Taylor & Francis
Taylor & Francis Group

Engineering Optimization

ISSN: 0305-215X (Print) 1029-0273 (Online) Journal homepage: https://www.tandfonline.com/loi/geno20

A hybrid iterated local search algorithm with adaptive perturbation mechanism by success-history based parameter adaptation for differential evolution (SHADE)

Fuqing Zhao, Xuan He, Guoqiang Yang, Weimin Ma, Chuck Zhang & Houbin Song

ENGINEERING OPTIMIZATION
https://doi.org/10.1080/0305215X.2019.1595611

A hybrid iterated local search algorithm with adaptive perturbation mechanism by success-history based parameter adaptation for differential evolution (SHADE)

Fuqing Zhao [a], Xuan He[a], Guoqiang Yang[a], Weimin Ma[b], Chuck Zhang[c] and Houbin Song[a]

[a]School of Computer and Communication Technology, Lanzhou University of Technology, Lanzhou, People's Republic of China; [b]School of Economics and Management, Tongji University, Shanghai, People's Republic of China; [c]H.Milton Stewart School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT

The iterated local search (ILS) is exceptionally successful in combinatorial solution spaces. However, few research works have reported on the application of ILS in continuous problems. In this article, a new hybrid population-based iterated local search (HILS) algorithm is proposed for solving numerical optimization problems. The proposed hybrid method introduces both success-history based parameter adaptation for differential evolution (SHADE) and limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) as the perturbation and local search strategy, respectively, and integrates the benefits of exploration capability of SHADE and local search performance of LBFGS. The simulated annealing type of acceptance criterion is adopted to balance the exploration and exploitation of ILS. The proposed HILS is tested against the CEC2017 benchmark functions, which were used to evaluate the performance of the proposed algorithm in solving numerical optimization problems. The experimental results show the effectiveness and efficiency of the HILS.

ARTICLE HISTORY

Received 4 March 2018
Accepted 11 March 2019

KEYWORDS

Iterated local search; SHADE; LBFGS; numerical optimization

1. Introduction

Optimization algorithms have been widely applied to solve many numerical optimization problems originating from real-world problems. For real-parameter numerical global optimization problems, heuristic and meta-heuristic algorithms have already shown excellent search performance (Zhao *et al.* 2017). Most of these algorithms have been widely used in real-life engineering optimization (Shao, Pi, and Shao 2017; Zhao *et al.* 2017, 2018). Classical meta-heuristics include the genetic algorithm (Holland 1992), simulated annealing (SA) (Javidrad and Nazari 2017), differential evolution (DE) (Storn and Price 1997; Cui, Li, Lin, Chen and Lu 2016; Li *et al.* 2016), particle swarm optimization (PSO) (Kennedy and Eberhart 1995; Zhao *et al.* 2015), artificial bee colony (Cui, Li, Lin, Du *et al.* 2016; Cui *et al.* 2017) and other typical hybrid evolution computation algorithms.

Iterated local search (ILS) Lourenço, Martin, and Stützle (2010) is a meta-heuristic providing a simple but powerful framework for improving the performance of local search methods. It has attracted the attention of researchers because of its simplicity, effectiveness and efficiency. The key idea underlying ILS is to focus the search not on the full space of all candidate solutions but on the

CONTACT Fuqing Zhao Fzhao2000@hotmail.com

Supplemental data for this article can be accessed at https://doi.org/10.1080/0305215X.2019.1595611

solutions that are returned by some underlying algorithm, typically a local search heuristic (Lourenço, Martin, and Stützle 2010). The search behaviour is characterized as iteratively building a chain of solutions of this embedded algorithm. There have been three main modifications to the ILS:

- *Population-based ILS*. A population-based ILS heuristic was proposed by Ballestin and Trautmann (2008), with the principle of extracting useful information about the search space by keeping a population of solutions, to deal with the weighted earliness–tardiness resource-constrained project scheduling problem with minimum and maximum time lags. The experimental performance analysis confirmed that the population-based ILS heuristic outperformed some state-of-the-art methods.
- *ILS with guided mutation*. Qingfu and Jianyong (2006) used the guided mutation operator as a perturbation operator in ILS for the quadratic assignment problem. Guided mutation uses the idea of estimation of distribution algorithms to improve conventional mutation operators (Shao and Pi 2016; Shao *et al.* 2014). It provides a mechanism for combining global statistical information about the search space and the positional information of a good solution found during the previous search to generate a new trial solution.
- *Hybrid ILS with mathematical methods*. The traditional mathematical methods have certain limitations, such as premature convergence and poor global search ability, and require derivable fitness functions, so it is difficult to address complex optimization problems. However, some of them show strong local search capabilities, *e.g.* Powell's method, quasi-Newton method, simplex method and pattern search (Wu *et al.* 2014). Kramer (2010) introduced a hybrid ILS that integrates Powell's direct search method with adaptive control of mutation strengths. It combines a direct search with elements from population-based evolutionary optimization. Experimental comparison with the restart covariance matrix adaptation evolution strategy (G-CMA-ES) (Auger and Hansen 2005) reveals a competitive performance.
- *Evolutionary ILS-perturbation technique*. Evolutionary algorithms have been used to generate perturbations for ILS algorithms (Lozano and Garcia-Martinez 2008). The new perturbation technique exploits a micro-evolutionary algorithm capable of effectively exploring the neighbourhood of the solution that should undergo the perturbation operator. This techniques plays the same role as the standard ILS-perturbation operator, but is more powerful than the standard version.

According to the literature, ILS has been widely used to tackle a variety of combinatorial optimization problems owing to its simplicity combined with its powerful search ability (Molina, Latorre, and Herrera 2018). A review and a comprehensive introduction of ILS techniques are presented by Lourenço, Martin, and Stützle (2010) in the context of tackling combinatorial optimization problems, especially the travelling salesman problem. Martins *et al.* (2015) introduced an approach combining the ILS and variable neighbourhood descent (VND) heuristic to solve the cell formation problem (CFP). Differently from several meta-heuristics proposed for the CFP, the proposed algorithm is based on a general and flexible framework that is adapted for dealing with minimum cell size constraints. Li *et al.* (2015) proposed an ILS embedded adaptive neighbourhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pick-ups. The proposed algorithm integrates an adaptive neighbourhood selection mechanism into ILS, employs different structural neighbourhoods in the improvement and perturbation steps, and uses the probability rule to accept a worse solution based on the number of repetitions. Afshar-Nadjafi (2014) proposed an algorithm for estimating the parameters of the gamma/Gompertz distribution based on the maximum likelihood estimation method. ILS is proposed to maximize the likelihood function. An ILS algorithm for the team orienteering problem with variable profits was proposed by Gunawan *et al.* (2018). The ILS obtains good-quality solutions that have significantly better objective value than those found by the commercial solver CPLEX under reasonable computational times. Zhou and Hao (2017) introduced an effective ILS minimum differential dispersion problem, which includes three sequential search phases: a fast descent-based neighbourhood search phase to find a local optimum from a given starting solution, a local optima exploring phase to visit nearby high-quality solutions around a given local optimum, and a local optima escaping phase to move away from the current search region. A multi-start ILS algorithm combined with VND and an adaptive perturbation mechanism is proposed by Avci and Topaloglu (2017) to solve the quadratic multiple-knapsack problem. The multi-start implementation together with the adaptive perturbation mechanism enables the search process to explore different search regions in the search space while VND is applied to obtain high-quality solutions from the examined regions. Xu *et al.* (2017) designed a multi-objective iterated local search (MOILS) method to solve the multi-objective permutation flow-shop scheduling problem, which has properties with sequence-dependent set-up times, and minimize the makespan and total weighted tardiness of all jobs. A Pareto-based variable depth search was applied in the multi-objective local search phase of MOILS. The search depth is dynamically adjusted during the search process of MOILS to strike a balance between exploration and exploitation. The computational results show that the proposed MOILS outperforms several multi-objective evolutionary algorithms.

In addition, ILS has been applied to the job-shop problem (Essafi, Mati, and Dauzere-Peres 2008; Nguyen *et al.* 2015), flow-shop problem (Dong *et al.* 2013; Dong, Huang, and Chen 2009), single-machine earliness/tardiness scheduling (Qin *et al.* 2015), quadratic assignment problem (Stutzle 2006), timetabling problems (Soria-Alcaraz *et al.* 2016), *etc.*

However, the ILS algorithm is rarely used in real-valued search domains, although the way in which ILS handles combinatorial optimization problems has been extended straightforwardly to continuous optimization problems (Lourenço, Martin, and Stützle 2010). Nguyen and Yao (2008) introduced a cultural algorithms–ILS continuous optimization algorithm to solve the multimodal test functions. The latest progress in using ILS to solve continuous optimization problems was the SHADE-ILS algorithm, proposed by Molina, Latorre, and Herrera (2018). The authors introduced a hybrid ILS that combined, in an iterative way, a modern DE algorithm with one local search method chosen dynamically from a set of different search methods. The proposed success-history based parameter adaptation for differential evolution–iterated local search (SHADE-ILS) was tested and compared with various recent state-of-the-art methods based on CEC2013 benchmark functions. The experimental results demonstrate the effectiveness and efficiency of the SHADE-ILS.

In this article, a hybrid population-based iterated local search algorithm, named HILS, is presented to solve single-objective numerical optimization problems. The experimental results under the CEC2017 test suite show that the HILS has a strong global optimization ability compared with several state-of-the-art algorithms: the single-objective real-parameter optimization algorithm (jSO) (Brest, Maučec, and Bošković 2017), increasing population size with midpoint–covariance matrix adaptation evolution strategy (RB-IPOP-CMA-ES) (Biedrzycki 2017), proactive particles in swarm optimization (PPSO) (Tangherloni, Rundo, and Nobile 2017), effective butterfly optimizer with covariance matrix adapted retreat phase (EBOwithCMAR) (Kumar, Misra, and Singh 2017) and success-history based adaptive differential evolution with linear population size reduction–semi-parameter adaptation covariance matrix adaptation evolution strategy (LSHADE_SPACMA) (Mohamed *et al.* 2017). The contributions of this article are described as follows:

- Research on the ILS for solving numerical optimization problems is still limited. In this article, ILS is extended to solve real-parameter numerical global optimization problems.
- Compared with the original ILS, HILS uses a population instead of a single solution in the search space.
- Many heuristic algorithms have been used to enhance the local search ability of ILS in the literature. In the proposed algorithms, a mathematical method named the limited-memory quasi-Newton's algorithm [using the limited-memory quasi-Newton's Broyden–Fletcher–Goldfarb–Shanno (LBFGS) method] (Nocedal and Wright 1999) is utilized to achieve an excellent exploitation capability.
- An adaptive evolutionary ILS-perturbation technique based on SHADE (Tanabe and Fukunaga 2013), which effectively explores the neighbourhood of the solution, is employed.

- The SA acceptance criterion strategy (Javidrad and Nazari 2017) is adopted to enhance the robustness and avoid premature convergence during the iterations of the HILS.
- The Markov model of HILS is presented to analyse the convergence performance of the HILS mathematically. The evolutionary process is mapped to the state-transition process in the Markov model, and the convergence performance is proved.

The remainder of the article is organized as follows. Section 2 summarizes the framework of the original ILS algorithm. The proposed methodology is described and analysed in Section 3. Section 4 provides some experimental studies and statistical analysis regarding the CEC2017 single-objective real-parameter numerical optimization benchmark functions, along with their analysis and discussion. Finally, conclusions are summarized, and future work is highlighted, in Section 5.

2. Iterated local search algorithm

ILS is based on a simple, easy-to-implement, robust, and highly effective idea. Unlike random restart-type algorithms, in which multiple local searches are performed starting from different positions, ILS starts from an initial solution s_0, and successively applies local search and perturbation of the local optimal solution s^*. This procedure is repeated iteratively until a termination condition is satisfied. Starting from a good initial solution can be important if high-quality solutions need to be reached as quickly as possible. Most local search operators are deterministic. Consequently, the perturbation mechanism should introduce non-deterministic components to explore the solution space. The perturbation mechanism performs a global random search in the space of local optima that are approximated by the local search method. Lourenço, Martin, and Stützle (2010) pointed out that the balance of the perturbation mechanism is important. The best perturbation size depends strongly on the particular instance. The perturbation should be strong enough to allow escape from attraction basins, but low enough to exploit knowledge from previous iterations. There is no *a priori* single best size for the perturbation strength. This observation motivates the possibility of modifying the perturbation strength and adapting it during the run (Lourenço, Martin, and Stützle 2010). The acceptance criterion has a strong influence on the nature and effectiveness of the ILS. The diversity of the population is reduced if only satisfactory solutions are accepted. The overall efficiency of the ILS is sensitive to the acceptance criterion applied. The ILS search process is shown in Figure 1. If the fitness value of $s^{*'}$ is less than that of s^*, $s^{*'}$ is accepted. Otherwise, $s^{*'}$ is accepted with a certain probability.

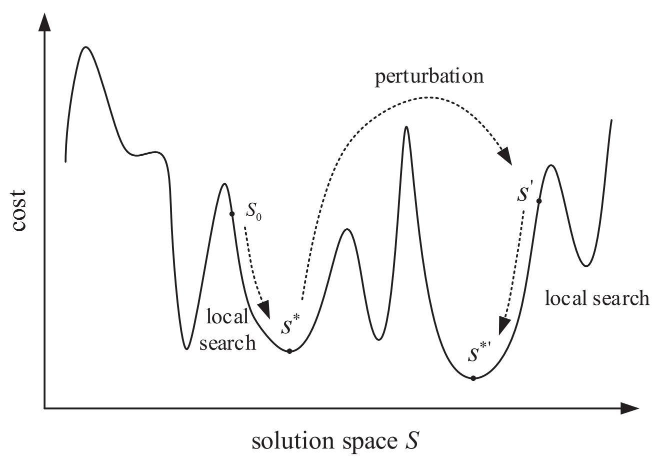

Figure 1. Pictorial representation of the iterated local search. (Adapted from Lourenço, Martin, and Stützle 2010.)

3. Hybrid population-based iterated local search algorithm

The four basic operators in HILS are described below.

- *Initial solution:* This is generated at random.
- *LS operator:* The LBFGS method.
- *Perturbation operator:* The SHADE perturbation mechanism.
- *Acceptance criterion operator:* The SA acceptance criterion.

3.1. Local search operator: the limited-memory quasi-Newton's method

Although the Newton method has a relatively fast convergence speed, the calculation of the second order partial derivative of the objective function has high computational complexity. The Hessian matrix of the objective function is not always positive definite. The quasi-Newton method is proposed in this article to address the various problems relating to the performance of the algorithm. The basic idea of the method is to construct a positive definite symmetrical matrix which is the approximation of the Hessian matrix without second order partial derivatives. The objective function under the condition of 'quasi-Newton' is optimized by Broyden–Fletcher–Goldfarb–Shanno (BFGS). Different construction methods produce different quasi-Newton methods. LBFGS is based on the BFGS updating formula.

Each step of the BFGS method has the form:

$$\begin{aligned} d_k &= -H_k \nabla f_k \\ x_{k+1} &= x_k + \alpha_k d_k \end{aligned} \tag{1}$$

where α_k is the step length and α_k satisfies the Wolfe conditions:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \beta' \alpha_k (\nabla f_k)^T d_k \tag{2}$$

$$(\nabla f_{k+1})^T d_k \ge \beta (\nabla f_k)^T d_k \tag{3}$$

$$0 < \beta' < \frac{1}{2}, \beta' < \beta < 1 \tag{4}$$

Moreover, if $\alpha_k = 1$ satisfies Equations (2)–(4), then $\alpha_k = 1$.

H_k is updated at every iteration by means of the formula:

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T \tag{5}$$

where

$$\rho_k = \frac{1}{y_k^T s_k}, \quad V_k = I - \rho_k y_k s_k^T,$$

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f_{k+1} - \nabla f_k.$$

The main idea of LBFGS is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information from earlier iterations, which is less likely to be relevant to the actual behaviour of the Hessian at the current iteration, is discarded in the interest of saving storage. The LBFGS is an instant method and has linear memory requirements. It is especially well suited to optimization problems with a large number of variables, which makes it a successful local search method in the optimization field (Shi, Yang, and Xiao 2016; Biglari 2015; Badem *et al.* 2017; Xiao, Wei, and Wang 2008).

3.2. Perturbation operator: the SHADE algorithm

SHADE (Tanabe and Fukunaga 2013) is a success history-based parameter adaptation technique of DE, which introduced a novel mechanism for parameter adaptation based on the historical memory of successful parameter settings. Success-history based adaptation uses a historical memory M_{CR}, M_F, which stores a set of CR, F values that have performed well in the past, and generates new CR, F pairs by directly sampling the parameter space close to one of these stored pairs.

The overall SHADE algorithm includes the external archive with size H. In each generation, the DE control parameters scaling factor F and crossover rate CR are generated based on the success history. Trial vectors are generated by the current-to-pbest/1/bin rule (to maintain diversity, SHADE uses an external archive $\boldsymbol{A}$ to generate the trial vectors), selection is applied and the historical memory is updated. This process is repeated until some termination criteria are achieved.

A brief description of DE is as follows. The population of DE consists of a set of real parameter vectors $x_i = (x_1, \ldots, x_D)$, $i = 1, \ldots, N$, where D is the dimensionality of the target problem, and N is the population size. At the beginning of the search, the individual vectors in the population are initialized randomly.

The mutation operator is:

$$v_{i,G} = x_{i,G} + F_{i,G} \cdot (x_{pbest,G} - x_{i,G}) + F_{i,G} \cdot (x_{r1,G} - x_{r2,G}) \tag{6}$$

The indices $r1, r2$ are selected from $[0 : N]$ randomly. $r1, r2$ and i are different from each other. $x_{pbest,G}$ denotes the best individual in the population in generation G. The parameter $F_{i,G} \epsilon [0 : 1]$ is the scaling factor of the ith individual in generation G; it controls the magnitude of the differential mutation operator. For the jth dimension, if the mutant vector element $v_{i,j,G}$ is outside the boundaries $[x_j^{\min}, x_j^{\max}]$, the following operation is performed:

$$v_{i,j,G} = \begin{cases} (x_j^{\min} + x_{i,j,G})/2 & \text{if} \quad v_{i,j,G} < x_j^{\min} \\ (x_j^{\max} + x_{i,j,G})/2 & \text{if} \quad v_{i,j,G} > x_j^{\max} \end{cases} \tag{7}$$

The crossover operator is:

$$u_{i,j,G} = \begin{cases} v_{i,j,G} & \text{if } \operatorname{rand}[0,1) < CR \text{ or } j = j_{\text{rand}} \\ x_{i,j,G} & \text{otherwise} \end{cases} \tag{8}$$

The selection operator is:

$$x_{i,G+1} = \begin{cases} u_{i,G} & \text{if} f(u_{i,G}) \le f(x_{i,G}) \\ x_{i,G} & \text{otherwise} \end{cases} \tag{9}$$

For more details, the reader is referred to Tanabe and Fukunaga (2013), which includes a further description of the SHADE method.

3.3. Acceptance criterion operator: the SA algorithm

The overall efficiency of ILS is sensitive to the acceptance criterion applied. SA (Kirkpatrick, Gelatt, and Vecchi 1983) is a probabilistic technique for approximating the global optimum of a given function. In the SA algorithm, the Metropolis Monte Carlo procedure was used to describe a collection of atoms in balance at a specified temperature. The Metropolis program allows states with high energy to be accepted with a certain probability. The acceptance probability of a trial solution point X_{i+1} from the current solution point X_i is stated as:

$$P = \begin{cases} 1 & \text{if} f(X_{i+1}) \le f(X_i) \\ e^{-\Delta f/T} & \text{Otherwise} \end{cases} \tag{10}$$

where f is the objective, $\Delta f = f(X_{i+1}) - f(X_i)$, and T is the temperature. A low temperature means a relatively low acceptance probability.

It is theoretically proven that the probability of finding the global minimum can approach 1 when very slow rates of temperature decrease are used (Michiels, Aarts, and Korst 2007). Many cooling schedules for temperature change have been introduced in the literature, using linear, logarithmic and exponential functions. The most common temperature decrement rule has the form:

$$T_{i+1} = R_T \cdot T_i \tag{11}$$

where T_0 is the initial temperature and R_T is a positive constant, usually taken in the interval 0.8–0.999 (Javidrad and Nazari 2017).

SA has been widely used in numerical optimization problems (Chen, Sarosh, and Dong 2012; Javidrad and Nazari 2017) and discrete optimization problems (Akram, Kamal, and Zeb 2016; Wei *et al*. 2018; Ezugwu, Adewumi, and Frîncu 2017).

3.4. Description of the HILS algorithm

In this article, the HILS is proposed based on five key concepts:

- Limited-memory quasi-Newton's method as a local search method: this is a fast direct search optimization method, in particular for unimodal problems. There are two termination conditions for LBFGS: (1) the difference in the quality of solutions between two adjacent iterations is less than 10^{-8}; and (2) the number of iterations is greater than 200.
- Iterative local search: this prevents the limited-memory quasi-Newton's method becoming stuck in local optima. The ILS approach is based on the successive repetition of the limited-memory quasi-Newton's method as local search technique and a perturbation mechanism.
- Population-based ILS: a population of candidate solutions is adopted to explore the search space more efficiently.
- Adaptive perturbation strategy: the strength of the ILS perturbation is controlled by the adaptive perturbation strategy in SHADE, which effectively explores the neighbourhood of the solution that should undergo the perturbation operator.
- Acceptance criterion: an SA-type acceptance criterion is applied. This diversification mechanism balances the intensification and diversification of the HILS.

The flowchart of the proposed HILS algorithm is shown in Figure 2. The pseudo-code of HILS is presented in Algorithm 1 in the online Supplementary material, which consists of the following main steps.

Step 1: Initialization

Four parameters need to be initialized: population size $N = 50$, memory size $H = 50$, initial temperature $T_0 = 1$ and temperature decrement parameter $R_T = 0.999$. Generate a random initial population $P_0 = (X_{1,0}, \ldots, X_{N,0})$, set all values in M_{CR}, M_F to 0.5, generation counter $G = 0$, external archive $A = \emptyset$, index counter $k = 1$. The values of the parameters are the results of the experiments in Section S2 of the Supplementary material.

Step 2: Record three current solutions

Obtain the best solution $X^{\text{current}}_{\text{best},G}$, the worst solution $X^{\text{current}}_{\text{worst},G}$ and a random solution $X^{\text{current}}_{\text{rand},G}$ (rand $\neq$ best $\neq$ worst) in P_G. Three variables, rand, best and worst, are used to record the position of the three solutions in P_G.

Step 3: Perturbation phase

The mutation and crossover strategy of SHADE is adopted to perturb the population P_G to produce the population U^{current}_G.

Step 4: Local search phase

The LBFGS algorithm is used to obtain the local optimum $U^{\text{trial}}_{\text{best},G}$, $U^{\text{trial}}_{\text{worst},G}$, $U^{\text{trial}}_{\text{rand},G}$ from the three solutions $U^{\text{current}}_{\text{best},G}$, $U^{\text{current}}_{\text{worst},G}$, $U^{\text{current}}_{\text{rand},G}$ in the population U^{current}_G.

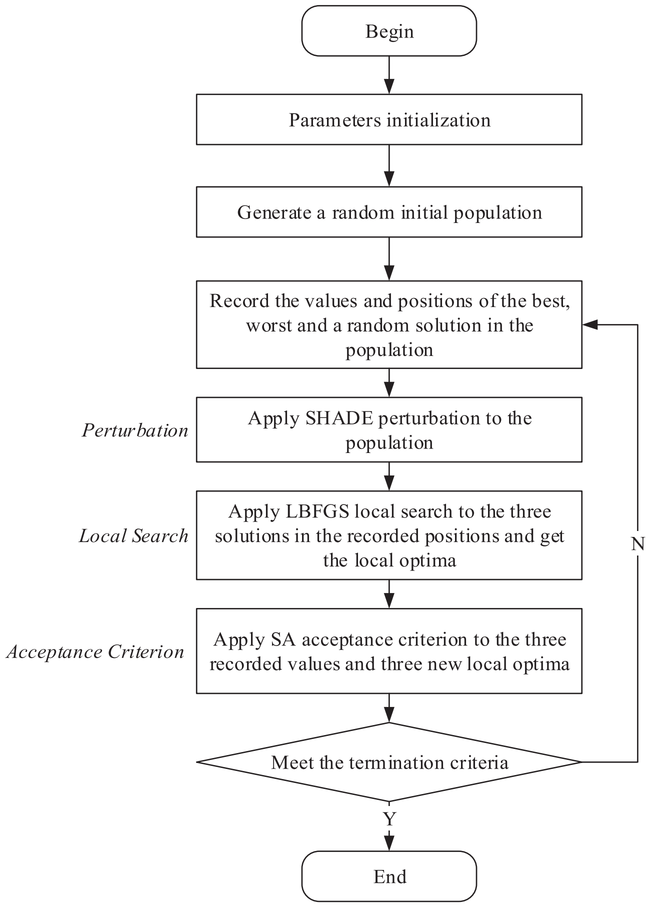

Figure 2. Flowchart of the hybrid population-based iterated local search (HILS) algorithm. SHADE = success-history based parameter adaptation for differential evolution; LBFGS = limited-memory Broyden–Fletcher–Goldfarb–Shanno; SA = simulated annealing.

Step 5: Acceptance criterion phase

The acceptance strategy from SA is used to decide whether the current or trial solution survives to the next iteration.

Step 6: Check the termination condition

If the number of function evaluations exceeds the maximum evaluations, then terminate the iteration and output the optimal solution R. Otherwise, $T_{G+1} = R_T \cdot T_G$, $G = G + 1$ and go back to Step 3.

4. Experiments and comparisons

In this section, the performance of the proposed algorithm is evaluated on the CEC2017 benchmark set (Awad *et al.* 2016). Subsection 4.1 describes the experimental conditions under which the simulation is run. Subsection 4.2 discusses the time complexity of HILS. The comparison of HILS with some proposed state-of-the-art algorithms is described in Subsection 4.3. It is necessary to point out that according to the evaluation criteria of CEC2017, f_2 has been excluded because it shows unstable behaviour, especially for higher dimensions, and significant performance variations for the same algorithm implemented in MATLAB® and C program.

Moreover, owing to limited space, the convergence analysis, time complexity analysis, and parameter analysis of HILS are presented in the online Supplementary material. The final parameters used in the proposed HILS are set as follows: $N = 50, H = 50, \quad R_T = 0.999$.

4.1. Experimental conditions

The experiments are run on a Windows Server 2012 R2 under the hardware environment of Intel® Core™ i7-6700 CPU 3.40 GHz processor and 8.0 GB of RAM. The proposed algorithm is implemented using C++ programming language. A cross-platform numerical analysis and data processing library, ALGLIB (http://www.alglib.net/), was used to deal with the LBFGS local search.

To compare the effect of each operator on HILS, the experiments compared HILS, ILS with jSO perturbation operator, HILS without LBFGS and HILS without SA-based acceptance criterion strategy. The experimental results are shown in Table 1. SHADE is more suitable than jSO to act as a perturbation operator in the ILS framework. LBFGS performed a key role in the HILS. SA is important in the hybrid and composition functions.

4.2. Time complexity of HILS

This subsection deals with an analysis of the HILS time complexity as defined in Awad *et al.* (2016). The running time obtained by evaluating the benchmark function f_{18} is compared with the running time of the test program.

The computing time of the test program is denoted as $T0$. Assume that variable $T1$ represents the time required for evaluating the benchmark function f_{18} and variable $T2$ the time of HILS execution for function f_{18} within 200,000 evaluations for each dimension. Variable $\widehat{T2}$ is an average of $T2$ values obtained in five independent runs, and shows the relationship between the complexity of the algorithm and dimension. The computational complexity of the HILS algorithm is reflected by the measured and calculated variables $T0$, $T1$, $\widehat{T2}$ and $(\widehat{T2} - T1)/T0$ for each of the observed dimensions, $D = \{10, \ 30, \ 50, \ 100\}$. This calculation is independent of the computing system and programming language in which the measured algorithm is implemented. From Table 2, it is easy to conclude that more computational cost is required with an increasing number of dimensions for the benchmark functions.

Table 1. Mean error of different algorithm combinations on 50 dimensions.

Functions	HILS	ILS with jSO	HILS without LBFGS	HILS without SA
Unimodal functions	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
Simple multimodal functions	**6.95E+02**	7.51E+02	4.59E+04	5.07E+03
Hybrid functions	**1.90E+02**	3.73E+03	3.25E+03	6.01E+04
Composition functions	**1.30E+04**	2.73E+04	6.65E+04	9.63E+05

Note: HILS = hybrid population-based iterated local search; ILS = iterated local search; jSO = single-objective real-parameter optimization algorithm; LBFGS = limited-memory Broyden–Fletcher–Goldfarb–Shanno; SA = simulated annealing.

Table 2. Computational complexity of the hybrid population-based iterated local search (HILS) algorithm (all times are in seconds).

D	$T0$	$T1$	$\widehat{T2}$	$(\widehat{T2} - T1)/T0$
10	0.148	0.270	3.906	24.568
30		0.915	5.136	28.520
50		1.995	13.601	78.419
100		6.742	72.879	446.872

4.3. Comparison of HILS with some proposed state-of-the-art algorithms

In this section, the proposed HILS is compared with five state-of-the-art algorithms, namely jSO (Brest, Maučec, and Bošković 2017), RB-IPOP-CMA-ES (Biedrzycki 2017), PPSO (Tangherloni, Rundo, and Nobile 2017), EBOwithCMAR (Kumar, Misra, and Singh 2017) and LSHADE_SPACMA (Mohamed *et al.* 2017). The five algorithms are variants of SHADE, PSO, effective butterfly optimizer (EBO) and the covariance matrix adaptation evolution strategy (CMA-ES). A brief description of these algorithms is as follows.

- jSO is an improved variant of the iL-SHADE algorithm (Brest, Maučec, and Bošković 2016, 2017), mainly with a new weighted version of the mutation strategy. On the CEC2017 test suite, the solving performance of jSO is ranked second out of 11 algorithms.
- RB-IPOP-CMA-ES is an enhanced version of IPOP-CMA-ES (Auger and Hansen 2005), which uses the midpoint of the population as an approximation of the optimum. Simulation results over the CEC2017 test suite demonstrate the effectiveness and improvement of the RB-IPOP-CMA-ES compared with IPOP-CMA-ES.
- PPSO is a self-tuning version of PSO, which leverages fuzzy logic to dynamically determine the best settings for the inertia weight, cognitive factor and social factor (Tangherloni *et al.* 2018). The robustness of PPSO is proved by conducting the CEC2017 test suite (Nobile *et al.* 2016; Spolaor *et al.* 2017).
- EBOwithCMAR uses a covariance matrix to generate a new solution and thus improves the local search capability of EBO. The experimental results on CEC2017 benchmarks show that EBOwithCMAR is the top method among 11 algorithms.
- LSHADE_SPACMA uses a new semi-parameter adaptation approach to effectively adapt the values of the scaling factor of the DE algorithm. The modified version of CMA-ES undergoes the crossover operation to improve the exploration capability of the proposed framework. The experimental results show that LSHADE_SPACMA is competitive with state-of-the-art evolutionary algorithms on the CEC2017 benchmarks.

To evaluate the performance of the proposed HILS, four groups of simulations are executed on 10, 30, 50 and 100 dimensions over the CEC2017 test suite. All experimental algorithms are run independently 51 times on each test problem, and mean and standard deviation metrics are calculated. The experimental results are shown. Owing to limited space, the experimental results of all algorithms on 10, 30, 50 and 100 dimensions are described in the online Supplementary material.

The convergence rate of the HILS algorithm is also compared with that of jSO, RB-IPOP-CMA-ES, PPSO, LSHADE_SPACMA and EBOwithCMAR. Figure 3 describes the convergence plots of f_1, f_2, f_4, f_5, f_8, f_{10}, f_{18} and f_{22} which include the unimodal, simple multimodal, hybrid and composition functions. It can be observed that the convergence rate of HILS is better than that of the other algorithms in the majority of the test functions. The efficient local search strategy in HILS plays a key role, as it effectively guides the three solutions in the population to the local optimum and helps to improve the convergence rate.

To verify the performance of the HILS algorithm, statistical tests demonstrate HILS shows significant improvement over the compared algorithms. The multiple-problem Wilcoxon's test (Garcia *et al.* 2009) was performed to check the behaviours of the five algorithms which were introduced as comparison algorithms. Holm's procedure and Hochberg's procedure were used as *post hoc* procedures in Wilcoxon's test.

Table 3 summarizes the statistical analysis results, considering HILS as the control algorithm. Where HILS is significantly better than the compared algorithm with a certain confidence level, the test results are marked in bold. From Table 3, it can be found that HILS is significantly better than PPSO on 30 benchmark functions, with $\alpha = 0.05$ for all dimensions, and significantly better than RB-IPOP-CMA-ES on 30 benchmark functions, with $\alpha = 0.1$ except for 100 dimensions. HILS is

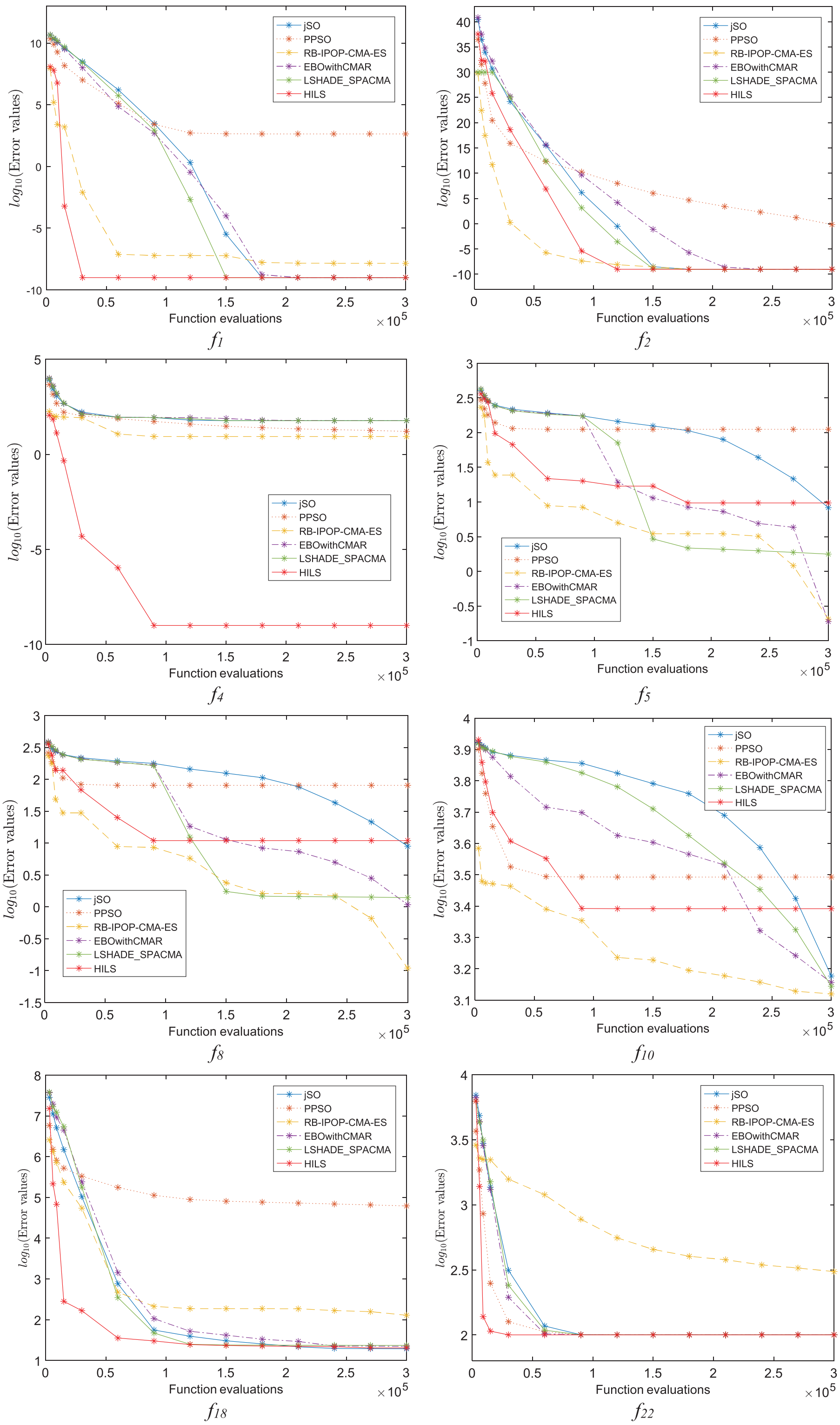

Figure 3. Convergence plots of single-objective real-parameter optimization algorithm (jSO), proactive particles in swarm optimization (PPSO), increasing population size with midpoint–covariance matrix adaptation evolution strategy (RB-IPOP-CMA-ES), effective butterfly optimizer with covariance matrix adapted retreat phase (EBOwithCMAR), success-history based adaptive differential evolution with linear population size reduction–semi-parameter adaptation covariance matrix adaptation evolution strategy (LSHADE_SPACMA) and hybrid population-based iterated local search (HILS) on some typical benchmark functions (30 dimensions).

Table 3. Results of the multiple-problem Wilcoxon's test for HILS, jSO, RB-IPOP-CMA-ES, PPSO, LSHADE_SPACMA and EBOwithCMAR at $\alpha = 0.05$ and $\alpha = 0.1$ significance levels on 10, 30, 50 and 100 dimensions.

						Adjusted *p*-value			
D	HILS *vs*	R+	R−	*Z*	*p*-Value	Holm	Hochberg	$\alpha = 0.1$	$\alpha = 0.05$
10	jSO	33.00	220.00	−3.036	0.002	0.004	0.004	**Yes**	**Yes**
	RB-IPOP-CMA-ES	23.00	302.00	−3.754	0.000	0.000	0.000	**Yes**	**Yes**
	PPSO	3.00	375.00	−4.469	0.000	0.000	0.000	**Yes**	**Yes**
	LSHADE_SPACMA	24.00	252.00	−3.467	0.001	0.003	0.003	**Yes**	**Yes**
	EBOwithCMAR	49.00	182.00	−2.311	0.021	0.021	0.021	**Yes**	**Yes**
30	jSO	106.50	218.50	−1.507	0.132	0.264	0.264	No	No
	RB-IPOP-CMA-ES	58.00	320.00	−3.147	0.002	0.008	0.008	**Yes**	**Yes**
	PPSO	0.00	406.00	−4.623	0.000	0.000	0.000	**Yes**	**Yes**
	LSHADE_SPACMA	78.00	222.00	−2.057	0.040	0.120	0.120	No	No
	EBOwithCMAR	112.00	188.00	−1.086	0.278	0.278	0.278	No	No
50	jSO	154.00	197.00	−0.546	0.585	1.000	0.585	No	No
	RB-IPOP-CMA-ES	81.00	297.00	−2.595	0.009	0.036	0.036	**Yes**	**Yes**
	PPSO	0.00	435.00	−4.703	0.000	0.000	0.000	**Yes**	**Yes**
	LSHADE_SPACMA	111.00	189.00	−1.114	0.265	0.759	0.585	No	No
	EBOwithCMAR	150.00	201.00	−0.648	0.517	1.000	0.585	No	No
100	jSO	170.50	82.50	−1.429	0.153	0.459	0.400	No	No
	RB-IPOP-CMA-ES	244.00	81.00	−2.193	0.028	0.112	0.112	No	No
	PPSO	334.00	44.00	−3.484	0.000	0.000	0.000	**Yes**	**Yes**
	LSHADE_SPACMA	166.00	87.00	−1.282	0.200	0.459	0.400	No	No
	EBOwithCMAR	121.00	155.00	−0.517	0.605	0.605	0.605	No	No

Note: HILS = hybrid population-based iterated local search; jSO = single-objective real-parameter optimization algorithm; RB-IPOP-CMA-ES = increasing population size with midpoint–covariance matrix adaptation evolution strategy; PPSO = proactive particles in swarm optimization; LSHADE_SPACMA = success-history based adaptive differential evolution with linear population size reduction–semi-parameter adaptation covariance matrix adaptation evolution strategy; EBOwithCMAR = effective butterfly optimizer with covariance matrix adapted retreat phase.

significantly better than all the comparison algorithms in 10 dimensions. Although significant differences cannot be observed between HILS and jSO and EBOwithCMAR, the R− value is better than R+ for jSO, LSHADE_SPACMA and EBOwithCMAR in 30 and 50 dimensions. That is, the HILS can obtain better solutions in the above cases.

To check the significant differences between HILS and the five competitors, Friedman's test (Garcia *et al.* 2009) was carried out to rank the algorithms, for use in the experimental evaluations, and Bonferroni–Dunn's procedure was used as a *post hoc* procedure. Table 4 summarizes the average ranking of the six algorithms obtained by Friedman's test. From Table 4, it can be seen that the proposed HILS ranks first in 10 and 30 dimensions. To evaluate the significance level of all the algorithms, an additional Bonferroni–Dunn's procedure is applied as a *post hoc* procedure to calculate the critical difference (CD in Equation 12) to compare their differences with $\alpha = 0.05$ and $\alpha = 0.1$:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{12}$$

In Equation (12), parameters k and N are the number of algorithms to be compared and the number of benchmarks, respectively. They are $k = 6$ and $N = 30$ in the experimental evaluations. When $\alpha = 0.05$, q_α is 2.576, and when $\alpha = 0.1$, q_α is 2.327, from Table B.16 (two-tailed $\alpha(2)$) of Zar (1999). Figure 4 sketches the results of Bonferroni–Dunn's test, considering HILS as the control algorithm. From Figure 4, it can be seen that there is a significant difference between HILS and PPSO, with $\alpha = 0.1$ and $\alpha = 0.05$ for all dimensions. There is a significant difference between HILS and LSHADE_SPACMA, with $\alpha = 0.1$ and $\alpha = 0.05$ for 10 dimensions. There is a significant difference between HILS and RB-IPOP-CMA-ES, with $\alpha = 0.1$ and $\alpha = 0.05$ for 10 and 30 dimensions. There is no significant difference between HILS and the other algorithms in other cases.

Table 4. Ranking of HILS, jSO, RB-IPOP-CMA-ES, PPSO, LSHADE_SPACMA and EBOwithCMAR obtained through Friedman's test and critical difference of Bonferroni-Dunn's procedure.

	Mean rank			
Algorithm	10D	30D	50D	100D
jSO	3.10	2.90	2.83	3.43
RB-IPOP-CMA-ES	4.21	3.97	3.69	3.55
PPSO	4.93	5.62	5.86	5.81
LSHADE_SPACMA	3.76	3.31	3.21	1.76
EBOwithCMAR	2.91	2.74	2.81	3.29
HILS	2.09	2.47	2.60	3.16
CD $\alpha = 0.05$	1.2443			
CD $\alpha = 0.1$	1.1240			

Note: HILS = hybrid population-based iterated local search; jSO = single-objective real-parameter optimization algorithm; RB-IPOP-CMA-ES = increasing population size with midpoint–covariance matrix adaptation evolution strategy; PPSO = proactive particles in swarm optimization; LSHADE_SPACMA = success-history based adaptive differential evolution with linear population size reduction–semi-parameter adaptation covariance matrix adaptation evolution strategy; EBOwithCMAR = effective butterfly optimizer with covariance matrix adapted retreat phase; CD = critical difference.

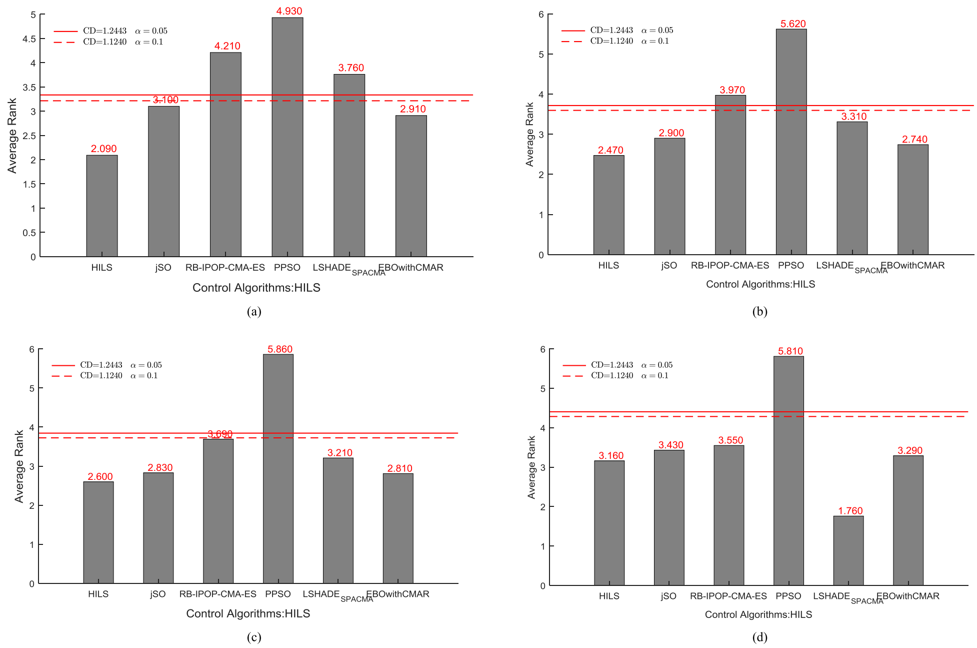

Figure 4. Rankings obtained through the Friedman test and graphical representation of Bonferroni-Dunn's test, taking hybrid population-based iterated local search (HILS) as a control method: (a) 10 dimensions (10D); (b) 30D; (c) 50D; (d) 100D. jSO = single-objective real-parameter optimization algorithm; PPSO = proactive particles in swarm optimization; RB-IPOP-CMA-ES = increasing population size with midpoint–covariance matrix adaptation evolution strategy; EBOwithCMAR = effective butterfly optimizer with covariance matrix adapted retreat phase; LSHADE_SPACMA = success-history based adaptive differential evolution with linear population size reduction–semi-parameter adaptation covariance matrix adaptation evolution strategy; CD = critical difference.

In summary, the statistical analysis of the results obtained by the algorithms in the comparative study showed that the proposed HILS is the best for 10, 30 and 50 dimensions. Even though HILS cannot obtain the best results on some test problems, it can obtain the suboptimal results compared with all experimental algorithms. The proposed HILS is superior to RB-IPOP-CMA-ES, PPSO, LSHADE_SPACMA on the most of the benchmark problems with different dimensions. Compared with other methods, HILS can obtain the best results on the unimodal and multimodal functions with different dimensions. On the rest of the test problems, the proposed algorithm still maintains a stable solving performance. The convergence rate of HILS is much better than that of the other algorithms in most of the functions in the early stage of the search, which benefited from the LBFGS local search strategy.

5. Conclusions

This article presented a hybrid population-based iterated local search algorithm (HILS) to solve numerical optimization problems. In the proposed HILS, the population-based SHADE perturbation operator was used to improve the social behaviour of ILS. Then, the LBFGS was used as the local search operator to enhance the local search ability of ILS. Afterwards, the SA acceptance criterion was developed to balance the exploration and exploitation of ILS, and avoid trapping into local optima. In addition, the global convergence performance of HILS was analysed theoretically. The parameters of HILS were also calibrated by numerical experiments. Finally, the proposed HILS was evaluated on the CEC2017 benchmark functions and compared with five state-of-the-art algorithms. The comparison results showed that the proposed HILS significantly outperforms these algorithms. The source code of HILS is published at https://github.com/fenyuguren/HILS_source_code, which can provide a reference for future studies.

Further research will be conducted in the following directions. First, the authors intend to continue improving the performance of HILS and apply the improved HILS to solve some scheduling problems, *e.g.* job-shop, flow-shop and lot-streaming flow-shop scheduling problems. Secondly, ILS will be combined with other meta-heuristics for multi-objective scheduling problems. HILS will also be used in machine learning in classification tasks, data mining and certain related research fields.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was financially supported by the National Natural Science Foundation of China [grant number 61663023]. It was also supported by the Key Science and Technology Foundation of Gansu Province [2017GS10817], Lanzhou Science and Technology [2018-rc-98], Zhejiang Provincial Natural Science Foundation [LGJ19E050001] Wenzhou Science and Technology Bureau [G20170016].

ORCID

Fuqing Zhao http://orcid.org/0000-0002-7336-9699

References

Afshar-Nadjafi, B. 2014. "An Iterated Local Search Algorithm for Estimating the Parameters of the Gamma/Gompertz Distribution." *Modelling and Simulation in Engineering* 2014 (1): 1–7. doi:10.1155/2014/629693.

Akram, K., K. Kamal, and A. Zeb. 2016. "Fast Simulated Annealing Hybridized with Quenching for Solving Job Shop Scheduling Problem." *Applied Soft Computing* 49: 510–523. doi:10.1016/j.asoc.2016.08.037.

Auger, A., and N. Hansen. 2005. "A Restart CMA Evolution Strategy with Increasing Population Size." In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, 1769–1776. Piscataway, NJ: IEEE Press.

Avci, M., and S. Topaloglu. 2017. "A Multi-start Iterated Local Search Algorithm for the Generalized Quadratic Multiple Knapsack Problem." *Computers & Operations Research* 83: 54–65. doi:10.1016/j.cor.2017.02.004.

Awad, N. H., M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan. 2016. "Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization." Technical Report, Nanyang Technological University, Singapore.

Badem, H., A. Basturk, A. Caliskan, and M. E. Yuksel. 2017. "A New Efficient Training Strategy for Deep Neural Networks by Hybridization of Artificial Bee Colony and Limited–Memory BFGS Optimization Algorithms." *Neurocomputing* 266: 506–526. doi:10.1016/j.neucom.2017.05.061.

Ballestin, F., and N. Trautmann. 2008. "An Iterated-Local-Search Heuristic for the Resource-Constrained Weighted Earliness-Tardiness Project Scheduling Problem." *International Journal of Production Research* 46 (22): 6231–6249. doi:10.1080/00207540701420560.

Biedrzycki, R. 2017. "A Version of IPOP-CMA-ES Algorithm with Midpoint for CEC 2017 Single Objective Bound Constrained Problems." In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation*, 1489–1494. Piscataway, NJ: IEEE.

Biglari, F. 2015. "Dynamic Scaling on the Limited Memory BFGS Method." *European Journal of Operational Research* 243 (3): 697–702. doi:10.1016/j.ejor.2014.12.050.

Brest, J., M. S. Maučec, and B. Bošković. 2016. "iL-SHADE: Improved L-SHADE Algorithm for Single Objective Real-Parameter Optimization." In *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, 1188–1195. Piscataway, NJ: IEEE.

Brest, J., M. S. Maučec, and B. Bošković. 2017. "Single Objective Real-Parameter Optimization: Algorithm jSO." In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, 1311–1318. Piscataway, NJ: IEEE.

Chen, S.-M., A. Sarosh, and Y.-F. Dong. 2012. "Simulated Annealing Based Artificial Bee Colony Algorithm for Global Numerical Optimization." *Applied Mathematics and Computation* 219 (8): 3575–3589. doi:10.1016/j.amc.2012.09.052.

Cui, L. Z., G. H. Li, Q. Z. Lin, J. Y. Chen, and N. Lu. 2016. "Adaptive Differential Evolution Algorithm with Novel Mutation Strategies in Multiple Sub-populations." *Computers & Operations Research* 67: 155–173. doi:10.1016/j.cor.2015.09.006.

Cui, L. Z., G. H. Li, Q. Z. Lin, Z. H. Du, W. F. Gao, J. Y. Chen, and N. Lu. 2016. "A Novel Artificial Bee Colony Algorithm with Depth-First Search Framework and Elite-Guided Search Equation." *Information Sciences* 367–368: 1012–1044. doi:10.1016/j.ins.2016.07.022.

Cui, L. Z., G. H. Li, X. Z. Wang, Q. Z. Lin, J. Y. Chen, N. Lu, and J. Lu. 2017. "A Ranking-Based Adaptive Artificial Bee Colony Algorithm for Global Numerical Optimization." *Information Sciences* 417: 169–185. doi:10.1016/j.ins.2017.07.011.

Dong, X. Y., P. Chen, H. K. Huang, and M. Nowak. 2013. "A Multi-restart Iterated Local Search Algorithm for the Permutation Flow Shop Problem Minimizing Total Flow Time." *Computers & Operations Research* 40 (2): 627–632. doi:10.1016/j.cor.2012.08.021.

Dong, X. Y., H. K. Huang, and P. Chen. 2009. "An Iterated Local Search Algorithm for the Permutation Flowshop Problem with Total Flowtime Criterion." *Computers & Operations Research* 36 (5): 1664–1669. doi:10.1016/j.cor.2008.04.001.

Essafi, I., Y. Mati, and S. Dauzere-Peres. 2008. "A Genetic Local Search Algorithm for Minimizing Total Weighted Tardiness in the Job-Shop Scheduling Problem." *Computers & Operations Research* 35 (8): 2599–2616. doi:10.1016/j.cor.2006.12.019.

Ezugwu, A. E.-S., A. O. Adewumi, and M. E. Frîncu. 2017. "Simulated Annealing Based Symbiotic Organisms Search Optimization Algorithm for Traveling Salesman Problem." *Expert Systems with Applications* 77: 189–210. doi:10.1016/j.eswa.2017.01.053.

Garcia, S., D. Molina, M. Lozano, and F. Herrera. 2009. "A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization." *Journal of Heuristics* 15 (6): 617–644. doi:10.1007/s10732-008-9080-4.

Gunawan, A., K. M. Ng, G. Kendall, and J. H. Lai. 2018. "An Iterated Local Search Algorithm for the Team Orienteering Problem with Variable Profits." *Engineering Optimization* 50 (7): 1148–1163. doi:10.1080/0305215x.2017.1417398.

Holland, J. H. 1992. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA: MIT Press.

Javidrad, F., and M. Nazari. 2017. "A New Hybrid Particle Swarm and Simulated Annealing Stochastic Optimization Method." *Applied Soft Computing* 60: 634–654. doi:10.1016/j.asoc.2017.07.023.

Kennedy, J., and R. Eberhart. 1995. "Particle Swarm Optimization." Proceedings of ICNN'95 - international conference on neural networks, November 27–December 1.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598): 671–680. doi:10.1126/science.220.4598.671.

Kramer, O. 2010. "Iterated Local Search with Powell's Method: A Memetic Algorithm for Continuous Global Optimization." *Memetic Computing* 2 (1): 69–83. doi:10.1007/s12293-010-0032-9.

Kumar, A., R. K. Misra, and D. Singh. 2017. "Improving the Local Search Capability of Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat Phase." In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, 1835–1842. Piscataway, NJ: IEEE.

Li, G. H., Q. Z. Lin, L. Z. Cui, Z. H. Du, Z. P. Liang, J. Y. Chen, N. Lu, and Z. Ming. 2016. "A Novel Hybrid Differential Evolution Algorithm with Modified CoDE and JADE." *Applied Soft Computing* 47: 577–599. doi:10.1016/j.asoc.2016.06.011.

Li, J., P. M. Pardalos, H. Sun, J. Pei, and Y. Zhang. 2015. “Iterated Local Search Embedded Adaptive Neighborhood Selection Approach for the Multi-depot Vehicle Routing Problem with Simultaneous Deliveries and Pickups.” *Expert Systems with Applications* 42 (7): 3551–3561. doi:10.1016/j.eswa.2014.12.004.

Lourenço, H. R., O. C. Martin, and T. Stützle. 2010. “Iterated Local Search: Framework and Applications.” In *Handbook of Metaheuristics*, 363–397. Boston, MA: Springer.

Lozano, M., and C. Garcia-Martinez. 2008. “An Evolutionary ILS-Perturbation Technique.” Paper presented at the 5th International Workshop on Hybrid Metaheuristics, HM 2008, Malaga, Spain, October 8–9.

Martins, I. C., R. G. S. Pinheiro, F. Protti, and L. S. Ochi. 2015. “A Hybrid Iterated Local Search and Variable Neighborhood Descent Heuristic Applied to the Cell Formation Problem.” *Expert Systems with Applications* 42 (22): 8947–8955. doi:10.1016/j.eswa.2015.07.050.

Michiels, W., E. Aarts, and J. Korst. 2007. *Theoretical Aspects of Local Search*. Berlin Heidelberg: Springer Science & Business Media.

Mohamed, A. W., A. A. Hadi, A. M. Fattouh, and K. M. Jambi. 2017. “LSHADE with Semi-Parameter Adaptation Hybrid with CMA-ES for Solving CEC 2017 Benchmark Problems.” In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, 145–152. Piscataway, NJ: IEEE.

Molina, D., A. Latorre, and F. Herrera. 2018. “SHADE with Iterative Local Search for Large-Scale Global Optimization.” In *2018 IEEE Congress on Evolutionary Computation (CEC)*, 1–8. Piscataway, NJ: IEEE.

Nguyen, T. T., and X. Yao. 2008. “An Experimental Study of Hybridizing Cultural Algorithms and Local Search.” *International Journal of Neural Systems* 18 (1): 1–17. doi:10.1142/S0129065708001397.

Nguyen, S., M. J. Zhang, M. Johnston, and K. C. Tan. 2015. “Automatic Programming via Iterated Local Search for Dynamic Job Shop Scheduling.” *IEEE Transactions on Cybernetics* 45 (1): 1–14. doi:10.1109/TCYB.2014.2317488.

Nobile, M. S., A. Tangherloni, D. Besozzi, and P. Cazzaniga. 2016. “GPU-Powered and Settings-Free Parameter Estimation of Biochemical Systems.” In *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, 32–39. Piscataway, NJ: IEEE.

Nocedal, J., and S. J. Wright. 1999. *Numerical Optimization*. Berlin and New York: Springer-Verlag.

Qin, T., B. Peng, U. Benlic, T. C. E. Cheng, Y. Wang, and Z. P. Lü. 2015. “Iterated Local Search Based on Multi-type Perturbation for Single-Machine Earliness/Tardiness Scheduling.” *Computers & Operations Research* 61: 81–88. doi:10.1016/j.cor.2015.03.005.

Shao, W. S., S. Li, M. Li, Z. S. Shao, and X. F. Ying. 2014. “Hybrid Differential Evolution/Estimation of Distribution Algorithm Based on Adaptive Incremental Learning.” *Journal of Computational Information Systems* 10 (12): 5355–5364. doi:10.12733/jcis10739.

Shao, W. S., and D. C. Pi. 2016. “A Self-Guided Differential Evolution with Neighborhood Search for Permutation Flow Shop Scheduling.” *Expert Systems with Applications* 51: 161–176. doi:10.1016/j.eswa.2015.12.001.

Shao, Z. S., D. C. Pi, and W. S. Shao. 2017. “An Extended Continuous Estimation of Distribution Algorithm for Solving the Permutation Flow-Shop Scheduling Problem.” *Engineering Optimization* 49 (11): 1868–1889. doi:10.1080/0305215x.2016.1275605.

Shi, Z. W., G. Y. Yang, and Y. H. Xiao. 2016. “A Limited Memory BFGS Algorithm for Non-Convex Minimization with Applications in Matrix Largest Eigenvalue Problem.” *Mathematical Methods of Operations Research* 83 (2): 243–264. doi:10.1007/s00186-015-0527-8.

Soria-Alcaraz, J. A., E. Özcan, J. Swan, G. Kendall, and M. Carpio. 2016. “Iterated Local Search Using an Add and Delete Hyper-heuristic for University Course Timetabling.” *Applied Soft Computing* 40: 581–593. doi:10.1016/j.asoc.2015.11.043.

Spolaor, S., A. Tangherloni, L. Rundo, M. S. Nobile, and P. Cazzaniga. 2017. “Reboot Strategies in Particle Swarm Optimization and their Impact on Parameter Estimation of Biochemical Systems.” In *Proceedings of the 2017 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 1–8. Piscataway, NJ: IEEE.

Storn, R., and K. Price. 1997. “Differential Evolution – A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces.” *Journal of Global Optimization* 11 (4): 341–359. doi:10.1023/a:1008202821328.

Stutzle, T. 2006. “Iterated Local Search for the Quadratic Assignment Problem.” *European Journal of Operational Research* 174 (3): 1519–1539. doi:10.1016/j.ejor.2005.01.066.

Tanabe, R., and A. Fukunaga. 2013. “Success-History Based Parameter Adaptation for Differential Evolution.” In *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, 71–78. Piscataway, NJ: IEEE.

Tangherloni, A., L. Rundo, and M. S. Nobile. 2017. “Proactive Particles in Swarm Optimization: A Settings-Free Algorithm for Real-Parameter Single Objective Optimization Problems.” In *Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC)*, 1940–1947. Piscataway, NJ: IEEE.

Tangherloni, A., L. Rundo, S. Spolaor, P. Cazzaniga, and M. S. Nobile. 2018. “GPU-Powered Multi-swarm Parameter Estimation of Biological Systems: A Master-Slave Approach.” In *Proceedings of the 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)*, 698–705. Piscataway, NJ: IEEE.

Wei, L. J., Z. Z. Zhang, D. F. Zhang, and S. C. H. Leung. 2018. "A Simulated Annealing Algorithm for the Capacitated Vehicle Routing Problem with Two-Dimensional Loading Constraints." *European Journal of Operational Research* 265 (3): 843–859. doi:10.1016/j.ejor.2017.08.035.

Wu, G. H., D. Qiu, Y. Yu, W. Pedrycz, M. H. Ma, and H. F. Li. 2014. "Superior Solution Guided Particle Swarm Optimization Combined with Local Search Techniques." *Expert Systems with Applications* 41 (16): 7536–7548. doi:10.1016/j.eswa.2014.06.005.

Xiao, Y. H., Z. X. Wei, and Z. G. Wang. 2008. "A Limited Memory BFGS-Type Method for Large-Scale Unconstrained Optimization." *Computers & Mathematics with Applications* 56 (4): 1001–1009. doi:10.1016/j.camwa.2008.01.028.

Xu, J. Y., C.-C. Wu, Y. Q. Yin, and W.-C. Lin. 2017. "An Iterated Local Search for the Multi-objective Permutation Flowshop Scheduling Problem with Sequence-Dependent Setup Times." *Applied Soft Computing* 52: 39–47. doi:10.1016/j.asoc.2016.11.031.

Zar, J. H. 1999. *Biostatistical Analysis*. 4th ed. New Delhi: Pearson Education India.

Zhang, Q. F., and J. Y. Sun. 2006. "Iterated Local Search with Guided Mutation." In *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation*, 924–929. Piscataway, NJ: IEEE.

Zhao, F. Q., Y. Liu, Y. Zhang, W. M. Ma, and C. Zhang. 2017. "A Hybrid Harmony Search Algorithm with Efficient Job Sequence Scheme and Variable Neighborhood Search for the Permutation Flow Shop Scheduling Problems." *Engineering Applications of Artificial Intelligence* 65: 178–199. doi:10.1016/j.engappai.2017.07.023.

Zhao, F. Q., H. Liu, Y. Zhang, W. M. Ma, and C. Zhang. 2018. "A Discrete Water Wave Optimization Algorithm for No-Wait Flow Shop Scheduling Problem." *Expert Systems with Applications* 91: 347–363. doi:10.1016/j.eswa.2017.09.028.

Zhao, F. Q., Y. Liu, C. Zhang, and J. B. Wang. 2015. "A Self-Adaptive Harmony PSO Search Algorithm and its Performance Analysis." *Expert Systems with Applications* 42 (21): 7436–7455. doi:10.1016/j.eswa.2015.05.035.

Zhou, Y. M., and J.-K. Hao. 2017. "An Iterated Local Search Algorithm for the Minimum Differential Dispersion Problem." *Knowledge-Based Systems* 125: 26–38. doi:10.1016/j.knosys.2017.03.028.