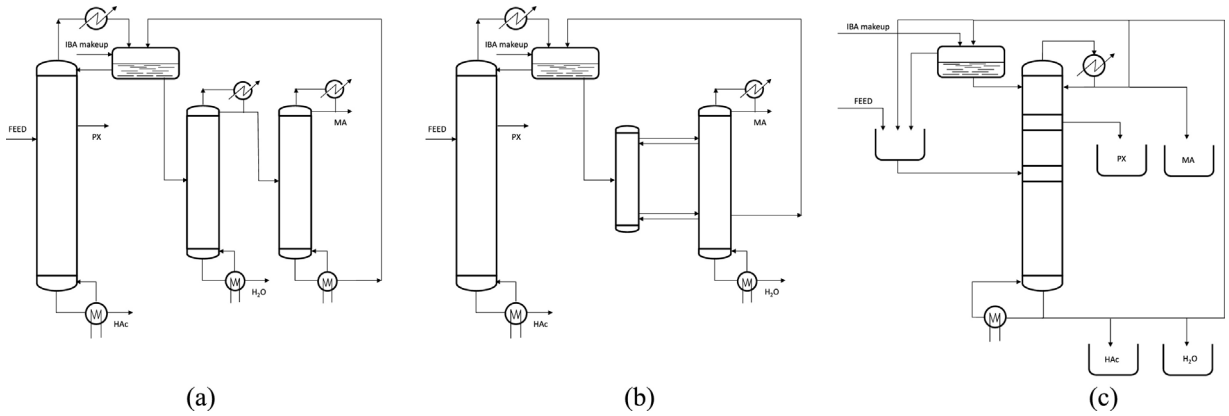

Fig. 11. Flowsheet of different distillation systems for HAc dehydration process: (a) CDC, (b) DWC, (c) SHAD.

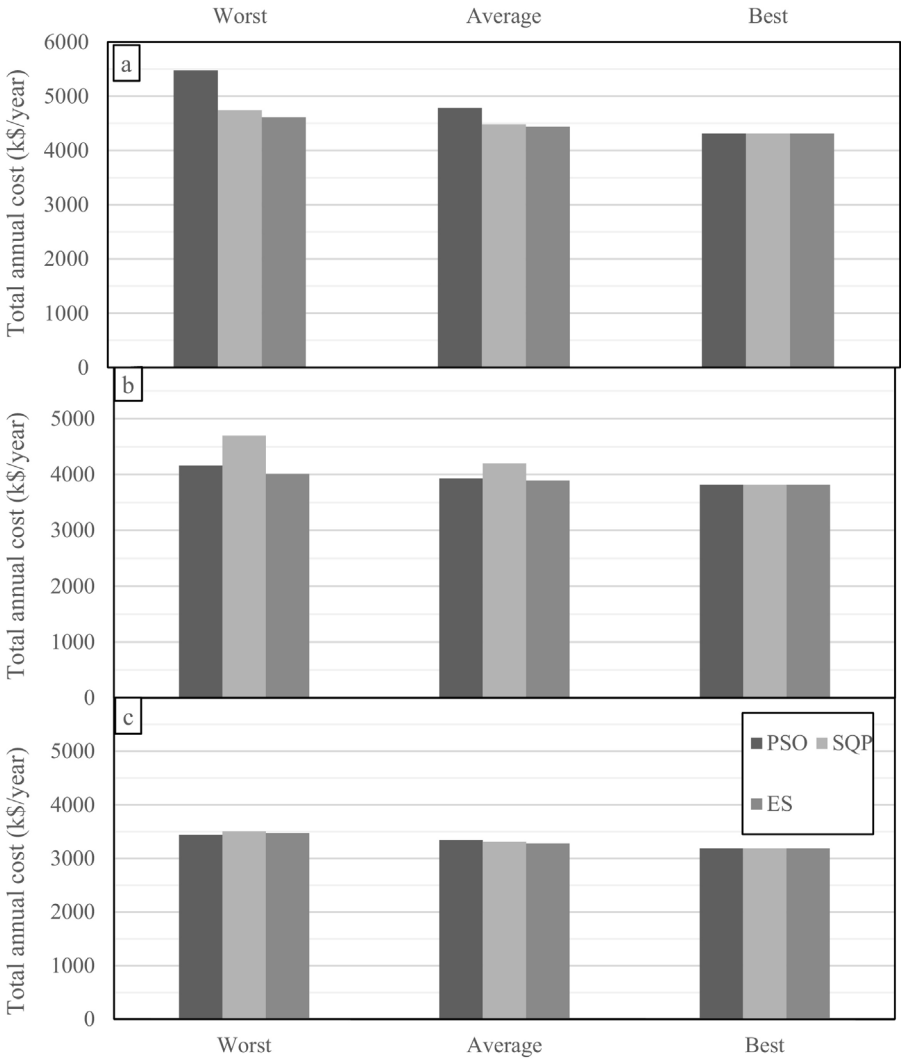

Fig. 12. Comparative results for the optimization of HAc dehydration process: values of the total annual cost for (a) CDC, (b) DWC, (c) SHAD.

number of function evaluations) is met. The results show that the ES algorithm has reached the same best optimal and slightly better average values than PSO and SQP for all three designs. The SQP, by contrast, had the worst performance for the average and worst cases on DWC design. For the CDC design, PSO showed higher TAC in average and worst cases compared to other algorithms. On optimizing the SHAD design, PSO and SQP had the higher TAC in average and worst cases, respectively. Overall, these results show that ES algorithm is efficiently applicable in optimization of HAc dehydration process, making it a promising alternative to current methods for optimization of industrial problems.

5. Conclusions

A new meta-heuristic optimization algorithm, denoted as Electro-Search (ES) algorithm, was proposed inspired by behavior of orbiting electrons in a molecular space. A three-phase scheme was conducted based on some basic physical principles to simulate the progressive atoms' trajectory towards global optimum. A variety of optimization problems, including multimodal benchmark test functions and constrained NLP, were presented to examine the effectiveness and robustness of the new algorithm in comparison with other meta-heuristic optimization methods. A real-world industrial problem was also investigated to reveal the performance of the new algorithm in obtaining the optimal design of heterogeneous azeotropic distillation process with different configurations. The results showed the capability of the ES algorithm to reach the global optima with greater performance compared to well-established optimization algorithms in terms of success rate, number of function evaluations, and computation time. The results of the industrial optimization problem showed that the new ES algorithm is superior to the conventional optimization algorithms for reaching the optimal design of chemical processes, and is potentially a powerful and promising optimization method for solving complicated engineering optimization problems.

Acknowledgments

This study is partly supported by the Ministry of Education, Government of Malaysia through the Prototype Research Grant Scheme and Universiti Teknologi Malaysia through RUGS-07H12 and RUGS 13H95.

References

Biegler, L.T., Grossmann, I.E., 2004. Retrospective on optimization. Computers and Chemical Engineering 28, 1169–1192.
Boggs, P.T., Tolle, J.W., 2000. Sequential quadratic programming for large-scale nonlinear optimization. Journal of Computational and Applied Mathematics 124, 123–137.
Bohr, N., 1913. On the constitution of atoms and molecules. Part I. Philosophical Magazine 26, 1–24.
Cha, J.Z., Mayne, R.W., 1989. Optimization with discrete variables via recursive quadratic programming: part 1 – concepts and definitions. Transactions of the American Society of Mechanical Engineers 111, 124–129.
Chien, I.L., Kuo, C.L., 2006. Investigating the need of a pre-concentrator column for acetic acid dehydration system via heterogeneous azeotropic distillation. Chemical Engineering Science 61, 569–585.
Clerc, M., Kennedy, J., 2002. The particle swarm: explosion stability and convergence in a multi-dimensional complex space. IEEE: Transactions on Evolutionary Computation 6, 58–73.
Edgar, T.F., Himmelblau, D.M., Lasdon, L.S., 2002. Optimization of Chemical Processes. McGraw-Hill, New York.
Engelbrecht, A.P., 2007. Computational Intelligence: An Introduction. John Wiley and Sons.
Faber, R., Jockenhövel, T., Tsatsaronis, G., 2005. Dynamic optimization with simulated annealing. Computers & Chemical Engineering 29, 273–290.
Feo, T.A., Resende, M.G.C., 1995. Greedy randomized adaptive search procedures. Journal of Global Optimization 6, 109–133.
Floudas, C.A., Gounaris, C.E., 2009. A review of recent advances in global optimization. Journal of Global Optimization 45, 3–38.
Floudas, C.A., Pardalos, P.M., 2009. Encyclopedia of Optimization, 2nd ed. Springer, US.
Gau, T.K., 2005. Effect of feed impurity on the design of and control of heterogeneous azeotropic distillation for acetic acid dehydration (II). In: Master's Thesis. Taiwan University, Taiwan.
Gendreau, M., Potvin, J.Y., 2010. Handbook of Metaheuristics. Springer.
Glover, F., Kochenberger, G.A., 2003. Handbook of Metaheuristics. Kluwer Academic Publishers, New York.
Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.
Grossmann, I.E., Biegler, L.T., 2004. Part II. Future perspective on optimization. Computers and Chemical Engineering 28, 1193–1218.
Hansen, P., Mladenović, N., 2001. Variable neighborhood search: principles and applications. European Journal of Operational Research 130, 449–467.
Hedar, A.-R., Fukushima, M., 2006. Tabu Search directed by direct search methods for nonlinear global optimization. European Journal of Operational Research 170, 329–349.
Hock, V., Schittkowski, K., 1981. Test examples for nonlinear programming codes. Lecture Notes in Economics and Mathematical Systems. Spring-Verlag, Berlin.
Homaifar, A., Qi, C.X., Lai, S.H., 1994. Constrained optimization via genetic algorithms. Simulation 62, 242–254.
Hu, X., Zhang, J., Li, Y., 2008. Orthogonal methods based ant colony search for solving continuous optimization problems. Journal of Computer Science and Technology 23, 2–18.
Karaboga, D., Bahriye, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39, 459–471.
Kennedy, J., Eberhart, R.C., 1995. Particle swarm optimization. In: Proceeding of IEEE International Conference on Neural Networks, Piscataway, NJ.
Leboreiro, J., Acevedo, J., 2004. Processes synthesis and design of distillation sequences using modular simulators: a genetic algorithm framework. Computers & Chemical Engineering 28, 1223–1236.
Lee, K.S., Geem, Z.W., 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering 194, 3902–3933.
Lerner, R.G., Trigg, G.T., 2005. Encyclopedia of Physics, 3rd ed. Wiley-VCH, Weinheim.
Lourenc, H.R., Martin, O., Stützle, T., 2002. Iterated local search. In: Glover, F., Kochenberger, G. (Eds.), Handbook of Metaheuristics, International Series in Operations Research & Management Science. Kluwer Academic Publishers, Norwell, MA.
Martí, R., Laguna, M., Glover, F., 2006. Principles of scatter search. European Journal of Operational Research 169, 359–372.
Mezura-Montes, E., Coello, C.A.C., 2005. A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Transactions on Evolutionary Computation 9, 1–17.
Moscato, P.A., 1989. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Caltech Concurrent Computation Program Report, 826. Caltech, Pasadena, California.
Neumaier, A., 2004. Complete search in continuous global optimization and constraint satisfaction. Acta Numerica 13, 271.
Ochoa, S., Wozny, G., Repke, J.-U., 2010. A new algorithm for global optimization: molecular-inspired parallel tempering. Computers & Chemical Engineering 34, 2072–2084.
Parker, S.P., 1993. McGraw-Hill Encyclopedia of Physics, 2nd ed. McGraw-Hill, New York.
Patel, V.K., Rao, R.V., 2010. Design optimization of shell-and-tube heat exchanger using particle swarm optimization technique. Applied Thermal Engineering 30, 1417–1425.
Rangaiah, G.P., 2010. Stochastic Global Optimization: Techniques and Applications in Chemical Engineering. World Scientific.
Rangaiah, G.P., Bonilla-Petriciolet, A., 2013. Multi-Objective Optimization in Chemical Engineering: Developments and Applications. John Wiley.
Ribeiro, C.C., Resende, M.G.C., 2012. Path-relinking intensification methods for stochastic local search algorithms. Journal of Heuristics 18, 193–214.
Schneider, J.J., Kirkpatrick, S., 2006. Stochastic Optimization. Springer-Verlag, Berlin.
Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, L.M., 2005. An improved GA and a novel PSO-GA-based hybrid algorithm. Journal of Information Processing Letters 93, 255–261.
Shokrian, M., High, K.A., 2014. Application of a multi objective multi-leader particle swarm optimization algorithm on NLP and MINLP problems. Computers & Chemical Engineering 60, 57–75.
Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359.
Stuckman, B.E., 1988. A global search method for optimizing nonlinear systems. IEEE Transactions on Systems Man, and Cybernetics 18, 965–977.
Tabari, A., Ahmad, A., 2015. A semicontinuous approach for heterogeneous azeotropic distillation processes. Computers and Chemical Engineering 73, 183–190.
Tigli, O., Eksin, I., Ertem, S., Boz, O., 1994. Global optimization method for controlling a plant. Electric Power Systems Research 30, 257–262.
Vaughan, D.E., Jacobson, S.H., Hall, S.N., McLay, L.A., 2005. Simultaneous generalized hill-climbing algorithms for addressing sets of discrete optimization problems. INFORMS Journal on Computing 17, 438–450.
Wang, S., Lee, C., Jang, S., Shieh, S., 2008. Plant-wide design and control of acetic acid dehydration system via heterogeneous azeotropic distillation and divided wall distillation. Journal of Process Control 18, 45–60.
Yang, X.-S., 2010. A new metaheuristic bat-inspired algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Springer, Berlin, Heidelberg.
Yiqing, L., Xigang, Y., Yongjian, L., 2007. An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints. Computers & Chemical Engineering 31, 153–162.