Ann Oper Res
DOI 10.1007/s10479-009-0596-2

Hybrid scatter tabu search for unconstrained global optimization

Abraham Duarte · Rafael Martí · Fred Glover · Francisco Gortazar

Abstract The problem of finding a global optimum of an unconstrained multimodal function has been the subject of intensive study in recent years, giving rise to valuable advances in solution methods. We examine this problem within the framework of adaptive memory programming (AMP), focusing particularly on AMP strategies that derive from an integration of Scatter Search and Tabu Search. Computational comparisons involving 16 leading methods for multimodal function optimization, performed on a testbed of 64 problems widely used to calibrate the performance of such methods, disclose that our new Scatter Tabu Search (STS) procedure is competitive with the state-of-the-art methods in terms of the average optimality gap achieved.

Keywords Adaptive memory programming · Metaheuristics · Hybridization

1 Introduction

The unconstrained continuous global optimization problem may be formulated as follows:

$$(P) \quad \begin{array}{l} \text{Minimize } f(x) \\ \text{subject to } l \leq x \leq u, \quad x \in \Re^n, \end{array}$$

A. Duarte · F. Gortazar
Departamento de Ciencias de la Computación, E.T.S. Ingeniería Informática, Universidad Rey Juan Carlos, Ed. Dept. II. C/ Tulipán s/n. 28933 Móteles, Madrid, Spain

A. Duarte
e-mail: Abraham.Duarte@urjc.es

F. Gortazar
e-mail: Francisco.Gortazar@urjc.es

R. Martí (✉)
Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universitat de València, Dr. Moliner 50, 46100 Burjassot, Valencia, Spain
e-mail: Rafael.Marti@uv.es

F. Glover
University of Colorado and OptTek Systems, Inc., Boulder, USA
e-mail: Fred.Glover@colorado.edu

Published online: 08 August 2009

where $f(x)$ is a nonlinear function and x is a vector of continuous and bounded variables.

We investigate Adaptive Memory Programming (AMP) methods for P based on the Scatter Search (SS) and Tabu Search (TS) methodologies, and perform comparative computational testing against currently leading methods for unconstrained global optimization on a widely used benchmark set of problems for which global optima are known.

TS and SS have common roots in basic principles going back three decades (see, e.g., Glover 1977) and extended more recently in a variety of expositions (Glover and Laguna 1997), including settings that explore primal-dual relationships by effectively integrating adaptive memory principles derived from surrogate constraint relaxations with TS and SS to create what is generally called a relaxation adaptive memory programming (RAMP) approach (Rego 2005). Expositions that focus especially on SS (and its close relative, path relinking) can also be found in Glover et al. (2000) and Laguna and Martí (2003). The TS and SS metaheuristics are often combined, to marry the evolutionary strategy of Scatter Search with the adaptive memory focus of Tabu Search, as documented in the articles featured in Rego and Alidaee (2005).

In prior work on unconstrained global optimization, SS was applied as a stand-alone method (without the hybridization with Tabu Search) in Laguna and Martí (2005). This implementation was based on the "SS template" (Glover 1998) which consists of the following five components:

1. A *diversification-generation method* to generate a collection of diverse trial solutions, using an arbitrary trial solution (or seed solution) as an input.
2. An *improvement method* to transform a trial solution into one or more enhanced trial solutions. (Neither the input nor output solutions are required to be feasible, though the output solutions will more usually be expected to be so.)
3. A *reference-set update method* to build and maintain a *reference set* consisting of the b "best" solutions found (where the value of b is typically small, e.g. no more than 20). New solutions are obtained by combination of solutions in the reference set. Solutions gain membership to the reference set according to their quality or their diversity.
4. A *subset-generation method* to operate on the reference set, to produce several subsets of its solutions as a basis for creating combined solutions.
5. A *solution-combination method* to transform a given subset of solutions produced by the Subset Generation Method into one or more combined solution vectors.

Laguna and Martí (2005) experimented with an abridged version of SS that excluded the improvement method, and focused on testing several alternatives for generating diversification and updating the reference set. The combinations generated by their approach are linear and limited to joining pairs of solutions. They also tested the use of a two-phase intensification. In the present study, we compare our procedure with the best version of SS proposed in that work.

We also make reference to the following additional approaches to provide a basis for comparing our procedure with the current leading methods for P.

Direct Tabu Search, DTS, due to Hedar and Fukushima (2006), is based on the following three procedures:

- An exploration procedure, based on the simplex method (Nelder and Mead 1965) and on the adaptive pattern search strategy (Hedar and Fukushima 2004), to generate trial moves. Anti-cycling is prevented not only with the standard *tabu list* but also with the inclusion of *tabu regions* (Glover 1994).
- A diversification procedure to generate new points outside the visited regions. To do so, the visited regions list, VRL, is stored during the search.

- An intensification procedure, applied at a final stage, to refine the elite solutions and thereby obtain a solution still closer to the global optimum.

The computational experience reported in Hedar and Fukushima (2006) shows that DTS improves upon a previous implementation of Tabu Search for global optimization (Chelouah and Siarry 2005) as well as upon the SS method (Laguna and Martí 2005) described above.

Addis et al. (2005) proposed a smoothing transformation of the local search results as an approximation framework. A common approach to deal with complex multi-modal functions is to optimize the underlying smooth function, often called the *funnel structure*, instead of the original objective function. However, in this paper they considered an interesting variant: optimize the smoothed transform of the local search operator. A computational comparison over 4 well known functions, Rastrigin, Levy, Ackley and Schwefel, shows that the proposed method improves upon the previous Monotonic Basin-Hopping Method (MBH, Leary 2000). Addis and Leyffer (2006) improved the previous algorithm adding a trust-region framework to it. This is a reactive method that adaptively updates two key search parameters of the previous algorithm: the radius of the ball in which the local optimization takes place, and the sample size. A comparison over the four functions mentioned shows the contribution of this reactive mechanism in terms of percentage of success (i.e. number of times that the local search is able to improve the initial point) and CPU time.

Genetic algorithms (GAs) have also been applied to global optimization (see, e.g., Michalewicz 1996). Although the population-based character of the SS and GA methods qualifies them both as evolutionary procedures, there are fundamental differences between these two approaches (Laguna and Martí 2003). Hansen (2006) reports the results of the empirical comparison of different solving methods on 25 benchmark instances presented at the Congress on Evolutionary Computation CEC2005. We can find there the following 11 algorithms, most of them based on hybrid evolutionary strategies and methods. Specifically, they are:

- BLX-GL50. Hybrid real coded genetic algorithm (García-Martínez and Lozano 2005)
- BLX_MA. Real coded memetic algorithm (Molina et al. 2005)
- CoEVO. Cooperative co-evolutionary algorithm (Posik 2005)
- DE. Differential evolution algorithm (Rönkkönen et al. 2005)
- DMS-L-PSO. Particle multi-swarm optimizer (Liang and Suganthan 2005)
- EDA. Continuous estimation of distribution algorithm (Yuan and Gallagher 2005)
- G-CMA-ES. Covariance matrix adaptation evolution strategy (Auger and Hansen 2005a),
- k_PCX. Population based steady-state algorithm (Sinha et al. 2005)
- L-CMA-ES. Advanced local search evolutionary algorithm (Auger and Hansen 2005b)
- L-SaDE. Self adaptive differential evolution algorithm (Qin and Suganthan 2005)
- SPC-PNX. Real parameter genetic algorithm (Ballester et al. 2005)

Hirsch et al. (2007) introduced C-GRASP which implements a continuous heuristic based on the GRASP methodology (Resende and Ribeiro 2001). As in the standard GRASP, C-GRASP has two phases, a construction and an improvement phase, that are repeated for a specified maximum number of iterations. However, GRASP is based on an independent random sampling of the solution space and in C-GRASP most of the consecutive iterations (construction + improvement) are dependent upon each other since the solution obtained with the improvement method seeds the next construction.

Starting from a solution x, the construction phase in C-GRASP performs multiple line-searches in which only one variable is modified and the other $n-1$ remain fixed. The restricted candidate list is formed from the variables that have a good evaluation and the method randomly selects among the k best of these (where k is a parameter calibrated by preliminary testing) to perform a line search over the selected variable. The construction finishes after n steps when all the variables are fixed. The improvement phase generates a set of directions and determines in which direction, if any, the objective function value improves. The number of directions is set to 30 in C-GRASP to reduce the computational effort and the procedure stops after 200 global iterations (construction + improvement).

The next section describes our proposed approach for the unconstrained global optimization problem, and Sect. 3 provides some extensions to this design. We perform a computational study comparing our method to the leading methods previously indicated, applied to a set of 64 benchmark problems whose form is described in Sect. 4, where we also report our computational findings. Finally, we summarize our conclusions and offer suggestions about future research directions in Sect. 5.

2 Scatter search design

The fundamental structure of our Scatter Search procedure is sketched in Fig. 1. The method starts with the creation of an initial large set of diverse solutions with the Diversification Generation Method:

Controlled randomization and frequency memory are used to generate the set D of diverse solutions by dividing the range of each variable $u_i - l_i$ into sr sub-ranges of equal size. In the study of Laguna and Martí (2005) sr is set to 4. We investigate the effect on the SS algorithm of the changes in this search parameter. A solution is then constructed in two steps. First a sub-range is selected by a randomized rule using a probability inversely proportional to the frequency count associated with the sub-range. Then a value is randomly generated within the selected interval. The number of times sub-range j has been chosen to generate a value for variable i is accumulated in $\mathit{freq}(i, j)$.

2.1 Initialization

Since we want the solutions in D to be diverse, we do not directly admit a generated solution x to become part of D, but only admit those with a distance to the solutions already in D, $d(x, D)$, larger than a pre-established distance threshold value *dthresh*. In mathematical terms:

$$d(x, D) = \min_{y \in D} d(x, y) \geq \mathit{dthresh}.$$

The value of the *dthresh* parameter is set in our preliminary experimentation described in Sect. 4. In the expression above we use the Euclidean distance $d(x, y)$ between two given solutions x and y. Therefore, in the initialization we generate solutions with this frequency mechanism until *DSize* solutions qualify to enter in D.

Instead of the one-by-one selection of diverse solutions typically employed in previous Scatter Search applications to build the Reference Set, we propose solving the max-min

```
1. Start with $D = Pool = RefSet = \varnothing$.
while ($|D| < DSize$) {
    2. Use the Diversification Generation Method to construct a solution $x$.
    3. If $x \notin D$ and $d(x, D) > dthresh$ then add $x$ to $D$ (i.e., $D = D \cup \{x\}$).
4. Evaluate the solutions in $D$ and build $RefSet = \{x^1, \dots, x^{b1}\}$ with the best $b_1$ solutions according to $f$. $NumEval = |D|$
}
while ($NumEval < MaxEval$) {
    5. Solve the max-min diversity problem in $D$ to obtain $b_2$ diverse solutions ($b_2 = b - b_1$) with respect to the solutions already in $RefSet$.
    6. Build $RefSet = \{x^1, \dots, x^b\}$ with the $b_1$ quality and $b_2$ diverse solutions.
    7. Evaluate the solutions in $RefSet$ and order them according to their objective function value (where $x^1$ is the best solution).
    8. Make $NewSolutions = $ TRUE
    while ($NewSolutions$) {
        9. $Pool = \varnothing$. $NewSolutions = $ FALSE
        10. Generate $NewSubset$, which consists of all pairs of solutions in $RefSet$ that include at least one new solution.
        while ($NewSubset \neq \varnothing$) {
            11. Select the next subset $s$ in $NewSubset$.
            12. Apply the Solution Combination Method to $s$ to obtain a new solution $x$.
            13. Evaluate $x$. $NumEval++$
            14. Add $x$ to $Pool$ ($Pool = Pool \cup \{x\}$)
            15. Apply the Improvement Method to the best $b$ solutions in $Pool$. Replace these $b$ solutions with the outputs of the Improvement Method. Update $NumEval$ adding the number of evaluations performed.
            16. Delete $s$ from $NewSubset$
        }
        while ($Pool \neq \varnothing$) {
            17. Select the next solution $x$ in $Pool$.
            18. Let $y_x$ be the closest solution in $RefSet$ to $x$.
            if ($f(x) < f(x^1)$ or ($f(x) < f(x^b)$ & $d(x, y_x) > dthresh$))
                19. Add $x$ to the $RefSet$ and remove $x^b$ ($RefSet = RefSet \backslash \{x^b\} \cup \{x\}$).
                20. Make $NewSolutions = $ TRUE.
            21. Remove $x$ from $Pool$
        }
    }
    22. Remove the worst $b_2$ solutions from the $RefSet$
}
```

Fig. 1 Outline of the Scatter Search procedure

diversity problem (MMDP) in the step 4 of Fig. 1.

$$
\begin{aligned}
&\text{Maximize } z = \min_{x^i, x^j \in D} d(x^i, x^j) x_i x_j \\
&\text{subject to: } \sum_{i=1}^{n} x_i = b, \\
&\qquad\qquad\quad x_i \in \{0, 1\}, \quad i = 1, \dots, n.
\end{aligned}
$$

Since the MMDP is a computationally hard problem, we have chosen the D_2 method (Glover et al. 1998) because it provides a good balance between solution quality and speed, attributes that are important in order to embed it as part of the overall solution method. The MMDP consists of finding, from a given set of elements (D in this case) and corresponding distances between elements, the most diverse subset of a given size (the Reference Set size b). The diversity of the chosen subset is given by the minimum of the distances between every pair of elements: The assignment $x_i = 1$ indicates that the associated solution in D, x^i, is selected.

Starting with all the elements selected, D_2 unselects, at each step, the element with the minimum distance to the set Sel of selected elements. The method finishes when $n - b$ elements have been unselected. The distance between an element x^i and the set Sel is defined as follows:

$$d(x^i, Sel) = \sum_{y \in Sel} d(x^i, y).$$

The best b_1 solutions in terms of the objective function f are selected from D. Then, the b_2 ($b_2 = b - b_1$) most diverse solutions in D found with the D_2 method considering that b_1 solutions are already selected, are chosen to form the set *RefSet*. The initialization of the SS algorithm finishes with the construction of the Reference Set, $RefSet = \{x^1, \dots, x^b\}$ with the b_1 quality and b_2 diverse solutions from D. Note that the b solutions are selected from D and only generated solutions verifying the distance threshold *dthresh* are in D. Therefore we are considering diversity in two stages of the method.

2.2 Reference set update

The solutions in *RefSet* are ordered according to quality, where the best solution is the first one in the list. The search is then initiated by assigning the value of TRUE to the Boolean variable *NewSolutions*. In step 10, *NewSubset* is constructed with all pairs of solutions in *RefSet*. The pairs in *NewSubset* are selected one at a time in lexicographical order and the Solution Combination Method is applied to generate a new solution in step 12. The $(b^2 - b)/2$ combined solutions are stored in a new set called *Pool*. The Improvement Method is selectively applied to save computational effort. This is why we introduce the set *Pool* in the step 9 of the algorithm shown in Fig. 1. The Improvement Method is applied to the best b solutions in *Pool* in the step 14. Each of these b solutions is replaced with the output of the Improvement Method.

In our SS procedure we restrict the application of the Improvement Method. We consider three alternative algorithms for the Improvement. In this section we describe a local search based on line-searches for solution improvement. In Sect. 3 we propose two different extensions of the improvement procedure based on the Tabu Search methodology. Section 3.1 describes a short term Tabu Search algorithm based on line-searches, and Sect. 3.2 is devoted to a memory-based implementation of the Nelder-Mead simplex method. The selective application of the Improvement Method as well as these three alternative algorithms, are tested and compared in the computational experiments reported on Sect. 4.

If solution x qualifies to enter *RefSet*, then, the worst solution x^b is removed from it. The *NewSolutions* flag is switched to TRUE. If a new solution entered the *RefSet*, in the next main loop, when generating the pairs of solutions in the *RefSet* (step 10), only pairs containing new solutions are included in *NewSubset*. Finally, when no new solutions are admitted to the reference set in the main *while* loop in Fig. 1, the SS methodology dictates that the search either terminates or a *RefSet* rebuilding step is performed. The rebuilding step consists of eliminating all but the best b_2 reference solution and reinitializing the process from the step 5 in Fig. 1. In our implementation, we have chosen to terminate the SS method after a pre-specified number of solution evaluations *MaxEval* as in most of the previous applications referenced in Sect. 1.

In the final steps of the SS algorithm, we test whether the solutions in *Pool* qualify to enter the *RefSet*. Given a solution x in *Pool*, let y_x be the closest solution to x in *RefSet*. We admit x to *RefSet* if it improves upon the best solution in it, x^1, or alternatively, if it improves upon the worst solution, x^b, and its distance with the closest solution in the *RefSet*, y_x, is larger than the pre-established distance threshold *dthresh* introduced above. The criteria for admission to *RefSet* can be stated as:

$$f(x) < f(x^1) \quad \text{or} \quad (f(x) < f(x^b)\ \&\ d(x, y_x) > \mathit{dthresh}).$$

2.3 Improvement method

One of the most commonly used ways to improve solutions in global optimization consists of the so-called line-search. Given a solution x and an index variable i, the line-search from x in the i-direction can be represented as the set $ls(x, h, i)$ where h is the width of the uniform grid of the discretized search space. This set contains all the feasible solutions that can be obtained in the h-grid by modifying variable x_i in solution x. In mathematical terms:

$$ls(x, h, i) = \{y \in \Re^n \mid y = x + ke_i h,\ k \in Z,\ l \le y \le u\},$$

where e_i represents the unit vector with all 0s except a 1 in the i position. Our improvement method consists of a local search based on line-searches. We consider a typical implementation of line search. Given a solution x, we first order the variables in a random fashion and then perform the line search associated with each variable in this order. In other words, at a given step, we scan the set $ls(x, h, i)$ associated with variable i and if it contains a better solution than x, we move to it (and replace x with the improved solution). The improvement method performs iterations until no further improvement is possible (i.e., when all the variables have been examined and no improvement is found). Then, it returns the best solution found as the output of the method.

2.4 Combination method

Given two solutions x and y, the Combination Method considers the line through x and y given by the representation

$$z(\lambda) = x + \lambda(y - x)$$

where λ is a scalar weight. We consider three points in this line: the *convex* combination $z(1/2)$, and the *exterior* solutions $z(-1/3)$ and $z(4/3)$. We evaluate these three solutions and return the best as the result of the combination of x and y.

3 Tabu search extensions

Our method couples the SS design described in the previous section with Tabu Search elements in the following manner. We have considered two alternative extensions for the improvement method; the first introduces memory structures in the line-search scheme described above, and the second replaces the line searches with an implementation of the Nelder-Mead simplex method mentioned in the introduction, which we also modify by incorporating memory structures to improve the method's performance.

3.1 Tabu line search

Our improvement method in this instance consists of a short term Tabu Search procedure based on line-searches. A global iteration first orders the variables according to their attractiveness for movement in the current solution. Given a solution x, we compute for each variable i two associated solutions, x^{i+h} and x^{i-h} where:

$$x = (x_1, x_2, \dots, x_i, \dots, x_n), \qquad x^{i+h} = (x_1, x_2, \dots, x_i + h, \dots, x_n),$$
$$x^{i-h} = (x_1, x_2, \dots, x_i - h, \dots, x_n).$$

We evaluate the attractiveness for movement of each variable i in x, $A(x,i)$, as

$$A(x,i) = \max(f(x) - f(x^{i+h}), f(x) - f(x^{i-h})).$$

Then, given an initial solution x, a global iteration of the improvement method first computes $A(x,i)$ for $i = 1$ to n, and then orders the variables according to their A-values (where the variable with the largest A-value comes first). The first ts variables are selected one at a time in this order, and their corresponding line-searches are performed. Say for instance that j is the first variable in the list. Then we scan $ls(x,h,j)$ and select the best solution x' in this set. We then consider the second variable in the list, say k, and scan its associated line search from x': $ls(x',h,k)$ selecting the best solution x'' in this set if it improves upon the current solution, and so on.

As it is customary in Tabu Search implementations, we permit non-improving moves that deteriorate the objective value. Specifically, in the first step, the method selects the best solution x' in $ls(x,h,j)$ and the search moves from x to x', even if $f(x') > f(x)$. In a similar way, in the second step the method moves from x' to x'' thus defining the trajectory of the Tabu Search algorithm. Also note that when we select the second variable in the list, say k, to perform the move from x' to x'', its attractiveness, $A(x,i)$, is computed with respect to the initial solution x and we do not perform an update by computing $A(x',i)$, so that the attractiveness value $A(x,i)$ only represents an indicator. This is why the attractiveness information is updated after ts iterations and we do not explore additional variables in the list. At this point the search can either stop or continue with the next global iteration. In the latter case the A-values are first computed with respect to the solution obtained in the previous global iteration and the variables are ordered according to the values obtained.

Our Tabu Search algorithm implements a short term memory structure incorporating the following simple design. When a variable j is selected and we move to the best solution in its associated line-search, we labeled j as tabu and we do not allow the method to select it in the next *tenure* iterations. Therefore, in each global iteration of the TS algorithm, it selects the first ts non-tabu variables in the list computed with the A-values. When TS finishes it returns as the output of the method to the main algorithm the best solution visited during its application. If no improvement has been found, it returns the initial solution as the output.

3.2 Nelder-Mead simplex search

The simplex search procedure of Nelder and Mead (1965) is a popular method for unconstrained minimization which does not use derivatives. A good description is found in Avriel (1976). It maintains a set of $n+1$ points, located at the vertices of a n-dimensional simplex. Each major iteration attempts to replace the worst point by a new and better one using reflection, expansion, and contraction steps. The reflection step moves from the centroid of all points except for the worst, in a direction away from the worst toward the centroid, to a new point x_r. If x_r is better than all others, the expansion step moves farther in this direction. If x_r is worse than all points in the original simplex but the worst, then the contraction step replaces the original simplex by a new one that retains the best point, but with the other vertices some fraction of their original distances from this best point. Some variants can be shown to converge to a local minimum of a smooth function, but the rate of convergence is at most linear, like steepest descent. However, the customary implementations of this method have been shown by Hvattum and Glover (2007) to perform very poorly, and consequently we create a modified approach that includes a TS memory component.

Given a solution x, our Improvement Method starts by perturbing each variable to create an initial simplex from which the local search begins. Let pt be the amount in which we modify the value of x in each variable to obtain the n points $x^1, x^2, \dots, x^n$, according to:

$$x^i = (x_1, x_2, \dots, x_i + pt, \dots, x_n) \quad \text{for } i = 1, \dots, n.$$

Tabu restrictions in continuous spaces can be based both on directional considerations and locational considerations, where the latter are usually expressed in terms of proximity to solutions previously visited (see, e.g., Glover 1994). In our present work, we adopt a proximity criterion for generating tabu restrictions, which operates by reference to a distance threshold T. Accompanying this, we use a simple memory structure to record all trial solutions we operate on with our modified simplex method. Then, at a given iteration, before applying the improvement method to a given trial solution, we test whether the trial solution lies within a hypersphere of radius T centered at any solution previously submitted to the simplex method (or centered at any of the perturbed solutions x^i). If, so, the trial solution is considered tabu, and the improvement method is not applied. In order to reduce the computational effort associated with this memory structure, as in a customary short term tabu list, we limit the memory by maintaining a record only of the last *NumSol* solutions submitted to the simplex search.

4 Computational experiments

Our computational testing consists of several experiments. The first set of experiments determines the key search parameters of our method. Then we perform two additional sets of experiments, as detailed below, to compare our method with the best known procedures from the literature. We consider two sets of instances:

LM: This data set consists of 40 test problems with n ranging from 2 to 30. These instances were reported in Laguna and Martí (2005). The initialization ranges for each problem are the same as in their "original sources":
http://www.cyberiad.net/realbench.htm
http://solon.cma.univie.ac.at//glopt.html

CEC: This data set consist of 24 instances (12 with $n = 10$, and 12 with $n = 30$). These instances are extremely difficult, consisting of multimodal function mostly obtained by composition and hybridization of functions in the LM data set (biased, rotated, shifted and added). These instances are described in detail in Suganthan et al. (2005) and reported in Hansen (2006) under the section entitled "Never solved multimodal functions". The initialization ranges for each problem are the same as in their "original sources":
http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-05/CEC05.htm.

Table 1 Test problems

Set	Name of instance
LM	Branin, B2, Easom, Goldstein and Price, Shubert, Beale, Booth, Matyas, SixHumpCamelback, Schwefel(2), Rosenbrock (2), Zakharov(2), De Joung, Hartmann(3,4), Colville, Shekel(5), Shekel(7), Shekel(10), Perm(4,0.5), Perm0(4,10), Powersum (8,18,44,114), Hartmann(6,4), Schwefel(6), Trid(6), Trid(10), Rastrigin(10), Griewank(10), Sum Squares(10), Rosenbrock(10), Zakharov(10), Rastrigin(20), Griewank(20), Sum Squares(20), Rosenbrock(20), Zakharov(20), Powell(24), Dixon and Price(25), Levy(30), Sphere(30), Ackley(30)
CEC	F8—Shifted rotated Ackley, F13—Shifted expanded Griewank plus Rosenbrock, F14—Shifted rotated expanded Scaffer, F16—Rotated Rastrigin et al., F17—Rotated Rastrigin et al. biased with noise, F18—Rotated hybrid Ackley et al., F19—Hybrid F18 with narrow basin optimum, F20—Hybrid F18 with optimum on bound, F21—Rotated Rosenbrock plus Rastrigin, Weierstrass and Griewank, F22—Same composition of F21 with different condition numbers, F23—Non continuous F22, F24—Weierstrass plus rotated Scaffer plus Ackley plus Rastrigin plus Griewank.

In the initial experiments we compare the performance of the Scatter Tabu Search variants that result by experimentally varying the key search parameters of the method. Then, we compare the performance of the most effective Scatter Tabu Search design, STS, against previous algorithms reported with the instances in the LM data set. Specifically we target C-GRASP (Hirsch et al. 2007), DTS (Hedar and Fukushima 2006), SS (Laguna and Martí 2005), MCS (Huyer and Neumaier 1999) and AMBH (Addis and Leyffer 2006). In the final experiment we compare STS with the methods reported on the CEC instances (CEC 2005 competition). Specifically we consider BLX-GL50 (García-Martínez and Lozano 2005), BLX_MA (Molina et al. 2005), CoEVO (Posik 2005), DE (Rönkkönen et al. 2005), DMS-L-PSO (Liang and Suganthan 2005), EDA (Yuan and Gallagher 2005), G-CMA-ES (Auger and Hansen 2005a), k_PCX (Sinha et al. 2005), L-CMA-ES (Auger and Hansen 2005b), L-SaDE (Qin and Suganthan 2005) and SPC-PNX (Ballester et al. 2005). Table 1 shows the names of the 64 test problems with known optimum objective function values that we use in our experiments.

In our preliminary experiments we apply the SS design described in Sect. 2. We set the maximum number of solution evaluations *MaxEval* to 10,000 and we employ the following 9 functions in the LM set: Branin, Beale, Rosenbrock (2), Shekel(5), Powersum (8,18,44,114), Rastrigin(10), Rastrigin(20), Powell(24) and Ackley(30). We have implemented the SS algorithm in C and all the experiments were conducted on a Pentium 4 computer at 3 GHz with 3 GB of RAM.

We have identified the following 12 search parameters in our method: b_1, b_2, Improvement strategy (All/Selective), *sr*, *dthresh*, *dgrid*, SS design (standard/distance based), *h* (width), Improvement method (TS, Simplex, Tabu Simplex), *ts*, *tenure* and *pt*. Although a full factorial design would provide us complete information about the effect of each parameter and the interdependencies among them, it requires a huge amount of work. Moreover, according to our previous experience on this methodology and following the recommendations in Laguna and Martí (2003) we limit our preliminary experimentation to test some selected values on groups of key search parameters (testing the interdependencies within each group). We therefore consider the following six experiments.

The objective of the first experiment is to test the selective application of the improvement method and to study the sizes b_1 (quality) and b_2 (diversity) in the Reference Set. In order to isolate the contribution of these three parameters, we configure the Scatter Search with a standard design. Specifically, we set $sr = 4$, as suggested in Laguna and Martí (2003),

Table 2 RefSet size and selective improvement method

b	(b_1, b_2)	Improve all		Selective improvement	
		Avg. GAP	#Optima	Avg. GAP	#Optima
8	(2, 6)	0.401	5	0.050	5
	(4, 4)	0.367	5	0.075	6
	(6, 2)	0.436	5	0.050	6
10	(2, 8)	0.251	5	0.056	6
	(5, 5)	0.367	5	0.062	6
	(8, 2)	0.264	5	0.053	5
20	(4, 16)	0.281	5	0.276	5
	(10, 10)	0.267	5	0.270	5
	(16, 4)	0.267	5	0.264	5

$dthresh = dgrid = 0$ (i.e., it does not use distance information), $h = MinRange/50$ and with line searches as the improvement strategy.

In the standard SS designs (Laguna and Martí 2003) every solution generated by the diversification generation method or by the combination method is subjected to the improvement method. Since the execution of the improvement method is computationally expensive, applying it to every solution may prevent the search from visiting additional solutions during the allotted search time. Therefore, in our current design (summarized in Sect. 2) we selectively apply the improvement method to a subset of the solutions that are visited during the search. The column labeled with *Improve All* reports the results of the standard SS design in which all the solutions are improved, while the column *Selective Improvement* reports the results of the design described in Sect. 2. We also test the size b of the Reference Set and the sizes b_1 and b_2 of the component subsets devoted respectively to high quality solutions and to diverse solutions.

Table 2 summarizes the results of this experiment, in which we report the average gap, Avg. GAP, and the number of optimal solutions, #Optima, found with each variant across the 9 problems considered. We define the optimality gap as:

$$GAP = \left| f(x) - f(x^*) \right|$$

where x is a heuristic solution and x^* is the optimal solution. We then say that a heuristic solution x is effectively optimal if:

$$GAP \le \begin{cases} \varepsilon, & f(x^*) = 0, \\ \varepsilon * |f(x^*)|, & f(x^*) \neq 0 \end{cases}$$

where we set $\varepsilon = 0.001$ as in the work of Laguna and Martí (2005).

Regarding the average GAP, the results in Table 2 reveal that the best average performance when applying the improvement method to all solutions is achieved when b is set to $(b_1, b_2) = (2, 8)$. On the other hand, the best average deviation for the selective procedure is achieved when b is set to (2, 6). When considering average deviation and number of best solutions, the selective application of the improvement method outperforms the standard design reported in the “Improve All”. We applied the Friedman test for paired samples to the solutions obtained by the different variants tested in our preliminary experiments. This test

Table 3 Two alternative SS designs

Instance	Standard SS	Distance based SS
Branin	0.001	0.001
Beale	0.000	0.000
Rosenbrock(2)	0.000	0.000
Shekel(5)	0.000	0.000
Powersum(8,18,44,114)	0.028	0.005
Rastrigin(10)	0.000	0.000
Rastrigin(20)	0.000	0.000
Powell(24)	0.008	0.005
Ackley(30)	10.537	0.510

computes, for each instance, the rank-value of each method according to solution quality, and calculates the average rank values of each method across all the instances solved. If the averages are very different, the associated p-value or significance will be small. The resulting significance level of 0.000 obtained in this first experiment clearly indicates that there are statistically significant differences between the two strategies tested (*Improve all* versus *Selective improvement*).

As far as we know the number of sub-ranges sr associated with the diversification generator method has been arbitrarily set to 4 in previous SS implementations. In the second experiment we study the variation of this search parameter. Specifically we target $sr = 2$, 4, 10, 20. The average gap value obtained with these four sr values is 0.130, 0.050, 0.100 and 0.051 respectively. These results are in line with the value employed in previous SS implementations and confirm that four sub-ranges provides good solutions; thus we set $sr = 4$ in the following experiments.

In the third experiment we set the distance threshold parameter *dthresh* as a function of the distance grid *dgrid*. We test the four values $dgrid/2$, $dgrid/3$, $dgrid/5$, and $dgrid/10$, obtaining that $dthresh = dgrid/3$ provides the best results.

In most SS designs, diversity in the Reference Set is achieved with a one-by-one selection mechanism. In Sect. 2 we have proposed to use the MMDP model to obtain a larger level of diversity in the initial *RefSet*. On the other hand, in customary SS designs, new trial solutions obtained from combination operations become part of the *RefSet* according to their quality. As previously indicated in Sect. 2, we filter evaluations also according to diversity criteria, allowing a solution to enter the *RefSet* only when its distance to this set is larger than *dthresh* (allowing an exception for the best solution found so far). In the fourth preliminary experiment we compare both designs, the customary approach and the approach proposed in Sect. 2, in order to measure the contribution of these two distance based mechanisms. Table 3 shows the GAP of the best solution found with each method for the 9 instances considered.

As Table 3 shows, the distance based SS is uniformly as good as, or better than, the customary SS approach. Moreover, the significance level of 0.046 obtained with the Friedman test in this experiment, indicates that there are statistically significant differences between both designs.

In the fifth preliminary experiment we test different values of the parameter h that defines the width of the uniform grid in which we discretized the search space. For any variable x_i, let l_i and u_i be its lower and upper bounds respectively. Let *MinRange* the minimum range

Table 4 Preliminary experiment to determine h

h	Avg. GAP	#Optima
$MinRange/50$	0.035	5
$MinRange/100$	0.004	7
$MinRange/150$	0.115	6
$MinRange/200$	0.074	5

Table 5 Scatter Search variants

Method	Avg. GAP	#Optima
SS	0.0291	7
SS + TS	0.0035	7
SS + Sx	0.0014	8
SS + TSx	0.0011	8
STS	0.0001	9

across all the variables according to the following formula:

$$MinRange = \min_{i=1,\ldots,n} u_i - l_i.$$

In this experiment we test four values for the width grid h computed as a fraction of the *MinRange*. Table 4 reports the average gap, Avg. GAP, and the number of optimal solutions, #Optima, found with our SS solution method for alternative h-values across the 9 problems considered in the preliminary experimentation.

Results in Table 4 show that the best results, with respect to both average GAP and number of optima, are obtained with the value of h set to *MinRange*/100 (although other variants present similar values). Therefore, we adopted this value for the rest of our experimentation although the significance level of 0.675 obtained with the Friedman test indicates that there are no statistically significant differences among the tested variants.

In our final preliminary experiment we measure the relative contribution of the different improvement methods of the SS algorithm described in Sect. 3. We consider five alternatives based on the Adaptive Memory Programming framework represented by Tabu Search and Scatter Search:

SS:	The SS method described on Sect. 2
SS + TS:	SS with Tabu Line Search (Sect. 3.1) as improvement method
SS + Sx:	SS with the original simplex method (Sect. 3.2) as improvement method
SS + TSx:	SS with TS simplex method (memory-based) as improvement method
STS:	SS + TS coupled with a post-processing with the TS simplex method

After initial experimentation, the values of *ts*, *tenure* and *pt* are set to $\lceil n/2 \rceil$, $\lfloor n/2 \rfloor$ and $15*h$ respectively in the methods above. Table 5 shows the average gap value and the number of optima of each AMP variant. As in the previous experiments, we run the methods until they reach the limit of 10,000 solution evaluations.

Table 5 clearly shows that the five variants considered are able to provide high quality results for this problem, since the average gap values are, in all the cases, below 0.1%. Moreover, comparing SS with SS + TS we can see that significant marginal improvement is achieved by replacing the simpler form of the line-search with the tabu line-search in the

Table 6 Comparison with best known methods over the 40 LM test problems

Method	Avg. GAP	#Optima
C-GRASP	2.382	28
DTS	1.29	32
prevSS	3.460	30
STS	0.028	33

Scatter Search algorithm. Further, comparing SS + Sx with SS + TSx, we see the advantage of including a memory structure to modify the improvement method (the Nelder-Mead simplex method in this case), so that successive applications of the method are restricted to operate only with solutions relatively far from those already submitted to the improvement method. Finally, the STS method, which couples the SS + TS with the TS modified Nelder-Mead simplex procedure outperforms the other AMP variants. Thus, in sum, the combination of two different improvement methods provides the best results, producing a variant that is able to obtain optimal solutions to all of the 9 instances tested. Finally, the significance level of 0.010 obtained with the Friedman test in this experiment, indicates that there are statistically significant differences among the five SS variants tested.

Having determined the values of the key search parameters for our algorithm in the first set of experiments, we perform the second and third sets of experiments to compare the best variant, STS, with the best methods identified in previous studies. In the second set of experiments we consider the three methods reported on the LM instances, while in the third set we consider the eleven reported on the CEC instances. We employ in each experiment not only the instances but also the conditions and evaluation criteria found in the respective previous papers.

Table 6 reports the results found with C-GRASP (Hirsch et al. 2007),[1] Direct Tabu Search, DTS (Hedar and Fukushima 2004) and a previous implementation of Scatter Search, prevSS (Laguna and Martí 2005). Each method is executed a single time on each instance and we report the average GAP across the 40 instances described in Table 1 as well as the number of optima found from among the 40 instances. As with previous studies, we limit all the methods to performing at most 50,000 evaluations.

The results in Table 6 clearly show that our Adaptive Memory STS framework is able to outperform these previous solution methods. Moreover, since it combines SS with TS and improves upon previous pure SS (prevSS) and TS (DTS) designs, our findings illustrate the utility of hybrid memory based designs in this context. To complement the information in Table 5, we show the evolution of the methods' performance over time. Figure 2 shows the typical search profile for the methods compared, depicting the average GAP value from 5,000 to 50,000 evaluations over the 40 instances considered.

We now test the robustness of our STS algorithm and compare it with two other previous methods: MCS (Huyer and Neumaier 1999) and AMBH (Addis and Leyffer 2006). We do not include these two methods in Table 6 because we do not have their results on the 40 test problems reported in that table. Tables 7 and 8 include MCS and AMBH respectively with the results and instances reported in their respective source papers. We study in these experiments the behavior of the algorithms when they are executed several times. Table 7 shows the results of MCS, STS, DTS and C-GRASP on the eight instances for which we have results of the previous methods run for several executions. Specifically, MCS and STS

[1] We thank the authors for sharing their codes with us. (Some of these codes are freely available on the net.)

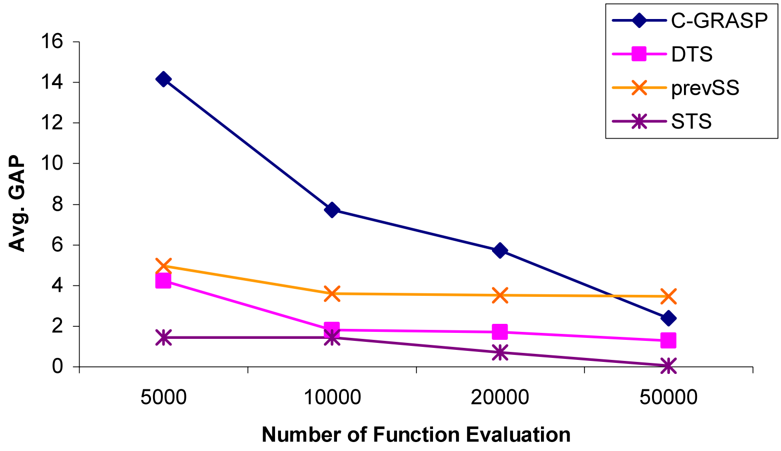

Fig. 2 Average optimal GAP value

Table 7 Average function evaluations and rate of success

Functions	MCS	STS	DTS	C-GRASP
Branin	41/100	1248/100	212/100	10090/100
Goldstein and Price	81/100	809/100	230/100	53/100
Shubert	69/100	1245/100	274/92	18608/100
Hartmann(3,4)	79/100	298/100	438/100	1719/100
Shekel(5)	83/100	9524/100	819/75	9274/100
Shekel(7)	129/100	3818/100	812/75	11766/100
Shekel(10)	103/100	3917/100	828/52	17612/100
Hartmann(6,4)	111/96	1263/100	1787/83	29894/100

are run 25 times while DTS and C-GRASP are run 100 times. Following the statistics used in Huyer and Neumaier (1999), we report for each function in Table 7 the average number of function evaluations needed for the convergence and the success rate (percentage of runs) of reaching global minima.

Table 7 shows that MCS outperforms the other three methods in terms of average function evaluations. Specifically, MCS needs 87 iterations (evaluations) on average over these 8 instances to converge to the global optimum while STS, DTS and C-GRASP need 2765, 675 and 12377 respectively. On the other hand, STS and C-GRASP obtain the global optimum in all the runs (100%) in the 8 instances while MCS and DTS obtain the global optimum in all the runs in 7 and 3 instances respectively.

Table 8 shows the results of the AMBH and our STS method on the eight instances reported in Addis and Leyffer (2006). The results of AMBH are taken from Addis and Leyffer (2006) where the method is replicated 1000 times, reporting the percentage of success and average local searches.

Table 8 shows that our STS method presents a 100% rate of success in 5 problems (i.e. in all the runs it obtains the global optimum). However, it is not able to obtain the optimum in any run in the other 3 problems of this experiment (Levy(50), Ackley(20) and Ackley(50)), exhibiting a 0.0% average rate of success over these three instances. On the other hand,

Table 8 Rate of success and average local searches

Functions	AMBH	STS
Rastrigin(20)	15.2/7711	100.0/2233
Rastrigin(50)	0.0/–	100.0/5583
Levy(20)	100.0/20	100.0/238
Levy(50)	55.8/45	0.0/–
Ackley(20)	100.0/346	0.0/–
Ackley(50)	69.5/538	0.0/–
Schwefel(5)	0.2/3500	100.0/2207
Schwefel(10)	0.1/44000	100.0/2309

AMBH presents a 100% rate of success in 2 problems and shows a 23.46% average rate of success over the other 6 instances in this experiment (with a value lower than 0.3% in three instances). Both methods present similar average number of local searches in their executions.

In our third (and concluding) set of experiments we compare the 11 methods reported on the CEC 2005 competition with our STS algorithm, first with $n = 10$ and then with $n = 30$. Following the guidelines in Suganthan et al. (2005) these 12 methods are run for 25 independent times on each instance. We then record the best value (minimum), the worst (maximum) and the average value over the 25 runs for each instance. Tables 12 to 14 show the optimality gap (minimum, maximum and average) of each method on each instance. For the sake of simplicity, Tables 9 and 10 only reports the average of the minimum (Min.) and average (Avg.) optimality gap across the 12 instances. We consider three different time horizons given by 1,000, 10,000 and 100,000 function evaluations as reported in Hansen (2006). We do not report the number of optima, as we did in previous experiments, since none of the methods considered is able to match any of them (which is why these problems are called ”Never solved instances”). As expected, the average gap values shown in Tables 9 and 10 are considerably larger than those reported in previous tables since these instances are extremely hard to solve (and no method has solved any of them to optimality).

The interpretation as to the quality of the outcomes obtained by a method compared to those obtained by the other methods depends strongly on how these outcomes are measured. If one is looking for a method that will succeed in finding a solution better than found by all competing methods, when applied over a series of trials or one is looking for a method whose average performance over the series of trials is better than that of competitors. Thus, the value of the method depends on whether one is interested in the best or in the “best average” (Johnson 2001). We elaborate these findings as follows.

Regarding the average of the minimum values (Min. columns) in Table 9, over a short term horizon STS produces high quality solutions, but not the best ones, compared with the other methods. Specifically, within the first 1,000 evaluations STS ranks 4th, within the 10,000 evaluations it has moved up to sixth place, and over the long term horizon (100,000 function evaluations) STS produces solutions with an average optimality gap of 198.3, which is superior to the gap obtained by all of the competing methods (whose gaps range from 202.7 to 306.2). If we consider relative gaps ($GAP/|f(x*)|$) instead of the absolute gaps shown in Table 9, STS still ranks first within 100,000 evaluations, with a value of 7.96 (while the other methods range from 7.98 to 16.36).

If we consider now the average of the average gap values (Avg. Columns) across the 12 instances, STS ranks 7th within the first 1,000 evaluations and 8th within the 10,000 and 100,000 evaluations. On the other hand, with respect to relative gaps, STS ranks in the

Table 9 Comparison over the 12 "Never solved" CEC test problems with $n = 10$

Method	1,000		10,000		100,000	
	Min.	Avg.	Min.	Avg.	Min.	Avg.
STS	616.1	759.4	348.9	576.6	198.3	413.4
G-CMA-ES	269.7	542.0	260.0	419.4	256.0	265.3
EDA	669.9	1059.1	287.1	335.1	269.4	300.6
BLX-MA	456.7	711.1	315.5	445.1	306.2	430.1
SPC-PNX	621.7	750.3	279.6	391.0	206.0	309.9
BLX-GL50	676.0	716.3	272.8	341.0	257.2	319.0
L-CMA-ES	289.0	825.7	225.9	655.8	202.7	411.1
DE	715.4	914.1	396.7	492.4	228.8	272.0
K-PCX	671.0	968.5	488.0	564.4	257.4	475.6
CoEVO	672.6	799.0	437.5	623.5	268.3	465.4
L-SaDE	636.0	729.2	300.2	438.6	205.6	369.9
DMS-L-PSO	651.7	734.0	356.9	477.0	244.4	392.3

same 8th place (with an average gap value of 19.24, while the other methods rank from 8.23 to 22.65). This different behavior of our STS method (comparing averages with minimum values) can be partially explained by the fact that the scatter search methodology is oriented to perform a diversified (scattered) exploration of the solution space and therefore some runs obtain high quality solutions and others low quality results but diverse solutions.

Table 10 shows the average results (Min. and Avg. as in Table 9) of the 10 methods when tackling the 12 "Never solved" CEC test problems with $n = 30$ (we did not find results for L-SaDE and DMS-L-PSO on these instances). As in the previous experiment we refer the reader to the detailed tables in the Appendix and we only represent here summarized information. Regarding the average of the minimum values (Min. columns), the results in Table 10 confirm the pattern previously indicated in Table 9. Our STS algorithm achieves high quality solutions in the short term, ranking 4th. In the middle term, between 10,000 and 100,000 evaluations, STS occupies the fifth place out of the 10 methods considered. Finally, over the long term horizon, after 300,000 function evaluations, only one method stands (slightly) ahead of STS, the L-CMA-ES with an average gap value of 392.64, as compared to a value of 394.07 obtained by the STS algorithm. Relative gaps again produce similar ranks than the absolute gaps shown in the tables. Specifically within 300,000 evaluations L-CMA-ES ranks first (with a value of 20.70) and STS ranks second (with a value of 21.31).

If we consider now the average across the 12 instances of the average gap values (Avg. Columns) we see that STS performs better than in the lower dimensional problems reported in Table 9. Within the first 1,000 evaluations STS ranks second and in the middle term, between 10,000 and 100,000 evaluations, STS occupies the fifth place out of the 10 methods considered. Over the long term horizon (300,000 function evaluations) it ranks third. As in the previous results, relative gaps (ranging in this experiment from 21.78 to 24.27) provide similar ranks than absolute gaps. In particular STS ranks third within 300,000 evaluations.

We applied the Friedman test for paired samples to the data used to generate Tables 9 to 14. The resulting p-value of 0.000 obtained in this experiment clearly indicates that there are statistically significant differences among the twelve methods tested (we are using the typical significance level of $\alpha = 0.05$ as the threshold between rejecting or not the null hypothesis). A typical post-test analysis consists of ranking the methods under comparison

Table 10 Comparison over the 12 ”Never solved” CEC test problems with $n = 30$

Method	1,000		10,000		100,000		300,000	
	Min.	Avg.	Min.	Avg.	Min.	Avg.	Min.	Avg.
STS	829.3	957.0	614.9	747.3	431.3	540.3	394.1	460.3
G-CMA-ES	570.3	658.4	414.4	526.8	405.7	493.0	402.1	485.1
EDA	39742	63491	11951.1	26418.8	653.6	934.7	454.6	503.5
BLX-MA	792.9	1198.7	443.9	502.4	410.7	457.2	407.2	453.9
SPC-PNX	29793.4	74050.0	637.6	850.1	414.8	430.0	410.6	418.8
BLX-GL50	8545.4	20008.7	474.8	545.9	433.0	507.5	408.7	484.2
L-CMA-ES	790.8	1009.8	447.6	722.6	404.6	617.0	392.6	595.6
DE	3473.3	14461.1	726.0	781.8	558.7	592.0	466.9	487.4
K-PCX	27749.8	108623.0	27719.7	108602.9	866.1	2257.2	419.3	521.9
CoEVO	908.5	1025.8	7496	822.0	625.3	734.5	549.2	652.0

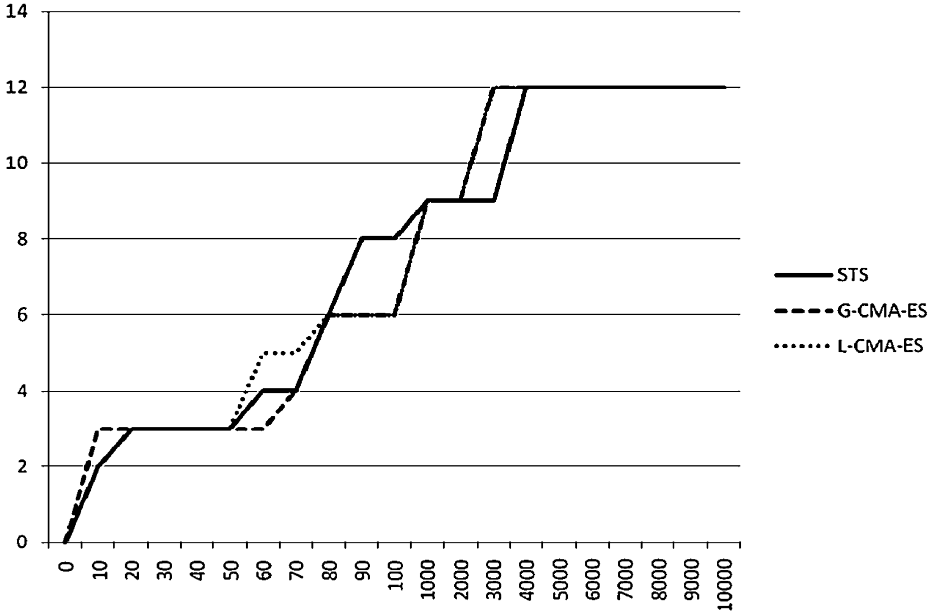

Fig. 3 Relative errors profile over the 12 CEC problems with $n = 10$

according to the average rank values computed with this test. According to this, we obtain that the L-CMA-ES method is the best overall with an average rank of 3.79. Then we obtain the following group of 6 methods with very similar rank-values: G-CMA-ES (4.23), STS (4.69), BLX-GL50 (4.72), BLX-MA (4.89), K-PCX (4.96) and SPC-PNX (5.06) and finally a group of three methods with larger rank values (as compared with the previous methods): DE (7.06), CoEVO (7.68) and EDA (7.90).

To complement the information above we can create a diagram in which the x-axis has relative errors (in percentage) and the y-axis the number of test functions within that percentage error. Figure 3 shows this diagram for the best three methods according to the Friedman

Table 11 Optimality gap values for f8, f13, f14, f16, f17, f18 and $n = 10$

Eval.	Method	f8			f13			f14			f16			f17			f18		
		min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
1000	STS	2.03E+1	2.09E+1	2.07E+1	1.10E+0	5.20E+0	2.45E+0	3.48E+0	4.51E+0	4.04E+0	1.95E+2	6.25E+2	3.10E+2	1.88E+2	7.38E+2	4.01E+2	9.37E+2	1.25E+3	1.13E+3
	G-CMA-ES	2.05E+1	2.09E+1	2.08E+1	2.26E+0	5.28E+0	3.84E+0	4.07E+0	4.52E+0	4.28E+0	1.12E+2	4.05E+2	2.19E+2	1.43E+2	1.01E+3	2.74E+2	3.04E+2	1.06E+3	8.43E+2
	EDA	2.04E+1	2.09E+1	2.07E+1	5.84E+1	1.24E+4	3.76E+3	3.90E+0	4.52E+0	4.35E+0	2.81E+2	4.42E+2	3.62E+2	3.49E+2	4.68E+2	4.07E+2	9.27E+2	1.21E+3	1.10E+3
	BLX-MA	2.01E+1	2.06E+1	2.05E+1	6.80E−1	1.33E+1	6.82E+0	3.92E+0	4.57E+0	4.33E+0	1.93E+2	4.20E+2	3.03E+2	2.66E+2	6.14E+2	4.24E+2	8.47E+2	1.15E+3	1.05E+3
	SPC-PNX	2.09E+1	2.13E+1	2.12E+1	8.93E+0	9.41E+1	2.80E+1	3.95E+0	4.49E+0	4.27E+0	2.86E+2	4.03E+2	3.33E+2	3.11E+2	4.37E+2	3.62E+2	1.01E+3	1.17E+3	1.11E+3
	BLX-GL50	2.03E+1	2.09E+1	2.07E+1	5.66E+1	3.43E+3	6.47E+2	3.73E+0	4.48E+0	4.29E+0	2.85E+2	3.99E+2	3.40E+2	2.91E+2	4.72E+2	3.92E+2	1.02E+3	1.16E+3	1.18E+2
	L-CMA-ES	2.04E+1	2.09E+1	2.07E+1	2.61E+0	5.92E+0	3.87E+0	4.11E+0	5.00E+0	4.73E+0	1.68E+2	2.27E+3	4.98E+2	2.72E+2	2.47E+3	1.46E+3	4.07E+2	2.14E+3	9.92E+2
	DE	2.04E+1	2.10E+1	2.08E+1	1.07E+2	4.62E+3	1.62E+3	3.99E+0	4.51E+0	4.34E+0	3.36E+2	5.03E+2	4.08E+2	2.91E+2	4.92E+2	3.86E+2	9.80E+2	1.24E+3	1.16E+3
	K-PCX	2.04E+1	2.09E+1	2.07E+1	8.93E+1	9.84E+3	2.38E+3	4.13E+0	4.63E+0	4.39E+0	3.78E+2	5.59E+2	4.47E+2	4.07E+2	6.67E+2	5.27E+2	1.09E+3	1.30E+3	1.22E+3
	CoEVO	2.03E+1	2.09E+1	2.06E+1	6.30E+0	1.41E+1	1.00E+1	3.66E+0	4.58E+0	4.29E+0	2.59E+2	5.01E+2	4.08E+2	3.67E+2	5.90E+2	4.67E+2	1.03E+3	1.33E+3	1.17E+3
	L-SaDE	2.04E+1	2.10E+1	2.07E+1	4.48E+0	9.38E+0	6.92E+0	3.77E+0	4.44E+0	4.26E+0	2.35E+2	3.67E+2	3.07E+2	3.07E+2	4.67E+2	3.66E+2	1.03E+3	1.18E+3	1.11E+3
	DMS-L-PSO	2.05E+1	2.09E+1	2.07E+1	3.95E+0	6.75E+0	5.59E+0	3.22E+0	4.41E+0	4.12E+0	2.58E+2	3.62E+2	2.97E+2	2.73E+2	4.32E+2	3.41E+2	1.06E+3	1.17E+3	1.11E+3
10000	STS	2.02E+1	2.05E+1	2.03E+1	4.63E−1	1.19E+0	7.40E−1	2.68E+0	4.07E+0	3.54E+0	1.23E+2	2.39E+2	1.71E+2	1.47E+2	2.53E+2	1.89E+2	5.17E+2	1.08E+3	9.53E+2
	G-CMA-ES	2.03E+1	2.07E+1	2.05E+1	4.90E−1	1.84E+0	9.70E−1	2.95E+0	4.51E+0	3.91E+0	7.95E+1	1.69E+2	1.11E+2	1.17E+2	9.99E+2	1.95E+2	3.00E+2	1.03E+3	6.02E+2
	EDA	2.03E+1	2.07E+1	2.05E+1	2.43E+0	5.45E+0	4.40E+0	3.90E+0	4.28E+0	4.08E+0	1.32E+2	2.38E+2	2.05E+2	1.95E+2	2.69E+2	2.36E+2	3.61E+2	8.03E+2	5.16E+2
	BLX-MA	2.01E+1	2.04E+1	2.02E+1	6.80E−1	2.52E+0	1.43E+0	2.91E+0	4.03E+0	3.51E+0	1.03E+2	1.32E+2	1.13E+2	1.31E+2	1.90E+2	1.57E+2	8.00E+2	8.99E+2	8.07E+2
	SPC-PNX	2.09E+1	2.12E+1	2.11E+1	7.73E−1	2.83E+0	1.54E+0	3.56E+0	4.02E+0	3.83E+0	1.50E+2	2.05E+2	1.85E+2	1.62E+2	2.23E+2	2.01E+2	3.03E+2	9.30E+2	5.42E+2
	BLX-GL50	2.03E+1	2.07E+1	2.05E+1	1.54E+0	3.69E+0	2.95E+0	3.31E+0	4.13E+0	3.85E+0	1.46E+2	1.98E+2	1.73E+2	1.68E+2	2.22E+2	2.03E+2	3.01E+2	9.92E+2	4.30E+2
	L-CMA-ES	2.00E+1	2.07E+1	2.04E+1	5.10E−1	1.60E+0	8.80E−1	3.84E+0	4.98E+0	4.57E+0	9.39E+1	5.00E+2	1.49E+2	1.39E+2	2.31E+3	1.08E+3	3.00E+2	1.10E+3	8.40E+2
	DE	2.02E+1	2.07E+1	2.05E+1	6.52E+0	1.18E+1	9.05E+0	3.68E+0	4.18E+0	4.02E+0	1.83E+2	2.61E+2	2.30E+2	2.16E+2	2.67E+2	2.41E+2	6.44E+2	1.01E+3	8.23E+2
	K-PCX	2.00E+1	2.02E+1	2.00E+1	2.59E+0	4.66E+0	3.63E+0	3.67E+0	4.51E+0	4.25E+0	1.73E+2	2.22E+2	1.95E+2	1.72E+2	2.42E+2	2.14E+2	8.48E+2	1.06E+3	9.52E+2
	CoEVO	2.01E+1	2.06E+1	2.04E+1	1.13E+0	3.67E+0	2.22E+0	3.56E+0	4.29E+0	4.01E+0	1.53E+2	3.03E+2	2.34E+2	2.40E+2	4.08E+2	3.12E+2	8.38E+2	1.12E+3	1.02E+3
	L-SaDE	2.03E+1	2.07E+1	2.05E+1	9.80E−1	1.88E+0	1.43E+0	3.19E+0	4.10E+0	3.84E+0	1.42E+2	1.89E+2	1.68E+2	1.72E+2	2.42E+2	2.00E+2	5.62E+2	9.70E+2	8.17E+2
	DMS-L-PSO	2.01E+1	2.05E+1	2.04E+1	6.24E−1	1.92E+0	1.38E+0	2.69E+0	3.85E+0	3.38E+0	1.17E+2	1.70E+2	1.41E+2	1.39E+2	1.86E+2	1.61E+2	8.00E+2	1.01E+3	8.97E+2

Table 11 (*Continued*)

Eval.	Method	f8			f13			f14			f16			f17			f18		
		min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
100000	STS	2.01E+1	2.02E+1	2.02E+1	2.27E−1	5.73E−1	4.54E−1	2.10E+0	3.56E+0	2.87E+0	9.19E+1	1.15E+2	1.01E+2	1.04E+2	1.45E+2	1.16E+2	3.00E+2	9.85E+2	7.72E+2
	G-CMA-ES	2.00E+1	2.00E+1	2.00E+1	4.70E−1	1.05E+0	7.00E−1	2.08E+0	3.51E+0	3.01E+0	7.92E+1	9.68E+1	9.13E+1	9.48E+1	2.04E+2	1.23E+2	3.00E+2	7.18E+2	3.32E+2
	EDA	2.02E+1	2.04E+1	2.05E+1	1.59E+0	3.29E+0	2.61E+0	2.92E+0	3.99E+0	3.61E+0	1.32E+2	1.79E+2	1.64E+2	1.50E+2	2.01E+2	1.83E+2	3.00E+2	8.00E+2	4.20E+2
	BLX-MA	2.01E+1	2.04E+1	2.02E+1	3.80E−1	1.23E+0	7.70E−1	6.80E−1	3.20E+0	2.03E+0	8.96E+1	1.11E+2	1.02E+2	1.04E+2	1.56E+2	1.27E+2	8.00E+2	8.84E+2	8.03E+2
	SPC-PNX	2.08E+1	2.11E+1	2.10E+1	3.49E−1	1.32E+0	8.38E−1	1.39E+0	3.61E+0	3.05E+0	9.11E+1	1.40E+2	1.10E+2	9.89E+1	1.51E+2	1.19E+2	3.00E+2	9.01E+2	4.40E+2
	BLX-GL50	2.02E+1	2.05E+1	2.04E+1	3.70E−1	1.12E+0	7.50E−1	1.38E+0	3.03E+0	2.17E+0	7.23E+1	1.06E+2	9.35E+1	9.72E+1	1.21E+2	1.09E+2	3.00E+2	8.00E+2	4.20E+2
	L-CMA-ES	2.00E+1	2.00E+1	2.00E+1	1.90E−1	8.20E−1	4.90E−1	3.36E+0	4.43E+0	4.01E+0	6.14E+1	1.21E+2	1.05E+2	1.23E+2	1.33E+3	5.49E+2	3.00E+2	8.00E+2	4.97E+2
	DE	2.02E+1	2.05E+1	2.04E+1	4.58E−1	4.05E+0	1.81E+0	2.73E+0	3.79E+0	3.52E+0	1.52E+2	1.96E+2	1.73E+2	1.09E+2	2.09E+2	1.72E+2	3.00E+2	3.00E+2	3.00E+2
	K-PCX	2.00E+1	2.00E+1	2.00E+1	3.28E−1	1.07E+0	6.53E−1	1.53E+0	3.30E+0	2.35E+0	8.75E+1	1.13E+2	9.59E+1	8.82E+1	1.14E+2	9.73E+1	3.00E+2	9.51E+2	7.52E+2
	CoEVO	2.01E+1	2.04E+1	2.03E+1	4.69E−1	2.15E+0	1.14E+0	3.26E+0	4.14E+0	3.71E+0	1.23E+2	2.73E+2	1.77E+2	1.45E+2	2.97E+2	2.12E+2	8.00E+2	1.03E+3	9.02E+2
	L-SaDE	2.00E+1	2.00E+1	2.00E+1	1.20E−1	3.12E−1	2.20E−1	2.58E+0	3.34E+0	2.92E+0	8.63E+1	1.12E+2	1.01E+2	9.90E+1	1.36E+2	1.14E+2	3.00E+2	9.01E+2	7.19E+2
	DMS-L-PSO	2.00E+1	2.00E+1	2.00E+1	2.54E−1	4.73E−1	3.69E−1	1.48E+0	2.89E+0	2.36E+0	5.16E+1	1.05E+2	9.48E+1	9.99E+1	1.19E+2	1.10E+2	3.00E+2	9.35E+2	7.61E+2

Table 12 Optimality gap values for f19, f20, f21, f22, f23, f24 and $n = 10$

Method	f19			f20			f21			f22			f23			f24		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
1000																		
STS	1.07E+3	1.29E+3	1.14E+3	8.89E+2	1.31E+3	1.13E+3	1.00E+3	1.44E+3	1.30E+3	8.73E+2	1.33E+3	1.08E+3	1.26E+3	1.41E+3	1.33E+3	9.55E+2	1.39E+3	1.25E+3
G-CMA-ES	3.04E+2	1.06E+3	8.86E+2	3.01E+2	1.07E+3	8.50E+2	5.00E+2	1.27E+3	9.05E+2	7.86E+2	9.49E+2	8.29E+2	5.59E+2	1.28E+3	1.08E+3	2.00E+2	9.94E+2	5.89E+2
EDA	9.92E+2	1.19E+3	1.11E+3	9.31E+2	1.20E+3	1.11E+3	1.28E+3	1.42E+3	1.35E+3	8.54E+2	1.05E+3	9.94E+2	1.31E+3	1.41E+3	1.36E+3	1.04E+3	1.21E+3	1.13E+3
BLX-MA	8.12E+2	1.16E+3	1.02E+3	8.82E+2	1.14E+3	1.16E+3	5.11E+2	1.34E+3	1.18E+3	7.03E+2	1.10E+3	9.81E+2	1.03E+3	1.45E+3	1.32E+3	2.09E+2	1.44E+3	1.07E+3
SPC-PNX	1.05E+3	1.19E+3	1.13E+3	1.05E+3	1.20E+3	1.13E+3	7.93E+2	1.30E+3	1.20E+3	8.53E+2	1.21E+3	1.07E+3	1.27E+3	1.43E+3	1.37E+3	8.06E+2	1.35E+3	1.24E+3
BLX-GL50	1.03E+3	1.18E+3	1.11E+3	9.97E+2	1.16E+3	1.12E+3	1.24E+3	1.37E+3	1.33E+3	8.99E+2	1.07E+3	1.01E+3	1.25E+3	1.36E+3	1.34E+3	1.02E+3	1.27E+3	1.17E+3
L-CMA-ES	3.94E+2	2.16E+3	9.81E+2	3.22E+2	2.24E+3	9.88E+2	4.11E+2	2.37E+3	1.10E+3	7.13E+2	2.26E+3	1.11E+3	5.54E+2	2.17E+3	1.36E+3	2.00E+2	2.34E+3	1.39E+3
DE	1.10E+3	1.21E+3	1.16E+3	1.08E+3	1.25E+3	1.18E+3	1.23E+3	1.41E+3	1.35E+3	9.86E+2	1.17E+3	1.07E+3	1.28E+3	1.42E+3	1.36E+3	1.17E+3	1.35E+3	1.25E+3
K-PCX	1.11E+3	1.32E+3	1.22E+3	1.06E+3	1.34E+3	1.21E+3	1.22E+3	1.40E+3	1.34E+3	9.37E+2	1.22E+3	1.09E+3	1.25E+3	1.45E+3	1.40E+3	4.86E+2	9.95E+2	7.63E+2
CoEVO	1.09E+3	1.32E+3	1.18E+3	1.08E+3	1.25E+3	1.17E+3	1.09E+3	1.44E+3	1.34E+3	1.07E+3	1.36E+3	1.20E+3	1.18E+3	1.47E+3	1.34E+3	8.79E+2	1.42E+3	1.28E+3
L-SaDE	1.06E+3	1.15E+3	1.11E+3	1.02E+3	1.16E+3	1.11E+3	1.07E+3	1.34E+3	1.30E+3	9.04E+2	1.86E+3	9.91E+2	1.19E+3	1.35E+3	1.30E+3	7.78E+2	1.32E+3	1.13E+3
DMS-L-PSO	1.06E+3	1.17E+3	1.12E+3	1.09E+3	1.19E+3	1.13E+3	1.10E+3	1.31E+3	1.27E+3	8.78E+2	1.09E+3	9.84E+2	9.88E+2	1.35E+3	1.27E+3	1.08E+3	1.35E+3	1.25E+3
10000																		
STS	7.24E+2	1.10E+3	9.59E+2	6.47E+2	1.10E+3	9.53E+2	4.11E+2	1.25E+3	9.46E+2	7.80E+2	1.10E+3	9.01E+2	6.16E+2	1.32E+3	1.17E+3	2.00E+2	1.29E+3	6.52E+2
G-CMA-ES	3.00E+2	1.03E+3	6.75E+2	3.00E+2	9.90E+2	6.85E+2	5.00E+2	1.25E+3	7.05E+2	7.41E+2	8.98E+2	7.81E+2	5.59E+2	1.28E+3	9.49E+2	2.00E+2	9.72E+2	3.04E+2
EDA	3.52E+2	8.01E+2	4.96E+2	3.40E+2	8.02E+2	4.75E+2	5.01E+2	5.41E+2	5.10E+2	7.78E+2	8.05E+2	7.92E+2	5.60E+2	5.66E+2	5.60E+2	2.01E+2	2.05E+2	2.02E+2
BLX-MA	3.69E+2	9.23E+2	7.72E+2	8.00E+2	8.01E+2	8.00E+2	5.00E+2	1.16E+3	7.41E+2	3.00E+2	8.00E+2	7.21E+2	5.59E+2	1.19E+3	9.81E+2	2.00E+2	5.00E+2	2.24E+2
SPC-PNX	3.25E+2	8.05E+2	4.96E+2	3.25E+2	8.17E+2	5.36E+2	3.00E+2	1.17E+3	6.80E+2	8.12E+2	9.05E+2	8.35E+2	7.53E+2	1.19E+3	9.84E+2	2.01E+2	2.17E+2	2.05E+2
BLX-GL50	3.01E+2	1.03E+3	4.53E+2	3.01E+2	9.54E+2	4.50E+2	5.00E+2	1.25E+3	6.95E+2	7.71E+2	8.65E+2	7.96E+2	5.59E+2	1.27E+3	6.65E+2	2.00E+2	2.00E+2	2.00E+2
L-CMA-ES	3.00E+2	1.45E+3	8.16E+2	3.00E+2	1.17E+3	7.75E+2	3.00E+2	1.85E+3	8.32E+2	5.00E+2	2.08E+3	9.12E+2	5.54E+2	2.17E+3	1.22E+3	2.00E+2	2.30E+3	1.22E+3
DE	6.21E+2	1.02E+3	8.30E+2	6.18E+2	1.03E+3	8.58E+2	7.08E+2	1.02E+3	8.10E+2	8.12E+2	8.74E+2	8.38E+2	6.50E+2	9.25E+2	8.47E+2	2.78E+2	4.94E+2	3.98E+2
K-PCX	8.47E+2	1.08E+3	9.76E+2	8.60E+2	1.07E+3	9.74E+2	9.42E+2	1.14E+3	1.12E+3	5.50E+2	9.30E+2	7.43E+2	1.03E+3	1.20E+3	1.16E+3	4.08E+2	4.13E+2	4.11E+2
CoEVO	6.87E+2	1.09E+3	9.74E+2	6.74E+2	1.10E+3	9.83E+2	6.19E+2	1.24E+3	1.00E+3	7.89E+2	1.03E+3	8.74E+2	7.77E+2	1.28E+3	1.15E+3	4.48E+2	1.23E+3	9.03E+2
L-SaDE	5.43E+2	9.86E+2	8.32E+2	5.10E+2	9.75E+2	8.13E+2	4.77E+2	1.07E+3	6.89E+2	4.12E+2	8.00E+2	7.69E+2	5.59E+2	1.12E+3	7.49E+2	2.00E+2	2.00E+2	2.00E+2
DMS-L-PSO	7.39E+2	1.00E+3	8.80E+2	8.01E+2	1.02E+3	9.01E+2	5.07E+2	1.04E+3	8.51E+2	3.98E+2	8.00E+2	7.57E+2	5.59E+2	1.11E+3	8.61E+2	2.00E+2	5.06E+2	2.49E+2

Table 12 (*Continued*)

Method	f19			f20			f21			f22			f23			f24		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
100000																		
STS	3.00E+2	9.72E+2	7.30E+2	3.00E+2	9.74E+2	7.16E+2	2.01E+2	1.11E+3	6.56E+2	3.01E+2	8.77E+2	7.58E+2	5.59E+2	1.20E+3	8.64E+2	2.00E+2	5.01E+2	2.24E+2
G-CMA-ES	3.00E+2	7.69E+2	3.26E+2	3.00E+2	3.00E+2	3.00E+2	5.00E+2	5.00E+2	5.00E+2	7.16E+2	7.44E+2	7.29E+2	5.59E+2	5.59E+2	5.59E+2	2.00E+2	2.00E+2	2.00E+2
EDA	3.00E+2	8.00E+2	4.00E+2	3.00E+2	8.00E+2	3.80E+2	5.00E+2	5.00E+2	5.00E+2	7.66E+2	7.78E+2	7.74E+2	5.60E+2	5.60E+2	5.60E+2	2.00E+2	2.00E+2	2.00E+2
BLX-MA	3.00E+2	8.70E+2	7.63E+2	8.00E+2	8.00E+2	8.00E+2	5.00E+2	8.00E+2	7.22E+2	3.00E+2	8.00E+2	6.71E+2	5.59E+2	1.05E+3	9.27E+2	2.00E+2	5.00E+2	2.24E+2
SPC-PNX	3.00E+2	8.00E+2	3.80E+2	3.00E+2	8.00E+2	4.40E+2	3.00E+2	1.17E+3	6.80E+2	3.00E+2	7.77E+2	7.49E+2	5.59E+2	9.71E+2	5.76E+2	2.00E+2	2.00E+2	2.00E+2
BLX-GL50	3.00E+2	1.03E+3	4.49E+2	3.00E+2	9.50E+2	4.46E+2	5.00E+2	1.23E+3	6.89E+2	7.35E+2	8.28E+2	7.59E+2	5.59E+2	1.27E+3	6.39E+2	2.00E+2	2.00E+2	2.00E+2
L-CMA-ES	3.00E+2	9.76E+2	5.16E+2	3.00E+2	9.07E+2	4.42E+2	2.00E+2	8.00E+2	4.04E+2	5.00E+2	9.00E+2	7.40E+2	4.25E+2	1.26E+3	7.91E+2	2.00E+2	2.18E+3	8.65E+2
DE	3.00E+2	3.00E+2	3.00E+2	3.00E+2	3.00E+2	3.00E+2	5.00E+2	5.00E+2	5.00E+2	3.02E+2	7.85E+2	7.34E+2	5.59E+2	5.59E+2	5.59E+2	2.00E+2	2.00E+2	2.00E+2
K-PCX	3.00E+2	9.42E+2	7.51E+2	3.00E+2	9.59E+2	8.13E+2	5.00E+2	1.09E+3	1.05E+3	5.27E+2	8.69E+2	6.59E+2	5.59E+2	1.11E+3	1.06E+3	4.05E+2	4.07E+2	4.06E+2
CoEVO	5.00E+2	1.02E+3	8.45E+2	5.00E+2	1.04E+3	8.63E+2	2.01E+2	1.13E+3	6.35E+2	3.00E+2	8.46E+2	7.79E+2	4.25E+2	1.09E+3	8.35E+2	2.00E+2	1.09E+3	3.14E+2
L-SaDE	3.00E+2	9.31E+2	7.05E+2	3.00E+2	9.07E+2	7.13E+2	3.00E+2	8.00E+2	4.64E+2	3.00E+2	8.00E+2	7.35E+2	5.59E+2	9.71E+2	6.64E+2	2.00E+2	2.00E+2	2.00E+2
DMS-L-PSO	3.00E+2	9.33E+2	7.14E+2	8.00E+2	9.39E+2	8.22E+2	3.00E+2	8.00E+2	5.36E+2	3.00E+2	8.00E+2	6.92E+2	5.59E+2	9.71E+2	7.30E+2	2.00E+2	5.00E+2	2.24E+2

Table 13 Optimality gap values for f8, f13, f14, f16, f17, f18 and $n = 30$

Method	f8			f13			f14			f16			f17			f18		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
1000																		
STS	2.10E+1	2.13E+1	2.12E+1	2.42E+1	1.03E+2	4.66E+1	1.34E+1	1.46E+1	1.40E+1	5.60E+2	1.15E+3	8.36E+2	5.67E+2	1.34E+3	1.06E+3	1.17E+3	1.43E+3	1.28E+3
G-CMA-ES	2.12E+1	2.13E+1	2.12E+1	3.05E+1	4.98E+2	1.14E+2	1.37E+1	1.44E+1	1.42E+1	2.75E+2	6.01E+2	3.75E+2	3.10E+2	6.83E+2	4.79E+2	9.18E+2	9.89E+2	9.45E+2
EDA	2.11E+1	2.13E+1	2.12E+1	4.66E+5	1.14E+6	7.50E+5	1.37E+1	1.44E+1	1.42E+1	7.60E+2	1.07E+3	9.16E+2	8.63E+2	1.14E+3	9.98E+2	1.22E+3	1.37E+3	1.30E+3
BLX-MA	2.07E+1	2.11E+1	2.09E+1	4.09E+2	1.06E+4	3.95E+3	1.29E+1	1.44E+1	1.39E+1	5.23E+2	8.40E+2	6.43E+2	7.12E+2	1.23E+3	1.02E+3	9.60E+2	1.32E+3	1.09E+3
SPC-PNX	2.10E+1	2.13E+1	2.12E+1	3.47E+5	1.36E+6	8.77E+5	1.40E+1	1.44E+1	1.42E+1	5.87E+2		7.87E+2	6.89E+2	1.07E+3	8.61E+2	1.21E+3		1.29E+3
BLX-GL50	2.11E+1	2.13E+1	2.12E+1	9.28E+4	4.67E+5	2.29E+5	1.39E+1	1.44E+1	1.42E+1	5.34E+2	8.63E+2	7.06E+2	5.89E+2	9.88E+2	7.98E+2	1.12E+3	1.27E+3	1.20E+3
L-CMA-ES	2.11E+1	2.13E+1	2.12E+1	2.98E+1	1.19E+3	2.06E+2	1.41E+1	1.50E+1	1.47E+1	3.25E+2	1.56E+3	5.96E+2	1.51E+3	2.36E+3	1.94E+3	1.02E+3	1.74E+3	1.22E+3
DE	2.11E+1	2.13E+1	2.12E+1	3.12E+4	4.33E+5	1.62E+5	1.38E+1	1.44E+1	1.41E+1	7.17E+2	9.97E+2	8.33E+2	6.18E+2	1.07E+3	9.05E+2	1.21E+3	1.33E+3	1.28E+3
K-PCX	2.11E+1	2.13E+1	2.12E+1	3.21E+5	2.35E+6	1.29E+6	1.40E+1	1.46E+1	1.44E+1	9.82E+2	1.36E+3	1.20E+3	1.16E+3	1.55E+3	1.36E+3	1.33E+3	1.48E+3	1.42E+3
CoEVO	2.11E+1	2.13E+1	2.12E+1	9.68E+1	4.30E+2	2.58E+2	1.38E+1	1.44E+1	1.41E+1	7.58E+2	1.15E+3	9.75E+2	8.19E+2	1.46E+3	1.10E+3	1.24E+3	1.41E+3	1.33E+3
10000																		
STS	2.06E+1	2.11E+1	2.09E+1	2.04E+0	6.61E+0	3.85E+0	1.29E+1	1.39E+1	1.34E+1	2.08E+2	5.48E+2	3.93E+2	2.74E+2	7.48E+2	4.68E+2	9.37E+2	1.20E+3	1.05E+3
G-CMA-ES	2.10E+1	2.12E+1	2.11E+1	2.46E+0	5.62E+0	3.80E+0	1.34E+1	1.40E+1	1.38E+1	5.75E+1	5.00E+2	1.96E+2	6.88E+1	6.08E+2	3.00E+2	9.04E+2	9.14E+2	9.08E+2
EDA	2.10E+1	2.12E+1	2.11E+1	1.35E+5	5.24E+5	3.08E+5	1.34E+1	1.41E+1	1.39E+1	3.44E+2	4.50E+2	4.02E+2	4.17E+2	5.22E+2	4.66E+2	1.09E+3	1.16E+3	1.13E+3
BLX-MA	2.07E+1	2.10E+1	2.09E+1	9.97E+0	1.96E+1	1.51E+1	1.22E+1	1.38E+1	1.34E+1	1.42E+2	5.20E+2	3.56E+2	2.44E+2	6.09E+2	3.62E+2	8.29E+2	9.09E+2	8.92E+2
SPC-PNX	2.09E+1	2.11E+1	2.11E+1	2.11E+2	5.97E+3	2.17E+3	1.35E+1	1.40E+1	1.38E+1	2.56E+2		3.07E+2	2.68E+2	3.87E+2	3.24E+2	9.04E+2		1.07E+3
BLX-GL50	2.09E+1	2.12E+1	2.11E+1	3.06E+1	6.99E+1	4.95E+1	1.35E+1	1.40E+1	1.39E+1	2.34E+2	4.36E+2	2.69E+2	2.43E+2	4.76E+2	3.14E+2	9.10E+2	9.13E+2	9.11E+2
L-CMA-ES	2.09E+1	2.12E+1	2.11E+1	2.61E+0	5.02E+0	3.64E+0	1.36E+1	1.49E+1	1.43E+1	5.77E+1	8.02E+2	2.52E+2	4.33E+2	2.28E+3	1.71E+3	9.10E+2	1.49E+3	9.64E+2
DE	2.09E+1	2.12E+1	2.11E+1	3.20E+1	2.47E+2	1.02E+2	1.36E+1	1.42E+1	1.39E+1	3.82E+2	5.71E+2	4.48E+2	4.03E+2	5.77E+2	5.07E+2	1.05E+3	1.12E+3	1.09E+3
K-PCX	2.03E+1	2.12E+1	2.10E+1	3.21E+5	2.35E+6	1.29E+6	1.40E+1	1.45E+1	1.43E+1	9.82E+2	1.36E+3	1.19E+3	1.16E+3	1.55E+3	1.36E+3	1.26E+3	1.48E+3	1.37E+3
CoEVO	2.10E+1	2.11E+1	2.11E+1	2.73E+1	3.86E+1	3.21E+1	1.34E+1	1.41E+1	1.38E+1	4.57E+2	6.47E+2	5.61E+2	4.30E+2	8.48E+2	6.96E+2	1.09E+3	1.20E+3	1.16E+3

Table 13 (*Continued*)

Method	f8			f13			f14			f16			f17			f18		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
100000																		
STS	2.04E+1	2.09E+1	2.06E+1	1.37E+0	2.58E+0	1.95E+0	1.13E+1	1.32E+1	1.26E+1	9.09E+1	4.02E+2	1.97E+2	1.32E+2	4.47E+2	2.47E+2	8.85E+2	9.74E+2	9.21E+2
G-CMA-ES	2.00E+1	2.11E+1	2.07E+1	2.43E+0	3.67E+0	2.89E+0	1.27E+1	1.40E+1	1.35E+1	2.69E+1	1.28E+2	5.34E+1	6.67E+1	5.95E+2	2.92E+2	9.03E+2	9.07E+2	9.04E+2
EDA	2.08E+1	2.11E+1	2.10E+1	1.84E+3	9.93E+3	4.52E+3	1.34E+1	1.39E+1	1.37E+1	2.20E+2	3.11E+2	2.77E+2	2.81E+2	3.49E+2	3.13E+2	9.67E+2	1.01E+3	9.80E+2
BLX-MA	2.06E+1	2.09E+1	2.08E+1	2.78E+0	1.28E+1	8.66E+0	1.18E+1	1.34E+1	1.27E+1	1.08E+2	5.00E+2	3.35E+2	1.16E+2	5.38E+2	2.97E+2	8.00E+2	8.89E+2	8.78E+2
SPC-PNX	2.08E+1	2.11E+1	2.10E+1	2.25E+0	6.46E+0	3.72E+0	1.32E+1	1.38E+1	1.35E+1	5.18E+1		1.33E+2	4.83E+1	3.24E+2	1.24E+2	9.06E+2		9.08E+2
BLX-GL50	2.08E+1	2.11E+1	2.10E+1	1.25E+1	1.58E+1	1.44E+1	1.30E+1	1.37E+1	1.34E+1	1.70E+2	4.00E+2	2.18E+2	1.84E+2	4.35E+2	2.59E+2	9.03E+2	9.06E+2	9.04E+2
L-CMA-ES	2.00E+1	2.11E+1	2.03E+1	2.09E+0	3.82E+0	2.84E+0	1.35E+1	1.47E+1	1.42E+1	5.05E+1	1.68E+2	7.41E+1	2.68E+2	2.17E+3	1.13E+3	8.00E+2	9.22E+2	9.07E+2
DE	2.08E+1	2.11E+1	2.10E+1	2.31E+0	1.39E+1	4.55E+0	1.30E+1	1.37E+1	1.36E+1	2.82E+2	3.32E+2	3.11E+2	3.09E+2	3.81E+2	3.50E+2	9.31E+2	9.52E+2	9.39E+2
K-PCX	2.00E+1	2.00E+1	2.00E+1	2.56E+3	4.97E+4	1.74E+4	1.40E+1	1.44E+1	1.42E+1	4.18E+2	1.32E+3	9.45E+2	1.08E+3	1.52E+3	1.31E+3	9.97E+2	1.14E+3	1.06E+3
CoEVO	2.08E+1	2.11E+1	2.10E+1	6.73E+0	1.82E+1	1.10E+1	1.28E+1	1.39E+1	1.36E+1	2.85E+2	5.24E+2	4.06E+2	3.74E+2	6.63E+2	5.31E+2	1.05E+3	1.19E+3	1.11E+3
300000																		
STS	2.04E+1	2.06E+1	2.05E+1	1.37E+0	2.01E+0	1.72E+0	1.04E+1	1.24E+1	1.18E+1	6.91E+1	3.69E+2	1.21E+2	6.62E+1	4.36E+2	1.69E+2	8.00E+2	9.17E+2	8.91E+2
G-CMA-ES	2.00E+1	2.00E+1	2.10E+1	1.10E+0	3.20E+0	2.49E+0	1.18E+1	1.37E+1	1.29E+1	1.53E+1	1.08E+2	3.50E+1	6.66E+1	5.95E+2	2.91E+2	9.03E+2	9.04E+2	9.04E+2
EDA	2.08E+1	2.10E+1	2.09E+1	3.82E+1	1.29E+2	7.36E+1	1.33E+1	1.38E+1	1.36E+1	2.20E+2	2.75E+2	2.56E+2	2.69E+2	3.07E+2	2.88E+2	9.21E+2	9.43E+2	9.28E+2
BLX-MA	2.01E+1	2.09E+1	2.07E+1	1.33E+0	1.03E+1	3.96E+0	1.18E+1	1.31E+1	1.26E+1	7.14E+1	5.00E+2	3.26E+2	1.16E+2	5.03E+2	2.79E+2	8.00E+2	8.89E+2	8.78E+2
SPC-PNX	2.08E+1	2.10E+1	2.09E+1	1.88E+0	6.42E+0	3.59E+0	1.23E+1	1.35E+1	1.31E+1	4.08E+1		7.40E+1	4.43E+1	1.60E+2	8.54E+1	9.03E+2		9.05E+2
BLX-GL50	2.08E+1	2.10E+1	2.09E+1	1.87E+0	1.39E+1	5.15E+0	1.05E+1	1.33E+1	1.21E+1	3.75E+1	4.00E+2	8.87E+1	4.76E+1	4.12E+2	1.35E+2	9.03E+2	9.06E+2	9.04E+2
L-CMA-ES	2.00E+1	2.00E+1	2.00E+1	1.48E+0	2.98E+0	2.32E+0	1.28E+1	1.45E+1	1.40E+1	4.34E+1	7.65E+1	5.84E+1	2.66E+2	2.17E+3	1.07E+3	8.00E+2	9.18E+2	8.90E+2
DE	2.08E+1	2.10E+1	2.09E+1	2.31E+0	1.39E+1	4.51E+0	1.22E+1	1.37E+1	1.33E+1	2.52E+2	2.99E+2	2.82E+2	2.76E+2	3.32E+2	3.09E+2	9.12E+2	9.15E+2	9.13E+2
K-PCX	2.00E+1	2.00E+1	2.00E+1	2.27E+0	1.49E+1	1.19E+1	1.32E+1	1.41E+1	1.38E+1	1.85E+1	4.00E+2	7.15E+1	3.39E+1	5.45E+2	1.56E+2	8.28E+2	8.37E+2	8.30E+2
CoEVO	2.07E+1	2.10E+1	2.09E+1	5.20E+0	1.68E+1	9.02E+0	1.25E+1	1.36E+1	1.32E+1	2.77E+2	4.43E+2	3.81E+2	3.03E+2	5.54E+2	4.54E+2	1.01E+3	1.13E+3	1.06E+3

Table 14 Optimality gap values for f19, f20, f21, f22, f23, f24 and $n = 30$

Method	f19			f20			f21			f22			f23			f24		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
1000																		
STS	1.21E+3	1.42E+3	1.29E+3	1.20E+3	1.39E+3	1.28E+3	1.27E+3	1.43E+3	1.35E+3	1.23E+3	1.82E+3	1.52E+3	1.32E+3	1.51E+3	1.38E+3	1.37E+3	1.45E+3	1.41E+3
G-CMA-ES	9.16E+2	1.03E+3	9.51E+2	9.17E+2	1.00E+3	9.50E+2	6.70E+2	1.11E+3	9.44E+2	1.00E+3	1.21E+3	1.08E+3	8.14E+2	1.13E+3	1.03E+3	9.58E+2	1.24E+3	9.97E+2
EDA	1.21E+3	1.37E+3	1.32E+3	1.23E+3	1.39E+3	1.30E+3	1.37E+3	1.55E+3	1.48E+3	1.39E+3	1.79E+3	1.63E+3	1.41E+3	1.56E+3	1.50E+3	1.43E+3	1.56E+3	1.51E+3
BLX-MA	9.45E+2	1.27E+3	1.08E+3	9.22E+2	1.37E+3	1.10E+3	1.24E+3	1.43E+3	1.32E+3	1.19E+3	1.56E+3	1.33E+3	1.29E+3	1.44E+3	1.37E+3	1.29E+3	1.51E+3	1.45E+3
SPC-PNX	1.24E+3		1.30E+3	1.21E+3		1.28E+3	1.12E+3		1.15E+3	1.48E+3		1.59E+3	1.35E+3		1.44E+3	1.42E+3		1.46E+3
BLX-GL50	1.13E+3	1.26E+3	1.20E+3	1.13E+3	1.28E+3	1.20E+3	1.31E+3	1.42E+3	1.37E+3	1.29E+3	1.67E+3	1.43E+3	1.30E+3	1.40E+3	1.35E+3	1.29E+3	1.46E+3	1.40E+3
L-CMA-ES	1.01E+3	1.78E+3	1.27E+3	1.04E+3	1.49E+3	1.17E+3	1.02E+3	1.25E+3	1.17E+3	1.08E+3	2.51E+3	1.41E+3	1.06E+3	1.93E+3	1.40E+3	1.36E+3	1.91E+3	1.70E+3
DE	1.17E+3	1.33E+3	1.28E+3	1.25E+3	1.34E+3	1.29E+3	1.37E+3	1.51E+3	1.44E+3	1.30E+3	1.72E+3	1.57E+3	1.38E+3	1.47E+3	1.42E+3	1.43E+3	1.52E+3	1.48E+3
K-PCX	1.33E+3	1.49E+3	1.42E+3	1.33E+3	1.49E+3	1.42E+3	1.39E+3	1.65E+3	1.56E+3	1.58E+3	2.29E+3	1.91E+3	1.38E+3	1.64E+3	1.56E+3	1.48E+3	1.65E+3	1.59E+3
CoEVO	1.18E+3	1.39E+3	1.31E+3	1.17E+3	1.41E+3	1.33E+3	1.35E+3	1.53E+3	1.43E+3	1.46E+3	1.80E+3	1.63E+3	1.37E+3	1.50E+3	1.43E+3	1.42E+3	1.53E+3	1.49E+3
10000																		
STS	9.47E+2	1.21E+3	1.05E+3	9.23E+2	1.19E+3	1.04E+3	8.22E+2	1.28E+3	1.15E+3	1.07E+3	1.39E+3	1.24E+3	9.11E+2	1.28E+3	1.21E+3	1.25E+3	1.43E+3	1.34E+3
G-CMA-ES	9.04E+2	9.21E+2	9.08E+2	9.04E+2	9.11E+2	9.06E+2	5.00E+2	1.09E+3	5.47E+2	8.63E+2	9.30E+2	9.00E+2	5.34E+2	1.10E+3	6.92E+2	2.00E+2	9.64E+2	9.26E+2
EDA	1.10E+3	1.16E+3	1.13E+3	1.06E+3	1.15E+3	1.12E+3	1.24E+3	1.33E+3	1.28E+3	1.11E+3	1.25E+3	1.17E+3	1.22E+3	1.37E+3	1.29E+3	1.19E+3	1.29E+3	1.25E+3
BLX-MA	8.85E+2	9.11E+2	8.95E+2	8.85E+2	9.08E+2	8.96E+2	5.01E+2	7.83E+2	5.19E+2	9.67E+2	1.05E+3	1.01E+3	6.01E+2	1.18E+3	7.34E+2	2.29E+2	5.44E+2	3.14E+2
SPC-PNX	9.90E+2		1.01E+3	9.90E+2		1.01E+3	1.12E+3		1.15E+3	1.11E+3		1.15E+3	1.10E+3		1.15E+3	6.69E+2		8.15E+2
BLX-GL50	9.09E+2	9.13E+2	9.11E+2	9.09E+2	9.13E+2	9.11E+2	5.80E+2	6.60E+2	6.13E+2	8.93E+2	9.36E+2	9.18E+2	6.10E+2	9.63E+2	6.94E+2	3.44E+2	1.01E+3	9.27E+2
L-CMA-ES	9.11E+2	1.73E+3	1.03E+3	9.09E+2	9.42E+2	9.24E+2	5.00E+2	1.16E+3	6.18E+2	8.79E+2	1.21E+3	9.63E+2	5.34E+2	1.78E+3	7.51E+2	2.00E+2	1.87E+3	1.42E+3
DE	1.04E+3	1.14E+3	1.10E+3	1.04E+3	1.12E+3	1.10E+3	1.18E+3	1.28E+3	1.25E+3	1.20E+3	1.33E+3	1.26E+3	1.18E+3	1.27E+3	1.24E+3	1.17E+3	1.29E+3	1.25E+3
K-PCX	1.28E+3	1.47E+3	1.37E+3	1.22E+3	1.47E+3	1.36E+3	1.39E+3	1.65E+3	1.56E+3	1.50E+3	2.18E+3	1.86E+3	1.38E+3	1.64E+3	1.56E+3	1.43E+3	1.65E+3	1.57E+3
CoEVO	1.09E+3	1.23E+3	1.15E+3	1.05E+3	1.26E+3	1.16E+3	1.20E+3	1.27E+3	1.24E+3	1.16E+3	1.35E+3	1.25E+3	1.18E+3	1.31E+3	1.26E+3	1.27E+3	1.37E+3	1.32E+3

Table 14 (*Continued*)

Method	f19			f20			f21			f22			f23			f24		
	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg	min	max	avg
100000																		
STS	8.97E+2	1.01E+3	9.23E+2	8.85E+2	9.41E+2	9.13E+2	5.00E+2	1.16E+3	5.51E+2	1.02E+3	1.17E+3	1.07E+3	5.34E+2	1.17E+3	6.18E+2	2.00E+2	1.29E+3	1.00E+3
G-CMA-ES	9.03E+2	9.06E+2	9.04E+2	9.03E+2	9.06E+2	9.04E+2	5.00E+2	5.00E+2	5.00E+2	7.97E+2	8.51E+2	8.27E+2	5.34E+2	1.10E+3	5.82E+2	2.00E+2	9.56E+2	9.13E+2
EDA	9.57E+2	9.95E+2	9.76E+2	9.53E+2	1.00E+3	9.76E+2	6.15E+2	1.13E+3	7.08E+2	8.98E+2	9.42E+2	9.23E+2	6.65E+2	1.12E+3	7.70E+2	4.09E+2	1.02E+3	7.43E+2
BLX-MA	8.74E+2	8.90E+2	8.82E+2	8.68E+2	8.89E+2	8.80E+2	5.00E+2	5.00E+2	5.00E+2	8.93E+2	9.27E+2	9.14E+2	5.34E+2	1.16E+3	5.59E+2	2.00E+2	2.00E+2	2.00E+2
SPC-PNX	9.06E+2		9.07E+2	9.06E+2		9.07E+2	5.00E+2		5.00E+2	8.88E+2		9.09E+2	5.34E+2		5.34E+2	2.00E+2		2.00E+2
BLX-GL50	9.03E+2	9.06E+2	9.04E+2	9.03E+2	9.04E+2	9.03E+2	5.00E+2	5.00E+2	5.00E+2	8.51E+2	9.07E+2	8.78E+2	5.34E+2	9.17E+2	5.87E+2	2.00E+2	9.94E+2	8.89E+2
L-CMA-ES	9.07E+2	9.30E+2	9.15E+2	8.00E+2	9.22E+2	9.05E+2	4.09E+2	5.00E+2	4.96E+2	8.51E+2	9.34E+2	8.85E+2	5.34E+2	1.56E+3	6.44E+2	2.00E+2	1.86E+3	1.41E+3
DE	9.24E+2	9.51E+2	9.36E+2	9.28E+2	9.48E+2	9.38E+2	8.50E+2	9.28E+2	8.89E+2	1.01E+3	1.09E+3	1.04E+3	8.35E+2	9.72E+2	9.15E+2	5.99E+2	8.17E+2	7.47E+2
K-PCX	9.83E+2	1.14E+3	1.06E+3	9.80E+2	1.14E+3	1.06E+3	9.71E+2	1.04E+3	1.01E+3	1.04E+3	2.05E+3	1.77E+3	9.78E+2	1.06E+3	1.02E+3	3.52E+2	5.09E+2	4.17E+2
CoEVO	1.05E+3	1.15E+3	1.10E+3	1.00E+3	1.18E+3	1.11E+3	7.09E+2	1.21E+3	9.94E+2	1.10E+3	1.27E+3	1.18E+3	7.62E+2	1.23E+3	1.10E+3	1.13E+3	1.33E+3	1.23E+3
300000																		
STS	8.84E+2	9.79E+2	9.02E+2	8.00E+2	9.15E+2	8.97E+2	5.00E+2	5.00E+2	5.00E+2	8.42E+2	1.05E+3	9.91E+2	5.34E+2	1.16E+3	5.75E+2	2.00E+2	1.24E+3	4.45E+2
G-CMA-ES	9.03E+2	9.04E+2	9.04E+2	9.03E+2	9.04E+2	9.04E+2	5.00E+2	5.00E+2	5.00E+2	7.67E+2	8.42E+2	8.03E+2	5.34E+2	5.34E+2	5.34E+2	2.00E+2	9.56E+2	9.10E+2
EDA	9.16E+2	9.31E+2	9.26E+2	9.20E+2	9.36E+2	9.27E+2	5.04E+2	1.10E+3	5.31E+2	8.83E+2	9.17E+2	8.97E+2	5.42E+2	1.10E+3	6.02E+2	2.07E+2	1.01E+3	5.79E+2
BLX-MA	8.71E+2	8.86E+2	8.80E+2	8.68E+2	8.89E+2	8.79E+2	5.00E+2	5.00E+2	5.00E+2	8.92E+2	9.23E+2	9.08E+2	5.34E+2	1.16E+3	5.59E+2	2.00E+2	2.00E+2	2.00E+2
SPC-PNX	9.04E+2		9.05E+2	9.04E+2		9.05E+2	5.00E+2		5.00E+2	8.61E+2		8.80E+2	5.34E+2		5.34E+2	2.00E+2		2.00E+2
BLX-GL50	9.03E+2	9.06E+2	9.04E+2	9.03E+2	9.04E+2	9.03E+2	5.00E+2	5.00E+2	5.00E+2	8.42E+2	9.01E+2	8.74E+2	5.34E+2	9.17E+2	5.87E+2	2.00E+2	9.89E+2	8.77E+2
L-CMA-ES	8.00E+2	9.23E+2	9.03E+2	8.00E+2	9.16E+2	8.89E+2	4.09E+2	5.00E+2	4.85E+2	8.25E+2	9.20E+2	8.71E+2	5.34E+2	5.41E+2	5.35E+2	2.00E+2	1.86E+3	1.41E+3
DE	9.11E+2	9.16E+2	9.13E+2	9.11E+2	9.15E+2	9.13E+2	5.36E+2	6.51E+2	5.81E+2	9.41E+2	9.92E+2	9.64E+2	5.73E+2	6.79E+2	6.21E+2	2.55E+2	3.82E+2	3.14E+2
K-PCX	8.28E+2	8.34E+2	8.31E+2	8.28E+2	8.33E+2	8.31E+2	8.58E+2	8.60E+2	8.59E+2	5.25E+2	1.96E+3	1.56E+3	8.65E+2	8.68E+2	8.66E+2	2.12E+2	2.14E+2	2.13E+2
CoEVO	1.01E+3	1.11E+3	1.05E+3	9.83E+2	1.16E+3	1.06E+3	5.00E+2	1.18E+3	6.04E+2	1.10E+3	1.23E+3	1.16E+3	6.07E+2	1.22E+3	9.22E+2	7.60E+2	1.27E+3	1.10E+3

test rank: L-CMA-ES, G-CMA-ES and STS, where we represent the relative percentage errors with respect to the optima for the 12 test instances with $n = 10$. This diagram shows that G-CMA-ES exhibits the largest number of test functions with an associated percentage error in the interval [0, 10], L-CMA-ES in [50, 80] and STS in [80, 1000]. In the rest of the relative errors the three methods perform very similar (they share most of the line-profile in the diagram); thus confirming the previous results.

5 Conclusions

We have described the development and implementation of an adaptive memory programming procedure integrating Scatter Search and Tabu Search for unconstrained nonlinear optimization. Based on a series of preliminary experiments to identify effective ways to coordinate the underlying strategies, we are able to produce a method that reaches good quality solutions on previously reported problems (including the methods reported in the well-known CEC2005 competition). Our extensive comparison with sixteen methods previously published shows that the STS method obtains good quality solutions for unconstrained global optimization problems.

Our experimentation shows that the improvement method can be significantly improved when a memory structure is introduced. This is especially true with the line search based method, but is still true with the popular Nelder and Mead simplex method. Moreover, our study reveals that a combination of line search with the simplex, both with a memory structure, produces high quality outcomes.

We anticipate that significant opportunity exists for creating an improved version of our method since a number of intensification and diversification mechanisms that have been previously proposed with Scatter Search and Tabu Search remain to be examined in the present context. The fact that the two most competitive previous methods made use of TS and SS strategies, and that the successes achieved by our present method derives from a marriage of such memory-based strategies, suggests that appropriately designed memory mechanisms will prove valuable in future explorations of methods for solving bound-constrained continuous optimization problems.

Acknowledgements This research has been partially supported by the *Ministerio de Educación y Ciencia* of Spain (Grant Refs. TIN2006-0269, TIN2009-07516), and by the Comunidad de Madrid, Universidad Rey Juan Carlos project (Ref. URJC-CM-2008-CET-3731).

References

Addis, B., & Leyffer, S. (2006). A trust-region algorithm for global optimization. *Computational Optimization and Applications*, *35*, 287–304.

Addis, B., Locatelli, M., & Schoen, F. (2005). Local optima smoothing for global optimization. *Optimization Methods and Software*, *20*, 417–437.

Auger, A., & Hansen, N. (2005a). A restart CMA evolution strategy with increasing population size. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 1769–1776).

Auger, A., & Hansen, N. (2005b). Performance evaluation of an advanced local search evolutionary algorithm. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 1777–1784).

Avriel, M. (1976). *Nonlinear programming, analysis and methods*. Prentice-Hall: Englewood Cliffs.

Ballester, P. J., Stephenson, J., Carter, J. N., & Gallagher, K. (2005). Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 498–505).

Chelouah, R., & Siarry, P. (2005). A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimization of multiminima functions. *European Journal of Operational Research*, *161*, 636–654.

García-Martínez, C., & Lozano, M. (2005). Hybrid real-coded genetic algorithms with female and male differentiation. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 896–903).

Glover, F. (1977). Heuristics for integer programming using surrogate constraints. *Decision Sciences*, *8*, 156–166.

Glover, F. (1994). Tabu search for nonlinear and parametric optimization (with links to genetic algorithms). *Discrete Applied Mathematics*, *49*, 231–255.

Glover, F. (1998). A template for scatter search and path relinking. In J. K. Hao, E. Lutton, E. Ronald, M. Schoenauer, & D. Snyers (Eds.), *Lecture notes in computer science: Vol. 1363. Artificial evolution* (pp. 13–54). Berlin: Springer.

Glover, F., & Laguna, M. (1997). *Tabu search*. Dordrecht: Kluwer Academic.

Glover, F., Kuo, C. C., & Dhir, K. S. (1998). Heuristic algorithms for the maximum diversity problem. *Journal of Information and Optimization Sciences*, *19*(1), 109–132.

Glover, F., Laguna, M., & Martí, R. (2000). Fundamentals of scatter search and path relinking. *Control and Cybernetics*, *29*(3), 653–684.

Hansen, N. (2006). *Compilation of results on the 2005 CEC benchmark function set* (Technical report). CoLAB Institute of Computational Sciences ETH, Zurich.

Hedar, A., & Fukushima, M. (2004). Heuristic pattern search and its hybridization with simulated annealing for nonlinear global optimization. *Optimization Methods and Software*, *19*, 291–308.

Hedar, A., & Fukushima, M. (2006). Tabu search directed by direct search methods for nonlinear global optimization. *European Journal of Operational Research*, *170*(2), 329–349.

Hirsch, M. J., Meneses, C. N., Pardalos, P. M., & Resende, M. G. C. (2007). Global optimization by continuous GRASP. *Optimization Letters*, *1*(2), 201–212.

Huyer, W., & Neumaier, A. (1999). Global optimization by multilevel coordinate search. *Journal of Global Optimization*, *14*, 331–355.

Hvattum, L.-M., & Glover, F. (2007). *Finding local optima of high-dimensional functions using direct search methods*. Research paper, Molde University College, Molde, Norway.

Johnson, D. S. (2001). *A theoretician's guide to the experimental analysis of algorithms*. AT&T Labs Research. http://www.research.att.com/~dsj/.

Laguna, M., & Martí, R. (2003). *Scatter search—methodology and implementations in C*. Boston: Kluwer Academic.

Laguna, M., & Martí, R. (2005). Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, *33*, 235–255.

Leary, R. H. (2000). Global optimization on funneling landscapes. *Journal of Global Optimization*, *18*(4), 367–383.

Liang, J. J., & Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer with local search. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 522–528).

Michalewicz, Z. (1996). *Genetic algorithms + data structures = evolution programs* (3rd ed.). Berlin: Springer.

Molina, D., Herrera, F., & Lozano, M. (2005). Adaptive local search parameters for real-coded memetic algorithms. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 888–895).

Nelder, J. A., & Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, *7*, 308–313.

Posik, P. (2005). Real-parameter optimization using the mutation step co-evolution. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 872–879).

Qin, A. K., & Suganthan, P. N. (2005). Self-adaptive differential evolution algorithm for numerical optimization. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 1785–1791).

Rego, C. (2005). RAMP: a new metaheuristic framework for combinatorial optimization. In C. Rego & B. Alidaee (Eds.), *Metaheuristic optimization via memory and evolution: tabu search and scatter search* (pp. 441–460). Dordrecht: Kluwer Academic.

Rego, C., & Alidaee, B. (2005). *Metaheuristic optimization via memory and evolution*. Boston: Kluwer Academic.

Resende, M. G. C., & Ribeiro, C. C. (2001). Greedy randomized adaptive search procedures. In F. Glover & G. Kochenberger (Eds.), *State-of-the-art handbook in metaheuristics* (pp. 219–250). Boston: Kluwer Academic.

Rönkkönen, J., Kukkonen, S., & Price, K. V. (2005). Real-parameter optimization with differential evolution. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 506–513).

Sinha, A., Tiwari, S., & Deb, K. (2005). A population-based, steady-state procedure for real-parameter optimization. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 514–521).

Suganthan, P. N., Hansen, N., Liang, J. J., 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). Nanyang technological university of Singapore.

Yuan, B., & Gallagher, M. (2005). Experimental results for the special session on real-parameter optimization at CEC 2005: a simple, continuous EDA. In *Proceedings of 2005 IEEE congress on evolutionary computation (CEC'2005)* (pp. 1792–1799).