J Ind Eng Int (2017) 13:93–105
DOI 10.1007/s40092-016-0170-x

ORIGINAL RESEARCH

A novel hybrid meta-heuristic technique applied to the well-known benchmark optimization problems

Amir-Reza Abtahi[1] · **Afsane Bijari**[1]

Received: 31 July 2016 / Accepted: 13 September 2016 / Published online: 17 September 2016

Abstract In this paper, a hybrid meta-heuristic algorithm, based on imperialistic competition algorithm (ICA), harmony search (HS), and simulated annealing (SA) is presented. The body of the proposed hybrid algorithm is based on ICA. The proposed hybrid algorithm inherits the advantages of the process of harmony creation in HS algorithm to improve the exploitation phase of the ICA algorithm. In addition, the proposed hybrid algorithm uses SA to make a balance between exploration and exploitation phases. The proposed hybrid algorithm is compared with several meta-heuristic methods, including genetic algorithm (GA), HS, and ICA on several well-known benchmark instances. The comprehensive experiments and statistical analysis on standard benchmark functions certify the superiority of the proposed method over the other algorithms. The efficacy of the proposed hybrid algorithm is promising and can be used in several real-life engineering and management problems.

Keywords Meta-heuristics · Imperialistic competition algorithm · Harmony search · Simulated annealing · Optimization

Introduction

Meta-heuristic algorithms are assumed as powerful tools for solving various optimization problems. They have been widely used to deal with complicated problems in different areas of science and engineering (Goldberg 1989). There are several methods to solve optimization problems, a great deal of them based on traditional mathematical programming (Goldberg 1989). These methods require several conditions that cannot be met by a lot of real-world optimization problems (Goldberg 1989). Moreover, a lot of real-world problems are hard to solve using limited amount of time resources.[1] Although traditional mathematical programming methods are good at dealing with small instances of such problems, they are not well posed for large-scale instances of NP-hard problems.

Traditional optimization methods usually require some assumptions to work properly. These assumptions are about continuity of solution space and differentiability of the objective function. Fortunately, meta-heuristic methods are not limited by such assumptions. We are usually seeking an optimum or even proper values for non-differentiable measurement functions in real-world problems. Moreover, calculating partial derivatives may be so hard and computationally expensive for multivariate functions in real-life problems. Searching discrete solution space is another challenge in real-world problems. All the aforementioned issues are assumed to be proper reasons to pursue the researchers to develop and apply meta-heuristic algorithms for real-life and NP-hard problems.

There are lots of meta-heuristic algorithms, which work with different mechanisms and inspired from various natural, social, economical, political, cultural, mechanical, and physical concepts. Among them, Tabu search (TS) (Glover 1989, 1990), simulated annealing (SA) (Kirkpatrick et al.

✉ Amir-Reza Abtahi
amir_abtahi@yahoo.com; abtahi@khu.ac.ir

Afsane Bijari
afsane.bijari@gmail.com

1 Department of Information Technology Management, Kharazmi University, Tehran 15999-64511, Iran

[1] These types of problems are called non-deterministic polynomial hard (NP-hard), which cannot be solved using a polynomial algorithm. In such problems, the required time for solving the problem increases exponentially as the dimension of the problem grows.

1983), genetic algorithm (GA) (Holland 1975), particle swarm optimization (PSO) (Eberhart and Kennedy 1995), ant colony optimization (ACO) (Colorni et al. 1991), artificial bee colony (ABC) (Karaboga and Basturk 2008), harmony search (HS) (Geem et al. 2001), and imperialistic competition algorithm (ICA) (Atashpaz-Gargari and Lucas 2007) have been successfully applied to various engineering, and management problems.

HS and ICA have been brilliantly used to solve various real-world problems (Geem 2006; Niknam et al. 2011; Talbi 2009; Kaveh and Talatahari 2010; Mohammadi-Ivatloo et al. 2012). The HS algorithm is based on the process of improvisation in the musical performance and ICA is inspired by economical, social, and cultural evolution of imperialists and their colonies. Although the ICA algorithm has successfully been applied to various problems, it suffers from lack of proper information flow between various empires. This will cause a considerable performance reduction of the ICA whenever the solution space contains several high-quality regions (i.e., multi-modal objective functions). In this paper, a hybrid algorithm is proposed based on ICA, SA, and HS to overcome the aforementioned pitfall. The proposed hybrid algorithm serves several properties which improve its performance. To test the performance of the proposed hybrid algorithm, several well-known and standard benchmark instances are chosen and a comparative analysis is accomplished. The comprehensive experiments on the well-known benchmark instances certify the superiority of the proposed hybrid algorithm in comparison with the other meta-heuristic algorithms.

The remaining part of this paper is organized as follows. A literature of recent applications of meta-heuristics methods, including GA, SA, ICA, and HS, are represented in "Literature on recent applications of meta-heuristics". The next section deals with the "Basics and fundamental of ICA, SA, and HS". The proposed hybrid algorithm is developed in the "Proposed algorithm". Benchmark instances and experimental results are presented and discussed in "Benchmark functions and experimental results", respectively. The conclusion remarks are drawn in "Conclusions".

Literature on recent applications of meta-heuristics

This section aims to take a short look at various applications of meta-heuristic methods for solving a broad range of engineering problems. Meta-heuristic methods have been widely used in almost all areas in engineering. There are many real-life engineering problems, including structural design, shape optimization, building design, energy-efficient design, scheduling, planning, and many others, which have been addressed using meta-heuristics (Yang 2010; Chen and Wang 2004). We focus on the applications of GA, SA, ICA, and HS to make a brief sense of their success story in engineering problems.

Applications of genetic algorithm (GA)

Genetic algorithm was initially proposed by Holland (1975). GA has been widely used to solve various optimization problems in different branches of engineering, and it is known as a successful tool to deal with complex real-world problems. Das et al. (2012) employed GA to sample the optimal set of local regions from which an optimal feature set can be extracted. The proposed method has been evaluated using data set of handwritten Bangla digits, and the results certified that the GA-based region sampling outperforms several existing methods. Bateni et al. (2012) proposed a hybrid GA—finite difference to estimate soil thermal properties using land surface temperature. It has been shown that the proposed method was capable of accurately estimating soil thermal properties. Qu et al. (2013) proposed an improved version of GA used for global path planning of multiple mobile robots. The proposed method resulted in an optimal or near-optimal collision-free path, and simulations demonstrated the efficiency of the proposed method. Yang and Koziel (2011) provided a detailed description of application of GA for various engineering problems such as estimation of input parameters in environmental emergency modeling, simulation of behavior of concrete materials, and network optimization. Rabbani et al. (2016) proposed a multi-objective version of GA for line balancing in assembly lines. The proposed method compared with a particle swarm optimization method (PSO). The operation of GA outperforms PSO in many large-scale problems. Mehmanpazir and Asadi (2016) used GA as a rule-filtering tool and tuning mechanism for membership function in designing an evolutionary fuzzy expert system. The results showed that the proposed expert system with GA provided more accuracy in stock price forecasting problems.

Applications of simulated annealing (SA)

Simulated annealing is a local search method inspired by the physical annealing process (Kirkpatrick et al. 1983). SA was employed to address various optimization problems in different branches of engineering. Zhang et al. (2013) applied SA to the robot path planning problem. The study focused on three classes of paths, namely polyline, Bézier curve, and spline interpolated curve. The experimental results showed the capability of SA in finding near-optimal solutions. Saraiva et al. (2011) formulated the scheduling problem of generator maintenance actions as a mixed integer optimization problem. The objective function was to minimize the operation cost along the scheduling period plus a penalty on energy not supplied. SA was used to solve the problem. The empirical study demonstrated that the SA was able to find low-cost maintenance schedules with short computation times.

Tsang and Wiese (2010) used SA to predict RNA secondary structure. The prediction accuracy of the proposed method, SARNA-Predict, was compared to eight state-of-the-art RNA prediction algorithms. Experiments on 33 individual known structures from 11 RNA classes demonstrated that SARNA-Predict can outperform the eight state-of-the-art RNA prediction algorithms in terms of prediction accuracy.

Jafari and Salmasi (2015) applied SA in the nurse scheduling problem. The results of experiments showed that SA algorithm provides more accurate solutions. Also, the applied SA offers meaningfully better solutions in a reasonable time compared to other methods.

Applications of harmony search (HS)

The harmony search algorithm was inspired by the musical process of searching for a perfect state of harmony (Geem et al. 2001). HS has been widely used to deal with complex real-world optimization problems (Geem 2006). Del Ser et al. (2012) applied HS to the spectrum channel allocation problem. The experimental results certified that the proposed method achieves near-optimum spectral channel assignments at a low computational cost, and also it outperformed genetically inspired allocation algorithms for the set of simulated scenarios. Diao and Shen (2012) used HS to deal with feature selection problem in which a subset of relevant features is selected for model construction. The superiority of the proposed approach was shown over several different methods, including hill climbing, GA, and particle swarm optimization (PSO).

Imperialistic competition algorithm (ICA)

Imperialistic competition algorithm was originally inspired from the political behavior of imperialists (Atashpaz-Gargari and Lucas 2007). Yousefi et al. (2012) utilized ICA to optimize a cross-flow plate fin heat exchanger. The ability of ICA to optimize the total weight and total annual cost was demonstrated using a case study. Moreover, in comparison with GA, ICA showed lower computational complexity and higher accuracy. Niknam et al. (2011) applied ICA to a K-means clustering algorithm. The experimental results on several datasets demonstrated the performance of the proposed method over several clustering algorithms, including, ACO, PSO, SA, GA, TS, honey bee mating optimization, and K-means clustering.

Basics and fundamental of ICA, SA, and HS

As the proposed hybrid method is based on ICA, SA, and HS, so in this section the basics and fundamental of these methods are briefly revisited.

Concept of imperialistic competition algorithm

Imperialistic competition algorithm was originally inspired from the political behavior of imperialists (Atashpaz-Gargari and Lucas 2007). ICA starts with an initial population, which are named countries. Some of the best countries in the initial population are selected as imperialists, and the remaining countries form the colonies of these imperialists. Each imperialist with its colonies is called an empire.

After creation of the initial empires, colonies start moving toward their relevant imperialists. This movement takes place based on a process called assimilation policy. Figure 1 shows a colony movement toward its relevant imperialist.

Figure 1 depicts the process of assimilation, where $x^{(t)}$ is the position of a colony at time step t; $y^{(t)}$ the position of the corresponding imperialist of x at time step t; θ, the deviation degree, is a random number with uniform distribution on the interval $[-\gamma,\ \gamma]$; $\mathrm{d}(x^{(t)}, y^{(t)})$ the distance between $x^{(t)}$ and its imperialist position at time step t, i.e., $y^{(t)}$; l, the distance between $x^{(t)}$ and $x^{(t+1)}$, is a random variable with uniform distribution on the interval $[0, \beta \times \mathrm{d}(x^{(t)},\ y^{(t)})]$.

As a result of the assimilation process, each colony moves l units with a θ angle deviation from the vector which connects the colony to its imperialist. Since in each empire the imperialist is the most powerful country, during the assimilation process a country may change its role from

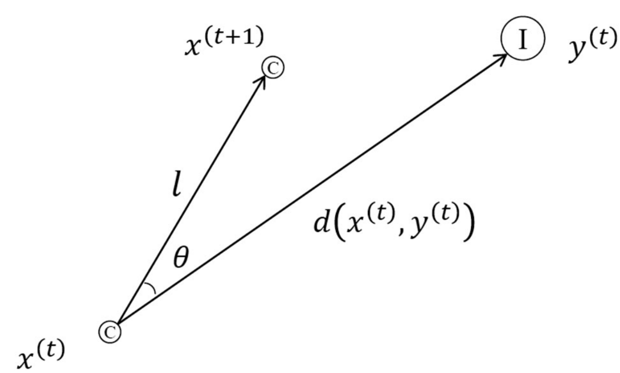


Fig. 1 Movement of a colony toward its imperialist during the assimilation process

imperialist to colony, and vice versa. It should be noted that the power of a country is calculated based on its fitness function value. For any optimization problem, the fitness function is defined to show the quality of a given solution (country). In the next step imperialistic competition takes place. First, the total power of each empire is calculated as follows:

$$\text{Power }(E) = \text{Fitness }(\text{imperialist}_E) + \delta \times \sum_{c \in E} \text{fitness }(c), \tag{1}$$

where E is an empire, c a colony of E, and δ a real number in the interval $(0, 1)$. The empire with the lowest power value and consequently its weakest colony is chosen. Then, all empires compete in a process to take possession of the chosen colony. Each empire has a chance to win the imperialistic competition proportional to its power. It should be noted that the imperialistic competition gradually results in a decrease in the power of weaker empires and an increase in the power of more powerful ones. When an empire lost all its colonies, its imperialist is assigned to another empire as a colony.

Revolution strategy, as another step of ICA, is applied to a certain percentage of countries belonging to each empire. In this strategy, some cultural, social, and political elements of a country, i.e., components of a solution, are changed randomly. The algorithm continues till a termination condition is met. In each iteration assimilation, imperialistic competition and revolution strategy are used. The flowchart of ICA algorithm is depicted in Fig. 2.

Concept of simulated annealing

Simulated annealing is a local search method inspired by the physical annealing process (Kirkpatrick et al. 1983). SA, while exploring solution space, offers the possibility to accept less fitted solutions in a controlled way to escape from local optima. More precisely, let $f(x)$ be the objective function for a minimization problem, x_t the solution at time step t, $f(x_t)$ the value of the objective function for x_t, and $N(x_t)$ the set of all immediate neighbors of x_t. Also, let $y \in N(x_t)$ and $\Delta(y, x_t) = (f(y) - f(x_t))$. The SA algorithm selects the next solution, i.e., x_{t+1}, according to the following formula:

$$x_{t+1} = \begin{cases} y & \Delta £ 0 \\ y & \Delta > 0 \text{ and } U(0,1) < e^{\frac{-\Delta(y,x_t)}{T_t}} \\ x_t & \Delta > 0 \text{ and } U(0,1) > e^{\frac{-\Delta(y,x_t)}{T_t}} \end{cases}, \tag{2}$$

where $U(0, 1)$ is a uniformly distributed random number in the interval $(0, 1)$, and T_t is the temperature at time step t. If $f(y) \leq f(x_t)$, i.e., y is a better solution in comparison to x_t, then $x_{t+1} = y$. Otherwise, y could also be accepted as x_{t+1}, with a probability of $p = e^{\frac{-\Delta(y,x_t)}{T_t}}$. The acceptance probability p is influenced by two factors, $\Delta(y, x_t)$ and T_t. Smaller values of $\Delta(y, x_t)$ induce greater acceptance probabilities and, therefore, more chance for y to be accepted as the value of x_{t+1}. Moreover, higher values of T_t give higher acceptance probability and, therefore, more chance for y to be accepted as the value of x_{t+1}. The temperature parameter T_t is controlled by a cooling schema. It has been shown that there are theoretical cooling schedules which guarantee asymptotic convergence toward the optimal solution, although they require infinite computing time steps (Kirkpatrick et al. 1983). In practice finite computing time steps are preferred even if they do not guarantee convergence to an optimum. Higher temperature values in the first iterations of the SA make it possible to explore different regions of the search space, and lower values in the final iterations of the algorithm make it more selective to neighborhoods of good solutions.

Concept of harmony search

HS algorithm was inspired by the musical process of searching for a perfect state of harmony (Geem et al. 2001). In HS, each potential solution for the problem is coded as a feature vector named harmony and the goal is to find a global optimum as determined by a fitness function. HS takes advantage of a limited subset of successful experiences, i.e., the fittest solutions. These harmonies are gathered in a memory called harmony memory (HM). In each iteration of HS a new harmony is generated. The new generated harmony, then, is compared with the worst harmony in HM based on the fitness function. If the new harmony dominates the worst harmony, it will be substituted for the worst one.

HM (*i, j*) shows the *j*th component of the *i*th harmony in HM. Each harmony is presented as a *d*-dimensional vector. To create a new harmony, all *d* components or features of a new harmony, called "*H*", should be computed. HS employs three strategies to compute each component of *H*, i.e., *H* (*j*). As the first strategy, one of the harmonies in HM is selected randomly, e.g., the *i*th harmony, and then the value of HM (*i,j*) is assigned to *H* (*j*). As the second strategy, one of the harmonies in HM is selected randomly, e.g., the *i*th harmony, and then an adjacent value of HM (*i*, *j*) is assigned to *H* (*j*). The adjustment value is conducted using a bandwidth bw, which is a parameter of the algorithm. Finally, as the third strategy, a random value from the possible range is used as *H* (*j*). To compute each *H* (*j*), HS uses one of these three strategies and therefore it needs to decide on one of them. HS uses two parameters named harmony memory consideration rate (HMCR) and pitch adjustment rate (PAR) to decide on each strategy. Figure 3

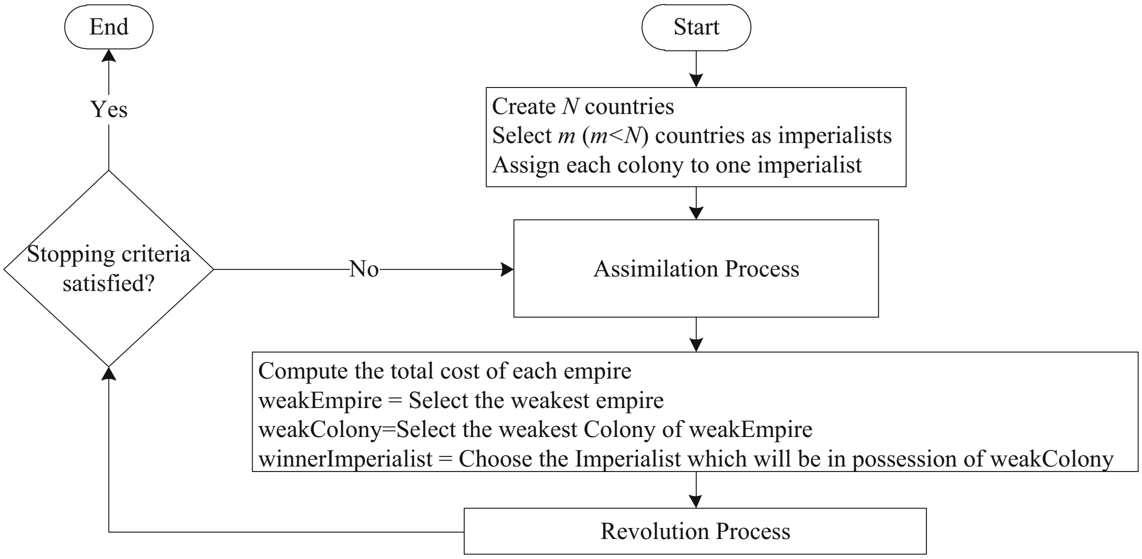


Fig. 2 Flowchart of ICA

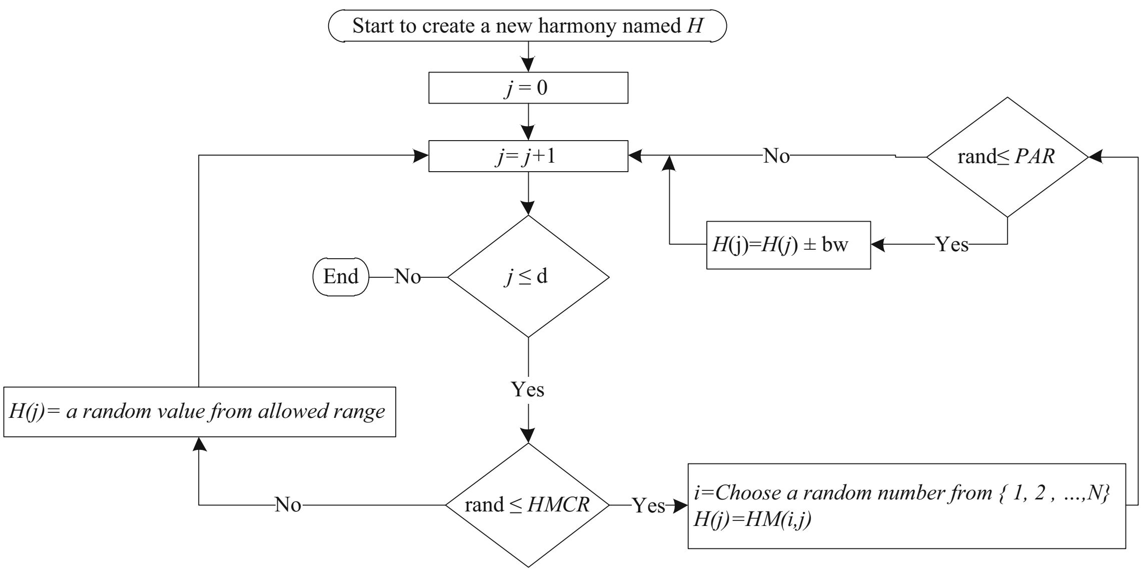


Fig. 3 Process of creation of a new harmony in the HS algorithm

depicts the process of creation of a new harmony in the HS algorithm.

Proposed algorithm

Exploration and exploitation are two essential concepts of meta-heuristic algorithms. The aim of exploration is to find new solutions in the search space, so it takes advantage of random steps to discover solutions from different regions of the search space. On the other hand, the aim of exploitation is to create new solutions based on the solutions which have previously been found during the search process. If a meta-heuristic algorithm just focuses on exploration, it acts like a random search algorithm which is a highly inefficient way to search the solution space. On the other hand, if a meta-heuristic algorithm just focuses on exploitation, it is not able to efficiently discover and search all regions of solution space. So, it may be involve in the local optima. Therefore, any meta-heuristic algorithm needs to make a balance between exploration and exploitation properties to find optimal solutions without searching the whole solution space.

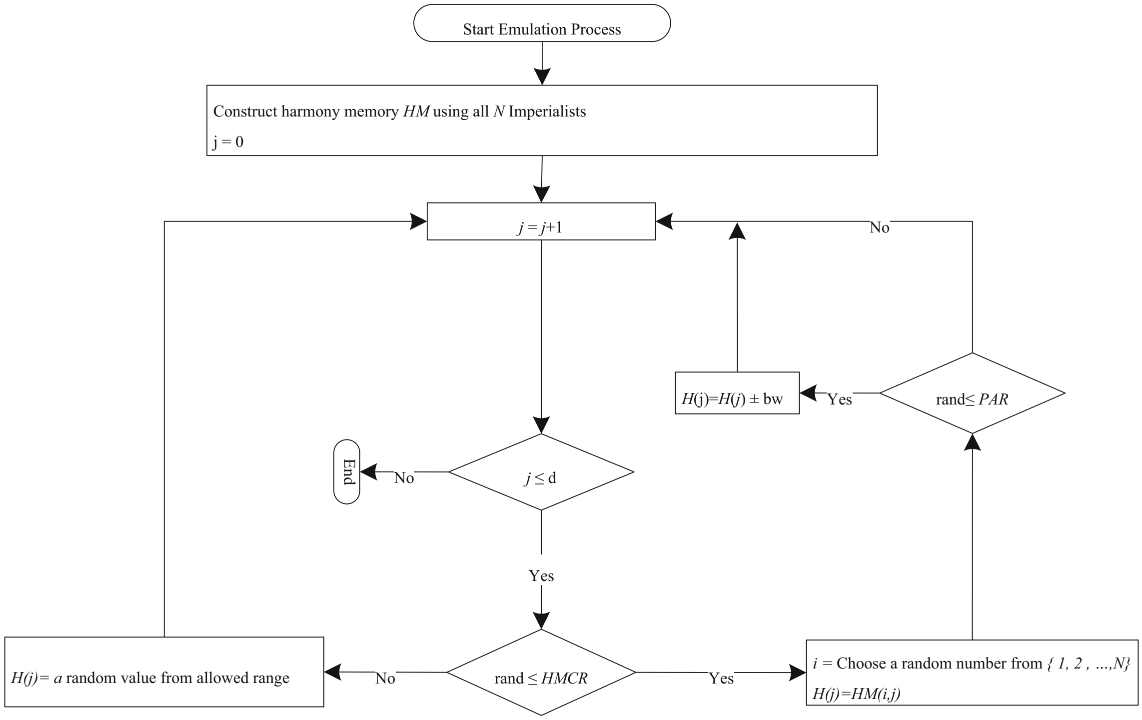


Fig. 4 The emulation process

Imperialistic competition algorithm uses revolution strategy to explore different regions of the search space. In addition, during the assimilation process in each iteration of IDA, the exploitation takes place inside each empire. Meanwhile, the imperialistic competition process, the weakest colony of the weakest empire, may move to another empire. Although this colony may help to diffuse information between empires, it usually is one of the worst countries (solutions) which has a negligible effect on the search process. Hence, ICA never exploits high-quality solutions from different empires to construct new solutions.

In the real world, sometimes developing countries emulate social, political, and economical characteristics of advanced countries. This strategy can be used in ICA to tune its exploitation ability. In the proposed hybrid algorithm, the harmony creation process of the HS algorithm is applied to use high-quality solutions (countries) for constructing a new solution. More formally, the proposed hybrid algorithm adds an emulation step to the ICA. Figure 4 represents the emulation process.

In the emulation step, first, imperialists assemble to construct harmony memory. Next, the harmony creation process is used to create a new harmony (country) H. Then, an empire is selected at random, and one of its colonies is randomly chosen. Let us denote this colony by C. After that, C and H are compared based on the fitness function. If H is a better solution in comparison with C, it will be replaced by C. Otherwise, H could also be accepted as a substitute for C, with a probability of $p = e^{\frac{-\Delta(H,C)}{T_t}}$, where T_t is defined by a cooling schema, and $\Delta(H,C)$ is the objective difference for H and C. Figure 5 illustrates the flowchart of the proposed hybrid algorithm.

The proposed hybrid algorithm uses a modified assimilation process to avoid trigonometry calculation and improve the computational efficiency. Let x be the position of a country C and y the position of the imperialist of C. Also, assume that the new position of C is illustrated by x'. The ith component of x', i.e., x'_i, is calculated as follows:

$$x'_i = x_i + \beta \times U(0,1) \times (y_i - x_i), \tag{3}$$

where x_i and y_i are the ith components of x and y, respectively; $\beta = 2$ and $U(0,1)$ is a uniformly distributed random number in the interval $(0,1)$.

Benchmark functions and experimental results

To evaluate the capability and efficiency of the proposed hybrid algorithm, its performance is compared with ICA, HS, and GA on a set of standard benchmark instances including ten well-known benchmark functions.

Benchmark functions

The standard benchmark functions are as follows.

Ackley function This function is defined as follows:

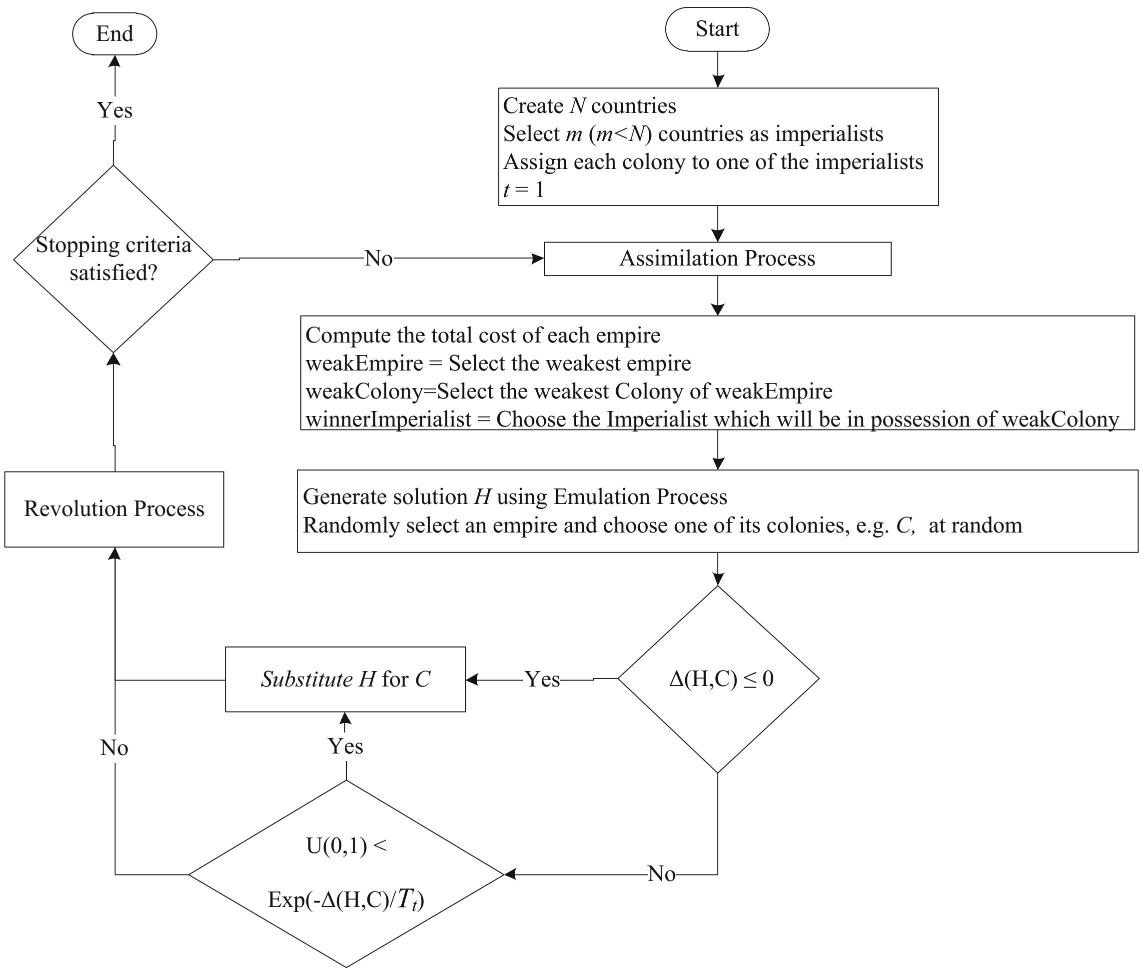


Fig. 5 Flowchart of the proposed hybrid algorithm

$$- 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}cos(2\pi x_i)\right) + 20 + e, \quad (4)$$

where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-32 \leq x(i) \leq 32$ (Fig. 6).

Griewank function This function is defined as follows:

$$\frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (5)$$

where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-600 \leq x(i) \leq 600$ (Fig. 7).

Rastrigin function This function is defined as follows:

$$\sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i) + 10), \quad (6)$$

where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-5.12 \leq x(i) \leq 5.12$ (Fig. 8).

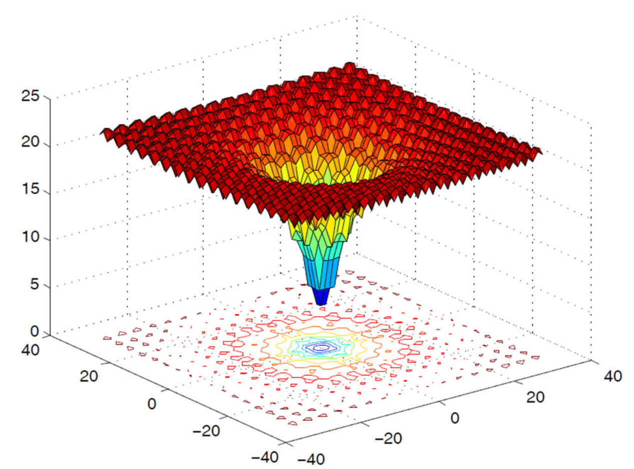


Fig. 6 The Ackley function

Rosenbrock function This function is defined as follows:

$$\sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), \quad (7)$$

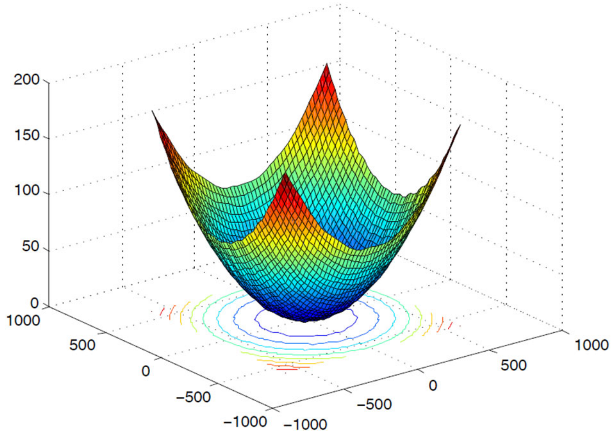


Fig. 7 The Griewank function

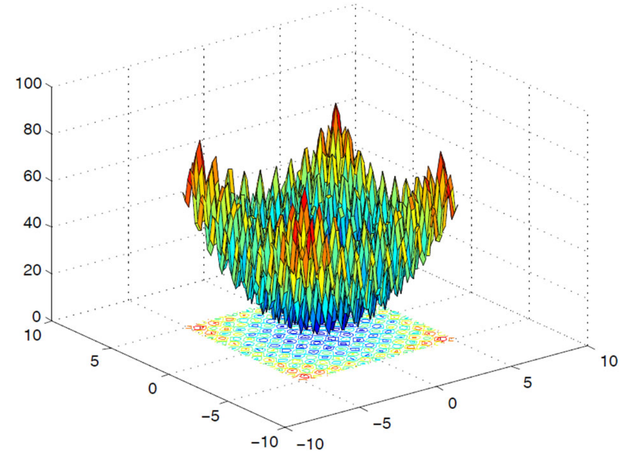


Fig. 8 The Rastrigin function

where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-30 \leq x(i) \leq 30$ (Fig. 9).

Rotated hyper-ellipsoid function This function is defined as follows:

$$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2, \tag{8}$$

where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-100 \leq x(i) \leq 100$ (Fig. 10).

Schwefel's problem This function, is defined as follows:

$$f(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|, \tag{9}$$

where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-10 \leq x(i) \leq 10$ (Fig. 11).

Schwefel's function This function, is defined as follows:

$$418.9829n - \sum_{i=1}^{n} (x_i sin(\sqrt{(|x_i|)})), \tag{10}$$

where the global optimum $x^* = (420.9687, \ldots, 420.9687)$ and $f(x^*) = 0$ for $-500 \leq x(i) \leq 500$ (Fig. 12).

Sphere function This function is defined as follows:

$$f(x) = \sum_{i=1}^{n} x^2(i), \tag{11}$$

where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-100 \leq x(i) \leq 100$ (Fig. 13).

Six-hump camel-back function This function is defined as follows:

$$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 + 4x_2^2 + 4x_2^4, \tag{12}$$

where the global optimum $x^* = (-0.08983,\ 0.7126)$ and $f(x^*) = -1.0316285$ for $-5 \leq x(i) \leq 5$ (Fig. 14).

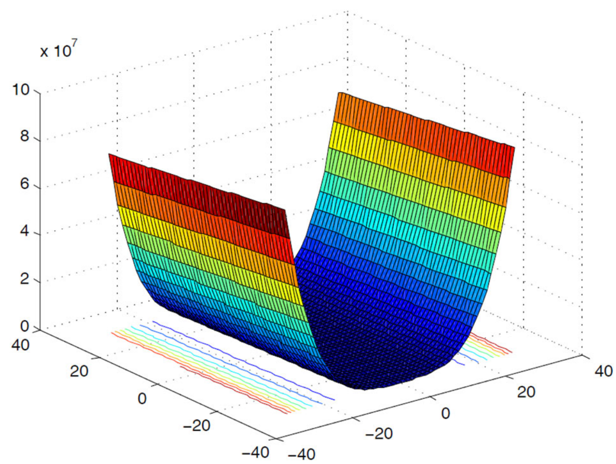


Fig. 9 The Rosenbrock function

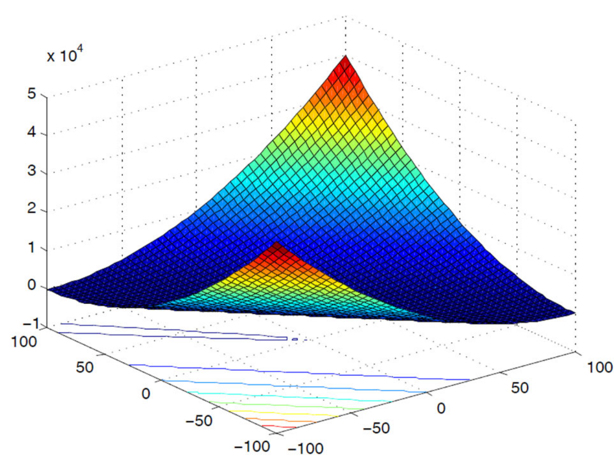


Fig. 10 The rotated hyper-ellipsoid function

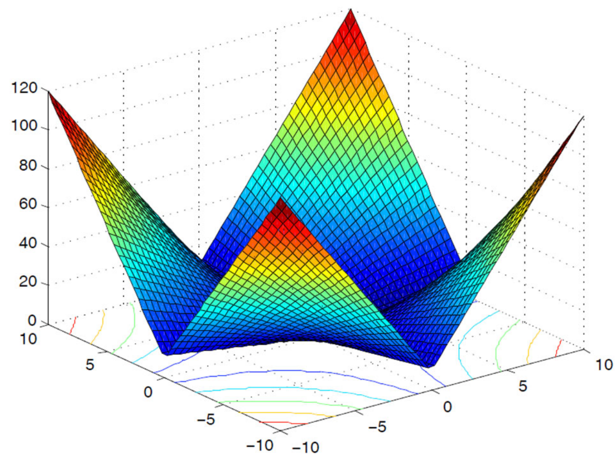


Fig. 11 The Schwefel's problem

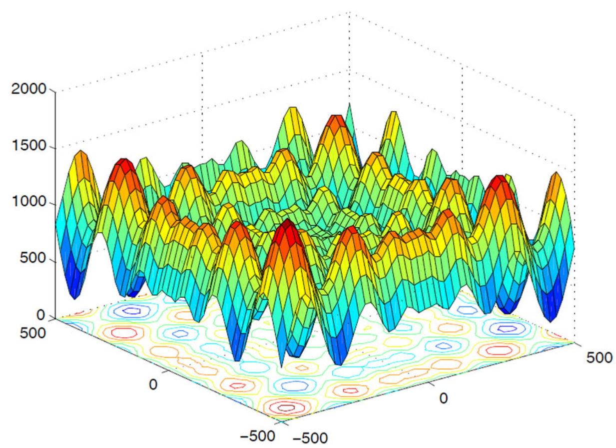


Fig. 12 The Schwefel's function

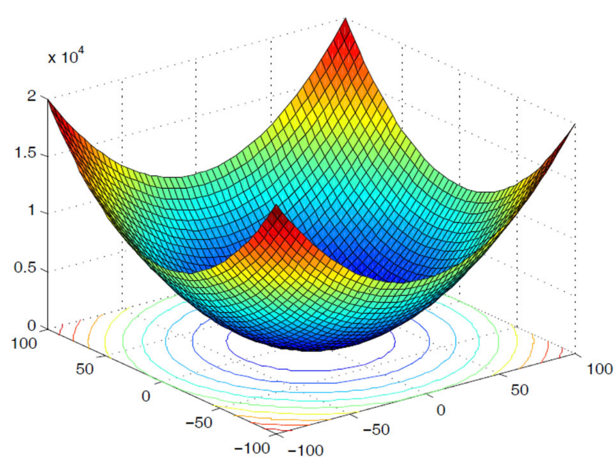


Fig. 13 The sphere function

Step function This function, is defined as follows:

$$\sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2, \tag{13}$$

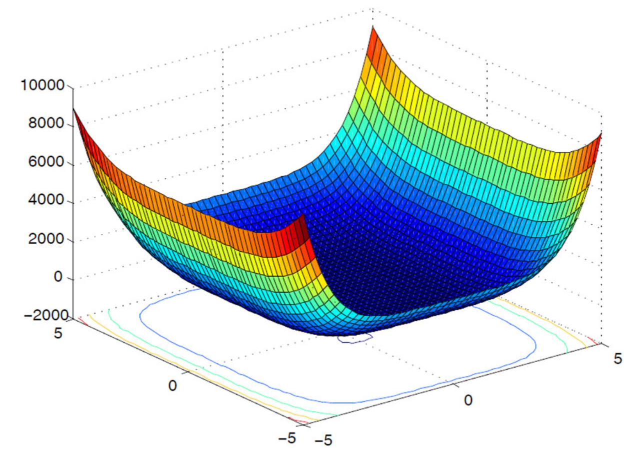


Fig. 14 The six-hump camel-back function

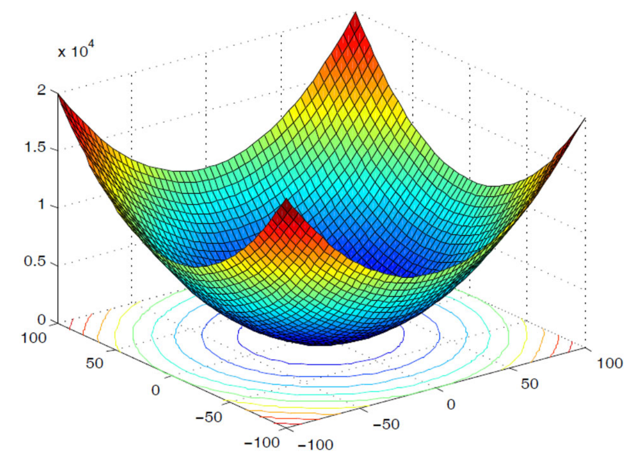


Fig. 15 The Step function

Table 1 Parameters of genetic algorithm

Population size	Number of iteration	Mutation rate	Crossover rate
100	200	0.01	0.90

Table 2 Parameters of harmony search algorithm

Number of iteration	HMCR	PAR	BW	HM size
20,000	0.9	0.1	0.01	5

Table 3 Parameters of ICA

Population size	Number of empires	Revolution rate	β	δ
100	10	0.01	2	0.1

Table 4 Parameters of the proposed hybrid algorithm

Population size	Number of empires	Revolution rate	β	δ	HMCR	PAR	BW
100	10	0.01	2	0.1	0.90	0.1	0.01

Table 5 Evaluation and comparison of the proposed algorithm with other methods

Function	Value	HS	GA	ICA	HSICA
Rastrigin	A	73.86800065	148.8863622	159.3491324	47.06123053
	B	52.87033715	100.5309011	88.65225145	22.99265435
	V	72.52436980	333.2909468	889.4490264	122.2813194
Step	A	1637.33	4318.1	305.185	27.55
	B	859	2048	54	9
	V	109931.5589	672372.4523	28936.53344	124.7211055
Griewank	A	15.74962645	39.29549023	2.386376350	1.117339723
	B	8.512656213	20.09827017	1.278180068	0.968977325
	V	7.831053920	63.33536810	0.689018954	0.004798309
Rotated hyper-ellipsoid	A	31033.64331	92431.90391	3413.176126	265.8344968
	B	16087.52566	44923.85352	559.4225231	63.48156505
	V	47934812.31	311528733.4	4328196.864	15979.10728
Six-hump camel-back	A	−1.0316284532	−1.0314847966	−1.0316284535	−1.0316284535
	B	−1.0316284535	−1.0316280965	−1.0316284535	−1.0316284535
	V	2.56E−19	3.23E−08	1.06E−29	1.04E−29
Schwefels problem	A	16.45869234	23.28643578	130.1712082	1.765889143
	B	11.12122981	16.07921931	90.82060767	0.635371822
	V	3.657980143	8.677513739	182.1192711	0.440279650
Schwefels problem	A	1788.095100	−96708.55994	5774.509383	1938.845204
	B	1206.869925	−122048.6442	3261.586477	977.4971647
	V	77826.43171	64928436.913	535486.4149	139629.2107
Ackley	A	7.712093734	10.90869984	16.65335679	3.935734989
	B	6.338061317	9.204203841	12.22135283	2.258496104
	V	0.265323799	0.489400176	1.500157784	1.111832527
Sphere	A	1664.487098	4201.984681	147.6140975	13.24124148
	B	884.0886831	2331.556985	27.36464552	2.906834961
	V	103804.7484	767589.5080	27.36464552	57.76724060
Rosenbrock	A	201565.93392	1766467.546834	20761.85366	4817.158306
	B	58634.209542	602588.6213100	20752.01576	920.0986773
	V	5026920423.1	485809094885.7	24.60486397	6111380.957

A average, *B* best, *V* variance

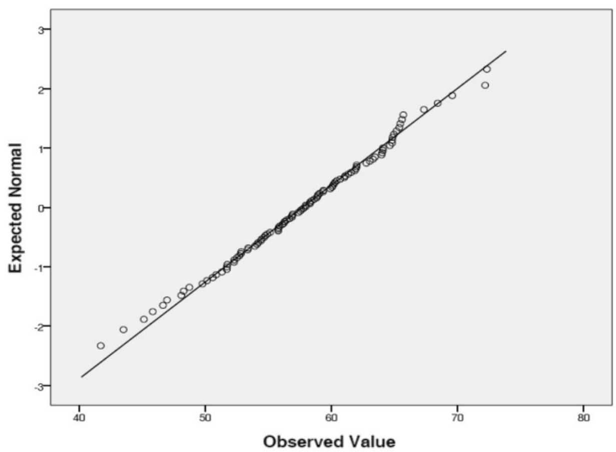
where the global optimum $x^* = (0, \ldots, 0)$ and $f(x^*) = 0$ for $-100 \le x(i) \le 100$ (Fig. 15).

Experimental results

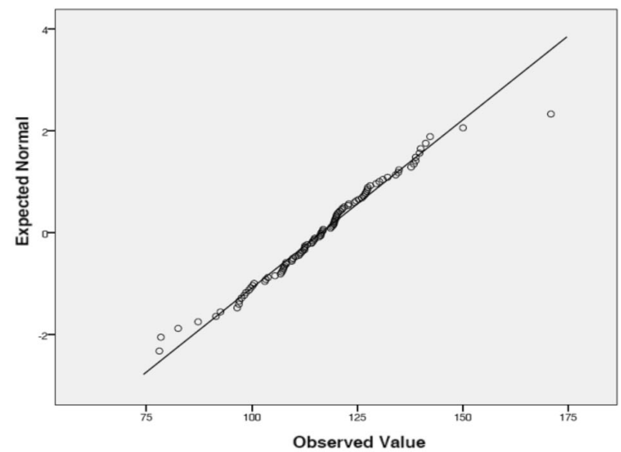
In this section, the comparative results of application of the proposed hybrid algorithm, GA, HS, and ICA on the standard benchmark instances are presented.

Experimental study using the standard benchmark functions

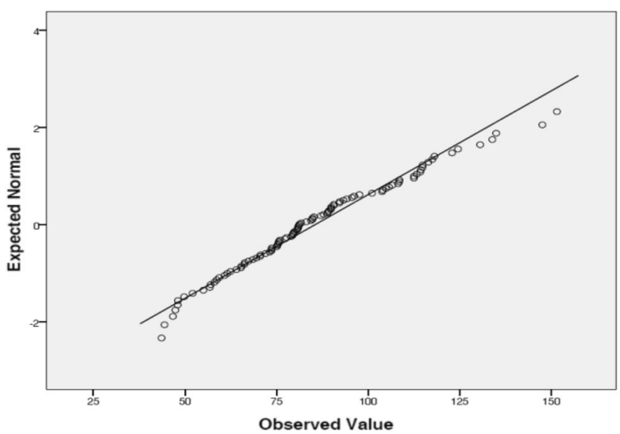
The performance of the proposed hybrid algorithm is compared with GA, HS, and ICA. To make an analogous condition for algorithms, similar parameters of these algorithms are similarly set. For example, GA, ICA, and the proposed hybrid algorithm use population size equal to 100. Table 1 shows the parameters of GA. Single-point



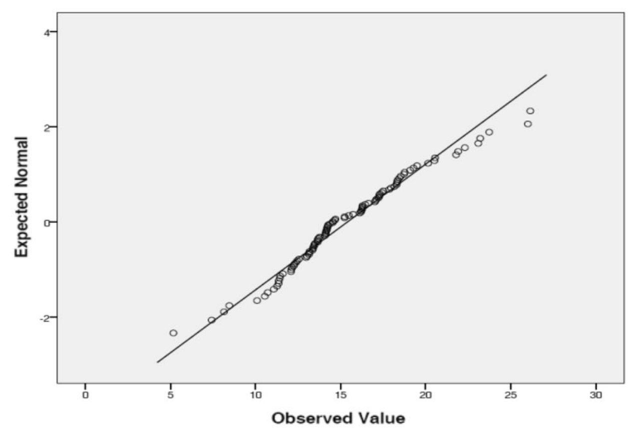
(a) HS Q-Q Plot



(b) GA Q-Q Plot



(c) ICA Q-Q Plot



(d) Hybrid Algorithm Q-Q Plot

Fig. 16 Q–Q plot for normality test

Table 6 Shapiro–Wilk test for normality

Variable	Statistic	Degree of freedom	p value
HS	0.994	100	0.938
GA	0.982	100	0.190
ICA	0.975	100	0.055
HSICA	0.976	100	0.069

crossover and roulette wheel selection function were used in GA.

It should be noted that GA, ICA, and the proposed hybrid algorithm evaluate n solutions in each iteration, where n is the population size. Hence, the same number of iterations, M, were used for them. HS, on the other hand, just evaluates one solution at each iteration. So, the number of iterations for HS were set equal to $n \times M$, where M is the number of iterations for GA, ICA, and the proposed hybrid algorithm. Tables 2, 3, and 4 show the parameters of HS, ICA, and the proposed hybrid algorithm, respectively.In addition, in the proposed hybrid algorithm, the geometric cooling schedule was used. In the geometric cooling schedule, the temperature decreases as follows:

$$T_{k+1} = \gamma \times T_k, \tag{14}$$

where γ, the cooling factor, is assumed to be a positive constant less than one. In this paper, γ and T_0 was assumed to be 0.95, and 100, respectively.

All benchmark instances except for six-hump camel-back function can be defined for arbitrary dimensions. Six-hump camel-back function is defined only for two dimensions. In experiments, 50-dimensional versions of benchmark functions were studied. Moreover, to achieve statistically reliable results, each algorithm was run 100 times on each problem and the best, the average, and the variance of solutions were reported in Table 5.

Table 7 ANOVA test results

	Sum of squares	Degree of freedom	Mean square	F	p value
Between groups	552882.463	3	184294.154	888.868	0.000
Within groups	82104.982	396	207.336		
Total	634987.445	399			

Table 8 Games and Howell's pairwise comparisons test

Variable	$\mu_{\mathrm{HSICA}} - \mu_{\mathrm{Variable}}$	Std. error	p value	95 % Confidence interval	
				Lower bound	Upper bound
HS	−42.39	0.721	0.000	−44.18	−40.44
GA	−101.1	1.560	0.000	−105.2	−97.04
ICA	−70.07	2.372	0.000	−76.26	−63.87

It can be concluded from Table 5 that the proposed hybrid algorithm is superior to HS, ICA, and GA on benchmark instances.

Statistical analysis

Although the measures presented in Table 5 certifies the relative dominance of the proposed hybrid algorithm, a statistical analysis was also supplied to verify the meaningful superiority of the proposed algorithm. Each method was executed 100 times independently. The outcome of the execution of each algorithm is assumed to be a random variable. A quantile–quantile plot (Q–Q plot) was supplied to determine whether parametric or non-parametric tests were appropriate. A Q–Q plot is a graphical data analysis technique for assessing whether the distribution of data follows a particular distribution or not. The Q–Q plot was applied to examine the normality of each random variable. Figure 16 depicts the Q–Q plot for the data resulting from the application of all methods for finding the minimum of Rastrigin's function.

The results support the normality of all variables under consideration. Moreover, Shapiro–Wilk test was used for tests of the normality of the population (Shapiro and Wilk 1965). Table 6 shows the results of Shapiro–Wilk test for all variables on finding the minimum of Rastrigin's function.

It can be concluded from both of Table 6 and Fig. 16 that there are not enough evidences to reject the normality of variables. Therefore, parametric tests were used to compare the performance of the algorithms. To examine whether there is a significant difference between the performance of the algorithms, the analysis of variance (ANOVA) was used. As depicted in Table 7, there is enough evidence to reject the null hypothesis, i.e., the equal means of outcomes of HS, GA, ICA, and the proposed algorithm.

Post hoc Games–Howell tests were carried out to further investigate how the competent methods differed from each other. Table 8 presents the results of Games and Howell's pairwise comparison test.

The confidence level, in all experiments, was set to 95 %. It can be inferred from Table 8 that there are significant differences between the mean of the proposed algorithm and the other methods. In addition, the confidence intervals show that the proposed algorithm provides solutions which are closer to the optimum solution. This certifies the superiority of the proposed algorithm in comparison with GA, HS, and ICA.

Conclusions

In this paper, a hybrid meta-heuristic algorithm was proposed. The proposed algorithm inherits the advantage of the process of harmony creation in harmony search algorithm to improve the exploitation phase of the ICA algorithm. In addition, the proposed algorithm uses simulated annealing to maintain a balance between the exploration and exploitation phases. To examine the efficiency and applicability of the proposed algorithm, several standard benchmark functions were used to compare the results of the proposed method with GA, HS, and ICA. Statistical analysis was supplied to investigate the performance of the proposed hybrid algorithm in comparison with GA, HS, and ICA. The experimental results and the statistical study certified the superiority of the proposed method over GA, HS, and ICA. The comprehensive experimental study of the proposed method demonstrates its promising efficiency and recommends it as a powerful tool to deal with real-life engineering and management problems.

Acknowledgments We thank the Editor-in-Chief Prof. Sadigh Raissi handling our paper and providing us valuable advice. We appreciate the constructive comments from the Area Editor and the anonymous reviewers. We have carefully considered all of the suggestions made by the reviewers in the revision. The quality of our paper has benefitted considerably from the comments made by the reviewers. All modifications and corrections have been highlighted in the revised report.

References

Atashpaz-Gargari E, Lucas C (2007) Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE congress on evolutionary computation, p 4661–4667

Bateni SM, Jeng DS, Naeini SM (2012) Estimating soil thermal properties from sequences of land surface temperature using hybrid genetic algorithm–finite difference method. Eng Appl Artif Intell 25(7):1425–1436

Chen SH, Wang PP (2004) Computational intelligence in economics and finance. In: Chen SH, Wang PP (eds) Computational intelligence in economics and finance. Springer, Berlin, Heidelberg, pp 3–55

Colorni A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life, vol 142. pp 134–142

Das N, Sarkar R, Basu S, Kundu M, Nasipuri M, Basu DK (2012) A genetic algorithm based region sampling for selection of local features in handwritten digit recognition application. Appl Soft Comput 12(5):1592–1606

Del Ser J, Matinmikko M, Gil-López S, Mustonen M (2012) Centralized and distributed spectrum channel assignment in cognitive wireless networks: a harmony search approach. Appl Soft Comput 12(2):921–930

Diao R, Shen Q (2012) Feature selection with harmony search. IEEE Trans Syst Man Cybern Part B (Cybern) 42(6):1509–1523

Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, vol 1. pp 39–43

Geem ZW (2006) Optimal cost design of water distribution networks using harmony search. Eng Optim 38(03):259–277

Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68

Glover F (1989) Tabu search—part I. ORSA J Comput 1(3):190–206

Glover F (1990) Tabu search—part II. ORSA J Comput 2(1):4–32

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addion Wesley 1989:102

Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. U Michigan Press, Ann Arbor

Jafari H, Salmasi N (2015) Maximizing the nurses' preferences in nurse scheduling problem: mathematical modeling and a meta-heuristic algorithm. J Ind Eng Int 11(3):439–458

Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. Appl Soft Comput 8(1):687–697

Kaveh A, Talatahari S (2010) Optimum design of skeletal structures using imperialist competitive algorithm. Comput Struct 88(21):1220–1229

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

Mehmanpazir F, Asadi S (2016) Development of an evolutionary fuzzy expert system for estimating future behavior of stock price. J Ind Eng Int 1–18. doi:10.1007/s40092-016-0165-7

Mohammadi-Ivatloo B, Rabiee A, Soroudi A, Ehsan M (2012) Imperialist competitive algorithm for solving non-convex dynamic economic power dispatch. Energy 44(1):228–240

Niknam T, Fard ET, Pourjafarian N, Rousta A (2011) An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering. Eng Appl Artif Intell 24(2):306–317

Qu H, Xing K, Alexander T (2013) An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. Neurocomputing 120:509–517

Rabbani M, Montazeri M, Farrokhi-Asl H, Rafiei H (2016) A multi-objective genetic algorithm for a mixed-model assembly U-line balancing type-I problem considering human-related issues, training, and learning. J Ind Eng Int 1–13. doi:10.1007/s40092-016-0158-6

Saraiva JT, Pereira ML, Mendes VT, Sousa JC (2011) A simulated annealing based approach to solve the generator maintenance scheduling problem. Electr Power Syst Res 81(7):1283–1291

Shapiro SS, Wilk MB (1965) An analysis of variance test for normality (complete samples). Biometrika 52(3/4):591–611

Talbi EG (2009) Metaheuristics: from design to implementation, vol 74. Wiley, Hoboken

Tsang HH, Wiese KC (2010) SARNA-predict: accuracy improvement of RNA secondary structure prediction using permutation-based simulated annealing. IEEE/ACM Trans Comput Biol Bioinf 7(4):727–740

Yang XS (2010) Engineering optimization: an introduction with metaheuristic applications. Wiley, Hoboken

Yang XS, Koziel S (eds) (2011) Computational optimization and applications in engineering and industry, vol 359. Springer, Berlin, Heidelberg

Yousefi M, Darus AN, Mohammadi H (2012) An imperialist competitive algorithm for optimal design of plate-fin heat exchangers. Int J Heat Mass Transf 55(11):3178–3185

Zhang Y, Gong DW, Zhang JH (2013) Robot path planning in uncertain environment using multi-objective particle swarm optimization. Neurocomputing 103:172–185