

Re-sampled inheritance search: high performance despite the simplicity

Fabio Caraffini · Ferrante Neri ·
Benjamin N. Passow · Giovanni Iacca

Published online: 18 August 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract This paper proposes re-sampled inheritance search (RIS), a novel algorithm for solving continuous optimization problems. The proposed method, belonging to the class of Memetic Computing, is very simple and low demanding in terms of memory employment and computational overhead. The RIS algorithm is composed of a stochastic

sample mechanism and a deterministic local search. The first operator randomly generates a solution and then recombines it with the best solution detected so far (inheritance) while the second operator searches in an exploitative way within the neighbourhood indicated by the stochastic operator. This extremely simple scheme is shown to display a very good performance on various problems, including hard to solve multi-modal, highly-conditioned, large scale problems. Experimental results show that the proposed RIS is a robust scheme that competitively performs with respect to recent complex algorithms representing the-state-of-the-art in modern continuous optimization. In order to further prove its applicability in real-world cases, RIS has been used to perform the control system tuning for yaw operations on a helicopter robot. Experimental results on this real-world problem confirm the value of the proposed approach.

Communicated by G. Acampora.

This research is supported by the Academy of Finland, Akatemiaturkija 130600, “Algorithmic design issues in Memetic Computing”. INCAS³ is co-funded by the Province of Drenthe, the Municipality of Assen, the European Fund for Regional Development and the Ministry of Economic Affairs, Peaks in the Delta.

F. Caraffini · F. Neri
Centre for Computational Intelligence (CCI),
School of Computer Science and Informatics, De Montfort
University, The Gateway, Leicester LE1 9BH, UK
e-mail: fabio.caraffini@email.dmu.ac.uk

F. Neri
e-mail: fneri@dmu.ac.uk

F. Caraffini · F. Neri (✉)
Department of Mathematical Information Technology, University
of Jyväskylä, P.O. Box 35 (Agora), 40014 Jyväskylä, Finland
e-mail: ferrante.neri@jyu.fi

F. Caraffini
e-mail: fabio.caraffini@jyu.fi

B. N. Passow
DMU’s Interdisciplinary Group in Intelligent Transport Systems
(DIGITS), School of Computer Science and Informatics,
De Montfort University, The Gateway, Leicester LE1 9BH, UK
e-mail: benpassow@dmu.ac.uk

G. Iacca
INCAS 3-Innovation Centre for Advanced Sensors and Sensor Systems,
P.O. Box 797, 9400 AT Assen, The Netherlands
e-mail: giovanniacca@incas3.eu

Keywords Memetic computing · Ockham’s Razor ·
Computational intelligence optimization · Large scale
optimization · Control system design · Autonomous
helicopter

1 Introduction

Memetic Computing (MC) is a subject in computational science that studies algorithmic structures composed of heterogeneous operators, (Neri et al. 2011b). On an abstract level, every set of multiple operators coordinated within a certain structure for solving a given problem can be seen as a MC approach. Although this definition may appear excessively broad (Ong et al. 2010; Neri and Cotta 2012), in our view the description of algorithms as structured collections of operators (*memes*) has both conceptual and practical implications.

The concept of MC originates with the definition of Memetic Algorithm (MA), (see Moscato and Norman 1989;

Moscato 1989). The first MAs were simple Genetic Algorithms (GAs) hybridized with a local search component for tackling the Travelling Salesman Problem (TSP). Although the idea of hybridizing algorithms was not totally new, (see e.g. Goldberg 1989), the visionary representation of the transmission of knowledge amongst subcomponents of an algorithm inspired a long-lasting discussion in the computer science community. An ex-post formalization of the definition of MA was given in Hart et al. (2004) where a MA is stated to be an algorithm composed of an evolutionary framework and one (or more) local search components activated within the generation cycle of the external framework. Several MAs (according to this definition) have been successfully used in various fields of applied science and engineering.

For example, in Joshi and Sanderson (1999) an ad-hoc MA based on a Differential Evolution (DE) framework has been proposed for solving the multi-sensor fusion problem while in Rogalsky and Derksen (2000) another DE based MA is designed to tackle an aerodynamic design problem. In Zamuda et al. (2011) a DE scheme for plant model reconstruction is proposed. A memetic solution for studying a material structure is given in Fan et al. (2007). In Caponio et al. (2007), Neri and Mininno (2010) domain-specific MAs are proposed for solving a control engineering problem with reference to electric motors and robotics. In Ong and Keane (2004) an aerodynamic design problem is considered. Biological and medical problems are addressed by means of MC approaches are also very popular, (see e.g. Abbass 2002; Neri et al. 2007a,b). MAs for scheduling and planning problems have been proposed (in e.g. Hasan et al. 2009; Lim et al. 2008; Tan et al. 2007).

During the latest years, MAs have become very popular in multiple contexts. As highlighted in Neri and Cotta (2012), one important reason behind the MA success is the diffusion of the No Free Lunch Theorems (NFLTs) (Wolpert and Macready 1997). These theorems prove that the average performance of any pair of algorithms A and B across all possible problems is identical. Strictly speaking, the proof of NFLTs (Auger and Teytaud 2007) are made for discrete problems and under the hypothesis that both the algorithms A and B are non-revisiting, i.e. the algorithms do not perform the fitness evaluation of the same candidate solution more often than once during the optimization run. Although a rigorous verification of these hypotheses is often not realistic, computer scientists accepted the idea that there is no universal optimizer because if an algorithm performs well on a certain class of problems, then it necessarily pays for that with degraded performance on the set of all remaining problems. As a trivial consequence, each problem should be analysed and a proper algorithm that addresses the specific features of that problem should be designed. Considering the non-specificity of the general structure of a MA, practitioners

soon realized that proper hybridizations aiming at addressing specific problem features (such as noise, multi-modality etc.) would have helped them to solve problems that appeared hard for traditional paradigms.

Memetic Computing (MC) extends the concept of MA taking into account algorithms that are not population-based, (see e.g. Neri et al. 2011a), or that employ external support operators such as machine learning components, (see e.g. Handoko et al. 2010). In other words MC is an umbrella name which includes many modern algorithms composed of multiple operators and would not fit within the MA definition given in Hart et al. (2004). From a philosophical viewpoint, however, the real breakthrough of MC is the view of algorithms as structures whose bricks composing them are operators: this idea opens, for example, the exciting perspective of automatic algorithm generation, (see Neri et al. 2011b; Ong et al. 2010; Meuth et al. 2009).

The presence of multiple components and coordination schemes usually makes MC approaches rather complex and demanding in terms of computational overhead. For example, paper (Molina et al. 2010a) proposes a MA that, although very efficient, requires the usage of multiple covariance matrices, thus resulting into a very high computational overhead and memory employment, especially in high dimensions. In Montes de Oca et al. (2009) a Particle Swarm Optimization (PSO) that includes multiple enhancing mechanisms collected by other variants in literature is proposed. In Vrugt et al. (2009); Peng et al. (2010) optimization algorithms composed of multiple popular meta-heuristics are proposed. The coordinated employment of multiple algorithms, and more generally multiple strategies, is encompassed within the idea of *ensemble* (Mallipeddi et al. 2010, 2011), where different strategies concur, by means of a self-adaptive or randomized mechanism, to the optimization of the same fitness function. A similar approach is proposed in Zamuda and Brest (2012) where multiple strategies are combined with a population size reduction to in order to tackle industrial problems. In Nguyen et al. (2009a), yet multiple algorithms are considered, while their coordination is performed by means of a success probability criterion (by following a principle similar to the meta-Lamarckian learning (Ong and Keane 2004). In Nguyen et al. (2009b) multiple algorithmic components are coordinated by means of the structural mapping of the population.

Although in some cases, complex algorithmic implementations can lead to successful results, unnecessary complexity during the design phase should be strictly avoided for the following four major reasons.

- (1) Algorithms composed of multiple parts and containing many parameters could be *hard to control*. The setting of many parameters whose values heavily affects the per-

formance (for a given problem) is often a complicated issue since the optimal setting of each parameter is likely to depend on the setting of the other parameters, (see [Eiben and Smit 2011](#)).

- (2) Complex algorithms can be *hard to understand* in terms of functioning. More specifically, some algorithms despite their performance on some problems are so complex that it is nearly impossible to interpret their functioning and understand the reasons behind their success. If there is no proper understanding of the working principles of an algorithm, the scheme risks to be highly specialized and unexpectedly fail when the problem changes. While the specialization is not necessarily a drawback of an algorithm, the lack of understanding of the algorithmic working principles and thus the difficulty of taking efficient countermeasures to adapt an algorithm to a new situation (e.g. a new dimensionality value) is definitely a limitation of complex schemes.
- (3) Some complex algorithms include computationally expensive components. This may result into a *high computational overhead* which may super-linearly depend on the problem dimensionality. For example, those algorithms that make use of covariance or distance matrices are characterized by a computational overhead that grows quadratically with the dimensionality of the problem. These algorithms may be unacceptably expensive in large scale optimization problems.
- (4) In other cases, modern algorithms require machine learning structures, archives, and learning components for the supervision of the operators. In such cases, the algorithm can be *expensive in terms of memory consumption*, thus being impractical for those problems that are characterized by a limited hardware, such as micro-controllers and embedded systems. Obviously, some algorithms can present both, a high computational overhead and high memory requirement.

Motivated by these reasons, paper ([Iacca et al. 2012a](#)) introduces the concept of Ockham's Razor in MC, stating that unnecessary complex structures should be avoided as properly designed, simple algorithms can perform as well as complex ones. In addition, in [Iacca et al. \(2012a\)](#) an implementation of a novel algorithm, namely Three Stage Optimal Memetic Exploration (3SOME), is proposed. The 3SOME algorithm is a fairly simple scheme that makes use of three operators to progressively perturb a single solution. Despite its simplicity, 3SOME displays a competitive performance with respect to other MC approaches, and modern population-based algorithms. This successful implementation has been further studied and improved. Paper ([Neri et al. 2012](#)) compares the simplistic meme coordination scheme of 3SOME with the meta-Lamarckian learning ([Ong and Keane 2004](#)) and concludes that the simplistic 3SOME coordination

is not worse than the adaptive leaning. Papers ([Poikolainen et al. 2012a](#); [Caraffini et al. 2012a](#)) propose the integration of components for handling non-separability within a 3SOME framework. A marginally improved version of the original 3SOME is presented in [Poikolainen et al. \(2012b\)](#), where a modified operator enhances the exploitation features of the algorithm. Finally, paper ([Caraffini et al. 2012b](#)) studies the 3SOME algorithmic structure by comparing 3SOME variants holding the same structure but encompassing different operators. As a result, algorithms with the same structure but different operators appeared to display an overall similar performance over various problems.

It is worth mentioning that a way before the formulation of the Ockham's Razor in algorithmic contexts and the diffusion of the MC terminology, simple structures perturbing single solutions by means of diverse operators have been proposed in the literature. For example, in [Yao et al. \(1999\)](#) a modified Evolutionary Programming (EP) scheme that cooperatively-competitively makes use of both Gaussian and Cauchy distributions in order to generate a new trial individuals. An evolution of this approach is presented in [Lee and Yao \(2004\)](#) where the EP is empowered by the Lévy distribution within the mutation operator.

The present paper, by following the Ockham's Razor principle presented in [Iacca et al. \(2012a\)](#) and the considerations of the importance of the algorithmic structures reported in [Caraffini et al. \(2012b\)](#), proposes a novel algorithmic implementation that further attempts to be a simple and efficient alternative to modern complex algorithms. The proposed algorithm, namely re-sampled inheritance search (RIS), makes use of only two operators that progressively perturb a single solution, combined in a simplified structure with respect to that studied in [Caraffini et al. \(2012b\)](#). One of these operators is a modified version of one local search component used in [Tseng and Chen \(2008\)](#) while the second is a re-sampling mechanism flowed by an exponential crossover implemented in the fashion of DE ([Price et al. 2005](#)).

In addition, this paper applies the proposed algorithm to the control problem of a helicopter robot. In this case, a real-world application, the autonomous nature of the hardware would impose an on-board implementation of the optimization algorithm. The system would benefit from a simple algorithm that is characterized by a modest memory requirement and computational overhead. This real-world example highlights how the proposed algorithm is a simple scheme yet capable to display a performance competitive with modern complex algorithms.

The remainder of this article is organized in the following way. Section 2 introduces the RIS components, their coordination, and explains the motivation behind the design choices. Section 3 presents, on a diverse set of test problems belonging to four popular benchmarks, the RIS performance with respect to that of modern algorithm. Section 4 describes

the application of the proposed algorithm to a real-world engineering problem in the field of mobile robotics. Finally, Sect. 5 gives the conclusions of this study.

2 Re-sampling inheritance search

Without loss of generality, in the following we refer to the minimization problem of an objective function $f(\mathbf{x})$, where the candidate solution \mathbf{x} is a vector of n design variables (or genes) in a hyper-box decision space $\mathbf{D} = [\mathbf{a}, \mathbf{b}]$, with \mathbf{a} and \mathbf{b} respectively lower and upper bound vectors. Let us indicate with $x[i]$ the i th element of the vector \mathbf{x} . At an abstract level, an optimization algorithm can be seen as a mathematical procedure that progressively perturbs one or more candidate solutions in order to detect the optimum of the objective function. Let us indicate with \mathbf{x}_e (where e stands for “elite”) the best solution (or population of best solutions) detected at a given moment of the search, and with \mathbf{x}_t the trial solution(s), i.e. the candidate solution perturbed by an operator (or a set of operators).

The proposed Re-sampling Inheritance Search (RIS) is an extremely simple algorithm that makes use of two operators to perturb a single solution. The proposed algorithm randomly samples an initial solution \mathbf{x}_e within the decision space \mathbf{D} . The two operators proposed in the following two sub-sections are applied in order to perturb \mathbf{x}_e .

2.1 Re-sampling with inheritance

This operator, at first, randomly generates a solution \mathbf{x}_t within the decision space \mathbf{D} . Then, a perturbation of \mathbf{x}_e is performed by means of the exponential crossover in the fashion of Differential Evolution (Neri and Tirronen 2010; Zaharie 2009). More specifically, one gene from \mathbf{x}_e is randomly selected. This gene replaces the corresponding gene within the trial solution \mathbf{x}_t . Then, a set of random numbers between 0 and 1 are generated. As long as $\text{rand}(0, 1) \leq Cr$, where the crossover rate Cr is a parameter affecting the number of transferred genes (see below), the design variables from the elite \mathbf{x}_e are copied into the corresponding positions of the trial solution \mathbf{x}_t , starting from the initial gene. As soon as $\text{rand}(0, 1) > Cr$, the copy process is interrupted. Thus, all the remaining design variables of the offspring are those initially sampled (belonging to the original \mathbf{x}_t). The individual is handled as a cyclic buffer, i.e. when the n th variable is reached during the copy process the next to be copied is the first one. When the trial solution \mathbf{x}_t has been generated, its fitness is compared with that of \mathbf{x}_e . If the newly generated solution outperforms the elite, an elite replacement occurs. The pseudo-code displaying the working principles of re-sampling with inheritance is shown in Fig. 1.

```

global  $\mathbf{x}_e$ 
generate a random (uniform distribution) solution  $\mathbf{x}_t$  within  $\mathbf{D}$ 
generate  $i = \text{round}(n \cdot \text{rand}(0, 1))$ 
compute  $Cr$  according to eq. (3)
 $x_t[i] = x_e[i]$ 
 $k = 1$ 
while  $\text{rand}(0, 1) \leq Cr$  AND  $k < n$  do
   $x_t[i] = x_e[i]$ 
   $i = i + 1$ 
  if  $i == n$  then
     $i = 1$ 
  end if
   $k = k + 1$ 
end while
if  $f(\mathbf{x}_t) \leq f(\mathbf{x}_e)$  then
   $\mathbf{x}_e = \mathbf{x}_t$ 
end if

```

Fig. 1 Pseudo-code of re-sampling with inheritance

As shown in Neri et al. (2011a), it can easily be observed that for a given value of Cr , the effect of the exponential crossover changes with the dimensionality of the problem. For low-dimensional problems, the trial solution would inherit most of the genes from the elite, while for higher dimensionalities only a small portion of \mathbf{x}_e would be copied into \mathbf{x}_t . In order to avoid this problem and make the crossover action independent on the dimensionality of the problem, the following quantity, namely inheritance factor, is fixed:

$$\alpha_e \approx \frac{n_e}{n} \quad (1)$$

where n_e is the number of genes we expect to copy from \mathbf{x}_e into \mathbf{x}_t in addition to the first gene, which is deterministically copied. The probability that n_e genes are copied is $Cr^{n_e} = Cr^{n\alpha_e}$. In order to control the approximate amount of copied genes and to achieve that about n_e genes are copied into the offspring with probability 0.5, we impose that:

$$Cr^{n\alpha_e} = 0.5. \quad (2)$$

It can easily be seen that, for a chosen α_e , the crossover rate can be set on the basis of the dimensionality as follows:

$$Cr = \frac{1}{n^{\alpha_e \sqrt{2}}}. \quad (3)$$

By means of formula (3), the expected quantity of information to be inherited from \mathbf{x}_e to \mathbf{x}_t is thus controlled.

2.2 Exploitative local search

This operator is a local search algorithm which perturbs a single solution along its n axes, i.e. separately perturbs each design variable. Other search operators that separately

perturb each variable have been extensively proposed in the literature. Some examples that, unlike in the present paper, make use of a randomization are given in Zhou et al. (2008), Ji and Klinowski (2006). The meme here proposed can be seen as a modification of a classical hill-descend algorithm and employs the perturbation logic proposed in Tseng and Chen (2008).

The implementation of this operator requires an additional solution, which will be here referred to as \mathbf{x}_s . The trial solution \mathbf{x}_t generated by the first operator is perturbed by computing, for each variable i :

$$x_s[i] = x_t[i] - \rho[i], \tag{4}$$

where ρ is an n -dimensional exploratory radius vector. The elements of ρ are reinitialized to a predetermined initial value whenever the local search is activated. Subsequently, if \mathbf{x}_s outperforms \mathbf{x}_t , the trial solution \mathbf{x}_t is updated (the values of \mathbf{x}_s are copied into it), otherwise a half step in the opposite direction is performed:

$$x_s[i] = x_t[i] + \frac{\rho[i]}{2}. \tag{5}$$

Again, \mathbf{x}_s replaces \mathbf{x}_t if it outperforms it. After all the variables have been perturbed, the elite \mathbf{x}_e is replaced by \mathbf{x}_t if it is outperformed by it. If the elite is not updated, i.e. the exploration was unsuccessful, the radius ρ is halved for all variables. The exploration is then repeated again for all the design variables, until a precision criterion is met. In particular, the operator is stopped when the 2-norm of ρ , normalized per each element by the corresponding search interval, is smaller than a fixed threshold, as follows:

$$\sqrt{\sum_{i=1}^n \left(\frac{\rho[i]}{b[i] - a[i]} \right)^2} < \varepsilon \tag{6}$$

where $(b[i] - a[i])$ is the width of the decision space \mathbf{D} along the i th dimension and ε is the pre-arranged constant. For the sake of clarity, Fig. 2 displays, in a pseudo-code, the working principles of the exploitative local search.

As a further remark, RIS applies a toroidal management of the bounds. This means that if, along the dimension i , the design variable $x[i]$ exceeds the bounds of a value ζ , it is reinserted from the other end of the interval at a distance ζ from the edge, i.e. given an interval $[a, b]$, if $x[i] = b + \zeta$ it takes the value of $a + \zeta$.

2.3 Algorithmic structure and philosophy

The combination of the two memes composing the RIS is arranged straightforwardly. More specifically, a solution is firstly processed by the re-sampling with inheritance and then the outcoming solution \mathbf{x}_t is processed by the exploitative local search. The elite solution \mathbf{x}_e , which is possibly

```

global  $\mathbf{x}_e$ 
while (condition (6) is not met
and stop criterion is not met) do
 $\mathbf{x}_s = \mathbf{x}_t$ 
for ( $i = 1 : n$  and stop criterion is not met) do
 $x_s[i] = x_t[i] - \rho[i]$ 
if  $f(\mathbf{x}_s) \leq f(\mathbf{x}_t)$  then
 $\mathbf{x}_t = \mathbf{x}_s$ 
else
 $x_s[i] = x_t[i] + \frac{\rho[i]}{2}$ 
if  $f(\mathbf{x}_s) \leq f(\mathbf{x}_t)$  then
 $\mathbf{x}_t = \mathbf{x}_s$ 
end if
end for
if  $f(\mathbf{x}_t) \leq f(\mathbf{x}_e)$  then
 $\mathbf{x}_e = \mathbf{x}_t$ 
else
 $\rho = \frac{\rho}{2}$ 
end if
end while
    
```

Fig. 2 Pseudo-code of the exploitative local search

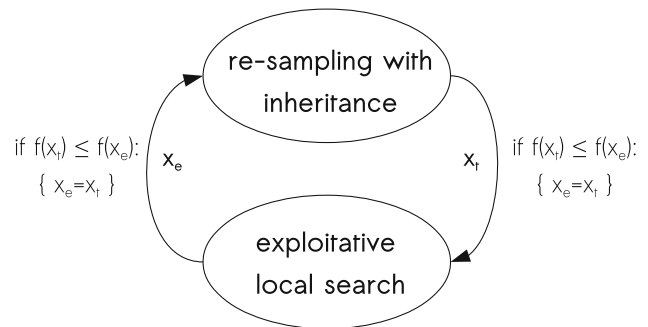


Fig. 3 Graphical representation of the RIS structure

```

randomly sample  $\mathbf{x}_e$  in the search space  $\mathbf{D}$ 
while stop criterion is not met do
perform re-sampling with inheritance on  $\mathbf{x}_e$  as in Fig. 1
perform exploitative local search on  $\mathbf{x}_t$  as in Fig. 2
end while
    
```

Fig. 4 Pseudo-code the RIS

the solution \mathbf{x}_t processed by the exploitative local search is then given back to the re-sampling inheritance for further improvement. A pseudo-code description of the RIS algorithmic structure is reported in Fig. 4 and graphically depicted in Fig. 3. The re-sampling mechanism is supposed

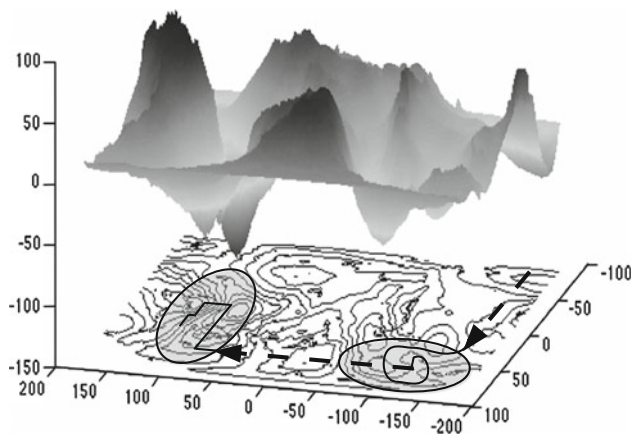


Fig. 5 Graphical representation of the RIS functioning

to generate a solution which is far away from the current elite while the local search exploits the area of the decision space suggested by the re-sampling operator. In this sense, the proposed RIS is nothing else but a simple multi-start local search. However, the proposed scheme is thought as a global optimization algorithm in the fashion of MC. The inheritance mechanism assures that a part of the genotype of the most promising candidate solution is used to enhance upon its performance, (see [Iacca et al. 2012a](#)). Although the re-sampling is an operation that is performed only occasionally and thus has a limited budget devoted to it (with respect to that allotted to the local search), the transmission of some variables from a promising solution to a newly sampled point appears to have a certain impact to the global performance of the algorithm, see results in Sect. 3.1 and Table 13. For a large experimental setup, it has been observed that the RIS version with inheritance is always at least as good as the version without inheritance.

Figure 5 shows the search mechanism of the RIS in a bi-dimensional case. Dashed lines show the search moves performed by the re-sampling mechanism with inheritance, while solid lines represent the search logic of the exploitative local search. The re-sampling mechanism in general performs diagonal moves. The movement along the axes are due to the fact that the representation is bi-dimensional and thus only one variable is perturbed into the trial solution. Obviously, in multi-dimensional cases a portion of the elite is inherited by \mathbf{x}_t while the moves are performed diagonally. As a fundamental remark, the proposed RIS has been designed by following a bottom-up strategy, (see [Iacca et al. 2012a](#)), i.e. building up the algorithmic structure from scratch and adding one operator at the time until a good performance is achieved. The resulting algorithm is indeed extremely simple and appears to be in line with the Ockham's Razor principle for MC structures formulated in [Iacca et al. \(2012a\)](#).

3 Numerical results

The proposed RIS algorithm has been run with the following parameter setting. The inheritance factor α_e , Eq. (1), has been set equal to 0.5. Regarding the local searcher, the initial search radius $\rho[i]$, see Eqs. (4) and (5), as in [Tseng and Chen \(2008\)](#) has been set equal to $0.4 \times (b[i] - a[i])$, i.e. 40 % of the domain width along each variable i , while the stop-threshold ε has been set equal to 10^{-6} . This configuration of the parameters ensures significant performances over a considerable variety of test problems. In particular, all the algorithms under study have been run over:

- The CEC2005 benchmark described in [Suganthan et al. \(2005\)](#) in 30 dimensions (25 test problems)
- The BBOB2010 benchmark described in [Hansen et al. \(2010\)](#) in 100 dimensions (24 test problems)
- The CEC2008 benchmark described in [Tang et al. \(2007\)](#) in 1000 dimensions (7 test problems)
- The CEC2010 benchmark described in [Tang et al. \(2010\)](#) in 1000 dimensions (20 test problems)

Thus, 76 test problems have been considered in this study. For each algorithm in this paper (see following subsections) 100 runs have been performed. Each run has been continued for $5000 \times n$ fitness evaluations, where n is the dimensionality of the problem. For each test problem and each algorithm, the average final fitness value \pm standard deviation over the 100 available runs has been computed. In order to strengthen the statistical significance of the results, for each test problem the Wilcoxon Rank-Sum test ([Wilcoxon 1945](#)) has been also applied, with a confidence level of 0.95.

The following algorithms with respective parameter setting have been considered for comparison against RIS.

- Three Stage Optimal Memetic Exploration (3SOME) proposed in [Iacca et al. \(2012a\)](#) with inheritance factor $\alpha_e = 0.05$, middle distance exploration hyper-cube size δ equal to 20 % of the total decision space width, coefficient of generated points at each activation of the middle distance exploration $k = 4$, short distance exploration radius $\rho = 0.4$ and local budget fixed to 150 iterations.
- Comprehensive Learning Particle Swarm Optimizer (CLP SO) proposed in [Liang et al. \(2006\)](#) with population size equal to 60 individuals.
- Adaptive Differential Evolution (JADE) proposed in [Zhang and Sanderson \(2009\)](#) with population size equal to 60 individuals, group size factor $p = 0.05$ and parameters adaptation rate factor $c = 0.1$.
- Cooperatively Coevolving Particle Swarms Optimizer (CCPSO2) proposed in [Li and Yao \(2012\)](#) with population size equal to 30 individuals, Cauchy/Gaussian

- sampling selection probability $p = 0.5$ and set of potential group sizes $S = \{2, 5, 10\}$, $S = \{2, 5, 10, 50, 100\}$, $S = \{2, 5, 10, 50, 100, 250\}$ for experiments in 30, 100 and 1000 dimensions, respectively.
- Memetic Algorithm with CMA-ES Chains (MA-CMA-Chains) proposed in [Molina et al. \(2010a\)](#) with population size equal to 60 individuals, probability of updating a chromosome by mutation equal to 0.125, local/global search ratio $r_{\frac{L}{G}} = 0.5$, BLX- α crossover with $\alpha = 0.5$, n_{ass} parameter for Negative Assortative Mating set to 3, LS intensity stretch $I_{str} = 500$ and threshold $\delta_{LS}^{min} = 10^{-8}$.
 - Modified Differential Evolution with p-Best Crossover (MDE-pBX) proposed in [Islam et al. \(2012\)](#) with population size equal to 100 individuals and group size q equal to 15 % of the population size.
 - Parallel Memetic Structure (PMS) proposed in [Caraffini et al. \(2013\)](#) with inheritance factor α_e set equal to 0.95, initial search radius ρ equal to 0.4 and computational budget for the first local searcher set to 150 iterations. Regarding the second local searcher h is initialised as a vector of 0.1, $\alpha = 2$, $\beta = 0.5$, while ε has been set equal to 10^{-5} .
 - compact Differential Evolution (cDE) proposed in [Mininno et al. \(2011\)](#) with rand/1 mutation and exponential crossover, cDE/rand/1/exp, virtual population size equal to 300, scale factor $F = 0.5$, and proportion of genes undergoing exponential crossover, (see [Neri et al. 2011a](#), $\alpha_m = 0.25$).

It must be remarked that the MA-CMA-Chains employs multiple covariance matrices, (see [Hansen et al. 2003](#)). Thus, its memory requirement and computational overhead dramatically grow with the problem dimensionality. In order to tackle large scale problems, ([Lozano et al. 2011](#)) suggested a tailored version which uses an efficient local search for high dimensional domains, namely:

- Memetic Algorithm with Subgrouping Solis Wets Chains (MA-SSW-Chains), originally proposed in [Molina et al. \(2010b\)](#), with population size equal to 100 individuals, probability of updating a chromosome by mutation equal to 0.125, local/global search ratio $r_{\frac{L}{G}} = 0.5$, BLX- α crossover with $\alpha = 0.5$, n_{ass} parameter for Negative Assortative Mating set to 3, LS intensity stretch $I_{str} = 500$ and threshold $\delta_{LS}^{min} = 0$.

Thus for the experiments in 1000 dimensions, MA-SSW-Chains has been used instead of MA-CMA-Chains.

The parameter setting for all the algorithms in this subsection has been carried out by using the parameters suggested in the respective original articles. As for CLPSO and JADE,

the population size was set in the original papers depending on the problem dimensionality (without a general rule). In this study, a tuning of the population size values has been performed, resulting in a size of 60 individuals which turned out to be the most suitable compromise in terms of overall performance. This value is in accordance with the study of [Lozano et al. \(2011\)](#) for setting the Differential Evolution population size.

Experimental results have been divided into groups. Tables 1, 2, 3, and 4 show the comparison against 3SOME, CLPSO, and JADE for the four benchmarks under consideration. Tables 5, 6, 7, and 8 show the comparison against CCPSO2, MA-CMA-Chains (MA-SSW-Chains), and MDE-pBX. Tables 9, 10, 11, and 12 show the comparison against cDE and PMS. The tables in this study display the average final fitness value over the 100 available runs and the corresponding standard deviation. The results of the Wilcoxon test are also reported in terms of pair-wise comparisons. The symbols “=” and “+” (“-”) indicate, respectively, a statistically equivalent performance and a better (worse) performance of RIS compared with the algorithm in the column label. The best results are highlighted in bold.

Despite its simple structure, the RIS algorithm outperforms both the popular meta-heuristics (CLPSO and JADE) under consideration. In particular, RIS overtakes CLPSO and JADE in 30 and 100 dimensions, while in 1000 dimensions CLPSO is competitive over the testbed in [Tang et al. \(2010\)](#). With reference to Table 1, it can be seen that for the testbed in [Suganthan et al. \(2005\)](#) the RIS algorithm is outperformed by all the other algorithms only in function f_4 which, being subjected to a Gaussian noise ($\mathcal{N}(0, 1)$), is more suited for a population-based algorithm ([Arnold and Beyer 2003](#)). It can also be observe that RIS achieves the best performance in 9 cases, 3SOME also in 9 cases, JADE in 8 case while CLPSO in only one case. Nonetheless, regarding problems f_1 , f_2 , f_9 , f_{18} , f_{19} , and f_{20} , despite the fact that 3SOME is possibly slightly more robust, RIS and 3SOME detect very similar solutions. On the contrary, in most of the cases where the RIS algorithm appears to be promising against the 3SOME scheme, there is an important margin of difference in terms of final fitness value. Equally relevant results are displayed in Table 2, for the testbed in [Hansen et al. \(2010\)](#) in 100 dimensions, and in Table 3 for the testbed in [Tang et al. \(2007\)](#) in 1000 dimensions. In particular, Table 3 highlights an extremely good behavior of the proposed algorithm over large-scale separable problems. In fact, RIS widely outperforms JADE on a regular basis and is significantly outperformed by CLPSO in only one case (see function f_4). With respect to the 3SOME algorithm, RIS appears to detect better solutions in most of the analysed cases. It should be remarked that RIS makes use of two of the operators contained in the 3SOME framework but combines them according to a slightly different logic/structure. In this sense, RIS

Table 1 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against its predecessor 3SOME and popular meta-heuristics on CEC2005 (Suganthan et al. 2005) in 30 dimensions

	RIS	3SOME		CLPSO		JADE	
f_1	$-4.50e + 02 \pm 1.31e - 13$	$-4.50e + 02 \pm 7.59e - 14$	-	$-4.50e + 02 \pm 6.12e - 05$	+	$-3.51e + 02 \pm 1.99e + 02$	+
f_2	$-4.50e + 02 \pm 1.23e - 10$	$-4.50e + 02 \pm 2.23e - 12$	-	$6.44e + 03 \pm 9.22e + 02$	+	$3.08e + 02 \pm 7.32e + 02$	+
f_3	$2.03e + 05 \pm 1.25e + 05$	$2.80e + 05 \pm 1.70e + 05$	+	$2.55e + 07 \pm 6.25e + 06$	+	$3.71e + 06 \pm 1.77e + 06$	+
f_4	$1.88e + 04 \pm 5.57e + 03$	$2.27e + 04 \pm 1.13e + 04$	+	$1.43e + 04 \pm 1.70e + 03$	-	$2.27e + 03 \pm 1.68e + 03$	-
f_5	$3.48e + 03 \pm 7.70e + 02$	$1.08e + 04 \pm 3.27e + 03$	+	$7.14e + 03 \pm 5.68e + 02$	+	$3.91e + 03 \pm 9.18e + 02$	+
f_6	$6.91e + 02 \pm 4.39e + 02$	$5.43e + 02 \pm 2.64e + 02$	-	$6.08e + 02 \pm 4.97e + 01$	-	$5.07e + 06 \pm 1.53e + 07$	+
f_7	$-1.80e + 02 \pm 3.38e - 03$	$-1.80e + 02 \pm 1.13e - 02$	+	$1.41e + 11 \pm 7.45e + 10$	+	$1.69e + 13 \pm 1.78e + 13$	+
f_8	$-1.20e + 02 \pm 4.05e - 04$	$-1.20e + 02 \pm 7.40e - 04$	+	$-1.19e + 02 \pm 5.05e - 02$	+	$-1.19e + 02 \pm 5.75e - 02$	+
f_9	$-1.18e + 02 \pm 1.63e - 05$	$-1.18e + 02 \pm 7.73e - 14$	-	$-1.15e + 02 \pm 8.97e - 01$	+	$-1.17e + 02 \pm 1.22e + 00$	+
f_{10}	$2.14e + 02 \pm 1.52e + 01$	$2.87e + 02 \pm 2.99e + 01$	+	$2.73e + 02 \pm 1.22e + 01$	+	$2.02e + 02 \pm 2.20e + 01$	-
f_{11}	$1.09e + 02 \pm 1.90e + 00$	$1.22e + 02 \pm 4.59e + 00$	+	$1.19e + 02 \pm 1.69e + 00$	+	$1.16e + 02 \pm 4.48e + 00$	+
f_{12}	$1.90e + 02 \pm 1.41e + 03$	$1.27e + 03 \pm 3.33e + 03$	+	$4.46e + 04 \pm 9.39e + 03$	+	$1.72e + 04 \pm 1.54e + 04$	+
f_{13}	$-1.21e + 02 \pm 1.92e + 00$	$-1.25e + 02 \pm 1.19e + 00$	-	$-1.20e + 02 \pm 8.21e - 01$	+	$-1.26e + 02 \pm 9.64e - 01$	-
f_{14}	$-2.86e + 02 \pm 2.79e - 01$	$-2.86e + 02 \pm 3.31e - 01$	-	$-2.87e + 02 \pm 1.65e - 01$	-	$-2.87e + 02 \pm 2.02e - 01$	-
f_{15}	$1.44e + 03 \pm 3.89e - 01$	$1.44e + 03 \pm 8.77e - 01$	=	$1.46e + 03 \pm 2.18e + 00$	+	$1.45e + 03 \pm 3.45e + 00$	+
f_{16}	$1.55e + 03 \pm 7.51e + 00$	$1.62e + 03 \pm 2.78e + 01$	+	$1.61e + 03 \pm 4.94e + 00$	+	$1.56e + 03 \pm 6.50e + 00$	+
f_{17}	$1.66e + 03 \pm 1.54e + 01$	$1.67e + 03 \pm 1.88e + 01$	+	$1.67e + 03 \pm 6.15e + 00$	+	$1.59e + 03 \pm 7.53e + 00$	-
f_{18}	$9.10e + 02 \pm 4.58e - 10$	$9.10e + 02 \pm 5.90e - 12$	-	$9.10e + 02 \pm 3.54e - 05$	+	$9.10e + 02 \pm 2.62e - 01$	+
f_{19}	$9.10e + 02 \pm 4.52e - 10$	$9.10e + 02 \pm 5.17e - 12$	-	$9.10e + 02 \pm 3.06e - 05$	+	$9.10e + 02 \pm 1.31e - 01$	+
f_{20}	$9.10e + 02 \pm 4.29e - 10$	$9.10e + 02 \pm 4.76e - 12$	-	$9.10e + 02 \pm 3.09e - 05$	+	$9.10e + 02 \pm 1.40e - 01$	+
f_{21}	$1.69e + 03 \pm 4.33e + 00$	$1.73e + 03 \pm 1.27e + 01$	+	$1.72e + 03 \pm 3.69e + 00$	+	$1.69e + 03 \pm 4.32e + 00$	=
f_{22}	$2.42e + 03 \pm 2.92e + 01$	$2.66e + 03 \pm 8.39e + 01$	+	$2.55e + 03 \pm 1.91e + 01$	+	$2.29e + 03 \pm 3.44e + 01$	-
f_{23}	$1.72e + 03 \pm 6.16e + 00$	$1.72e + 03 \pm 9.82e + 00$	=	$1.77e + 03 \pm 5.94e + 00$	+	$1.70e + 03 \pm 4.48e + 00$	-
f_{24}	$1.68e + 03 \pm 6.83e + 00$	$1.72e + 03 \pm 1.13e + 01$	+	$1.70e + 03 \pm 6.21e + 00$	+	$1.66e + 03 \pm 1.40e + 01$	-
f_{25}	$1.43e + 03 \pm 3.74e + 02$	$1.66e + 03 \pm 3.97e + 02$	+	$1.89e + 03 \pm 8.46e + 01$	+	$1.86e + 03 \pm 4.65e + 01$	+

can be seen as simpler algorithm with respect to its predecessor, thus further confirming the concept of Ockham's Razor in MC. The comparison against MDE-pBX shows that RIS displays a better performance for all the groups of dimensionality values considered in this article. Regarding the comparison against CCPSO2, RIS tends to outperform it at 30 and 100 dimension. For large scale problems, CCPSO2 displays a good performance and slightly outperforms the proposed RIS. A reversed situation occurs for the MA-CMA-Chains algorithm. The latter algorithm is very efficient in 30 and 100 dimensions where it clearly outperforms RIS. On the other hand, this trend is not confirmed in high dimensions, since RIS statistically outperforms MA-SSW-Chains for all the problems in 1000 dimensions.

Numerical results reported in Tables 9, 10, 11, and 12 show that RIS outperforms on most of the problems under analysis cDE. On the other hand, PMS appears to be a challenging competitor. The RIS and PMS algorithms are both based on a memetic philosophy and both employ the Ockham's Razor principle as a bottom line for the design. However, while RIS is extremely simple as it samples a new point

and exploit the outgoing search direction by means of a local search, PMS is more powerful and sophisticated since it makes use of two complementary search strategies, (see Caraffini et al. 2013). Although PMS is an excellent framework, it pays off the very good performance with a higher computational complexity and memory requirement with respect to RIS, see Sect. 3.3.

3.1 The effect of the inheritance

As mentioned above, the inheritance mechanism within the re-sampling appears to beneficially affect the performance of the algorithm. A direct comparison between RIS and its variant which is identical to it except that it does not employ the inheritance (the re-sampling occurs by simply generating another point within \mathbf{D}) has been performed. The RIS variant without inheritance is called Re-sampling Search (RS). In order to illustrate the advantages of the inheritance, Table 13 displays the results of RIS and RS for the CEC2005 benchmark proposed in Suganthan et al. (2005). It can be

Table 2 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against its predecessor 3SOME and popular meta-heuristics on BBOB2010 (Hansen et al. 2010) in 100 dimensions

	RIS	3SOME		CLPSO		JADE	
f_1	$7.95e + 01 \pm 1.59e - 12$	$7.95e + 01 \pm 3.29e - 14$	-	$7.95e + 01 \pm 4.12e - 10$	+	$8.77e + 01 \pm 7.64e + 00$	+
f_2	$-2.10e + 02 \pm 7.48e - 08$	$-2.10e + 02 \pm 5.69e - 14$	-	$-2.10e + 02 \pm 9.39e - 07$	+	$5.73e + 04 \pm 8.69e + 04$	+
f_3	$-3.35e + 02 \pm 2.15e + 01$	$-4.39e + 02 \pm 7.28e + 00$	-	$-4.62e + 02 \pm 4.45e - 01$	-	$-3.11e + 02 \pm 4.95e + 01$	+
f_4	$-2.85e + 02 \pm 3.40e + 01$	$-4.27e + 02 \pm 8.70e + 00$	-	$-4.55e + 02 \pm 1.50e + 00$	-	$-1.43e + 02 \pm 9.83e + 01$	+
f_5	$-9.21e + 00 \pm 1.83e - 05$	$7.40e + 00 \pm 1.65e + 02$	+	$2.58e + 02 \pm 1.17e + 01$	+	$1.24e + 02 \pm 5.08e + 01$	+
f_6	$3.59e + 01 \pm 3.75e - 07$	$3.59e + 01 \pm 8.86e - 08$	-	$4.25e + 02 \pm 2.65e + 01$	+	$3.92e + 02 \pm 1.18e + 02$	+
f_7	$2.09e + 02 \pm 1.93e + 01$	$5.97e + 02 \pm 2.83e + 02$	+	$2.25e + 02 \pm 1.07e + 01$	+	$3.08e + 02 \pm 6.50e + 01$	+
f_8	$2.30e + 02 \pm 4.51e + 01$	$1.83e + 02 \pm 3.31e + 01$	-	$2.82e + 02 \pm 3.53e + 01$	+	$7.24e + 03 \pm 6.15e + 03$	+
f_9	$1.77e + 02 \pm 2.64e + 01$	$1.76e + 02 \pm 1.36e + 01$	-	$2.21e + 02 \pm 5.45e - 01$	+	$1.78e + 03 \pm 1.33e + 03$	+
f_{10}	$2.44e + 03 \pm 5.46e + 02$	$2.68e + 03 \pm 6.96e + 02$	+	$5.32e + 05 \pm 6.46e + 04$	+	$1.94e + 05 \pm 8.87e + 04$	+
f_{11}	$9.47e + 02 \pm 1.10e + 02$	$3.83e + 02 \pm 8.22e + 01$	-	$3.18e + 02 \pm 1.67e + 01$	-	$2.11e + 02 \pm 2.76e + 01$	-
f_{12}	$-6.18e + 02 \pm 4.16e + 00$	$-6.09e + 02 \pm 1.83e + 01$	+	$-6.03e + 02 \pm 5.15e + 00$	+	$2.22e + 07 \pm 2.13e + 07$	+
f_{13}	$3.21e + 01 \pm 1.68e + 00$	$3.35e + 01 \pm 4.87e + 00$	=	$5.08e + 01 \pm 2.08e + 00$	+	$7.69e + 02 \pm 2.46e + 02$	+
f_{14}	$-5.23e + 01 \pm 4.89e - 05$	$-5.23e + 01 \pm 5.47e - 05$	+	$-5.23e + 01 \pm 2.77e - 03$	+	$-4.65e + 01 \pm 3.89e + 00$	+
f_{15}	$1.95e + 03 \pm 1.18e + 02$	$4.53e + 03 \pm 5.89e + 02$	+	$2.10e + 03 \pm 3.63e + 01$	+	$1.59e + 03 \pm 8.67e + 01$	-
f_{16}	$8.25e + 01 \pm 1.56e + 00$	$9.51e + 01 \pm 6.11e + 00$	+	$9.53e + 01 \pm 2.04e + 00$	+	$1.01e + 02 \pm 3.46e + 00$	+
f_{17}	$-8.97e + 00 \pm 1.62e + 00$	$-2.63e - 02 \pm 3.97e + 00$	+	$-1.21e + 01 \pm 2.96e - 01$	-	$-1.45e + 01 \pm 5.96e - 01$	-
f_{18}	$1.56e + 01 \pm 6.47e + 00$	$4.55e + 01 \pm 1.54e + 01$	+	$1.12e + 00 \pm 1.09e + 00$	-	$-8.79e + 00 \pm 2.11e + 00$	-
f_{19}	$-9.32e + 01 \pm 2.16e + 00$	$-9.08e + 01 \pm 3.39e + 00$	+	$-9.48e + 01 \pm 2.29e - 01$	-	$-9.50e + 01 \pm 2.26e - 01$	-
f_{20}	$-5.45e + 02 \pm 1.17e - 01$	$-5.46e + 02 \pm 9.61e - 02$	-	$-5.45e + 02 \pm 6.24e - 02$	+	$-5.12e + 02 \pm 9.92e + 01$	+
f_{21}	$4.32e + 01 \pm 2.63e + 00$	$5.19e + 01 \pm 1.21e + 01$	+	$4.17e + 01 \pm 1.21e + 00$	-	$4.93e + 01 \pm 6.25e + 00$	+
f_{22}	$-9.96e + 02 \pm 5.45e + 00$	$-9.82e + 02 \pm 1.47e + 01$	+	$-9.98e + 02 \pm 5.95e - 01$	-	$-9.94e + 02 \pm 6.66e + 00$	+
f_{23}	$7.52e + 00 \pm 1.07e - 01$	$8.21e + 00 \pm 4.93e - 01$	+	$1.03e + 01 \pm 2.74e - 01$	+	$1.07e + 01 \pm 3.78e - 01$	+
f_{24}	$1.26e + 03 \pm 1.41e + 02$	$2.79e + 03 \pm 4.75e + 02$	+	$1.21e + 03 \pm 4.50e + 01$	-	$1.03e + 03 \pm 4.44e + 01$	-

Table 3 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against its predecessor 3SOME and popular meta-heuristics on CEC2008 (Tang et al. 2007) in 1000 dimensions

	RIS	3SOME		CLPSO		JADE	
f_1	$-4.50e + 02 \pm 1.33e - 09$	$-4.50e + 02 \pm 4.03e - 08$	+	$-2.98e + 02 \pm 3.50e + 01$	+	$1.00e + 06 \pm 3.23e + 05$	+
f_2	$-4.50e + 02 \pm 2.43e - 02$	$-4.50e + 02 \pm 3.17e - 02$	+	$-3.72e + 02 \pm 6.49e - 01$	+	$-3.21e + 02 \pm 8.63e + 00$	+
f_3	$1.51e + 03 \pm 8.53e + 01$	$1.38e + 03 \pm 8.37e + 01$	-	$4.02e + 03 \pm 2.64e + 02$	+	$4.26e + 11 \pm 2.13e + 11$	+
f_4	$5.83e + 03 \pm 3.67e + 02$	$-3.30e + 02 \pm 4.11e - 04$	-	$-7.28e + 01 \pm 1.44e + 01$	-	$4.45e + 03 \pm 1.00e + 03$	-
f_5	$-1.80e + 02 \pm 1.89e - 03$	$-1.80e + 02 \pm 8.46e - 03$	+	$-1.78e + 02 \pm 2.17e - 01$	+	$8.45e + 03 \pm 3.11e + 03$	+
f_6	$-1.40e + 02 \pm 5.02e - 07$	$-1.40e + 02 \pm 4.49e - 04$	+	$-1.40e + 02 \pm 9.84e - 05$	+	$-1.22e + 02 \pm 6.01e - 01$	+
f_7	$-1.35e + 04 \pm 1.08e + 02$	$-1.39e + 04 \pm 6.56e + 01$	-	$-1.33e + 04 \pm 4.19e + 01$	+	$-1.19e + 04 \pm 4.24e + 02$	+

easily observed that RIS is at least as good as its variant without inheritance and, in some cases, appears significantly more promising. Similar results have been obtained for the other benchmarks/dimensionality values under consideration in this article but, for the sake of brevity, only the results in Table 13 are shown. For the sake of completeness Figs. 6, 7, 8, and 9 show the average (over 100 runs) performance trends for four optimization problems amongst the 76 under consideration.

3.2 Statistical ranking by means of Holm-Bonferroni procedure

In addition to the results presented above, the ranking among all the algorithms considered in this article has been performed by means of the Holm-Bonferroni procedure, (see Holm 1979; Garcia et al. 2008), for the 10 algorithms under study (MA-CMA-Chains and MA-SSW-Chains have been considered as a unique algorithm for performing this test

Table 4 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against its predecessor 3SOME and popular meta-heuristics on CEC2010 (Tang et al. 2010) in 1000 dimensions

	RIS	3SOME		CLPSO		JADE	
f_1	4.16e - 06 \pm 5.82e - 07	$1.88e - 02 \pm 5.82e - 02$	+	$6.15e + 05 \pm 1.42e + 05$	+	$1.40e + 10 \pm 6.91e + 09$	+
f_2	$5.80e + 03 \pm 4.17e + 02$	1.97e + 01 \pm 2.87e + 01	-	$1.99e + 02 \pm 1.27e + 01$	-	$4.56e + 03 \pm 1.04e + 03$	-
f_3	4.84e - 06 \pm 5.21e - 07	$3.93e - 01 \pm 3.61e - 01$	+	$9.04e - 01 \pm 1.06e - 01$	+	$1.76e + 01 \pm 6.75e - 01$	+
f_4	$2.13e + 13 \pm 4.05e + 12$	$8.57e + 12 \pm 2.82e + 12$	-	$1.22e + 13 \pm 2.82e + 12$	-	2.62e + 12 \pm 1.03e + 12	-
f_5	$4.62e + 08 \pm 1.11e + 08$	$7.17e + 08 \pm 1.22e + 08$	+	$1.93e + 08 \pm 2.20e + 07$	-	8.58e + 07 \pm 1.77e + 07	-
f_6	$1.95e + 07 \pm 2.27e + 06$	$1.98e + 07 \pm 1.16e + 05$	=	9.51e + 03 \pm 2.09e + 04	-	$3.48e + 06 \pm 1.40e + 06$	-
f_7	$1.89e + 10 \pm 4.23e + 09$	$1.57e + 09 \pm 4.02e + 08$	-	5.53e + 08 \pm 1.30e + 08	-	$3.37e + 09 \pm 3.66e + 09$	-
f_8	$2.39e + 10 \pm 1.33e + 10$	$4.80e + 08 \pm 1.78e + 09$	-	7.53e + 07 \pm 2.77e + 07	-	$6.31e + 13 \pm 1.80e + 14$	+
f_9	1.69e + 08 \pm 6.69e + 06	$4.01e + 08 \pm 6.95e + 07$	+	$9.54e + 08 \pm 5.86e + 07$	+	$1.67e + 10 \pm 5.87e + 09$	+
f_{10}	$7.22e + 03 \pm 2.90e + 02$	$6.75e + 03 \pm 3.73e + 02$	-	6.36e + 03 \pm 1.92e + 02	-	$7.50e + 03 \pm 1.07e + 03$	+
f_{11}	$1.34e + 02 \pm 3.80e + 01$	$1.99e + 02 \pm 7.14e - 01$	+	9.71e + 01 \pm 7.27e + 00	-	$1.94e + 02 \pm 7.49e + 00$	+
f_{12}	1.26e + 04 \pm 3.41e + 03	$1.59e + 05 \pm 7.73e + 04$	+	$7.77e + 05 \pm 3.78e + 04$	+	$2.32e + 06 \pm 4.55e + 05$	+
f_{13}	$2.54e + 05 \pm 4.03e + 04$	$1.49e + 04 \pm 5.82e + 03$	-	9.09e + 03 \pm 2.13e + 03	-	$8.02e + 10 \pm 4.76e + 10$	+
f_{14}	4.47e + 07 \pm 1.44e + 06	$1.22e + 08 \pm 3.18e + 07$	+	$1.45e + 09 \pm 9.35e + 07$	+	$1.31e + 10 \pm 4.64e + 09$	+
f_{15}	7.26e + 03 \pm 3.80e + 02	$1.38e + 04 \pm 5.34e + 02$	+	$1.25e + 04 \pm 3.21e + 02$	+	$8.51e + 03 \pm 1.03e + 03$	+
f_{16}	1.58e + 02 \pm 3.61e + 01	$3.71e + 02 \pm 7.82e + 01$	+	$2.42e + 02 \pm 1.61e + 01$	+	$3.83e + 02 \pm 1.19e + 01$	+
f_{17}	2.13e + 04 \pm 4.35e + 03	$2.77e + 05 \pm 2.15e + 05$	+	$1.80e + 06 \pm 6.51e + 04$	+	$2.63e + 06 \pm 7.56e + 05$	+
f_{18}	1.55e + 03 \pm 1.27e + 03	$2.68e + 04 \pm 1.42e + 04$	+	$7.08e + 04 \pm 1.22e + 04$	+	$4.42e + 11 \pm 1.91e + 11$	+
f_{19}	$2.47e + 06 \pm 2.79e + 05$	1.44e + 05 \pm 1.73e + 04	-	$5.82e + 06 \pm 2.81e + 05$	+	$3.59e + 06 \pm 7.17e + 05$	+
f_{20}	$1.18e + 03 \pm 1.64e + 02$	1.13e + 03 \pm 1.26e + 02	-	$3.65e + 04 \pm 7.28e + 03$	+	$5.48e + 11 \pm 2.10e + 11$	+

Table 5 Average fitness \pm SD and Wilcoxon Rank-Sum Test (reference = RIS) for RIS against state-of-the-art algorithms on CEC 2005 (Suganthan et al. 2005) in 30 dimensions

	RIS	CCPSO2		MA-CMA-Chains		MDE-pBX	
f_1	$-4.50e + 02 \pm 1.31e - 13$	$-4.50e + 02 \pm 6.40e - 13$	+	$-4.50e + 02 \pm 8.36e - 10$	+	-4.50e + 02 \pm 1.52e - 13	-
f_2	-4.50e + 02 \pm 1.23e - 10	$-4.38e + 02 \pm 3.92e + 01$	+	$-4.50e + 02 \pm 1.47e - 02$	+	$-4.50e + 02 \pm 2.54e - 03$	+
f_3	$2.03e + 05 \pm 1.25e + 05$	$1.56e + 06 \pm 9.03e + 05$	+	1.45e + 05 \pm 4.32e + 05	-	$2.81e + 05 \pm 1.99e + 05$	+
f_4	$1.88e + 04 \pm 5.57e + 03$	$1.81e + 04 \pm 3.74e + 03$	=	$3.21e + 02 \pm 5.06e + 02$	-	-1.29e + 02 \pm 9.67e + 02	-
f_5	$3.48e + 03 \pm 7.70e + 02$	$9.18e + 03 \pm 1.59e + 03$	+	5.60e + 02 \pm 5.63e + 02	-	$2.74e + 03 \pm 6.34e + 02$	-
f_6	$6.91e + 02 \pm 4.39e + 02$	$4.63e + 02 \pm 5.05e + 01$	=	4.00e + 02 \pm 3.07e + 01	-	$4.33e + 02 \pm 4.81e + 01$	-
f_7	-1.80e + 02 \pm 3.38e - 03	$-1.80e + 02 \pm 1.86e - 02$	+	$-1.41e + 02 \pm 1.16e + 02$	+	$2.03e + 06 \pm 1.88e + 07$	+
f_8	-1.20e + 02 \pm 4.05e - 04	$-1.19e + 02 \pm 5.34e - 02$	+	$-1.20e + 02 \pm 9.08e - 03$	+	$-1.19e + 02 \pm 4.23e - 01$	+
f_9	$-1.18e + 02 \pm 1.63e - 05$	$-1.18e + 02 \pm 1.98e - 01$	+	-3.30e + 02 \pm 7.35e - 01	-	$-1.17e + 02 \pm 1.14e + 00$	+
f_{10}	$2.14e + 02 \pm 1.52e + 01$	$2.55e + 02 \pm 1.98e + 01$	+	-2.96e + 02 \pm 2.17e + 01	-	$2.23e + 02 \pm 2.44e + 01$	+
f_{11}	1.09e + 02 \pm 1.90e + 00	$1.18e + 02 \pm 2.48e + 00$	+	$1.14e + 02 \pm 3.18e + 00$	+	$1.11e + 02 \pm 4.59e + 00$	+
f_{12}	1.90e + 02 \pm 1.41e + 03	$3.36e + 03 \pm 4.99e + 03$	+	$2.46e + 02 \pm 1.07e + 03$	=	$3.77e + 03 \pm 3.87e + 03$	+
f_{13}	$-1.21e + 02 \pm 1.92e + 00$	-1.27e + 02 \pm 1.90e - 01	-	$-1.27e + 02 \pm 1.96e + 00$	-	$-1.19e + 02 \pm 2.28e + 00$	+
f_{14}	$-2.86e + 02 \pm 2.79e - 01$	$-2.87e + 02 \pm 2.89e - 01$	-	$-2.87e + 02 \pm 3.18e - 01$	-	-2.87e + 02 \pm 4.50e - 01	-
f_{15}	$1.44e + 03 \pm 3.89e - 01$	$1.44e + 03 \pm 1.25e - 01$	-	4.27e + 02 \pm 2.92e + 01	-	$1.46e + 03 \pm 6.43e + 00$	+
f_{16}	$1.55e + 03 \pm 7.51e + 00$	$1.58e + 03 \pm 9.17e + 00$	+	2.61e + 02 \pm 1.62e + 02	-	$1.58e + 03 \pm 1.11e + 01$	+
f_{17}	$1.66e + 03 \pm 1.54e + 01$	$1.70e + 03 \pm 1.29e + 01$	+	3.05e + 02 \pm 1.68e + 02	-	$1.62e + 03 \pm 9.04e + 00$	-
f_{18}	$9.10e + 02 \pm 4.58e - 10$	9.10e + 02 \pm 1.58e - 12	-	$9.11e + 02 \pm 3.88e + 01$	+	$9.10e + 02 \pm 8.31e - 11$	-
f_{19}	$9.10e + 02 \pm 4.52e - 10$	$9.10e + 02 \pm 1.41e - 12$	-	9.07e + 02 \pm 4.09e + 01	-	$9.10e + 02 \pm 2.42e - 10$	-
f_{20}	$9.10e + 02 \pm 4.29e - 10$	$9.10e + 02 \pm 1.65e - 12$	-	9.09e + 02 \pm 3.94e + 01	-	$9.10e + 02 \pm 3.41e - 11$	-

Table 5 continued

	RIS	CCPSO2		MA-CMA-Chains		MDE-pBX	
f_{21}	$1.69e + 03 \pm 4.33e + 00$	$1.71e + 03 \pm 5.62e + 00$	+	$8.82e + 02 \pm 1.13e + 02$	-	$1.70e + 03 \pm 5.51e + 00$	+
f_{22}	$2.42e + 03 \pm 2.92e + 01$	$2.49e + 03 \pm 2.85e + 01$	+	$1.25e + 03 \pm 1.31e + 01$	-	$2.41e + 03 \pm 4.97e + 01$	=
f_{23}	$1.72e + 03 \pm 6.16e + 00$	$1.71e + 03 \pm 4.86e + 00$	-	$9.00e + 02 \pm 6.27e + 01$	-	$1.70e + 03 \pm 5.28e + 00$	-
f_{24}	$1.68e + 03 \pm 6.83e + 00$	$1.70e + 03 \pm 7.97e + 00$	+	$4.60e + 02 \pm 0.00e + 00$	-	$1.67e + 03 \pm 1.55e + 01$	-
f_{25}	$1.43e + 03 \pm 3.74e + 02$	$1.85e + 03 \pm 9.19e + 01$	+	$1.89e + 03 \pm 8.19e + 00$	+	$1.83e + 03 \pm 1.55e + 02$	+

Table 6 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against state-of-the-art algorithms on BBOB 2010 (Hansen et al. 2010) in 100 dimensions

	RIS	CCPSO2		MA-CMA-Chains		MDE-pBX	
f_1	$7.95e + 01 \pm 1.59e - 12$	$7.95e + 01 \pm 1.81e - 13$	-	$7.95e + 01 \pm 6.55e - 10$	+	$7.95e + 01 \pm 7.60e - 05$	+
f_2	$-2.10e + 02 \pm 7.48e - 08$	$-2.10e + 02 \pm 1.76e - 12$	-	$-2.10e + 02 \pm 5.31e - 06$	+	$-2.10e + 02 \pm 6.06e - 03$	+
f_3	$-3.35e + 02 \pm 2.15e + 01$	$-4.54e + 02 \pm 8.40e + 00$	-	$-4.14e + 02 \pm 9.03e + 00$	-	$3.29e + 01 \pm 7.94e + 01$	+
f_4	$-2.85e + 02 \pm 3.40e + 01$	$-4.40e + 02 \pm 1.31e + 01$	-	$-3.82e + 02 \pm 1.14e + 01$	-	$4.03e + 02 \pm 1.31e + 02$	+
f_5	$-9.21e + 00 \pm 1.83e - 05$	$-9.21e + 00 \pm 1.20e - 03$	+	$-9.21e + 00 \pm 0.00e + 00$	-	$-3.09e - 02 \pm 1.27e + 01$	+
f_6	$3.59e + 01 \pm 3.75e - 07$	$1.25e + 02 \pm 4.02e + 01$	+	$3.59e + 01 \pm 1.60e - 03$	+	$8.03e + 01 \pm 3.32e + 01$	+
f_7	$2.09e + 02 \pm 1.93e + 01$	$4.38e + 02 \pm 4.90e + 01$	+	$1.33e + 02 \pm 7.77e + 00$	-	$3.70e + 02 \pm 7.43e + 01$	+
f_8	$2.30e + 02 \pm 4.51e + 01$	$2.70e + 02 \pm 3.42e + 01$	+	$1.59e + 02 \pm 1.45e + 01$	-	$3.40e + 02 \pm 6.77e + 01$	+
f_9	$1.77e + 02 \pm 2.64e + 01$	$2.30e + 02 \pm 2.78e + 01$	+	$1.80e + 02 \pm 8.49e + 00$	+	$2.52e + 02 \pm 3.75e + 01$	+
f_{10}	$2.44e + 03 \pm 5.46e + 02$	$2.61e + 04 \pm 6.64e + 03$	+	$2.40e + 03 \pm 1.44e + 03$	=	$1.64e + 04 \pm 7.99e + 03$	+
f_{11}	$9.47e + 02 \pm 1.10e + 02$	$6.21e + 02 \pm 1.95e + 02$	-	$1.85e + 02 \pm 1.85e + 01$	-	$9.16e + 01 \pm 7.45e + 00$	-
f_{12}	$-6.18e + 02 \pm 4.16e + 00$	$-6.13e + 02 \pm 1.20e + 01$	+	$-6.21e + 02 \pm 6.95e - 01$	-	$-5.99e + 02 \pm 7.07e + 01$	+
f_{13}	$3.21e + 01 \pm 1.68e + 00$	$3.31e + 01 \pm 4.20e + 00$	=	$3.11e + 01 \pm 1.76e + 00$	-	$3.47e + 01 \pm 6.70e + 00$	=
f_{14}	$-5.23e + 01 \pm 4.89e - 05$	$-5.23e + 01 \pm 2.34e - 04$	+	$-5.23e + 01 \pm 6.00e - 05$	-	$-5.23e + 01 \pm 2.55e - 03$	+
f_{15}	$1.95e + 03 \pm 1.18e + 02$	$2.33e + 03 \pm 2.32e + 02$	+	$1.41e + 03 \pm 1.88e + 02$	-	$1.66e + 03 \pm 1.10e + 02$	-
f_{16}	$8.25e + 01 \pm 1.56e + 00$	$9.88e + 01 \pm 4.27e + 00$	+	$8.38e + 01 \pm 1.79e + 00$	+	$8.85e + 01 \pm 4.46e + 00$	+
f_{17}	$-8.97e + 00 \pm 1.62e + 00$	$-8.29e + 00 \pm 1.62e + 00$	+	$-1.69e + 01 \pm 4.95e - 02$	-	$-1.35e + 01 \pm 4.83e - 01$	-
f_{18}	$1.56e + 01 \pm 6.47e + 00$	$1.61e + 01 \pm 6.61e + 00$	=	$-1.64e + 01 \pm 1.89e - 01$	-	$-4.84e + 00 \pm 1.68e + 00$	-
f_{19}	$-9.32e + 01 \pm 2.16e + 00$	$-9.46e + 01 \pm 1.29e + 00$	-	$-1.00e + 02 \pm 1.17e + 00$	-	$-1.00e + 02 \pm 7.13e - 01$	-
f_{20}	$-5.45e + 02 \pm 1.17e - 01$	$-5.46e + 02 \pm 6.73e - 02$	-	$-5.45e + 02 \pm 9.48e - 02$	+	$-5.44e + 02 \pm 1.14e - 01$	+
f_{21}	$4.32e + 01 \pm 2.63e + 00$	$4.41e + 01 \pm 3.38e + 00$	+	$4.55e + 01 \pm 7.35e + 00$	+	$4.49e + 01 \pm 5.88e + 00$	+
f_{22}	$-9.96e + 02 \pm 5.45e + 00$	$-9.95e + 02 \pm 5.83e + 00$	+	$-9.92e + 02 \pm 8.63e + 00$	+	$-9.92e + 02 \pm 9.11e + 00$	+
f_{23}	$7.52e + 00 \pm 1.07e - 01$	$9.39e + 00 \pm 4.22e - 01$	+	$7.66e + 00 \pm 2.66e - 01$	+	$9.34e + 00 \pm 7.99e - 01$	+
f_{24}	$1.26e + 03 \pm 1.41e + 02$	$1.19e + 03 \pm 1.53e + 02$	-	$4.15e + 02 \pm 1.12e + 02$	-	$4.75e + 02 \pm 4.72e + 01$	-

Table 7 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against state-of-the-art algorithms on CEC 2008 (Tang et al. 2007) in 1000 dimensions

	RIS	CCPSO2		MA-SSW-Chains		MDE-pBX	
f_1	$-4.50e + 02 \pm 1.33e - 09$	$-4.50e + 02 \pm 3.54e - 12$	-	$6.18e + 04 \pm 6.12e + 05$	+	$1.20e + 05 \pm 4.41e + 04$	+
f_2	$-4.50e + 02 \pm 2.43e - 02$	$-4.09e + 02 \pm 2.46e + 01$	+	$-2.66e + 02 \pm 1.81e + 00$	+	$-3.33e + 02 \pm 4.09e + 00$	+
f_3	$1.51e + 03 \pm 8.53e + 01$	$1.80e + 03 \pm 1.12e + 02$	+	$1.45e + 11 \pm 1.01e + 12$	+	$3.13e + 10 \pm 1.65e + 10$	+
f_4	$5.83e + 03 \pm 3.67e + 02$	$-2.00e + 02 \pm 1.07e + 02$	-	$1.53e + 04 \pm 1.02e + 03$	+	$7.60e + 03 \pm 2.55e + 02$	+
f_5	$-1.80e + 02 \pm 1.89e - 03$	$-1.80e + 02 \pm 2.98e - 03$	+	$-1.72e + 02 \pm 1.09e + 01$	+	$1.08e + 03 \pm 4.60e + 02$	+
f_6	$-1.40e + 02 \pm 5.02e - 07$	$-1.40e + 02 \pm 9.34e - 11$	-	$-1.20e + 02 \pm 3.19e - 01$	+	$-1.21e + 02 \pm 5.10e - 02$	+
f_7	$-1.35e + 04 \pm 1.08e + 02$	$-1.44e + 04 \pm 7.46e + 01$	-	$-9.97e + 03 \pm 8.06e + 02$	+	$-1.11e + 04 \pm 1.63e + 02$	+

Table 8 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against state-of-the-art algorithms on CEC2010 (Tang et al. 2010) in 1000 dimensions

	RIS	CCPSO2		MA-SSW-Chains		MDE-pBX	
f_1	$4.16e - 06 \pm 5.82e - 07$	$6.47e - 14 \pm 1.41e - 13$	-	$2.45e + 11 \pm 4.07e + 10$	+	$1.05e + 09 \pm 6.58e + 08$	+
f_2	$5.80e + 03 \pm 4.17e + 02$	$1.36e + 02 \pm 1.11e + 02$	-	$1.97e + 04 \pm 1.74e + 03$	+	$7.02e + 03 \pm 2.38e + 02$	+
f_3	$4.84e - 06 \pm 5.21e - 07$	$7.34e - 11 \pm 1.05e - 10$	-	$2.02e + 01 \pm 4.21e - 01$	+	$1.93e + 01 \pm 4.76e - 02$	+
f_4	$2.13e + 13 \pm 4.05e + 12$	$2.14e + 12 \pm 1.27e + 12$	-	$1.24e + 15 \pm 5.69e + 14$	+	$3.21e + 12 \pm 9.76e + 11$	-
f_5	$4.62e + 08 \pm 1.11e + 08$	$3.92e + 08 \pm 7.98e + 07$	-	$8.09e + 08 \pm 6.58e + 07$	+	$1.54e + 08 \pm 2.77e + 07$	-
f_6	$1.95e + 07 \pm 2.27e + 06$	$1.71e + 07 \pm 4.45e + 06$	-	$2.03e + 07 \pm 2.24e + 05$	+	$3.65e + 06 \pm 1.75e + 06$	-
f_7	$1.89e + 10 \pm 4.23e + 09$	$7.60e + 09 \pm 9.72e + 09$	-	$7.64e + 11 \pm 5.09e + 11$	+	$6.79e + 06 \pm 1.01e + 07$	-
f_8	$2.39e + 10 \pm 1.33e + 10$	$5.46e + 07 \pm 4.16e + 07$	-	$5.34e + 16 \pm 3.07e + 16$	+	$2.03e + 08 \pm 1.63e + 08$	-
f_9	$1.69e + 08 \pm 6.69e + 06$	$5.01e + 07 \pm 7.68e + 06$	-	$2.81e + 11 \pm 4.08e + 10$	+	$1.68e + 09 \pm 1.00e + 09$	+
f_{10}	$7.22e + 03 \pm 2.90e + 02$	$4.57e + 03 \pm 2.75e + 02$	-	$1.85e + 04 \pm 1.76e + 03$	+	$7.33e + 03 \pm 2.55e + 02$	+
f_{11}	$1.34e + 02 \pm 3.80e + 01$	$2.00e + 02 \pm 5.98e + 00$	+	$2.20e + 02 \pm 3.44e + 00$	+	$2.06e + 02 \pm 2.40e + 00$	+
f_{12}	$1.26e + 04 \pm 3.41e + 03$	$6.12e + 04 \pm 8.14e + 04$	+	$1.69e + 07 \pm 4.07e + 06$	+	$2.92e + 05 \pm 6.60e + 04$	+
f_{13}	$2.54e + 05 \pm 4.03e + 04$	$1.14e + 03 \pm 5.42e + 02$	-	$1.27e + 12 \pm 5.77e + 11$	+	$2.88e + 09 \pm 3.17e + 09$	+
f_{14}	$4.47e + 07 \pm 1.44e + 06$	$1.60e + 08 \pm 3.35e + 07$	+	$3.23e + 11 \pm 3.62e + 10$	+	$1.04e + 09 \pm 1.97e + 08$	+
f_{15}	$7.26e + 03 \pm 3.80e + 02$	$9.31e + 03 \pm 5.52e + 02$	+	$1.86e + 04 \pm 1.58e + 03$	+	$7.44e + 03 \pm 2.80e + 02$	+
f_{16}	$1.58e + 02 \pm 3.61e + 01$	$3.95e + 02 \pm 1.45e + 00$	+	$4.00e + 02 \pm 7.55e + 00$	+	$3.84e + 02 \pm 1.22e + 00$	+
f_{17}	$2.13e + 04 \pm 4.35e + 03$	$1.41e + 05 \pm 1.44e + 05$	+	$4.64e + 07 \pm 1.30e + 07$	+	$4.35e + 05 \pm 8.33e + 04$	+
f_{18}	$1.55e + 03 \pm 1.27e + 03$	$5.62e + 03 \pm 4.13e + 03$	+	$5.38e + 12 \pm 7.78e + 11$	+	$3.73e + 10 \pm 1.95e + 10$	+
f_{19}	$2.47e + 06 \pm 2.79e + 05$	$1.14e + 06 \pm 1.22e + 06$	-	$1.24e + 08 \pm 3.39e + 07$	+	$9.22e + 05 \pm 1.06e + 05$	-
f_{20}	$1.18e + 03 \pm 1.64e + 02$	$1.42e + 03 \pm 1.19e + 02$	+	$6.07e + 12 \pm 8.88e + 11$	+	$4.18e + 10 \pm 2.02e + 10$	+

and indicated as MACH) and the 76 problems under consideration. The Holm-Bonferroni procedure consists of the following. Considering the results in the tables above, the 10 algorithms under analysis have been ranked on the basis of their average performance calculated over the 76 test problems. More specifically, a score R_i for $i = 1, \dots, N_A$ (where N_A is the number of algorithms under analysis, $N_A = 10$ in our case) has been assigned. The score has been assigned in the following way: for each problem, a score of 10 is assigned to the algorithm displaying the best performance, 9 is assigned to the second best, 8 to the third and so on. The algorithm displaying the worst performance scores 1. For each algorithm, the scores obtained on each problem are summed up averaged over the amount of test problems (76 in our case). On the basis of these scores the algorithms are sorted (ranked). With the calculated R_i values, RIS has been taken as a reference algorithm. Indicating with R_0 the rank of RIS, and with R_j for $j = 1, \dots, N_A - 1$ the rank of one of the remaining seven algorithms, the values z_j have been calculated as

$$z_j = \frac{R_j - R_0}{\sqrt{\frac{N_A(N_A+1)}{6N_{TP}}}} \tag{7}$$

where N_{TP} is the number of test problems in consideration ($N_{TP} = 76$ in our case). By means of the z_j values, the

corresponding cumulative normal distribution values p_j have been calculated. These p_j values have then been compared with the corresponding δ/j where δ is the level of confidence, set to 0.05 in our case. Table 14 displays the ranks, z_j values, p_j values, and corresponding δ/j obtained in this way. The rank of RIS is shown in parenthesis. Moreover, it is indicated whether the null-hypothesis (that the two algorithms have indistinguishable performances) is “Rejected”, i.e. RIS statistically outperforms the algorithm under consideration, or “Accepted” if the distribution of values can be considered the same (there is no out-performance).

As shown in Table 14, the proposed RIS is ranked third amongst all the algorithms considered in this study. The best two algorithms, PMS and CCPSO2, are characterized by the same ranking that is very close to the RIS ranking. The RS algorithm, which is as simple as RIS, is ranked right after RIS. The MACH, MDE-pBX, 3SOME, and CLPSO algorithms also displays respectable performance. As a general consideration, the algorithms exhibiting the highest rank are much simpler than other modern algorithms. This finding confirms, in accordance with the Ockham’s Razor in MC, that the performance of simple, properly designed algorithms can be as good as (or even better than) the performance of modern complex algorithms, composed of multiple and computationally expensive components.

Table 9 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against other state-of-the-art single solution algorithms on CEC2005 (Suganthan et al. 2005) in 30 dimensions

	RIS	PMS		cDE	
f_1	$-4.50e + 02 \pm 4.56e - 13$	$-4.50e + 02 \pm 2.36e - 13$	–	$6.59e + 03 \pm 4.31e + 03$	+
f_2	$-4.50e + 02 \pm 1.23e - 10$	$-4.50e + 02 \pm 3.43e - 05$	+	$3.68e + 04 \pm 1.16e + 04$	+
f_3	$2.03e + 05 \pm 1.25e + 05$	$2.09e + 05 \pm 1.23e + 05$	=	$1.49e + 08 \pm 8.08e + 07$	+
f_4	$1.88e + 04 \pm 5.57e + 03$	$1.33e + 04 \pm 5.32e + 03$	–	$6.65e + 04 \pm 1.91e + 04$	+
f_5	$3.48e + 03 \pm 7.70e + 02$	$9.47e + 03 \pm 4.35e + 03$	+	$1.30e + 04 \pm 3.01e + 03$	+
f_6	$6.91e + 02 \pm 4.39e + 02$	$6.89e + 02 \pm 5.60e + 02$	=	$7.17e + 08 \pm 1.06e + 09$	+
f_7	$-1.80e + 02 \pm 3.38e - 03$	$-1.80e + 02 \pm 1.37e - 02$	+	$4.63e + 03 \pm 8.41e + 01$	+
f_8	$-1.20e + 02 \pm 4.05e - 04$	$-1.20e + 02 \pm 2.97e - 02$	+	$-1.19e + 02 \pm 1.00e - 01$	+
f_9	$-1.18e + 02 \pm 1.63e - 05$	$-3.30e + 02 \pm 2.62e - 13$	–	$-1.60e + 02 \pm 3.38e + 01$	–
f_{10}	$2.14e + 02 \pm 1.52e + 01$	$-4.54e + 01 \pm 9.47e + 01$	–	$-2.68e + 01 \pm 5.58e + 01$	–
f_{11}	$1.09e + 02 \pm 1.90e + 00$	$1.21e + 02 \pm 5.30e + 00$	+	$1.23e + 02 \pm 3.42e + 00$	+
f_{12}	$1.90e + 02 \pm 1.41e + 03$	$1.26e + 03 \pm 2.83e + 03$	+	$3.22e + 05 \pm 1.37e + 05$	+
f_{13}	$-1.21e + 02 \pm 1.92e + 00$	$-1.23e + 02 \pm 4.12e + 00$	–	$1.48e + 03 \pm 2.29e + 03$	+
f_{14}	$-2.86e + 02 \pm 2.79e - 01$	$-2.86e + 02 \pm 5.81e - 01$	=	$-2.87e + 02 \pm 3.49e - 01$	–
f_{15}	$1.44e + 03 \pm 3.89e - 01$	$4.32e + 02 \pm 2.24e + 02$	–	$7.73e + 02 \pm 9.44e + 01$	–
f_{16}	$1.55e + 03 \pm 7.51e + 00$	$4.73e + 02 \pm 1.17e + 02$	–	$4.90e + 02 \pm 9.34e + 01$	–
f_{17}	$1.66e + 03 \pm 1.54e + 01$	$5.42e + 02 \pm 1.24e + 02$	–	$5.59e + 02 \pm 1.14e + 02$	–
f_{18}	$9.10e + 02 \pm 4.58e - 10$	$9.37e + 02 \pm 2.35e + 01$	+	$9.73e + 02 \pm 2.85e + 01$	+
f_{19}	$9.10e + 02 \pm 4.52e - 10$	$9.36e + 02 \pm 1.85e + 01$	+	$9.70e + 02 \pm 2.61e + 01$	+
f_{20}	$9.10e + 02 \pm 4.29e - 10$	$9.39e + 02 \pm 1.67e + 01$	+	$9.73e + 02 \pm 2.75e + 01$	+
f_{21}	$1.69e + 03 \pm 4.33e + 00$	$1.03e + 03 \pm 2.68e + 02$	–	$1.52e + 03 \pm 3.85e + 01$	–
f_{22}	$2.42e + 03 \pm 2.92e + 01$	$1.50e + 03 \pm 8.29e + 01$	–	$1.44e + 03 \pm 7.86e + 01$	–
f_{23}	$1.72e + 03 \pm 6.16e + 00$	$9.95e + 02 \pm 2.15e + 02$	–	$1.52e + 03 \pm 3.72e + 01$	–
f_{24}	$1.68e + 03 \pm 6.83e + 00$	$5.91e + 02 \pm 3.54e + 02$	–	$1.36e + 03 \pm 9.09e + 01$	–
f_{25}	$1.43e + 03 \pm 3.74e + 02$	$1.62e + 03 \pm 2.29e + 02$	+	$1.91e + 03 \pm 9.04e + 00$	+

3.3 Memory and computational overhead

In order to summarize the difference, in terms of computational requirements, of the algorithms under investigation, Table 15 displays the main features and required memory slots (candidate solutions saved in memory) of each of them. N_p indicates the population size.

Figure 10 displays the average (over 30 runs) computational overhead depending on the problem dimensionality n of the algorithms under examination. Each run has been continued until 10000 fitness evaluations. With computational overhead of an algorithm we mean here the calculation time of a run without the time required to perform the fitness evaluations.

It can be easily observed that RIS (as well 3SOME) is much less demanding in terms of memory usage with respect to the population-based algorithms. In addition, RIS is characterized by a more modest computational overhead than that of all the other algorithms. Most importantly, the overhead of RIS grows with the dimensionality slower than the overhead of the other algorithms. It is interesting to note that the

overhead of RIS is much lower even than the overhead of its predecessor 3SOME. In addition, as shown in Fig. 10, the algorithms characterized by the most modest overhead, i.e. RIS and CCPSO2, are also those that exhibit the best performance. This fact confirms that the computational cost is not directly correlated to the algorithmic performance.

Although a proof of algorithmic convergence of RIS is not given in this paper, a few further considerations are here offered to better understand its algorithmic functioning in the light of the displayed results. Since RIS, besides the re-sampling mechanism, attempts to improve upon the solution by an exploitative local search that performs movements along the axes, it is suitable to tackle separable problems. It can be observed from numerical results that RIS succeeds at solving several separable problems among those considered, such as f_1 and f_2 from CEC2005 testbed. In other separable cases, although RIS does not solve the problem, still achieves a better fitness values than that achieved by all the other algorithms. According to our interpretation, the RIS success is due to efficiency of the search along the axes. In the non-separable cases, especially multi-modal, PMS often displays

Table 10 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against other state-of-the-art single solution algorithms on BBOB2010 (Hansen et al. 2010) in 100 dimensions

	RIS	PMS		cDE	
f_1	$7.95e + 01 \pm 1.59e - 12$	$7.95e + 01 \pm 1.18e - 13$	–	$4.88e + 02 \pm 5.44e + 01$	+
f_2	$-2.10e + 02 \pm 7.48e - 08$	$-2.10e + 02 \pm 1.44e - 13$	–	$3.64e + 06 \pm 1.63e + 06$	+
f_3	$-3.35e + 02 \pm 2.15e + 01$	$-3.67e + 02 \pm 1.66e + 01$	–	$2.38e + 03 \pm 3.00e + 02$	+
f_4	$-2.85e + 02 \pm 3.40e + 01$	$-3.40e + 02 \pm 2.05e + 01$	–	$3.93e + 03 \pm 5.42e + 02$	+
f_5	$-9.21e + 00 \pm 1.83e - 05$	$-9.21e + 00 \pm 3.51e - 12$	–	$3.20e + 02 \pm 5.32e + 01$	+
f_6	$3.59e + 01 \pm 3.75e - 07$	$4.05e + 01 \pm 4.61e + 00$	+	$2.91e + 05 \pm 1.27e + 05$	+
f_7	$2.09e + 02 \pm 1.93e + 01$	$3.70e + 02 \pm 8.80e + 01$	+	$2.83e + 03 \pm 3.80e + 02$	+
f_8	$2.30e + 02 \pm 4.51e + 01$	$2.21e + 02 \pm 5.96e + 01$	=	$1.36e + 06 \pm 3.50e + 05$	+
f_9	$1.77e + 02 \pm 2.64e + 01$	$1.72e + 02 \pm 2.30e + 01$	–	$7.31e + 05 \pm 1.89e + 05$	+
f_{10}	$2.44e + 03 \pm 5.46e + 02$	$2.37e + 03 \pm 6.64e + 02$	=	$7.16e + 06 \pm 1.56e + 06$	+
f_{11}	$9.47e + 02 \pm 1.10e + 02$	$6.51e + 02 \pm 8.17e + 01$	–	$1.05e + 03 \pm 1.14e + 02$	+
f_{12}	$-6.18e + 02 \pm 4.16e + 00$	$-6.11e + 02 \pm 1.52e + 01$	+	$8.03e + 08 \pm 1.26e + 08$	+
f_{13}	$3.21e + 01 \pm 1.68e + 00$	$3.60e + 01 \pm 6.01e + 00$	+	$3.94e + 03 \pm 3.22e + 02$	+
f_{14}	$-5.23e + 01 \pm 4.89e - 05$	$-5.23e + 01 \pm 1.80e - 05$	–	$3.42e + 01 \pm 1.19e + 01$	+
f_{15}	$1.95e + 03 \pm 1.18e + 02$	$2.39e + 03 \pm 6.32e + 02$	+	$4.23e + 03 \pm 4.02e + 02$	+
f_{16}	$8.25e + 01 \pm 1.56e + 00$	$9.23e + 01 \pm 9.50e + 00$	+	$1.07e + 02 \pm 3.78e + 00$	+
f_{17}	$-8.97e + 00 \pm 1.62e + 00$	$-1.96e + 00 \pm 9.61e + 00$	+	$-3.49e + 00 \pm 1.82e + 00$	+
f_{18}	$1.56e + 01 \pm 6.47e + 00$	$4.89e + 01 \pm 4.41e + 01$	+	$3.48e + 01 \pm 6.97e + 00$	+
f_{19}	$-9.32e + 01 \pm 2.16e + 00$	$-5.08e + 01 \pm 4.81e + 01$	+	$-6.17e + 01 \pm 5.07e + 00$	+
f_{20}	$-5.45e + 02 \pm 1.17e - 01$	$-5.45e + 02 \pm 1.28e - 01$	=	$1.21e + 05 \pm 4.71e + 04$	+
f_{21}	$4.32e + 01 \pm 2.63e + 00$	$5.05e + 01 \pm 1.12e + 01$	+	$1.18e + 02 \pm 3.05e + 00$	+
f_{22}	$-9.96e + 02 \pm 5.45e + 00$	$-9.83e + 02 \pm 1.29e + 01$	+	$-9.25e + 02 \pm 4.82e + 00$	+
f_{23}	$7.52e + 00 \pm 1.07e - 01$	$8.70e + 00 \pm 8.16e - 01$	+	$1.01e + 01 \pm 4.85e - 01$	+
f_{24}	$1.26e + 03 \pm 1.41e + 02$	$2.14e + 03 \pm 6.37e + 02$	+	$2.83e + 03 \pm 1.66e + 02$	+

Table 11 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against other state-of-the-art single solution algorithms on CEC2008 (Tang et al. 2007) in 1000 dimensions

	RIS	PMS		cDE	
f_1	$-4.50e + 02 \pm 1.33e - 09$	$-4.50e + 02 \pm 5.91e - 13$	–	$4.80e + 06 \pm 1.57e + 05$	+
f_2	$-4.50e + 02 \pm 2.43e - 02$	$-3.88e + 02 \pm 5.17e + 01$	+	$-2.76e + 02 \pm 2.81e + 00$	+
f_3	$1.51e + 03 \pm 8.53e + 01$	$1.34e + 03 \pm 5.37e + 02$	=	$4.70e + 12 \pm 2.43e + 11$	+
f_4	$5.83e + 03 \pm 3.67e + 02$	$-3.30e + 02 \pm 1.33e - 12$	–	$2.06e + 04 \pm 5.08e + 02$	+
f_5	$-1.80e + 02 \pm 1.89e - 03$	$-1.80e + 02 \pm 1.50e - 02$	+	$4.29e + 04 \pm 1.21e + 03$	+
f_6	$-1.40e + 02 \pm 5.02e - 07$	$-1.38e + 02 \pm 6.36e + 00$	+	$-1.19e + 02 \pm 5.41e - 02$	+
f_7	$-1.35e + 04 \pm 1.08e + 02$	$-1.37e + 04 \pm 2.80e + 02$	–	$-7.24e + 03 \pm 1.57e + 02$	+

a better performance than that of RIS. In our opinion, these results can be explained considering that PMS combines two diverse search operators, the first moves along the axes, while the second employs the exploration logic of Rosenbrock algorithm, (see Rosenbrock 1960), and thus performs diagonal moves within the decision space, (see Caraffini et al. 2013). Although an operator that uses diagonal moves while following the variations of the gradient appears to have some impact on the robustness of the algorithm, its cost is a

$n \times n$ matrix and a non-negligible overhead, see Table 15 and Fig. 10. In addition, the simple combination of moves along the axes and re-sampling mechanism of RIS appears to lead to a respectable performance also for non-separable problems, as shown in Iacca et al. (2012a). In this light RIS can be a good scheme for those applications where (cost and space) hardware limitation as well as real-time requirements impose a fast algorithmic response and a modest memory usage.

Table 12 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS) for RIS against other state-of-the-art single solution algorithms on CEC2010 (Tang et al. 2010) in 1000 dimensions

	RIS	PMS		cDE	
f_1	$4.16e - 06 \pm 5.82e - 07$	$2.86e - 21 \pm 9.54e - 23$	–	$1.93e + 11 \pm 1.71e + 10$	+
f_2	$5.80e + 03 \pm 4.17e + 02$	$1.49e - 13 \pm 2.79e - 13$	–	$2.00e + 04 \pm 5.02e + 02$	+
f_3	$4.84e - 06 \pm 5.21e - 07$	$4.52e - 01 \pm 2.76e + 00$	+	$2.12e + 01 \pm 5.69e - 02$	+
f_4	$2.13e + 13 \pm 4.05e + 12$	$5.27e + 11 \pm 2.72e + 11$	–	$6.07e + 14 \pm 2.16e + 14$	+
f_5	$4.62e + 08 \pm 1.11e + 08$	$4.78e + 08 \pm 1.40e + 08$	=	$4.84e + 08 \pm 7.40e + 07$	=
f_6	$1.95e + 07 \pm 2.27e + 06$	$1.92e + 07 \pm 2.24e + 06$	–	$1.98e + 07 \pm 6.56e + 05$	=
f_7	$1.89e + 10 \pm 4.23e + 09$	$1.02e + 08 \pm 2.57e + 08$	–	$1.30e + 11 \pm 2.89e + 10$	+
f_8	$2.39e + 10 \pm 1.33e + 10$	$1.17e + 08 \pm 1.30e + 08$	–	$9.94e + 15 \pm 5.16e + 15$	+
f_9	$1.69e + 08 \pm 6.69e + 06$	$6.19e + 06 \pm 2.80e + 06$	–	$1.90e + 11 \pm 2.15e + 10$	+
f_{10}	$7.22e + 03 \pm 2.90e + 02$	$5.25e + 03 \pm 2.09e + 03$	–	$2.03e + 04 \pm 4.01e + 02$	+
f_{11}	$1.34e + 02 \pm 3.80e + 01$	$1.85e + 02 \pm 3.03e + 01$	+	$2.34e + 02 \pm 4.85e - 01$	+
f_{12}	$1.26e + 04 \pm 3.41e + 03$	$1.06e + 03 \pm 7.23e + 02$	–	$9.89e + 06 \pm 9.79e + 05$	+
f_{13}	$2.54e + 05 \pm 4.03e + 04$	$1.18e + 03 \pm 6.33e + 02$	–	$1.95e + 12 \pm 1.41e + 11$	+
f_{14}	$4.47e + 07 \pm 1.44e + 06$	$1.44e + 07 \pm 5.47e + 06$	–	$1.78e + 11 \pm 2.42e + 10$	+
f_{15}	$7.26e + 03 \pm 3.80e + 02$	$1.20e + 04 \pm 3.87e + 03$	+	$2.03e + 04 \pm 3.72e + 02$	+
f_{16}	$1.58e + 02 \pm 3.61e + 01$	$3.27e + 02 \pm 9.13e + 01$	+	$4.26e + 02 \pm 7.85e - 01$	+
f_{17}	$2.13e + 04 \pm 4.35e + 03$	$1.37e + 03 \pm 8.23e + 02$	–	$1.86e + 07 \pm 2.25e + 06$	+
f_{18}	$1.55e + 03 \pm 1.27e + 03$	$2.41e + 03 \pm 1.00e + 03$	+	$4.81e + 12 \pm 2.38e + 11$	+
f_{19}	$2.47e + 06 \pm 2.79e + 05$	$1.47e + 05 \pm 4.93e + 04$	–	$3.41e + 07 \pm 5.53e + 06$	+
f_{20}	$1.18e + 03 \pm 1.64e + 02$	$9.57e + 02 \pm 5.37e + 02$	=	$5.28e + 12 \pm 2.49e + 11$	+

3.4 Tuning of ρ

On the basis of our experience, the vector parameter ρ plays a crucial role on the RIS performance. In order to justify our choice and systematically offer a hint on the setting of this parameter the following procedure has been designed. We considered four test problems, i.e. f_1 , f_6 , f_{13} , and f_{15} from CEC2005. We chose these problems in order to have a diverse reduced testbed that includes separable and non-separable problems as well as uni-modal and multi-modal problems. For these optimization problem, RIS has been run with various ρ values. Since the problems are defined in hyper-cubical spaces, ρ is a vector of identical numbers. Thus, this vector parameter is here indicated as a scalar ρ . In correspondence to ρ values equal to 0.1, 0.2, 0.3, 0.4, and 0.5, respectively, RIS has been run 30 times for $5000 \times n$ fitness evaluations. Table 16 shows the average final error for the problems under consideration.

Numerical results show that small ρ values can be inadequate as each local search activation can be excessively exploitative. Since a proper global search is missing within RIS framework, a large initial radius appear to be beneficial to start the local search. On the other hand, an excessively large radius, such as 0.5 is likely to initially generate solutions outside the decision space (while the bounds are handled by the toroidal technique described above). Although a proper

setting of ρ obviously depends on the problem, as numerical results show, the setting of 0.4 appears to be a reasonable compromise that offers a reliable algorithmic behaviour.

4 Application case: tuning of a control system for a helicopter Robot

The proposed RIS has been also tested on a real-world problem, i.e. the tuning of a yaw (heading) controller of an autonomous small indoor helicopter. The problem under investigation consists of the design of a control system that allows the tail of a helicopter to keep a desired position, namely set-point, and to go back to this position when a disturbance occurs (e.g. wind). For this application a standard integral-state limited Proportional, Integral, and Derivative (PID) control system, (see Wescott 2000), has been selected while the RIS has been used to tune the parameters of the control system.

Our hardware setup consists of an autonomous small indoor helicopter namely *Flyper*. This device is characterized by a rotor span of 34 cm and a total weight of 191 g. Four actuators to enable the control of all six degrees of freedom are integrated into the system. Two motors which independently control the speed of each rotor, giving combined control over altitude and yaw, are present on-board. The two engines and

Table 13 Average fitness \pm SD and Wilcoxon Rank-Sum test on the fitness (reference = RIS) for RIS against its variant without inheritance, RS, on CEC 2005 (Suganthan et al. 2005) in 30 dimensions

	RIS	RS	
f_1	$-4.50e + 02 \pm 1.31e - 13$	$-4.50e + 02 \pm 1.54e - 13$	+
f_2	$-4.50e + 02 \pm 1.23e - 10$	$-4.50e + 02 \pm 1.29e - 10$	=
f_3	$2.03e + 05 \pm 1.25e + 05$	$2.30e + 05 \pm 1.39e + 05$	=
f_4	$1.88e + 04 \pm 5.57e + 03$	$3.42e + 04 \pm 6.04e + 03$	+
f_5	$3.48e + 03 \pm 7.70e + 02$	$4.04e + 03 \pm 8.22e + 02$	+
f_6	$6.91e + 02 \pm 4.39e + 02$	$6.45e + 02 \pm 6.52e + 02$	=
f_7	$-1.80e + 02 \pm 3.38e - 03$	$-1.80e + 02 \pm 3.00e - 03$	=
f_8	$-1.20e + 02 \pm 4.05e - 04$	$-1.20e + 02 \pm 7.51e - 03$	=
f_9	$-1.18e + 02 \pm 1.63e - 05$	$-1.17e + 02 \pm 3.69e - 01$	+
f_{10}	$2.14e + 02 \pm 1.52e + 01$	$2.15e + 02 \pm 1.49e + 01$	=
f_{11}	$1.09e + 02 \pm 1.90e + 00$	$1.10e + 02 \pm 1.76e + 00$	=
f_{12}	$1.90e + 02 \pm 1.41e + 03$	$1.25e + 02 \pm 1.20e + 03$	=
f_{13}	$-1.21e + 02 \pm 1.92e + 00$	$-1.22e + 02 \pm 1.76e + 00$	=
f_{14}	$-2.86e + 02 \pm 2.79e - 01$	$-2.86e + 02 \pm 3.43e - 01$	=
f_{15}	$1.44e + 03 \pm 3.89e - 01$	$1.45e + 03 \pm 5.00e - 01$	+
f_{16}	$1.55e + 03 \pm 7.51e + 00$	$1.55e + 03 \pm 6.53e + 00$	+
f_{17}	$1.66e + 03 \pm 1.54e + 01$	$1.70e + 03 \pm 1.30e + 01$	+
f_{18}	$9.10e + 02 \pm 4.58e - 10$	$9.10e + 02 \pm 4.47e - 10$	=
f_{19}	$9.10e + 02 \pm 4.52e - 10$	$9.10e + 02 \pm 4.72e - 10$	=
f_{20}	$9.10e + 02 \pm 4.29e - 10$	$9.10e + 02 \pm 4.79e - 10$	=
f_{21}	$1.69e + 03 \pm 4.33e + 00$	$1.70e + 03 \pm 4.06e + 00$	+
f_{22}	$2.42e + 03 \pm 2.92e + 01$	$2.45e + 03 \pm 2.98e + 01$	+
f_{23}	$1.72e + 03 \pm 6.16e + 00$	$1.73e + 03 \pm 5.51e + 00$	+
f_{24}	$1.68e + 03 \pm 6.83e + 00$	$1.68e + 03 \pm 6.59e + 00$	+
f_{25}	$1.43e + 03 \pm 3.74e + 02$	$1.44e + 03 \pm 2.64e + 02$	+

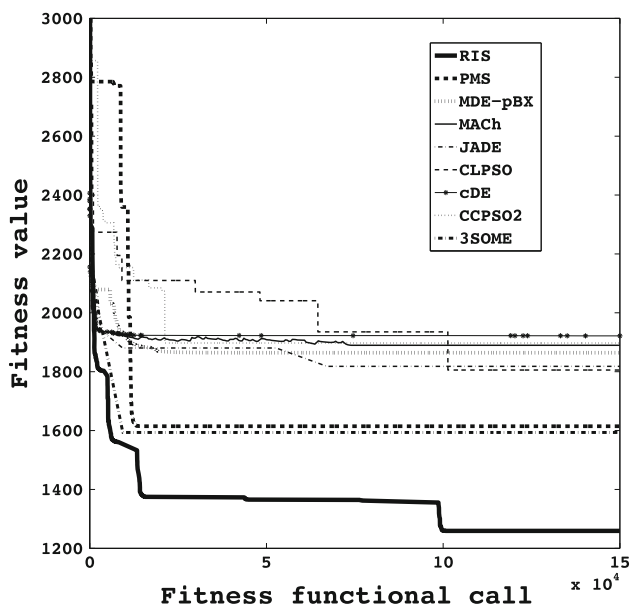


Fig. 6 Performance trend for f_{25} from CEC2005

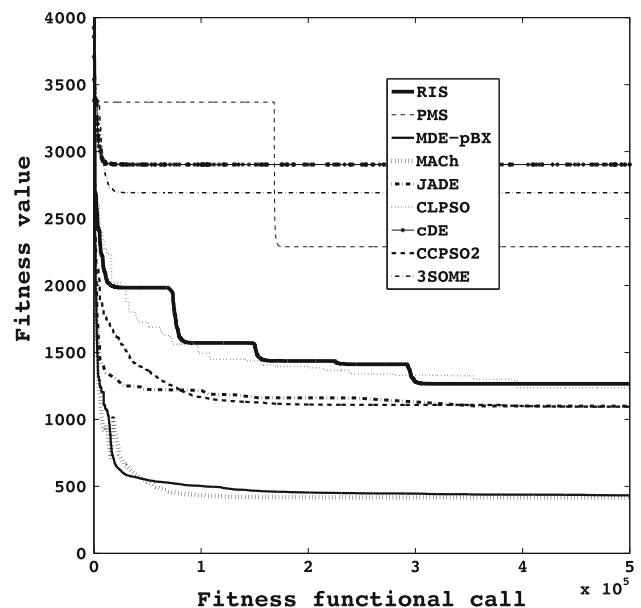


Fig. 7 Performance trend for f_{24} from BBOB2010

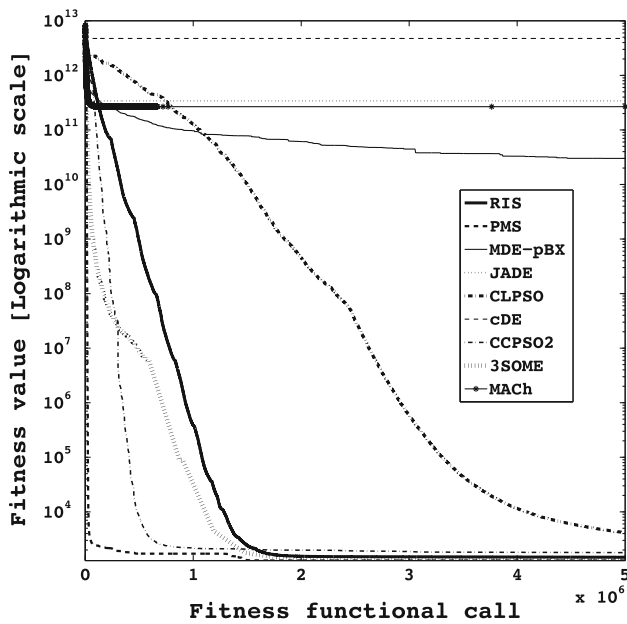


Fig. 8 Performance trend for f_3 from CEC2008

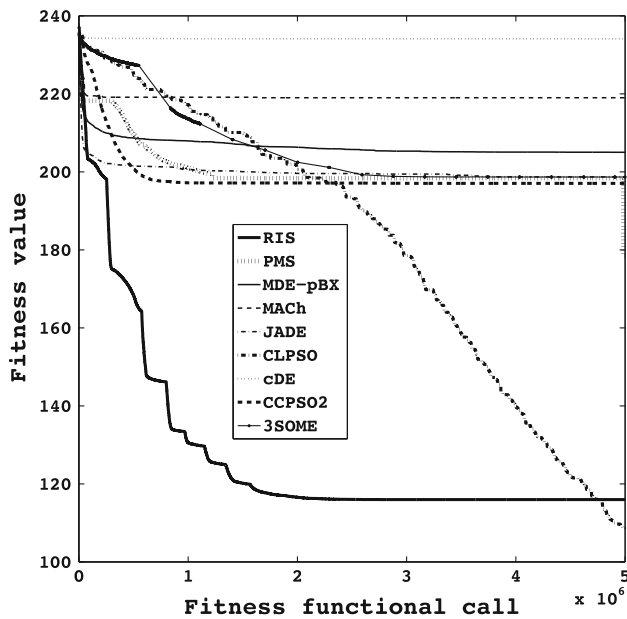


Fig. 9 Performance trend for f_{11} from CEC2010

rotors of this dual coaxial rotor helicopter spin in opposite directions, thus creating opposite torque effects that can cancel each other out. If one rotor’s speed is reduced whilst the other’s speed is increased by an identical amount, the heading will change whilst the amount of lift is maintained. As part of the embedded system, a digital compass is used to determine the current heading. The sensor is connected to a micro-controller which handles all on-board computation, sensor inputs, motor outputs, and serial communication used to transfer information to and from a base station.

In order to properly tune the control system, the dual rotor helicopter is attached to a ball bearing supported turn table, restricted to turn up to 90° and -90° from its middle position. A fan is used to cool down the helicopter’s motors and the embedded system in between controller fitness evaluations. Figure 11 gives an graphical representation of the controlled system, the base station running the algorithms, and the communication between them.

The main disadvantage of helicopter based robotic platforms is that they are highly nonlinear and unstable. As a consequence, the helicopter is prone to be very sensitive to external disturbances (Bagnell and Schneider 2001), and therefore difficult to control. For this reason, the control system for an autonomous helicopter must be able to quickly compute the control response and promptly react to any disturbances.

The PID control systems are characterized by an overall PID control response equal to:

$$PID_{out} = P_{out} + I_{out} + D_{out} \tag{8}$$

where P_{out} , I_{out} , and D_{out} are the individual control method outputs. The input to the individual methods is the error e at time t , which is defined as

$$e(t) = s - m(t) \tag{9}$$

where s is the set-point and m the actual angle measurement at time t . The standard form of the individual control method outputs is given by:

$$P_{out} = K_p e(t) \tag{10}$$

$$I_{out} = K_i \int_0^t e(t) dt \tag{11}$$

$$D_{out} = K_d \frac{de(t)}{dt} \tag{12}$$

where K_p , K_i , and K_d are the proportional, integral, and derivative gains respectively.

The integral-state limited PID controller is a modern variant of the traditional PID controller that limits the integral state variable to a lower and upper bound I_l and I_u respectively, (see Wescott 2000). If the bound values are properly selected, the limit introduction allows a mitigation of the undesired overshoots from integrator windup.

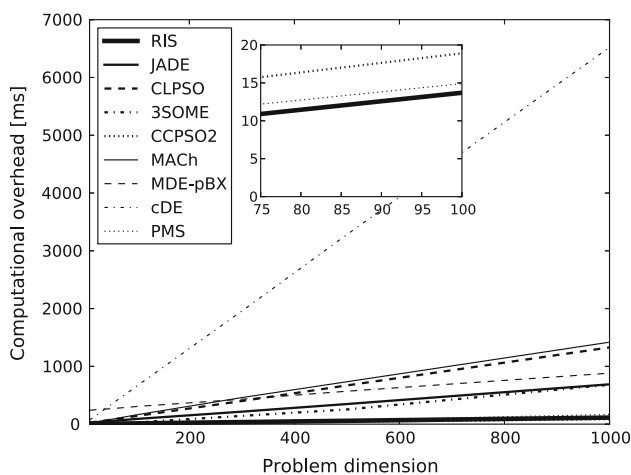
Optimization algorithms have been widely used in the literature to tune PID control systems, (see e.g. Fleming and Purshouse 2002; Passow et al. 2008; Iacca et al. 2012b). In the majority of the cases, a simulation model of the corresponding system is used to evaluate the fitness of each candidate solution (this kind of optimization is also known as off-line), (see De Moura Oliveira 2005). In order to build up a model, an extensive knowledge of the system and a thorough parameter identification are obviously required. Since, due to the presence of disturbances and unforeseeable events,

Table 14 Holm-Bonferroni procedure on the algorithms under consideration, reference algorithm RIS (Rank = $6.46e + 00$)

j	Optimizer	Rank	z_j	p_j	δ/j	Hypothesis
1	PMS	$6.58e + 00$	$2.67e-01$	$6.05e-01$	$5.00e-02$	Accepted
2	CCPSO2	$6.58e + 00$	$2.67e-01$	$6.05e-01$	$2.50e-02$	Accepted
3	RS	$6.09e + 00$	$-8.29e-01$	$2.03e-01$	$1.67e-02$	Accepted
4	MACH	$5.74e + 00$	$-1.63e + 00$	$5.17e-02$	$1.25e-02$	Accepted
5	MDE-pBX	$5.74e + 00$	$-1.63e + 00$	$5.17e-02$	$1.00e-02$	Accepted
6	3SOME	$5.70e + 00$	$-1.72e + 00$	$4.29e-02$	$8.33e-03$	Accepted
7	CLPSO	$5.64e + 00$	$-1.84e + 00$	$3.32e-02$	$7.14e-03$	Accepted
8	JADE	$4.62e + 00$	$-4.15e + 00$	$1.69e-05$	$6.25e-03$	Rejected
9	cDE	$1.79e + 00$	$-1.05e + 01$	$3.72e-26$	$5.56e-03$	Rejected

Table 15 Components and memory requirement of the algorithms under consideration

Algorithm	Features	Memory slots
CLPSO	PSO structure modified velocity rule	$2 \times N_p$
JADE	DE structure samples from distribution archive	$N_p + \text{archive}$
MA-CMA-Chains	GA structure covariance matrix driven search for multiple individuals	$N_p (1 + n^2)$
MA-SSW-Chains	GA structure Solis-Wets Local Search	$N_p (1 + n)$
CCPSO2	PSO structure variable decomposition	$2 \times N_p$
MDE-pBX	DE structure multiple mutation strategies self adaptive parameters	$N_p + \text{neighborhood}$
3SOME	Single-solution structure 3 sequential operators trial and error coordination	3
RIS	Single-solution structure 2 operators	3
cDE	Compact DE based structure 3 samplings, rand/1/bin scheme	4
PMS	Single-solution structure long distance exploration + short distance + Rosenbrock	$3 + n^2$

**Fig. 10** Average computational complexity (overhead vs. dimensionality) of the algorithms under consideration

some physical systems can be very hard to model, the optimization might preferably be performed on the actual plant (on-line optimization) and the fitness values can be measured as the result of one or more experiments, (see Caponio et al. 2010). The difference in terms of control response of a PID controller in off-line and on-line optimization is reported in

Caponio et al. (2007). In the present case, a realistic model taking into account all the dynamics of the helicopter can be extremely hard to build up as explained in Cai et al. (2010). In this work, rather than optimizing the controller using a simulation of the system (which would likely be unreliable), the robot itself is used for the optimization and evaluation of its controller. It must be remarked that a very good optimization algorithm run over an inaccurate model would detect a control system that would likely not be efficient on the actual robot. Thus, on-line optimization implicitly performs a system identification which can be subsequently used to build-up an accurate and realistic simulator.

Each fitness evaluation takes about 20 s with 20 additional seconds to cool down the system. The fan is switched off while candidate solutions are evaluated. For each fitness evaluation the following operations have been performed. At first the motors are switched on in the proximity of the set-point, which is set to zero. Then, a perturbation of $+90^\circ$ is performed (the helicopter tail is forced in one of the extreme position) and 100 control cycles (equivalent to 10 s) are allowed to the control system for its reaction. Finally, the symmetrical perturbation of -90° is performed followed by 10 s of waiting time to allow the control response. Then for 20 s the helicopter is cooled down by the fan. The fitness to be minimized

Table 16 Average fitness \pm SD and Wilcoxon Rank-Sum test (reference = RIS(0.4)) for RIS with various ρ values on CEC2005 (Suganthan et al. 2005) in 30 dimensions

	RIS(0.4)	RIS(0.5)	RIS(0.3)	RIS(0.2)	RIS(0.1)
f_1	$-4.50e + 02 \pm 4.44e - 13$	$-4.50e + 02 \pm 5.58e - 13$	$-4.50e + 02 \pm 7.17e - 13$	$-4.50e + 02 \pm 4.96e - 13$	$-4.50e + 02 \pm 4.42e - 13$
f_6	$6.89e + 02 \pm 4.27e + 02$	$6.35e + 02 \pm 3.71e + 02$	$6.96e + 02 \pm 4.25e + 02$	$5.74e + 02 \pm 3.23e + 02$	$6.82e + 02 \pm 8.17e + 02$
f_{13}	$-1.26e + 02 \pm 1.17e + 00$	$-1.26e + 02 \pm 1.06e + 00$	$-1.26e + 02 \pm 1.17e + 00$	$-1.26e + 02 \pm 1.09e + 00$	$-1.26e + 02 \pm 1.02e + 00$
f_{15}	$2.11e + 02 \pm 1.01e + 02$	$3.05e + 02 \pm 4.74e + 01$	$3.04e + 02 \pm 5.13e + 01$	$2.33e + 02 \pm 1.11e + 02$	$4.97e + 02 \pm 5.79e + 01$

is the quadratic position error related to the helicopter tail. More formally, our optimization problem consists of finding those values for K_p , K_i , and K_d from Eqs. (10), (11), and (12) as well as the values for lower and upper bounds I_l and I_u , such that following error function is minimized:

$$f(K_p, K_i, K_d, I_l, I_u) = \sum_{t=1}^{100} (h_t - s)^2 + \sum_{t=101}^{200} (h_t - s)^2 \tag{13}$$

where t is time in control cycles, h_t is the aircraft heading at time t (measured by the compass sensor), and s is the desired set-point. From $t = 1$ to $t = 100$ (the above-mentioned 10 s) the reaction of the control system to a $+90^\circ$ perturbation is tested, while from $t = 101$ to $t = 200$ the reaction to a -90° is considered. Between the two perturbations, the control system is stopped and its performance not evaluated. The two terms of the sum in Eq. (13) have been intentionally left apart in order to emphasize the separation of the two perturbation actions. The decision space is a hyper-rectangle identified by the Cartesian product generated by the intervals limited by the range of variability of each parameter. More specifically, since $K_p \in [0, 2]$, $K_i \in [0, 1]$, $I_l \in [-400, 0]$, $I_u \in [0, 400]$, and $K_d \in [0, 4]$, the decision space $D = [0, 2] \times [0, 1] \times [-400, 0] \times [0, 400] \times [0, 4]$.

4.1 Optimization results

In order to minimize the fitness function in Eq. (13), the RIS has been run for 600 fitness evaluations with the parameter setting reported in Sect. 3. The performance of the RIS algorithm has been compared against one of the most promising optimization algorithms considered in this study according to the Holm-Bonferroni ranking (see Table 14), i.e. CCPSO2, and against its predecessor 3SOME. As for the 3SOME the same parameter setting shown in Sect. 3 has been used. Regarding the CCPSO2, variable decomposition of size 1 and 5 (the only two possible values) have been allowed. The final results in terms of average values and standard deviation, as well as the Wilcoxon test, are reported in Table 17. The average performance trend of the three algorithms considered for this real-world application is displayed in Fig. 12.

The best fitness value on single run has been achieved by RIS. Table 18 displays the best parameters detected by each algorithm and the corresponding fitness value. Finally, Fig. 13 shows the control response associated to the parameters reported in Table 18. More specifically, the time evolution of the aircraft heading h_t corresponding to the three best control parameters are plotted. It must be observed how the helicopter heading tends to return to oscillate around the set-point after each perturbation.

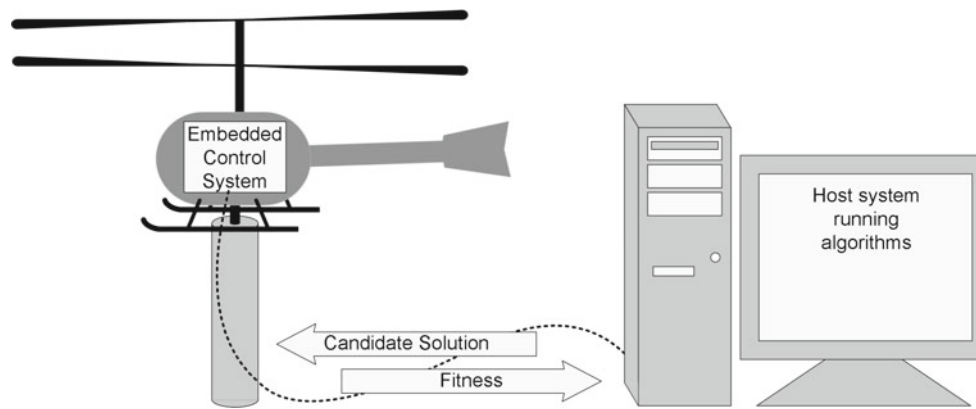


Fig. 11 Experimental setup of evolutionary heading controller tuning

Table 17 Average final fitness value obtained \pm SD and Wilcoxon test outcome (reference: RIS) on the real-world helicopter control problem

Algorithm	Average	SD	W
RIS	19321	3.2534e + 03	
CCPSO2	23167	4.2491e + 03	=
3SOME	24856	1.1367e + 04	=

Table 18 Best PID parameters obtained on the real-world experiments and associated fitness values

	K_p	K_i	I_l	I_u	K_d	f
RIS	0.8104	0.0628	-31.272	50.846	1.2199	13485
CPSO2	0.4031	0.0903	-0.2620	15.976	0.7581	14489
3SOME	0.2103	0.0032	-188.804	340.008	0.7432	16818

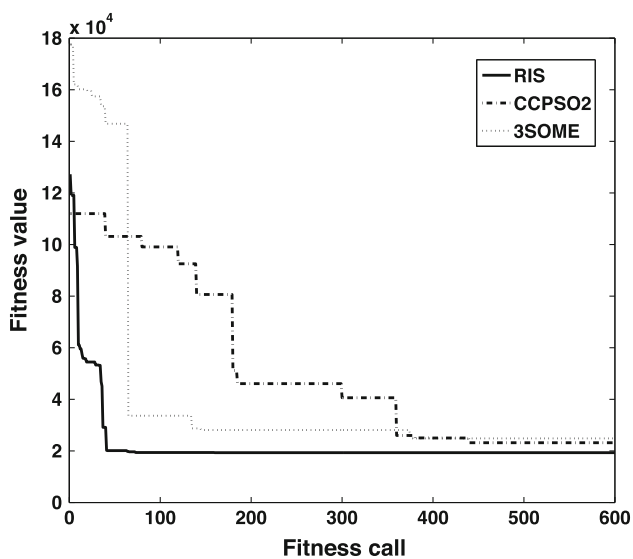


Fig. 12 Performance trend of RIS, 3SOME, and CCPSO2 for the helicopter control problem

Experimental results show then that RIS outperforms the 3SOME and CCPSO2 algorithms, thus confirming the value of the proposed approach. It must be remarked that, due to its modest computational overhead and memory requirement, RIS can be easily implemented directly within the micro-controller thus avoiding the necessity of using an external computer. The embedded implementation of the optimization algorithm within the micro-controller would allow a real-time parameter tuning within the helicopter hardware

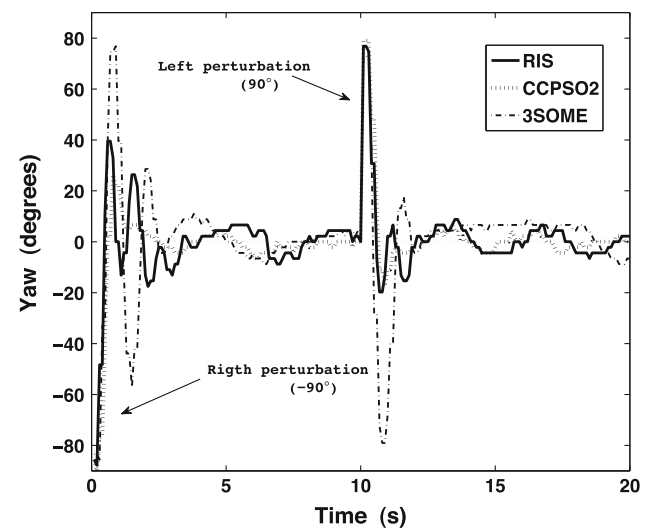


Fig. 13 Control signal related to the best solutions detected by RIS, 3SOME, and CCPSO2 for the helicopter control problem

thus making it completely autonomous. In the future, powerful implementations of simple but yet highly performing algorithms would allow an in-flight optimization.

5 Conclusion

This paper proposes a simple memetic approach for continuous optimization problems. The proposed algorithm (RIS) is composed of two memes, the first one sampling solutions

within the entire decision space, the second performing a deterministic local search to exploit the solutions suggested by the first. Thus, the proposed algorithm can be seen as a local search which incorporates a re-starting technique. However, this re-start is not performed purely random. On the contrary, it combines a certain degree of randomization with part of the genotype of the most promising solution (inheritance). The presence of inheritance appeared to be beneficial within the proposed scheme as it allowed to perform as good or better than its variant which does not make use of it.

An extensive comparison with the state-of-the-art algorithms (including complex schemes) over various problems showed that the proposed algorithm, despite its simplicity, is extremely efficient. The statistical ranking highlights that the proposed RIS displays the best performance with respect to modern and much more sophisticated approaches. This finding confirms the validity of the Ockham's Razor in Memetic Computing. In addition, RIS appears to be especially efficient to tackle large scale problems. This result can be seen as a consequence that exploitative approaches are likely more successful than exploratory ones in high dimensions. The real-world example reported in this study further confirms the efficiency of the proposed approach in industrial problems. Most importantly, due to its minimalistic structure, RIS is characterized by a modest memory requirement and computational overhead. These features, together with its high performance, make the RIS algorithm an appealing candidate for addressing those real-world problems that, as the one considered in this study, would benefit from an embedded and real-time implementation.

Acknowledgments The numerical experiments have been carried out on the computer network of the De Montfort University by means of the software for distributed optimization Kimeme (Cyber Dyne Srl Home Page 2012) and the Memenet project. We thank Dr. Lorenzo Picinali, Dr. Nathan Jeffery, and Mr. David Tunnicliffe for the technical support of the computer network.

References

- Abbass HA (2002) An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artif Intell Med* 25(3):265–281
- Arnold DV, Beyer H-G (May 2003) On the benefits of populations for noisy optimization. *Evol Comput* 11:111–127
- Auger A, Teytaud O (2007) Continuous lunches are free! In: Proceedings of the 9th annual conference on genetic and evolutionary computation. ACM, pp 916–922
- Bagnell J, Schneider J (2001) Autonomous helicopter control using reinforcement learning policy search methods. In: Proceedings of IEEE international conference on robotics and automation, vol 2
- Cai G, Chen B, Lee T (2010) An overview on development of miniature unmanned rotorcraft systems. *Fronti Electr Electron Eng China* 5(1):1–14
- Caponio A, Cascella GL, Neri F, Salvatore N, Sumner M (2007) A fast adaptive memetic algorithm for on-line and off-line control design of PMSM drives. *IEEE Trans Syst Man Cybern part B* 37(1):28–41
- Caponio A, Kononova A, Neri F (2010) Differential evolution with scale factor local search for large scale problems. In: Tenne Y, Goh C-K (eds) *Computational intelligence in expensive optimization problems*, vol 2 of studies in evolutionary learning and optimization, chap. 12. Springer, Berlin, pp 297–323
- Caraffini F, Iacca G, Neri F, Mininno E (2012) Three variants of three stage optimal memetic exploration for handling non-separable fitness landscapes. In: Proceedings of the UK workshop on computational intelligence
- Caraffini F, Iacca G, Neri F, Mininno E (2012) The importance of being structured: a comparative study on multi stage memetic approaches. In: Proceedings of the UK workshop on computational intelligence
- Caraffini F, Neri F, Iacca G, Mol A (2013) Parallel memetic structures. *Inf Sci* 227:60–82
- Cyber Dyne Srl Home Page (2012) Kimeme. <http://cyberdynesoft.it/>
- De Moura Oliveira P (2005) Modern heuristics review for pid control systems optimization: A teaching experiment. In: Proceedings of the 5th international conference on control and automation, ICCA'05, pp 828–833
- Eiben AE, Smit SK (2011) Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm Evol Comput* 1(1):19–31
- Fan XF, Zhu Z, Ong YS, Lu YM, Shen ZX, Kuo J-L (2007) A direct first principle study on the structure and electronic properties of bezn1-xo . *Appl Phys Lett* 91:121
- Fleming P, Purshouse R (2002) Evolutionary algorithms in control systems engineering: a survey. *Control Eng Pract* 10(11):1223–1241
- Garcia S, Fernandez A, Luengo J, Herrera F (2008) A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Comput* 13(10):959–977
- Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley Publishing Co., Reading
- Handoko SD, Kwoh CK, Ong YS (2010) Feasibility structure modeling: an effective chaperon for constrained memetic algorithms. *IEEE Trans Evol Comput* 14(5):740–758
- Hansen N, Müller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol Comput* 11(1):1–18
- Hansen N, Auger A, Finck S, Ros R et al (2010) Real-parameter black-box optimization benchmarking 2010: noiseless functions definitions. Technical Report, RR-6829, INRIA,
- Hart WE, Krasnogor N, Smith JE (2004) Memetic evolutionary algorithms. In: Hart WE, Krasnogor N, Smith JE (eds) *Recent advances in memetic algorithms*. Springer, Berlin, pp 3–27
- Hasan SMK, Sarker R, Essam D, Cornforth D (2009) Memetic algorithms for solving job-shop scheduling problems. *Memetic Comput* 1(1):69–83
- Holm S (1979) A simple sequentially rejective multiple test procedure. *Scand J Stat* 6(2):65–70
- Iacca G, Neri F, Mininno E, Ong YS, Lim MH (2012) Ockham's razor in memetic computing: three stage optimal memetic exploration. *Inf Sci* 188:17–43
- Iacca G, Caraffini F, Neri F (2012) Compact differential evolution light. *J Comput Sci Technol* 27(5):1056–1076
- Islam S, Das S, Ghosh S, Roy S, Suganthan P (2012) An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Trans Syst Man Cybern Part B Cybern* 42:482–500
- Ji M, Klinowski J (2006) Taboo evolutionary programming: a new method of global optimization. *Proc R Soc Lond Ser A Math Phys Eng Sci* 462(2076):3613–3627
- Joshi R, Sanderson AC (1999) Minimal representation multisensor fusion using differential evolution. *IEEE Trans Syst Man Cybern Part A* 29(1):63–76
- Lee C-Y, Yao X (2004) Evolutionary programming using mutations based on the levy probability distribution. *IEEE Trans Evol Comput* 8(1):1–13
- Li X, Yao X (2012) Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans Evol Comput* 16:210–224

- Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10(3):281–295
- Lim KK, Ong Y-S, Lim MH, Chen X, Agarwal A (2008) Hybrid ant colony algorithms for path planning in sparse graphs. *Soft Comput* 12(10):981–994
- Lozano M, Herrera F, Molina D (2011) Scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems. *Soft Comput* 15(11)
- Mallipeddi R, Mallipeddi S, Suganthan PN (2010) Ensemble strategies with adaptive evolutionary programming. *Inf Sci* 180(9):1571–1581
- Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput* 11(2):1679–1696
- Meuth R, Lim MH, Ong YS, Wunsch-II DC (2009) A proposition on memes and meta-memes in computing for higher-order learning. *Memetic Comput J* 1(2):85–100
- Mininno E, Neri F, Cupertino F, Naso D (2011) Compact differential evolution. *IEEE Trans Evol Comput* 15(1):32–54
- Molina D, Lozano M, Garcia-Martinez C, Herrera F (2010) Memetic algorithms for continuous optimization based on local search chains. *Evol Comput* 18(1):27–63
- Molina D, Lozano M, Herrera F (2010) MA-SW-Chains: memetic algorithm based on local search chains for large scale continuous global optimization. In: *IEEE congress on, evolutionary computation* pp 1–8
- Montes de Oca MA, Stutzle T, Birattari M, Dorigo M (2009) Frankenstein's PSO: a composite particle swarm optimization algorithm. *IEEE Trans Evol Comput* 13(5):1120–1132
- Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report 826
- Moscato P, Norman M (1989) A competitive and cooperative approach to complex combinatorial search. Technical report 790
- Neri F, Mininno E (2010) Memetic compact differential evolution for cartesian robot control. *IEEE Comput Intell Mag* 5(2):54–65
- Neri F, Tirronen V (2010) Recent advances in differential evolution: a review and experimental analysis. *Artif Intell Rev* 33(1–2):61–106
- Neri F, Cotta C (2012) Memetic algorithms and memetic computing optimization: a literature review. *Swarm Evol Comput* 2:1–14
- Neri F, Toivanen J, Mäkinen RAE (2007a) An adaptive evolutionary algorithm with intelligent mutation local searchers for designing multidrug therapies for HIV. *Appl Intell* 27:219–235
- Neri F, Toivanen JI, Cascella GL, Ong YS (2007b) An adaptive multi-meme algorithm for designing HIV multidrug therapies. *IEEE/ACM Trans Comput Biol Bioinform* 4(2):264–278
- Neri F, Iacca G, Mininno E (2011) Disturbed exploitation compact differential evolution for limited memory optimization problems. *Inf Sci* 181(12):2469–2487
- Neri F, Cotta C, Moscato P (2011) Handbook of memetic algorithms, vol 379 of *Studies in Computational Intelligence*. Springer, Berlin
- Neri F, Weber M, Caraffini F, Poikolainen I (2012) Meta-lamarckian learning in three stage optimal memetic exploration. In: *Proceedings of the UK workshop on computational intelligence*
- Nguyen QC, Ong YS, Lim MH (2009a) A probabilistic memetic framework. *IEEE Trans Evol Comput* 13(3):604–623
- Nguyen QH, Ong YS, Hiot LM, Krasnogor N (2009b) Adaptive cellular memetic algorithms no access. *Evol Comput* 17(2):231–256
- Ong YS, Keane AJ (2004) Meta-Lamarckian learning in memetic algorithms. *IEEE Trans Evol Comput* 8(2):99–110
- Ong Y-S, Lim M-H, Chen X (2010) Memetic computation-past, present and future. *IEEE Comput. Intell Mag* 5(2):24–31
- Passow BN, Gongora MA, Coupland S, Hopgood AA (2008) Real-time evolution of an embedded controller for an autonomous helicopter. In: *Proceedings of the IEEE international congress on evolutionary computation (CEC'08)*, (Hong Kong), pp 2538–2545
- Peng F, Tang K, Chen G, Yao X (2010) Population-based algorithm portfolios for numerical optimization. *IEEE Trans Evol Comput* 14(5):782–800
- Poikolainen I, Caraffini F, Neri F, Weber M (2012) Handling non-separability in three stage memetic exploration. In: *Proceedings of the fifth international conference on bioinspired optimization methods and their applications*, pp 195–205
- Poikolainen I, Neri F, Mininno E, Iacca G, Weber M (2012) Shrinking optimal three stage memetic exploration. In: *Proceedings of the fifth international conference on bioinspired optimization methods and their applications*, pp 61–74
- Price KV, Storn R, Lampinen J (2005) *Differential Evolution: a practical approach to global optimization*. Springer, Berlin
- Rogalsky T, Derksen RW (2000) Hybridization of differential evolution for aerodynamic design. In: *Proceedings of the 8th annual conference of the computational fluid dynamics society of Canada*, pp 729–736
- Rosenbrock HH (1960) An automatic Method for finding the greatest or least value of a function. *Comput J* 3(3):175–184
- Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y-P, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report, 2005005. Nanyang Technological University and KanGAL, Singapore and IIT Kanpur, India
- Tan KC, Cheong CY, Goh CK (2007) Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation. *Eur J Oper Res* 177(2):813–839
- Tang K, Yao X, Suganthan PN, MacNish C, Chen YP, Chen CM, Yang Z (2007) Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. Technical report, Nature Inspired Computation and Applications Laboratory, USTC, China
- Tang K, Li X, Suganthan PN, Yang Z, Weise T (2010) Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization. Technical report. University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), Hefei, Anhui, China
- Tseng L-Y, Chen C (2008) Multiple trajectory search for large scale global optimization. In: *Proceedings of the IEEE congress on, evolutionary computation*, pp 3052–3059
- Vrugt JA, Robinson BA, Hyman JM (2009) Self-adaptive multimethod search for global optimization in real-parameter spaces. *IEEE Trans Evol Comput* 13(2):243–259
- Wescott T (2000) Pid without a phd. *Embed Syst Program* 13(11):86–108
- Wilcoxon F (1945) Individual comparisons by ranking methods. *Biometr Bull* 1(6):80–83
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1(1):67–82
- Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. *IEEE Trans Evol Comput* 3(2):82–102
- Zaharie D (2009) Influence of crossover on the behavior of differential evolution algorithms. *Appl Soft Comput* 9(3):1126–1138
- Zamuda A, Brest J (2012) Population reduction differential evolution with multiple mutation strategies in real world industry challenges. In: *ICAISC (SIDE-EC)*, pp 154–161
- Zamuda A, Brest J, Bošković B, Zumer V (2011) Differential evolution for parameterized procedural woody plant models reconstruction. *Appl Soft Comput* 11(8):4904–4912
- Zhang J, Sanderson AC (2009) JADE: adaptive differential evolution with optional external archive. *IEEE Trans Evol Comput* 13:945–958
- Zhou J, Ji Z, Shen L, (2008) Simplified intelligence single particle optimization based neural network for digit recognition. In: *Proceedings of the Chinese conference on, pattern recognition*, pp 1–5 (1031–1847)