Path Relinking for Large Scale Global Optimization

ABRAHAM DUARTE
Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Spain.
Abraham.Duarte@urjc.es

RAFAEL MARTÍ
Departamento de Estadística e Investigación Operativa, Universidad de Valencia, Spain.
Rafael.Marti@uv.es

FRANCISCO GORTAZAR
Departamento de Ciencias de la Computación, Universidad Rey Juan Carlos, Spain.
Francisco.Gortazar@urjc.es

ORIGINAL VERSION: March 25, 2010
REVISED VERSION: July 2, 2010

Abstract

In this paper we consider the problem of finding a global optimum of a multimodal function applying path relinking. In particular, we target unconstrained large scale problems and compare two variants of this methodology: the static and the evolutionary path relinking. Both are based on the strategy of creating trajectories of moves passing through high quality solutions in order to incorporate their attributes to the explored solutions.

Computational comparisons are performed on a test-bed of 19 global optimization functions previously reported with dimensions ranging from 50 to 1000, totalizing 95 instances. Our results show that the evolutionary path relinking procedure is competitive with the state-of-the-art methods in terms of the average optimality gap achieved. Statistical analysis is applied to draw significant conclusions.

Key words: Evolutionary Algorithms, Path Relinking, Metaheuristics, Global Optimization.

1. Introduction

Path-relinking (PR) is an intensification strategy to explore trajectories connecting elite solutions obtained by heuristic methods (Glover and Laguna 1997). It can be considered as an extension of the combination methods applied in most evolutionary algorithms. Instead of directly producing a new solution when combining two or more original solutions, PR generates paths between and beyond the selected solutions in the neighbourhood space. In particular, in global optimization, where solutions are represented as real vectors, most evolutionary algorithms perform linear combinations between pairs of solutions. Alternatively, in problems where solutions are represented as a permutation, integer or binary vectors, such as ordering or knapsack-type problems, other kinds of combination methods have been applied. In all these settings, path-relinking provides a unified approach to produce combination methods for all types of problems. In this paper we explore the application of the path relinking methodology, in its variant known as evolutionary path relinking, to the global optimization problem.

The strategy of creating trajectories of moves passing through high quality solutions was first proposed in connection with tabu search in Glover (1989). The approach was then elaborated in greater detail in Glover (1994), as a means of integrating TS intensification and diversification strategies, and given the name path relinking (PR). PR generally operates by starting from an **initiating solution**, selected from a subset of high quality solutions, and generating a path in the neighborhood space that leads toward the other solutions in the subset, which are called **guiding solutions**. This is accomplished by selecting moves that introduce attributes contained in the guiding solutions.

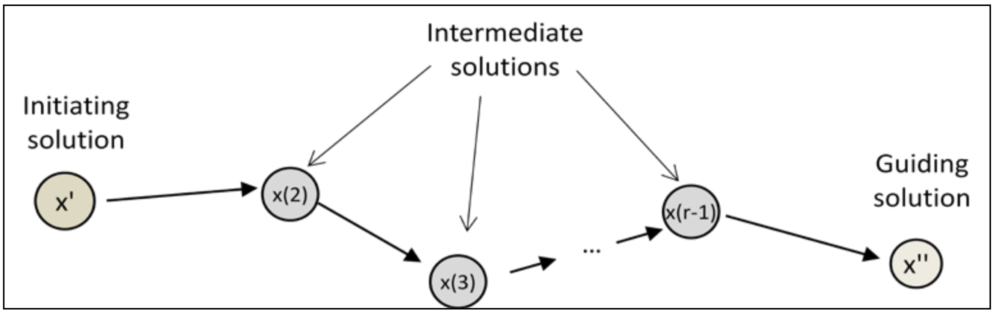

Figure 1. Path relinking representation

To generate the desired paths, it is only necessary to select moves that perform the following role: upon starting from an initiating solution, the moves must progressively introduce attributes contributed by a guiding solution as it is shown in Figure 1. The roles of the initiating and guiding solutions are interchangeable; each solution can also be induced to move simultaneously toward the other as a way of generating combinations. First consider the creation of paths that join two selected solutions x′ and x″, restricting attention to the part of the path that lies "between" the solutions, producing a sequence x′ = x(1), x(2), …, x(r) = x″ of **intermediate solutions**. The relinked path may encounter solutions that may not be better than the initiating or guiding solution, but that provide fertile "points of access" for reaching other, somewhat better, solutions. For this reason it is valuable to examine neighboring solutions along a relinked path, and keep track of those of high quality which may provide a starting point for launching additional searches.

Laguna and Martí (1999) proposed the adaptation of path relinking to the context of multi-start methods in which the solutions are not previously linked. Specifically, they coupled GRASP with PR as a form of intensification. The relinking in this method consists in finding a path between a solution found with GRASP and a chosen elite solution. Resende and Ribeiro (2003) present numerous examples of GRASP with PR. Resende and Werneck (2004) introduced evolutionary path relinking (EvoPR) as a post-processing phase for GRASP with PR (see also Andrade and Resende 2007). In EvoPR, the solutions in the elite set are evolved in a similar way that the reference set evolves in scatter search (Laguna and Martí, 2003).

In this paper we explore the adaptation of the PR and EvoPR methodologies to obtain high quality solutions to the unconstrained global optimization problem. This problem can be formulated as follows:

$$(P) \quad Minimize \;\; f(x)$$
$$l \leq x \leq u$$
$$x \in R^n$$

where $f(x)$ is a nonlinear function and x is a vector of continuous and bounded variables. We investigate the path relinking methods for (P) and perform comparative computational testing with currently leading methods for unconstrained global optimization on a benchmark set of high-dimensional problems for which global optima are known.

In prior work on unconstrained global optimization, scatter search was applied as a stand-alone method (without local optimization) in Laguna and Martí (2005). It focused on testing several alternatives for generating diversification and updating the reference set. However, the combinations generated by their approach are linear and limited to joining pairs of solutions. This study is extended in Duarte et al. (2010), where a scatter tabu search method, STS, is presented. It basically hybridizes the scatter search methodology with two tabu search improvement methods. Based on extensive experimentation with the CEC2005 instances (Suganthan et al. 2005) and sixteen previous methods, twelve of them from Hansen (2006), the study identified two leading methods: the proposed STS and, the covariance matrix adaptation evolution strategy, G-CMA-ES (Auger and Hansen 2005). On the other hand, Herrera et al. (2010b) considered three previous methods as the state-of-the-art on unconstrained global optimization: Differential Evolution DE (Storn and Price 1997), G-CMA-ES (Auger and Hansen 2005) and Real coded CHC (Eshelman and Schaffer 1993). We include these four methods (STS, DE, G-CMA-ES and CHC) in our computational testing.

The next section describes the basic path relinking approach for the unconstrained global optimization problem. Section 3 provides insight on the evolutionary path relinking, which can be considered as an extension of the basic design. We perform a computational study comparing our method to the four leading methods previously indicated, applied to the set of 19 scalable functions proposed in Herrera et al. (2010a) whose form is described in Section 4, where we also report our computational findings. Finally, we summarize our conclusions in Section 5.

2. Path Relinking

Figure 2 shows the pseudo-code of a simple PR procedure for a minimization problem. It starts with the generation of solutions. The reference set (RefSet) contains b elite solutions previously generated. It can be constructed with a diversification generator method as in scatter search (Laguna and Martí 2003), where we build a large set of diverse solutions D and then extract the b bests (according to quality and diversity). However, path relinking is not restricted to this design and can start from a set of elite solutions obtained during any previous search process.

To reduce the computational effort, we limit the application of the improvement method to the best solution in *RefSet* (step 3). In step 4, *NewSubsets* is constructed with the sets of solutions in *RefSet* to be submitted to the relinking process. It must be noted that the path relinking methodology is not limited to relink pairs of solutions; but it permits to relink an arbitrary number of solutions. In this paper we explore the relinking from one solution a to a pair of solutions, x and y. We will denote it as (a, x, y).

The sets (a, x, y) in *NewSubsets* are selected one at a time in lexicographical order and the Relinking Method is applied to generate a path of solutions from a to x and y in steps 5 and 6 of Figure 2. The Improvement Method is applied to the best solution in the path (step 7). The improved solution is checked to see whether it improves upon the best solution found x^1. If so, the new solution replaces it. The search finishes when all the sets in *NewSolutions* have been examined.

---

1. Create a *RefSet* of *b* elite solutions.
2. Evaluate the solutions in *RefSet* and order them. Let x^1 be the best one.
3. Apply the improvement method to x^1 and replace it with the improved solution.
4. Generate *NewSubsets*, which consists of the sets (*a, x, y*) of solutions in *RefSet*.

while (*NewSubsets* $\neq \emptyset$) **do**

    5. Select the next set (*a, x, y*) in *NewSubSets.*
    6. Apply the Relinking Method to produce the sequence from *a* to *x* and *y*.
    7. Apply the Improvement Method to the best solution in the sequence. Let *w* be the improved solution.

    if ($f(w) < f(x^1)$) **then**

        8. Make $x^1 = w$

    end if

    9. Delete (*a, x, y*) from *NewSubsets*

end while

Figure 2. Path relinking procedure

To generate the sets of solutions (a, x, y) in *RefSet* to be submitted to the relinking process, we adapt a method typically implemented in scatter search (Martí et al. 2006). It generates subsets of three reference solutions by expanding pairs into subsets of larger size. The objective is to select representative subsets of different compositions while limiting the number of them. Specifically, considering the *RefSet* $= \{x^1, x^2, \dots, x^b\}$ where the solutions are ordered by quality (i.e., $f(x^i) \leq f(x^{i+1})$ for $i = 1, \dots, b-1$), we limit the relinking sets to 3-tuples of the form (x^i, x^j, x^{j+1}) with $i < j$. Consider for example that $b = 6$ and the ordered *RefSet* is $\{x^1, x^2, \dots, x^6\}$ where x^1 is the best solution. Then, to create a path from, for example x^1, we can consider the guiding solutions x^2 and x^3, thus obtaining the relinking set (x^1, x^2, x^3). On the other hand, to create a path from, for example x^3, we can consider, the guiding solutions x^5 and x^6 thus obtaining the relinking set (x^3, x^5, x^6).

The path relinking approach subordinates other considerations, such as the objective function value, to the goal of choosing moves that introduce the attributes of the guiding solutions, in order to create a "good attribute composition" in the current solution. The approach is called path relinking either by virtue of generating a new path between solutions previously linked by a series of moves executed during a search, or by generating a path between solutions previously linked to other solutions but not to each other. However, in the context of GRASP with Path Relinking (Laguna and Martí 1999) the solutions are not previously linked, since they are independently obtained by strategically sampling the solution space. From this point of view, PR can be simply considered as a population-based method that operates on a set of reference or elite solutions by combining them in a specific way. To unify the notation with scatter search we will also let *RefSet* refer to this set of reference solutions that have been selected or generated with an embedded search method.

2.1 Reference Set Initialization

We have implemented a generator of solutions based on techniques from the area of statistics known as Design of Experiments. One of the most popular design of experiments is the factorial design k^n, where n is the number of factors (in our case variables) and k is the number of levels (in our case possible variable values). A full factorial design considers that all combinations of the factors and levels will be tested. However, it can quickly become impractical even for a small number of levels, because the number of experiments exponentially increases with the number of factors. We therefore consider a factorial design, in which we draw conclusions based on a fraction of experiments, which are strategically selected from the set of all possible experiments in the corresponding full factorial design. One of the most notable proponents of the use of fractional factorial designs is Genichi Taguchi (Roy, 1990), who proposed a special set of orthogonal arrays to lay out experiments associated with quality improvement in manufacturing. These orthogonal arrays are the result of combining orthogonal Latin squares in a unique manner. We use Taguchi's arrays as a mechanism for generating diversity, as previously introduced in Laguna and Martí (2005). Table 1 shows the $L_9(3^4)$ orthogonal array that can be used to generate 9 solutions for a 4-variable problem.

Experiment	**Factors**			
	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	2	3
8	3	2	1	3
9	3	3	2	1

Table 1. $L_9(3^4)$ orthogonal array.

The values in Table 1 represent the levels at which the factors are set in each experiment. For the purpose of creating a diversification generator based on Taguchi tables, we translate each level setting as follows:

$$1 := mid\ value = \ l_i + \frac{1}{2}(u_i - l_i)$$

$$2 := lower\ value = \ l_i + \frac{1}{4}(u_i - l_i)$$

$$3 := upper\ value = \ l_i + \frac{3}{4}(u_i - l_i)$$

Since we are facing high dimensional problems $(50 \le n \le 1000)$ and we have found Taguchi tables up to $n = 40$, we will split the set of variables into subsets of 40 and complete the rest of variables with the value assigned to levels 1, 2 or 3 . The table with 40 variables and three levels contains 81 experiments. Then, we generate 81 solutions by assigning the values in the table to the first 40 variables and the mid value (level 1) to the rest of the variables. We generate 81 more solutions by keeping the values in the table to the first 40 variables and assigning the lower value (level 2) to the rest of the variables. Similarly, assigning the upper value (level 3) we obtain another 81 solutions. In this way we generate 243 solutions applying the values in the table to the first 40 variables. We now move to the next set of variables to assign the Taguchi levels. Specifically, we apply the Taguchi values to variables from 21 to 60. We have experimentally found that with this "shifting", in which we move 20 positions in the list of variables to assign the next 40 variables to the Taguchi levels, we obtain good results with a low computational effort (i.e., evaluating a relatively reduced number of solutions). Therefore, we generate three groups of 81 solutions by assigning the values in the table to the variables from 21 to 60 and the mid, lower and upper values respectively to the rest of the variables, thus obtaining another 243 solutions. We proceed in the same way, generating a total of $DSize = 243\lceil n/20 \rceil$ initial solutions.

The method then starts by generating a set D with $DSize$ solutions strategically distributed in the solution space. The *RefSet* is populated with the best b solutions in D. It must be noted that this is not the standard way to create the *RefSet*, in which diversity is usually considered. However, the application of the Taguchi strategy directly provides the desired level of diversity in the generated solutions, and we have found that there is no need for including an additional diversification strategy.

2.2 Improvement Method: Two stage line search

We implement the so-called line-search coupled with the simplex method as our improvement method. The combination of these two procedures was successfully applied as the improvement method in Duarte et al. (2010), in which memory structures were also included. Their method first orders the variables according to their attractiveness and then selects the first *ts* (where *ts* is a search parameter) to perform the associated line searches. As it is customary in tabu search, the method permits non-improving moves that deteriorates the objective function value. In this way, the best solution in the line search is selected even if it does not improve the original solution. When a variable is selected and we move to the best solution in its associated line-search, we labelled it as tabu and we do not allow the method to select it in the next *tenure* iterations. After this part based on local searches, the Simplex method is applied if the final solution obtained lies within a hypersphere of radius *T* centered at any solution previously submitted to the simplex method. We implement here a variant of that procedure with two stages but with no memory structures. Figure 3 shows a pseudo-code of the method.

1. Let x be the initial solution. Set $h = range_{min}/100$.
2. For $k = 1$ to 10
    3. For each variable i compute $x + he_i$ and $x - he_i$ and consider the best of both values.
    4. Order the variables according to these values in increasing order.
    5. For $s = 1$ to $n/2$
        6. Select the next variable i in the ordered list.
        7. Perform a line search along $x + qhe_i$.
        8. Make $x := $ the best solution in the line search.
9. Apply the Simplex method to x, the best solution found.
    10. Select α randomly in $[-h, h]$.
    11. Generate the n points $x^1, x^2, \dots, x^n$ where $x^i = (x_1, \dots, x_i + \alpha, \dots, x_n)$.
        12. Apply the Simplex Method for a maximum of 1000 evaluations.
        13. Let x be the best solution found.
14. Return x

Figure 3. Improvement method

Our improvement method searches in a discretized space (as it is typically done in global optimization). According to the experimentation in Duarte et al. (2010) we use a grid of size $h = range_{min}/100$ where $range_{min}$ is the minimum range of the variables (i.e., the minimum difference between the upper and the lower bounds).

The first stage of the improvement method applies consecutive line searches, each of which consists of modifying the value of a variable i (which is equivalent of moving in the direction of a vector unit e_i). Given a solution x we first evaluate the potential contribution of each variable to improve its value. Specifically, for each variable $i = 1, \ldots, n$, we evaluate two solutions $x + he_i$ and $x - he_i$, where h is the width of the grid, and consider the best of the two values. Then, we order the variables according to these values (where the variable i with better associated value comes first). We then select the $n/2$ first variables in the ordered list and perform a line search for each of them.

Given a variable i a line search consists of examining the feasible solutions with the form $x + qhe_i$ where q is an integer value in $[-20,20]$. Considering that only bound constraints on the variables are present, the feasibility condition only requires to verify $l \leq x + qhe_i \leq u$. To reduce the number of evaluations in this process we apply a *first improving strategy*, randomly scanning the solutions of the form $x + qhe_i$ and selecting the first one improving the current solution. We then resort to the next variable in the ordered list and perform a line search from the best solution found in the previous line search. After $n/2$ consecutive line searches we recalculate the potential contribution of each variable by computing again $x + he_i$ and $x - he_i$ for the current solution x, where $i = 1$ to n. Then, we order the variables according to these new values (starting now with the first variable in the new ordered list). We repeat this process until no further improvement is found or for a maximum of 10 iterations, thus performing a maximum of $10n/2$ line searches.

The second stage is applied to the best solution $x = (x_1, x_2, \ldots, x_n)$ found in the first stage. It starts by perturbing the value of each variable in x in the amount α to create an initial simplex. Specifically, we generate the n points $x^1, x^2, \ldots, x^n$ where $x^i = (x_1, \ldots, x_i + \alpha, \ldots, x_n)$. The value of α is randomly selected in $[-h, h]$. Then, the simplex method performs iterations attempting to replace the worst point in the simplex by a new and better one using reflection, expansion, and contraction steps (Avriel 1976). The method finishes when the number of function evaluations reaches 1000 or if the improvement achieved is lower than 0.001.

It must be noted that the line search only explores solutions in the discretized grid. This permits to efficiently examine scattered solutions but at the same time limits the method. On the other hand, the Simplex is not limited to the grid but only explores a small region. Therefore the combination of both methods complements each other.

2.3 Linking Solutions

We consider two different ways to link solutions with a path of intermediate solutions (as shown in Figure 1). The first one, called **orthogonal linking**, sequentially replaces coordinates of the guiding solutions into the initiating solution. Specifically, in (a, x, y) we create a path of intermediate solutions starting in $a = (a_1, a_2, \ldots, a_n)$ by replacing alternatively its coordinates in blocks of size m with those in $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, obtaining the sequence of solutions $a(1), a(2), a(3), \ldots$, where:

$a(1) = (x_1, x_2, \ldots, x_m, a_{m+1}, a_{m+2}, \ldots, a_n)$,
$a(2) = (x_1, x_2, \ldots, x_m, y_{m+1}, y_{m+2}, \ldots, y_{2m}, a_{2m+1}, a_{2m+2}, \ldots, a_n)$,
$a(3) = (x_1, x_2, \ldots, x_m, y_{m+1}, y_{m+2}, \ldots, y_{2m}, x_{2m+1}, x_{2m+2}, \ldots, x_{3m}, a_{3m+1}, a_{3m+2}, \ldots, a_n)$.

To limit the number of points in the path we compute $m = n/k$ where k is a low value that will be set in our computational study. The diagram in Figure 4 illustrates this method with a simple case with two variables and $m = 1$. We represent an initiating solution $a = (a_1, a_2)$ and two guiding solutions, $x = (x_1, x_2)$ and $y = (y_1, y_2)$. In the first step, we replace the first coordinate in a with the first coordinate in x, keeping the second coordinate as it is, thus obtaining $a(1) = (x_1, a_2)$. In the second step we keep the added coordinate in $a(1)$ and replace its second coordinate with its value in y, obtaining $a(2) = (x_1, y_2)$.

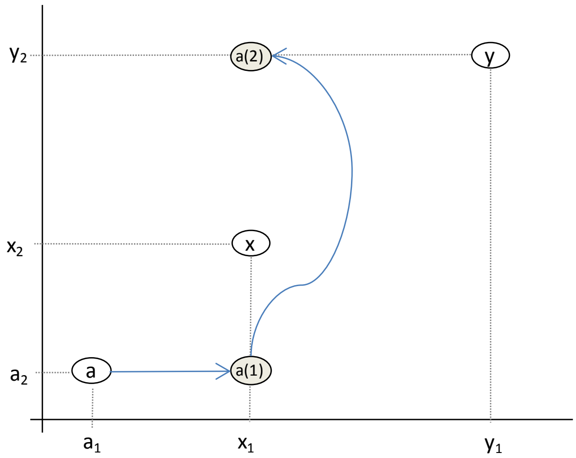

Figure 4. Orthogonal Linking

Our second option for relinking consists of moving from the initiating solution a, in the direction given by the vector from a to the first guiding solution x. We consider the intermediate solutions $a(1), a(2), \ldots, a(k-1)$ obtained as the convex combination of a and x in the first half segment joining them as:

$$a(1) = a + \frac{1}{k}(x - a)$$

$$a(2) = a + \frac{1}{k-1}(x - a)$$

……

$$a(k-1) = a + \frac{1}{2}(x - a)$$

Then, we select the best intermediate solution above, say $a(j)$, and repeat the process from $a(j)$ to y. In particular, we examine:

$$a(k) = a(j) + \frac{1}{k}(y - a(j))$$

$$a(k+1) = a(j) + \frac{1}{k-1}(y - a(j))$$

……

$$a(2k-2) = a(j) + \frac{1}{2}(y - a(j))$$

Figure 5 shows a representation of this process that we call **straight linking** on a small example with two coordinates.

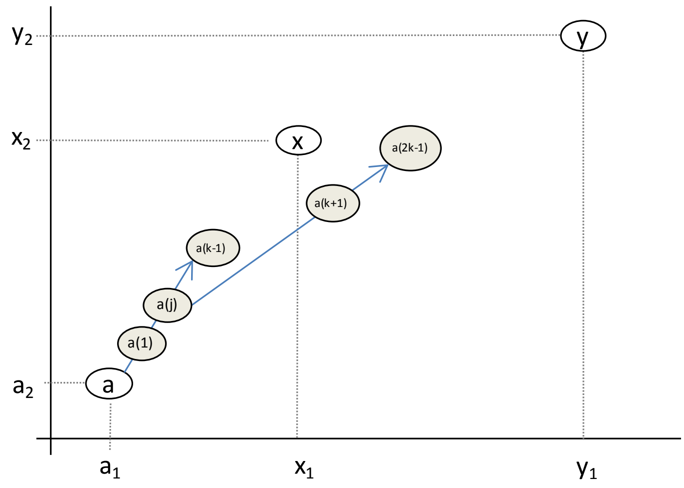

Figure 5. Straight Linking

In our computational experiments we compare both types of relinking and select the best one in terms of solution quality for our final design.

3. Evolutionary Path Relinking

Figure 6 shows the pseudo-code of an Evolutionary PR procedure for a minimization problem. It starts with the creation of an initial set of b elite solutions (*RefSet*). As in the SS method, the solutions in *RefSet* are ordered according to quality, and the search is initiated by assigning the value of TRUE to the Boolean variable *NewSolutions*. In step 3, *NewSubsets* is constructed with the (a, x, y) sets of solutions (described in Section 2) in *RefSet*, and *NewSolutions* is switched to FALSE.

The sets in *NewSubsets* are selected one at a time in lexicographical order and the Relinking Method is applied to generate a path of solutions in step 5 of Figure 6. The Improvement Method is applied to the best solution found in the relinking process (step 6). The improved solution, w, is added to the *Pool*. When all the sets (a, x, y) in *NewSubSets* have been explored, we examine in steps 8 to 10 the improved solutions added to *Pool* to check whether they qualify to enter the *RefSet*. In this way, we evolve the *RefSet* alternating both phases, relinking the solutions (steps 3 to 7) and updating it (steps 8 to 10) until the maximum number of function evaluations *MaxEvaluations* is reached. If at some point no new solution in *Pool* qualifies to enter the *RefSet* or the number of global iterations (relinking+updating) reaches the maximum value, *MaxIter*, the *RefSet* is rebuilt (step 12) and the search continues.

It must be noted that the criterion to enter a solution to the *RefSet* depends on both, quality and diversity. Given a solution w in *Pool*, let x^w be the closest solution to w in the *RefSet* solutions with a worse value than w. In mathematical terms,

$$x^w = \operatorname{argmin}_{x \in RefSet} \{d(x, w) : f(x) > f(w)\}.$$

We admit w into *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 x^w, is larger than the preestablished distance threshold *dthresh* (see the If- statement between steps 8 and 9 in Figure 6).

1. Obtain a *RefSet* of *b* elite solutions.
2. Evaluate the solutions in *RefSet* and order them according to their objective function value such that x^1 is the best solution and x^b the worst. Make *NewSolutions* = TRUE and *GlobalIter*=0.

while (*NumEvaluations* < *MaxEvaluations*) **do**

3. Generate *NewSubsets*, which consists of the sets (*a, x, y*) of solutions in *RefSet* that include at least one new solution. Make *NewSolutions* = FALSE and *Pool* = ∅.

while (*NewSubsets* ≠ ∅) **do**

4. Select the next set (*a, x, y*) in *NewSubSets.*
5. Apply the Relinking Method to produce the sequence from *a* to *x* and *y*.
6. Apply the Improvement Method to the best solution in the sequence. Let *w* be the improved solution. Add *w* to *Pool*.
7. Delete (*a, x, y*)from *NewSubsets*

end while

for (each solution $w \in Pool$)

8. Let x^w be the closest solution to *w* in *RefSet*

if ($f(w) < f(x^1)$ *or* ($f(w) < f(x^b)$ & $d(w, x^w)$>*dthresh*) **then**

9. Make $x^w = w$ and reorder *RefSet*
10. Make *NewSolutions* = TRUE

end if

end for

11. *GlobalIter* = *GlobalIter* +1

If (*GlobalIter* = *MaxIter* **or** *NewSolutions*= FALSE**)**

12. Rebuild the *RefSet*. *GlobalIter* =0

end while

Figure 6. Evolutionary Path Relinking procedure

When no new solution in *Pool* qualifies to enter the *RefSet*, or *GlobalIter* reaches the maximum value, *MaxIter*, we invoke in step 12 of the algorithm the rebuilding of the *RefSet*. It basically consists of resorting again to the set of solutions *D* initially generated with the Taguchi strategy. In the initial construction of the *RefSet* we used the best b solutions in *D*. Now we continue exploring the solutions in *D* ordered according to their quality. In particular, we consider the solution a in position $b + 1$ and directly subject it to the PR algorithm, which is applied between a and two solutions x and y selected from *RefSet* (Resende et al. 2010). The selection is probabilistically made according to the value of the solutions. The improvement method is applied to the output of PR, but now, the resulting solution is directly tested for inclusion in *RefSet* (we apply here the same criterion formulated in the If-statement between steps 8 and 9 in Figure 6). If succeeds, it replaces a solution in the *RefSet* and can be used as guiding solution in later applications of PR. We repeat this process b times; i.e. we consider b solutions in *D* and check whether their associated relinked+improved solution qualifies to become part of the *RefSet*. Then the search continues evolving the *RefSet* until the maximum number of evaluations is reached.

GRASP with evolutionary path relinking (EvoPR) and scatter search (SS) are evolutionary methods based on evolving a small set of selected solutions (elite set in the former and reference set in the latter). We can therefore observe similarities between them, as pointed out in Resende et al. (2010). In some implementations of SS, GRASP is used to populate the reference set, but note that other constructive methods can be used as well. Similarly, PR can be used to combine solutions in SS, but we can use any other combination method (Laguna and Martí 2003). From an algorithmic point of view, we may find two main differences between these methods. The first one is that in SS we do not apply PR to the solutions obtained with GRASP (as we do in GRASP with EvoPR), but rather, we only apply PR as a combination method between solutions already in the reference set. The second difference is that in SS when none of the new solutions obtained with combinations are admitted to the reference set, it is rebuilt, removing some of its solutions, as specified in the reference set update method. In GRASP with EvPR we do not remove solutions from *RefSet*, but rather, we again apply GRASP and use the same rules for inclusion in the *RefSet*.

4. Computational Experiments

This section describes the computational experiments that we performed to test the efficiency of our PR procedures as well as to compare them with the previous methods identified to be the state-of-the-art for unconstrained global optimization. We implement the methods in Java SE6 and run the algorithms on a Pentium 4 computer at 3GHz with 6GB of RAM. We have employed 11 simple scalable functions, called F1 to F11, and 8 hybrid composition functions, called F12 to F19. Herrera et al. (2010a) describe in detail these 19 functions, all of them with optimum known. Figure 7 contains the mathematical expressions of the simple functions (we can see how they are biased to make them even harder to solve). The hybrid composition functions are obtained by combining them.

Function	Name	Definition
F_1	Shif. Sphere	$\sum_{i=1}^{D} z_i^2 + f\_bias,\ z = x - o$
F_2	Shif. Schwefel 2.21	$max_i\{\lvert z_i\rvert, 1 \le i \le D\} + f\_bias,\ z = x - o$
F_3	Shif. Rosenbrock	$\sum_{i=1}^{D-1} (100(z_i^2 + z_{i+1})^2 + (z_i - 1)^2) + f\_bias,\ z = x - o$
F_4	Shif. Rastrigin	$\sum_{i=1}^{D} (z_i^2 - 10\cos(2\pi z_i) + 10) + f\_bias,\ z = x - o$
F_5	Shif. Griewank	$\sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos(\frac{z_i}{\sqrt{i}}) + 1 + f\_bias,\ z = x - o$
F_6	Shif. Ackley	$-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi z_i))$ $+20 + e + f\_bias$
F_7	Shif. Schwefel 2.22	$\sum_{i=1}^{D} \lvert z_i\rvert + \prod_{i=1}^{D} \lvert z_i\rvert,\ z = x - o$
F_8	Shif. Schwefel. 1.2	$\sum_{i=1}^{D} (\sum_{j=1}^{i} z_j)^2,\ z = x - o$
F_9	Shif. Extended f_{10}	$\left(\sum_{i=1}^{D-1} f_{10}(z_i, z_{i+1})\right) + f_{10}(z_D, z_1),\ z = x - o$ $f_{10} = (x^2 + y^2)^{0.25} \cdot \left(\sin^2(50 \cdot (x^2 + y^2)^{0.1}) + 1\right)$
F_{10}	Shif. Bohachevsky	$\sum_{i=1}^{D} \left(z_i^2 + 2z_{i+1}^2 - 0.3\cos(3\pi z_i) - 0.4\cos(4\pi z_{i+1}) + 0.7\right),\ z = x - o$
F_{11}	Shif. Schaffer	$\sum_{i=1}^{D-1} (z_i^2 + z_{i+1}^2)^{0.25} \left(\sin^2(50 \cdot (z_i^2 + z_{i+1}^2)^{0.1}) + 1\right),\ z = x - o$

Figure 7. Simple scalable functions

In all the experiments we report the error with respect to the optimum. In mathematical terms, given a solution x the error is defined as $f(x) - f(op)$, where op is the optimum of the function. We now describe the preliminary experimentation to set the values of the key search parameters as well as to test the different elements of our path relinking methods. After that, we compare our final method with the state-of-the-art procedures for global optimization.

4.1 Construction and Local Search

In the first preliminary experiment, we test the Taguchi constructive method. Specifically, we compare it with the constructive method in Duarte et al. (2010), called Frequency. We perform 1000 constructions with each method and report the best solution value obtained overall. We employ three functions in this experiment, F3, F8, and F13, each one with dimensions 50, 100, 200, 500, and 1000, thus totalizing 15 instances. Table 2 reports the average error achieved with each constructive method over the three instances for each dimension value.

	50	100	200	500	1000
Frequency	5.34E+10	1.13E+11	2.71E+11	7.32E+11	1.45E+12
Taguchi	3.74E+10	6.03E+10	1.33E+11	3.65E+11	7.63E+11

Table 2. Constructive methods

Table 2 clearly shows that the Taguchi-based approach results in lower error values than the Frequency method. In particular, its average error value is approximately half of value obtained with the Frequency constructive method. We compare both methods with two well-known nonparametric tests for pairwise comparisons: the Wilcoxon test and the Sign test. The former one answers the question: Do the two samples (solutions obtained with Frequency and Taguchi in our case) represent two different populations? The resulting p-value of 0.001 indicates that the values compared come from different methods. On the other hand, the Sign test computes the number of instances on which an algorithm supersedes another one. The resulting p-value of 0.000 indicates that there is a clear winner between both methods. We therefore consider the Taguchi based method as the constructive procedure of our final algorithm.

In our second preliminary experiment we test several state-of-the-art local search methods to improve the solutions obtained with the Taguchi procedure, including our two stage line search described in Section 2.2. According to Hvattum et al. (2010), some of the best local search methods in global optimization are the following three procedures: Compass search (Kolda et al. 2003), Solis and Wets (1981) and Tabu line search (Duarte et al., 2010). Since our two stage line search includes the application of the well-known Simplex method for unconstrained global optimization, we also apply this method as a post-processing of these three procedures in order to report a fair comparison. We run these methods to solve functions F3, F13 and F17 and report the average error obtained in these three instances per dimension.

	50	100	200	500	1000
Compass search	8.22E+13	2.82E+14	7.99E+14	1.69E+15	2.07E+16
Solis and Wets	3.19E+10	6.17E+10	1.44E+11	1.91E+11	8.44E+11
Tabu line search	1.70E+09	5.15E+09	1.28E+10	2.28E+11	9.68E+10
Two stage line search	1.76E+03	4.06E+03	1.17E+04	2.15E+10	4.60E+04

Table 3. Local search methods

Results in Table 3 clearly show that our two stage line search method consistently outperforms the other three local search methods tested, when improving the solutions generated with our constructive method. We apply a Friedman test for paired samples to the data used to generate this table. The resulting *p*-value of 0.000 obtained in this experiment clearly indicates that there are statistically significant differences among the six 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 according to the average rank values computed with this test. According to this, the best method is the Two-stage line search (with a rank value of 1.13), followed by the Tabu line search (2.47), while the Compass search and the Solis and Wets methods rank in lower positions (with 3.00 and 3.40 rank values respectively). We therefore consider the Two-stage line search as the improvement method of the path relinking and evolutionary path relinking procedures in the next experiments.

4.2 Path Relinking Elements

In this subsection we discuss and compare the parameters and elements of the path relinking variants. Specifically we first compare the straight linking and the orthogonal linking, each one with three different values of the k search parameter, and then we compare the path relinking with the evolutionary path relinking algorithm run with three different sizes of the *RefSet*.

The third preliminary experiment tests the effect of the path relinking approaches presented in Section 2.3. Specifically, we compare the straight linking and the orthogonal linking, each one with the parameter $k \in \{2, 3, 4\}$ in the algorithm outlined in Figure 2. Table 4 summarizes the average error results obtained over the functions F3, F13 and F17.

		50	**100**	**200**	**500**	**1000**	**Average**
Straight	2	264.55	1487.97	4204.76	12650.73	25526.22	8826.85
	3	198.78	1077.50	3978.56	12216.51	24426.14	8379.50
	4	250.02	1102.69	4193.04	12612.82	23886.72	8409.06
Orthogonal	2	166.53	1479.34	6229.81	17213.48	36570.62	12331.96
	3	349.65	1286.24	5561.30	18508.22	37495.35	12640.15
	4	204.43	1239.76	5102.37	17865.76	35661.70	12014.80

Table 4. Relinking methods

Results in Table 4 indicate that the straight linking is, in general, a better approach than the orthogonal linking in the context of global optimization. Moreover, the best value for the search parameter in the straight linking turns out to be $k = 3$. The Friedman test obtains a p-value of 0.00 and the associated ranking is: “Straight with $k = 3$” (with a rank value of 2.13), “Straight with $k = 4$” (2.33), “Straight with $k = 2$” (3.40), “Orthogonal with $k = 4$” (4.07), “Orthogonal with $k = 2$“ (4.53), and “Orthogonal with $k = 3$“ (4.53). Thus, we select the straight linking with $k = 3$ as the relinking method in both, the path relinking and the evolutionary path relinking procedures.

In our last preliminary experiment we compare the path relinking method (outlined in Figure 2) with the evolutionary path relinking method (outlined in Figure 6). In the path relinking method, the *RefSet* size, b, is set to 10; while in the evolutionary path relinking, we test three different values of the *RefSet* size, 4, 8, and 12. Table 5 reports the average error results obtained over the functions F3, F13 and F17. Column eight in Table 5 reports the average value of columns 3 to 7.

	|*RefSet*|	**50**	**100**	**200**	**500**	**1000**	**Average**
PR	10	61.71	128.78	514.13	2286.31	5751.37	1748.46
EvoPR	4	75.61	183.97	484.40	1614.59	3920.29	1255.77
	8	52.20	252.96	978.13	2716.49	6656.13	2131.18
	12	79.72	611.52	1697.12	4298.49	8908.66	3119.10

Table 5. Path relinking and Evolutionary path relinking

Table 5 reports an interesting result. The path relinking algorithm improves the evolutionary path relinking algorithm when the size of the reference set is relatively large ($b = 8, 12$); however, with a low *RefSet* value, $b = 4$, the evolutionary path relinking is able to produce the best results (its average error value of 1255.77 compares favorably with the 1748.46 of the path relinking). Moreover the Friedman test shows a p-value of 0.00 and the ranking EvoPR-4 (1.50), PR (1.67), EvoPR-8 (2.83), and EvoPR-12 (4.00). We therefore consider EvoPR with $b = 4$ as our best algorithm and compare it with the state-of-the-art methods in the next experiment.

In the final experiment we consider our evolutionary algorithm, called EvoPR, run for 20 global iterations (*MaxIter*=20). For an aggressive search of the solution space, the size of the grid h, is multiplied by 0.01 after each global iteration. As mentioned in the introduction, we compare our method with four algorithms identified in previous studies as the state-of-the art methods: STS (Duarte et al. 2010), DE (Storn and Price 1997), G-CMA-ES (Auger and Hansen 2005), and CHC (Eshelman and Schaffer 1993).

4.3 Comparison with Previous Methods

Following the guidelines in Herrera et al. (2010a) and Herrera and Lozano (2009) we run our final experiment with the following requirements:

- Each algorithm is run 25 times for each test function.
- All the methods stop when the maximum number of evaluations reaches $5000n$, where n is the problem dimension.

We report the results of this experiment in four tables where averages across the 19 functions are reported. Specifically, Table 6 reports, for each method and each dimension, the average error of the best solution found in the 25 runs. Tables 7 and 8 report respectively the maximum and minimum error achieved in the 25 runs. Finally, Table 9 reports the median of the error achieved in the 25 runs. We complement this information with Table 10 where the number of optima that each method is able to match is reported, and the tables in the Appendix with the individual results for each dimension and function.

	DE	CHC	G-CMA-ES	STS	EvoPR
50	1.74E+001	1.76E+005	1.01E+002	6.92E+001	2.41E+001
100	5.44E+001	3.70E+005	2.29E+002	5.02E+002	1.62E+002
200	3.64E+002	1.37E+006	5.87E+002	2.26E+003	7.45E+002
500	3.39E+003	1.80E+006	1.57E+261	1.38E+004	3.82E+003
1000	1.33E+004	4.60E+006	-	6.27E+004	1.16E+004
Average	3.43E+03	1.67E+06	3.91E+260	1.59E+04	3.27E+03

Table 6. Average error over the 25 runs

	DE	CHC	G-CMA-ES	STS	EvoPR
50	1.70E+01	2.60E+01	7.68E+01	2.89E+001	1.22E+01
100	4.89E+01	1.09E+02	1.86E+02	3.10E+002	1.15E+02
200	3.26E+02	4.69E+02	4.49E+02	1.71E+003	6.02E+02
500	3.08E+03	3.70E+03	6.93E+215	1.04E+004	3.45E+03
1000	1.26E+04	1.51E+04	-	5.20E+004	1.06E+04
Average	3.21E+03	3.87E+03	1.73E+215	1.29E+04	2.96E+03

Table 7. Minimum error over the 25 runs

	DE	CHC	G-CMA-ES	STS	EvoPR
50	1.79E+01	3.76E+06	1.27E+02	1.42E+002	6.72E+01
100	6.33E+01	6.87E+06	2.83E+02	7.71E+002	3.45E+02
200	4.25E+02	2.82E+07	7.97E+02	3.11E+003	3.45E+02
500	3.62E+03	1.38E+07	2.18E+262	1.77E+004	4.28E+03
1000	1.40E+04	1.04E+08	-	7.73E+004	1.33E+04
Average	3.62E+03	3.14E+07	5.46E+261	1.98E+04	3.67E+03

Table 8. Maximum error over the 25 runs

	DE	CHC	G-CMA-ES	STS	EvoPR
50	1.74E+01	1.76E+05	9.97E+01	5.31E+001	2.42E+01
100	5.32E+01	3.70E+05	2.27E+02	5.67E+002	1.45E+02
200	3.53E+02	1.37E+06	5.77E+02	2.57E+003	7.97E+02
500	3.40E+03	1.80E+06	3.10E+257	1.61E+004	3.60E+03
1000	1.33E+04	4.60E+06	-	7.11E+004	1.12E+04
Average	3.43E+03	1.67E+06	7.74E+256	1.81E+04	3.14E+03

Table 9. Median error over the 25 runs

	DE	CHC	G-CMA-ES	STS	EvoPR
50	7	2	4	0	8
100	6	0	4	0	8
200	6	0	5	0	6
500	6	0	2	0	5
1000	6	0	-	0	4
Sum	31	2	15	0	31

Table 10. Number of optima over the 25 runs

Tables 6 to 9 show that our EvoPR method consistently produces the best average results, since it is able to obtain lower error values (avg., min., and median) than DE, CHC and G-CMA-ES. Moreover, considering the number of optima shown in Table 10, EvoPR and DE obtain 31 out of the 95 test functions while CHC, G-CMA-ES and STS obtain 2, 15 and 0 respectively.

We now focus on the average error values, since they are the most informative in statistical terms, and apply a Friedman test for paired samples to the data used to generate Table 6. The resulting *p*-value of 0.000 obtained in this experiment clearly indicates that there are statistically significant differences among the five methods tested. According to the post-test analysis, the ranking of the methods under comparison is: DE (1.39), EvoPR (2.45), STS (3.37), G-CMA-ES (3.80) and CHC (4.00). If we consider now the average of the minimum errors reported in Table 7 and apply the Friedman test for paired samples to the data used to generate this table, we obtain the following rank: DE (1.88), EvoPR (2.34), G-CMA-ES (3.11), CHC (3.68), and STS (3.99). The associated *p*-value in this test is 0.00 indicating, as above, that there are statistically significant differences among the six methods tested.

As a result of the analysis above, it is difficult to establish a clear winner overall. On one hand, our EvoPR method obtains an average and minimum error values overall of 3.27E+03 and 2.96E+03 respectively, which compare favorably with the 3.43E+03 and 3.21E+03 of the DE method. On the other hand, the rank values of the Friedman test favor the DE method with respect to the EvoPR (1.39 and 2.45 for the average errors, and 1.88 and 2.34 for the minimum errors). Finally, both are able to match the same number of optima, equal to 31. We could conclude that both methods, DE and EvoPR, obtain the best results overall. G-CMA-ES also obtains good solutions considering the difficulty of the instances tested. Finally, CHC and STS are not able to produce quality solutions in this kind of large and difficult instances.

We finish our experimentation with the analysis of the number of evaluations consumed by the improvement method. Table 11 reports the number of times that the improvement method is invoked, Imp. Method calls, the average number of evaluations consumed by the line searches and the simplex on a run of the improvement method, Line searches eval., and Simplex eval. respectively. Finally, the last column in Table 11 reports the percentage of the total number of evaluations consumed by the Improvement Method, %Imp Method eval. This table confirms what is well known about heuristic algorithms: the improvement method is a key element and justifies a relatively large percentage of the total running time (number of evaluations in our case).

	Imp. Method calls	Line searches eval.	Simplex eval.	%Imp Method eval.
50	125.7	777.5	566.6	67.6%
100	139.2	1601.5	616.8	61.7%
200	143.0	3304.3	704.5	57.3%
500	144.8	8476.6	970.4	54.7%
1000	151.9	16765.3	1425.5	55.3%
Average	140.9	6185.0	856.8	59.3%

Table 11. Number of optima over the 25 runs

5. Conclusions

We have described the development and implementation of path relinking (PR) for the optimization of large scale unconstrained functions. 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 high quality solutions on previously reported problems. These strategies include two different methods to perform the relinking of solutions, which can be applied to different types of problems. Our extensive comparison with previous methods shows that the PR method is very competitive for unconstrained global optimization problems.

Acknowledgments

This research has been partially supported by the *Ministerio de Educación y Ciencia* of Spain (TIN2009-07516). We would like to thank Profs. Glover and Resende for their descriptions and suggestions on the Path Relinking and Evolutionary Path Relinking methodologies respectively.

References

Andrade D.V. and M.G.C. Resende. 2007. *GRASP with evolutionary path-relinking*. In Proc. of Seventh Metaheuristics International Conference (MIC 2007).

Auger. A.. N. Hansen. 2005. A Restart CMA Evolution Strategy With Increasing Population Size. In *Procs. of 2005 IEEE Congress on Evol. Comput*. (CEC'2005). 1769-1776.

Avriel. M. 1976. Nonlinear Programming. Analysis and Methods. Prentice-Hall. Englewood Cliffs. New Jersey.

Duarte. A.. R. Martí. and F. Glover. F. Gortazar. 2010. Hybrid Scatter Tabu Search for Unconstrained Global Optimization. *Annals of Operations Research*. DOI: 10.1007/s10479-009-0596-2.

Eshelman. L. J. and J. D. Schaffer. Real-coded genetic algorithm and interval schemata. Foundation of Genetic Algorithms. pages 187–202. 1993.

Glover. F. and M Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers. Boston.

Glover. F.. 1989. Tabu Search. Part I. *ORSA Journal on Computing* 1. 190-206.

Glover. F.. 1994. Tabu Search for Nonlinear and Parametric Optimization (with Links to Genetic Algorithms). *Discrete Applied Mathematics* 49. 231-255.

Hansen. N.. 2006. Compilation of results on the 2005 CEC benchmark function set. *Technical report*. CoLAB institute of computational sciences ETH. Zurich.

Herrera. F. and M. Lozano. 2009. Workshop on Evolutionary Algorithms and other Metaheuristics for Continuous Optimization Problems - A Scalability Test. http://sci2s.ugr.es/programacion/workshop/Scalability.html.

Herrera. F. M. Lozano and D. Molina. 2010a. Test Suite for the Special Issue of Soft Computing on Scalability of Evolutionary Algorithms and other Metaheuristics for Large Scale Continuous Optimization Problems. Technical report. University of Granada.

Herrera. F. M. Lozano and D. Molina. 2010b. Components and Parameters of DE. Real-coded CHC. and G-CMAES. Technical report. University of Granada.

Hvattum. L.M.. A. Duarte. R. Martí and F. Glover. 2010. The Improvement Method in Scatter Search: An experimental study on global optimization. Technical report. University of Valencia.

Kolda. T.G.. R.M. Lewis and V.J. Torczon. 2003. Optimization by Direct Search: New perspectives on some Classical and Modern Methods. SIAM Review 45. 385-482.

Laguna. M. and Martí. R.. 1999. GRASP and Path Relinking for 2-Layer Straight Line Crossing Minimization. *INFORMS Journal on Computing* 11(1). 44-52.

Laguna. M. and R. Martí. 2003. *Scatter Search. Methodology and Implementations in C*. Kluwer Academic Publishers. Boston.

Laguna. M. and R. Martí. 2005. Experimental Testing of Advanced Scatter Search Designs for Global Optimization of Multimodal Functions. *Journal of Global Optimization* 33. 235-255.

Martí. R.. F. Glover and M. Laguna. 2006. Principles of Scatter Search. *European Journal of Operational Research* 169. pp. 359-372.

Resende. M.G.C. and C.C. Ribeiro. 2003. Greedy randomized adaptive search procedures. In F. Glover and G. Kochenberger. editors. *Handbook of Metaheuristics*. 219–250. Kluwer Academic Publishers.

Resende, M., R. Martí, M. Gallego and A. Duarte. 2010. GRASP and PR for the Max-min diversity problem, Computers and Operations Research 37, 498-508.

Resende. M.G.C. and R.F. Werneck. 2004. A hybrid heuristic for the p-median problem. *Journal of Heuristics* 10. 59–88.

Roy. R.K. (1990) A Primer on the Taguchi Method. Van Nostrand Reinhold. New York.

Solis. F.J. and R.J.-B. Wets. 1981. Minimization by random search techniques. Mathematical Operations Research 6. 19-30.

Storn. R. and K. Price. 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 11:341–359.

Suganthan. P.N.. N. Hansen. J.J. Liang. K. Deb. Y.-P. Chen. A. Auger. S. Tiwari. 2005. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *Technical report*. Nanyang technological university of Singapore.

Appendix

		F1				F2				F3				F4			
		Avg	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**
	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.60E-01	2.56E-01	8.49E-01	3.29E-01	2.89E+01	2.55E+01	3.10E+01	2.90E+01	3.98E-02	0.00E+00	9.95E-01	1.51E-13
	CHC	1.67E-11	1.23E-11	2.33E-11	1.67E-11	6.19E+01	5.13E+01	8.43E+01	6.19E+01	1.25E+06	9.74E-01	2.01E7	1.25E+06	7.43E+01	5.47E+01	1.00E+02	7.43E+01
50	G-CMA-ES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.75E-11	2.08E-11	3.82E-11	2.64E-11	7.97E-01	0.00E+00	3.99E+00	0.00E+00	1.05E+02	7.16E+01	1.33E+02	1.08E+02
	EvoPR	1.22E-02	0.00E+00	1.53E-01	0.00E+00	3.71E-01	9.27E-02	7.80E-01	2.56E-01	1.12E+02	4.46E+01	5.06E+02	1.24E+02	4.96E-02	0.00E+00	9.95E-01	0.00E+00
	STS	2.15E-01	6.99E-02	5.51E-01	1.73E-01	4.52E+01	3.46E+01	5.84E+01	3.88E+01	1.84E+02	4.85E+01	4.24E+02	1.49E+02	3.80E+01	2.34E+01	6.01E+01	3.06E+01
	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.45E+00	3.82E+00	5.59E+00	4.34E+00	8.01E+01	7.55E+01	1.25E+02	7.81E+01	7.96E-02	1.37E-13	9.95E-01	4.23E-13
	CHC	3.56E-11	2.64E-11	4.8E-11	3.56E-11	8.58E+01	7.30E+01	9.74E+01	8.58E+01	4.19E+06	9.12E+01	7.26E7	4.19E+06	2.19E+02	1.64E+02	2.93E+02	2.19E+02
100	G-CMA-ES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.51E-10	7.71E-11	2.72E-10	1.62E-10	3.88E+00	0.00E+00	1.84E+01	2.27E+00	2.50E+02	1.86E+02	3.50E+02	2.50E+02
	EvoPR	4.34E-02	0.00E+00	2.97E-01	0.00E+00	3.30E+00	1.97E+00	4.68E+00	3.19E+00	3.98E+02	9.55E+01	2.42E+03	2.32E+02	1.07E-01	0.00E+00	1.14E+00	5.04E-02
	STS	6.31E-01	2.29E-01	9.66E-01	9.17E-01	5.97E+01	4.82E+01	6.95E+01	4.84E+01	5.21E+02	2.24E+02	1.69E+03	5.06E+02	1.10E+02	5.84E+01	1.75E+02	1.23E+02
	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.92E+01	1.74E+01	2.10E+01	1.93E+01	1.78E+02	1.74E+02	2.27E+02	1.77E+02	1.27E-01	7.44E-13	9.95E-01	3.58E-12
	CHC	8.34E-01	1.36E-11	2.09E+01	8.34E-01	1.03E+02	9.34E+01	1.15E+02	1.03E+02	2.01E7	2.07E+02	4.31E8	2.01E7	5.40E+02	4.02E+02	7.72E+02	5.40E+02
200	G-CMA-ES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.16E-9	4.9E-10	5.79E-9	9.91E-10	8.91E+01	0.00E+00	1.19E+02	8.95E+01	6.48E+02	0.00E+00	8.31E+02	6.68E+02
	EvoPR	8.03E-02	0.00E+00	2.97E-01	3.0E-5	8.03E+00	6.29E+00	4.68E+00	7.50E+00	2.91E+02	1.92E+02	2.42E+03	1.94E+02	3.52E-01	0.00E+00	1.14E+00	1.08E+00
	STS	4.51E+00	3.08E+00	6.42E+00	3.70E+00	7.35E+01	6.76E+01	8.18E+01	7.98E+01	2.17E+03	1.49E+03	6.72E+03	1.72E+03	3.19E+02	1.23E+02	4.60E+02	4.04E+02
	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.35E+01	5.13E+01	5.59E+01	5.33E+01	4.76E+02	4.70E+02	5.22E+02	4.74E+02	3.20E-01	4.64E-12	2.25E+00	9.22E-03
	CHC	2.84E-12	1.93E-12	4.38E-12	2.84E-12	1.29E+02	1.16E+02	1.41E+02	1.29E+02	1.14E+06	4.94E+02	2.85E7	1.14E+06	1.91E+03	1.46E+03	2.22E+03	1.91E+03
500	G-CMA-ES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.48E-4	1.52E-4	5.7E-4	3.31E-4	3.58E+02	2.50E+02	8.31E+02	3.55E+02	2.10E+03	1.88E+03	2.31E+03	2.07E+03
	EvoPR	0.00E+00	0.00E+00	1.0E-5	0.00E+00	2.04E+01	1.66E+01	2.28E+01	2.19E+01	5.97E+02	4.91E+02	1.17E+03	6.22E+02	1.45E+00	0.00E+00	5.97E+00	0.00E+00
	STS	2.85E+01	2.34E+01	3.49E+01	2.47E+01	8.50E+01	8.01E+01	9.02E+01	9.02E+01	7.82E+03	5.45E+03	1.44E+04	7.78E+03	7.57E+02	5.57E+02	1.11E+03	6.66E+02
	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.46E+01	8.22E+01	8.65E+01	8.44E+01	9.69E+02	9.66E+02	9.71E+02	9.69E+02	1.44E+00	2.76E-11	4.69E+00	1.32E+00
	CHC	1.36E-11	7.56E-12	2.33E-11	1.36E-11	1.44E+02	1.38E+02	1.57E+02	1.44E+02	8.75E+03	1.22E+03	1.80E+05	8.75E+03	4.76E+03	4.13E+03	5.36E+03	4.76E+03
1000	G-CMA-ES																
	EvoPR	4.0E-5	0.00E+00	3.4E-4	0.00E+00	3.21E+01	3.08E+01	3.41E+01	3.16E+01	1.12E+03	1.03E+03	1.22E+03	1.09E+03	4.08E+02	1.99E+00	1.21E+03	1.21E+03
	STS	6.42E+01	5.25E+01	7.66E+01	7.04E+01	9.05E+01	8.76E+01	9.46E+01	9.02E+01	1.51E+04	1.23E+04	1.83E+04	1.55E+04	1.55E+03	7.86E+02	2.40E+03	2.22E+03

Table 12.Results for functions F1-F4

		F5				F6				F7				F8			
		Avg	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**
50	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.43E-13	1.14E-13	1.71E-13	1.42E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.44E+00	1.89E+00	4.62E+00	3.54E+00
	CHC	1.67E-03	9.92E-12	2.21E-02	1.67E-03	6.15E-7	4.72E-7	7.33E-7	6.15E-7	2.66E-9	4.58E-10	9.92E-9	2.66E-9	2.24E+02	3.19E+01	6.27E+02	2.24E+02
	G-CMA-ES	2.96E-4	0.00E+00	7.40E-03	0.00E+00	2.09E+01	2.00E+01	2.12E+01	2.11E+01	1.01E-10	6.16E-11	2.32E-10	7.67E-11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	EvoPR	5.13E-02	0.00E+00	1.56E-01	4.74E-02	6.85E-03	0.00E+00	8.59E-02	0.00E+00	2.63E-02	0.00E+00	1.73E-01	1.0E-5	2.08E+02	1.43E+02	2.94E+02	2.09E+02
	STS	1.01E+00	9.13E-01	1.05E+00	1.02E+00	1.16E-01	6.97E-02	2.03E-01	1.71E-01	1.56E-01	1.26E-01	2.09E-01	1.44E-01	7.90E+02	3.20E+02	1.56E+03	5.20E+02
100	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.1E-13	2.84E-13	3.41E-13	3.13E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.69E+02	2.86E+02	4.84E+02	3.47E+02
	CHC	3.83E-03	7.16E-12	4.16E-02	3.83E-03	4.1E-7	3.02E-7	5.46E-7	4.1E-7	1.40E-02	2.05E-10	3.50E-01	1.40E-02	1.69E+03	9.32E+02	3.26E+03	1.69E+03
	G-CMA-ES	1.58E-03	0.00E+00	1.48E-02	0.00E+00	2.12E+01	2.00E+01	2.14E+01	2.13E+01	4.22E-4	2.78E-9	9.41E-03	6.98E-7	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	EvoPR	3.92E-02	0.00E+00	1.06E-01	1.06E-01	2.5E-4	0.00E+00	7.2E-4	0.00E+00	9.17E-02	0.00E+00	4.76E-01	0.00E+00	2.27E+03	1.95E+03	2.78E+03	2.34E+03
	STS	1.07E+00	9.99E-01	1.11E+00	1.07E+00	6.05E-01	4.50E-01	9.19E-01	5.58E-01	7.28E-01	5.75E-01	8.61E-01	6.53E-01	7.99E+03	5.15E+03	1.09E+04	9.27E+03
200	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.54E-13	5.97E-13	7.11E-13	6.54E-13	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.53E+03	4.82E+03	6.63E+03	5.33E+03
	CHC	8.76E-03	4.41E-12	4.67E-02	8.76E-03	1.23E+00	1.3E-7	4.41E+00	1.23E+00	2.59E-01	1.2E-9	2.65E+00	2.59E-01	9.38E+03	5.91E+03	1.49E+04	9.38E+03
	G-CMA-ES	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.14E+01	2.14E+01	2.15E+01	2.14E+01	1.17E-01	4.62E-5	7.85E-01	2.61E-02	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	EvoPR	2.68E-02	0.00E+00	1.06E-01	2.99E-02	6.22E-01	0.00E+00	7.2E-4	2.33E+00	3.82E-02	0.00E+00	4.76E-01	3.0E-5	1.34E+04	1.09E+04	2.78E+03	1.45E+04
	STS	1.12E+00	1.07E+00	1.19E+00	1.11E+00	5.67E-01	4.66E-01	6.69E-01	5.36E-01	3.03E+00	2.53E+00	3.93E+00	3.02E+00	3.77E+04	2.90E+04	4.75E+04	4.40E+04
500	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.65E-12	1.59E-12	1.71E-12	1.65E-12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.09E+04	5.51E+04	6.51E+04	6.11E+04
	CHC	6.98E-03	8.53E-14	4.42E-02	6.98E-03	5.16E+00	2.83E+00	8.05E+00	5.16E+00	1.27E-01	7.76E-9	1.66E+00	1.27E-01	7.22E+04	6.09E+04	8.86E+04	7.22E+04
	G-CMA-ES	2.96E-4	0.00E+00	7.40E-03	0.00E+00	2.15E+01	2.15E+01	2.16E+01	2.15E+01					2.36E-6	7.68E-7	3.91E-6	2.31E-6
	EvoPR	3.03E-02	0.00E+00	1.33E-01	9.86E-03	1.21E+00	0.00E+00	2.62E+00	1.0E-5	8.06E-03	0.00E+00	2.60E-02	2.51E-02	7.05E+04	6.40E+04	7.75E+04	6.61E+04
	STS	1.28E+00	1.21E+00	1.33E+00	1.31E+00	5.28E-01	4.62E-01	6.18E-01	4.91E-01	9.54E+00	8.73E+00	1.04E+01	9.07E+00	2.45E+05	1.84E+05	3.05E+05	2.90E+05
1000	DE	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.29E-12	3.18E-12	3.41E-12	3.3E-12	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.46E+05	2.31E+05	2.58E+05	2.46E+05
	CHC	7.02E-03	1.14E-13	3.83E-02	7.02E-03	1.38E+01	1.03E+01	1.63E+01	1.38E+01	3.52E-01	2.28E-7	2.90E+00	3.52E-01	3.11E+05	2.61E+05	3.43E+05	3.11E+05
	G-CMA-ES																
	EvoPR	3.72E-02	0.00E+00	2.00E-01	1.97E-02	1.97E+00	0.00E+00	2.68E+00	2.45E+00	1.5E-4	0.00E+00	3.21E-03	0.00E+00	2.15E+05	1.98E+05	2.45E+05	2.06E+05
	STS	1.60E+00	1.53E+00	1.72E+00	1.60E+00	5.57E-01	4.88E-01	6.57E-01	5.73E-01					1.09E+06	9.07E+05	1.34E+06	1.24E+06

Table 13. Results for functions F5-F8

		F9				F10				F11				F12			
		Avg	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**
	DE	2.73E+02	2.72E+02	2.74E+02	2.73E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.23E-5	3.35E-5	1.04E-4	5.6E-5	5.35E-13	2.72E-13	7.19E-13	5.27E-13
	CHC	3.10E+02	2.99E+02	3.20E+02	3.10E+02	7.30E+00	4.62E-11	1.62E+01	7.30E+00	2.16E+00	3.68E-4	1.34E+01	2.16E+00	9.57E-01	6.04E-11	2.39E+01	9.57E-01
50	G-CMA-ES	1.66E+01	4.38E+00	3.36E+01	1.61E+01	6.81E+00	2.10E+00	1.26E+01	6.71E+00	3.01E+01	7.83E+00	6.94E+01	2.83E+01	1.88E+02	1.15E+02	2.51E+02	1.87E+02
	EvoPR	8.02E+00	2.78E+00	1.81E+01	9.69E+00	4.80E-02	0.00E+00	8.45E-01	0.00E+00	9.68E+00	1.54E+00	2.60E+01	1.18E+01	2.27E+00	4.19E-02	8.26E+00	1.02E-01
	STS	3.07E+01	2.48E+01	3.93E+01	3.10E+01	3.29E-03	1.17E-03	9.60E-03	7.35E-03	3.24E+01	2.84E+01	3.70E+01	3.45E+01	1.02E+00	3.54E-01	3.14E+00	8.44E-01
	DE	5.06E+02	5.04E+02	5.07E+02	5.06E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.28E-4	7.89E-5	1.7E-4	1.29E-4	5.99E-11	3.42E-11	8.27E-11	6.18E-11
	CHC	5.86E+02	5.69E+02	5.99E+02	5.86E+02	3.30E+01	1.42E+01	9.61E+01	3.30E+01	7.32E+01	1.64E+00	1.55E+02	7.32E+01	1.03E+01	6.43E-11	5.19E+01	1.03E+01
100	G-CMA-ES	1.02E+02	4.31E+01	1.56E+02	1.06E+02	1.66E+01	9.28E+00	2.46E+01	1.68E+01	1.64E+02	8.07E+01	2.60E+02	1.51E+02	4.17E+02	3.46E+02	4.77E+02	4.20E+02
	EvoPR	2.91E+01	1.57E+01	4.53E+01	1.77E+01	2.05E-01	0.00E+00	1.44E+00	2.84E-01	2.60E+01	9.96E+00	3.77E+01	2.98E+01	5.01E+00	3.99E-01	2.07E+01	1.18E+00
	STS	7.73E+01	6.48E+01	9.41E+01	7.86E+01	1.75E-01	9.63E-02	2.51E-01	2.47E-01	7.34E+01	5.69E+01	8.52E+01	6.66E+01	4.89E+00	2.80E+00	7.23E+00	3.73E+00
	DE	1.01E+03	1.01E+03	1.01E+03	1.01E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.62E-4	2.26E-4	3.1E-4	2.59E-4	9.76E-10	6.65E-10	1.49E-9	9.36E-10
	CHC	1.19E+03	1.16E+03	1.22E+03	1.19E+03	7.13E+01	3.57E+01	1.42E+02	7.13E+01	3.85E+02	1.19E+02	5.96E+02	3.85E+02	7.44E+01	1.60E+01	1.59E+02	7.44E+01
200	G-CMA-ES	3.75E+02	2.95E+02	4.92E+02	3.81E+02	4.43E+01	3.04E+01	5.88E+01	4.41E+01	8.03E+02	6.37E+02	1.04E+03	7.93E+02	9.06E+02	8.30E+02	1.01E+03	9.08E+02
	EvoPR	6.22E+01	4.31E+01	4.53E+01	5.11E+01	1.04E+00	3.0E-5	1.44E+00	4.78E-01	5.93E+01	4.72E+01	3.77E+01	6.78E+01	1.00E+01	1.76E+00	2.07E+01	3.87E+00
	STS	1.63E+02	1.46E+02	1.78E+02	1.60E+02	4.23E+00	3.12E+00	5.99E+00	5.28E+00	1.65E+02	1.52E+02	1.87E+02	1.58E+02	1.95E+01	1.20E+01	3.33E+01	2.93E+01
	DE	2.52E+03	2.52E+03	2.53E+03	2.52E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.76E-4	6.13E-4	7.83E-4	6.71E-4	7.07E-9	5.95E-9	9.29E-9	6.98E-9
	CHC	3.00E+03	2.97E+03	3.03E+03	3.00E+03	1.86E+02	1.08E+02	5.18E+02	1.86E+02	1.81E+03	1.50E+03	2.47E+03	1.81E+03	4.48E+02	3.63E+02	5.52E+02	4.48E+02
500	G-CMA-ES	1.74E+03	1.58E+03	1.85E+03	1.76E+03	1.27E+02	1.03E+02	1.55E+02	1.27E+02	4.16E+03	3.50E+03	4.54E+03	4.18E+03	2.58E+03	2.41E+03	2.76E+03	2.59E+03
	EvoPR	1.75E+02	1.41E+02	2.41E+02	1.41E+02	3.29E+01	1.05E+01	6.63E+01	1.05E+01	1.77E+02	1.51E+02	2.48E+02	2.11E+02	1.73E+01	7.13E+00	3.67E+01	1.13E+01
	STS	4.44E+02	4.32E+02	4.69E+02	4.36E+02	2.75E+01	2.39E+01	3.12E+01	2.60E+01	4.43E+02	4.17E+02	4.81E+02	4.35E+02	7.23E+01	5.76E+01	8.93E+01	8.93E+01
	DE	5.13E+03	5.12E+03	5.14E+03	5.13E+03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.35E-03	1.25E-03	1.48E-03	1.35E-03	1.68E-8	1.4E-8	1.94E-8	1.7E-8
	CHC	6.11E+03	6.06E+03	6.16E+03	6.11E+03	3.83E+02	2.15E+02	7.40E+02	3.83E+02	4.82E+03	4.42E+03	5.42E+03	4.82E+03	1.05E+03	8.80E+02	1.21E+03	1.05E+03
1000	G-CMA-ES																
	EvoPR	4.07E+02	3.26E+02	5.68E+02	3.66E+02	3.86E+02	1.47E+02	4.78E+02	4.25E+02	3.96E+02	3.21E+02	5.86E+02	3.50E+02	3.23E+01	2.24E+01	5.76E+01	3.06E+01
	STS	9.06E+02	8.58E+02	9.37E+02	9.05E+02	5.60E+01	4.51E+01	6.61E+01	5.23E+01	9.15E+02	8.66E+02	9.58E+02	9.27E+02	1.75E+02	1.53E+02	2.01E+02	1.78E+02

Table 14. Results for functions F9-F12

		F13				F14				F15				F16			
		Avg	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**
50	DE	2.45E+01	2.28E+01	2.64E+01	2.44E+01	4.16E-8	1.32E-8	1.8E-7	2.58E-8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.56E-9	9.33E-10	2.68E-9	1.51E-9
	CHC	2.08E+06	1.14E+01	5.08E7	2.08E+06	6.17E+01	3.99E+01	1.48E+02	6.17E+01	3.98E-01	1.11E-8	2.12E+00	3.98E-01	2.95E-9	0.00E+00	9.17E-9	2.95E-9
	G-CMA-ES	1.97E+02	1.36E+02	2.32E+02	1.97E+02	1.09E+02	7.42E+01	1.50E+02	1.05E+02	9.79E-4	1.73E-4	3.85E-03	8.12E-4	4.27E+02	3.19E+02	5.27E+02	4.22E+02
	EvoPR	4.22E+01	3.39E+01	9.41E+01	3.54E+01	9.97E-01	4.48E-02	2.48E+00	2.01E+00	6.38E-02	0.00E+00	8.24E-01	8.24E-01	5.63E+00	2.00E-01	1.51E+01	2.27E+00
	STS	9.16E+01	3.68E+01	2.57E+02	1.15E+02	2.75E+01	1.22E+01	5.51E+01	1.70E+01	6.76E-02	4.84E-02	9.16E-02	7.86E-02	2.03E+00	7.44E-01	5.73E+00	2.46E+00
100	DE	6.17E+01	5.95E+01	6.45E+01	6.17E+01	4.79E-02	5.65E-8	9.95E-01	1.3E-7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.58E-9	2.63E-9	4.52E-9	3.53E-9
	CHC	2.70E+06	2.41E+01	5.61E7	2.70E+06	1.66E+02	1.19E+02	2.24E+02	1.66E+02	8.13E+00	2.42E-8	6.42E+01	8.13E+00	2.23E+01	1.05E+00	6.71E+01	2.23E+01
	G-CMA-ES	4.21E+02	3.48E+02	5.52E+02	4.12E+02	2.55E+02	2.16E+02	3.04E+02	2.52E+02	6.30E-01	2.39E-4	2.51E+00	4.13E-01	8.59E+02	7.47E+02	9.75E+02	8.48E+02
	EvoPR	1.40E+02	7.37E+01	5.09E+02	7.46E+01	1.24E+00	1.36E-01	5.55E+00	1.05E+00	6.56E-02	0.00E+00	3.81E-01	3.81E-01	8.29E+00	2.03E+00	2.29E+01	9.31E+00
	STS	3.19E+02	1.16E+02	6.64E+02	3.52E+02	8.69E+01	5.12E+01	1.64E+02	5.23E+01	4.16E-01	3.44E-01	5.13E-01	4.20E-01	7.61E+00	3.28E+00	1.50E+01	1.07E+01
200	DE	1.36E+02	1.34E+02	1.38E+02	1.36E+02	1.38E-01	1.24E-7	9.95E-01	2.71E-7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.46E-9	5.54E-9	9.6E-9	7.26E-9
	CHC	5.75E+06	1.62E+02	1.01E8	5.75E+06	4.29E+02	3.52E+02	5.14E+02	4.29E+02	2.14E+01	7.42E-8	1.23E+02	2.14E+01	1.60E+02	6.19E+00	2.94E+02	1.60E+02
	G-CMA-ES	9.43E+02	8.02E+02	1.08E+03	9.34E+02	6.09E+02	5.08E+02	7.05E+02	6.24E+02	1.75E+00	4.81E-03	4.92E+00	2.10E+00	1.92E+03	1.66E+03	2.12E+03	1.90E+03
	EvoPR	1.71E+02	1.48E+02	5.09E+02	2.05E+02	3.75E+00	4.59E-01	5.55E+00	6.84E-01	3.80E-01	0.00E+00	3.81E-01	0.00E+00	1.74E+01	4.58E+00	2.29E+01	9.41E+00
	STS	1.37E+03	9.80E+02	2.36E+03	1.52E+03	2.30E+02	1.48E+02	3.46E+02	2.16E+02	3.13E+00	2.73E+00	3.86E+00	3.86E+00	2.89E+01	1.56E+01	4.98E+01	1.56E+01
500	DE	3.59E+02	3.57E+02	3.78E+02	3.58E+02	1.35E-01	5.55E-7	1.12E+00	9.01E-7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.04E-8	1.71E-8	2.3E-8	2.05E-8
	CHC	3.22E7	3.35E+02	2.21E8	3.22E7	1.46E+03	1.15E+03	1.77E+03	1.46E+03	6.01E+01	6.53E+00	2.66E+02	6.01E+01	9.55E+02	2.32E+01	1.11E+03	9.55E+02
	G-CMA-ES	2.87E+03	2.59E+03	3.55E+03	2.87E+03	1.95E+03	1.80E+03	2.15E+03	1.95E+03	2.82E262	1.25E217	3.93E263	5.57E258	5.45E+03	5.23E+03	5.85E+03	5.43E+03
	EvoPR	5.75E+02	4.39E+02	8.26E+02	7.01E+02	9.00E+00	3.28E+00	1.85E+01	6.03E+00	2.25E+00	4.75E-02	9.88E+00	1.32E+00	4.87E+01	2.96E+01	7.69E+01	7.69E+01
	STS	5.24E+03	4.19E+03	8.77E+03	4.39E+03	6.48E+02	4.41E+02	8.82E+02	5.38E+02	1.06E+01	9.59E+00	1.19E+01	1.02E+01	1.27E+02	9.78E+01	1.53E+02	1.28E+02
1000	DE	7.30E+02	7.28E+02	7.31E+02	7.29E+02	6.90E-01	1.3E-6	2.77E+00	9.95E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.18E-8	3.61E-8	4.71E-8	4.19E-8
	CHC	6.66E7	1.32E+03	1.67E9	6.66E7	3.62E+03	3.21E+03	3.98E+03	3.62E+03	8.37E+01	2.83E+01	1.31E+02	8.37E+01	2.32E+03	5.46E+01	2.75E+03	2.32E+03
	G-CMA-ES																
	EvoPR	1.13E+03	8.73E+02	1.54E+03	9.99E+02	4.31E+02	1.06E+01	9.16E+02	5.47E+02	1.26E+02	4.42E+01	2.01E+02	8.12E+01	8.44E+01	5.56E+01	1.46E+02	1.00E+02
	STS	1.15E+04	8.85E+03	2.04E+04	1.01E+04	1.22E+03	8.63E+02	1.62E+03	1.54E+03	2.18E+01	1.88E+01	2.50E+01	2.11E+01	3.13E+02	2.66E+02	3.59E+02	3.03E+02

Table 15. Results for functions F13-F16*

		F17				F18				F19			
		Avg	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**	**Avg**	**Min**	**Max**	**Med**
50	DE	7.98E-01	1.18E-02	2.24E+00	6.83E-01	1.22E-4	6.13E-5	2.36E-4	1.2E-4	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	CHC	2.26E+04	9.55E-01	5.59E+05	2.26E+04	1.58E+01	3.98E+00	2.94E+01	1.58E+01	3.59E+02	0.00E+00	5.26E+03	3.59E+02
	G-CMA-ES	6.89E+02	5.96E+02	8.03E+02	6.71E+02	1.31E+02	1.13E+02	1.59E+02	1.27E+02	4.76E+00	4.13E-01	9.28E+00	4.03E+00
	EvoPR	6.77E+01	5.36E+00	3.03E+02	5.86E+01	1.62E+00	2.04E-01	4.94E+00	4.32E+00	5.03E-02	0.00E+00	1.10E+00	1.10E+00
	STS	6.33E+01	1.42E+01	1.80E+02	6.10E+01	8.87E+00	4.71E+00	1.59E+01	7.09E+00	3.15E-02	1.90E-02	4.85E-02	3.17E-02
100	DE	1.23E+01	1.49E-01	1.47E+01	1.28E+01	2.98E-4	1.98E-4	4.98E-4	2.86E-4	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	CHC	1.47E+05	4.14E+01	1.78E+06	1.47E+05	7.00E+01	3.97E+01	9.51E+01	7.00E+01	5.45E+02	4.20E+00	5.67E+03	5.45E+02
	G-CMA-ES	1.51E+03	1.27E+03	1.74E+03	1.52E+03	3.07E+02	2.62E+02	3.41E+02	3.13E+02	2.02E+01	6.71E+00	1.55E+02	1.47E+01
	EvoPR	1.97E+02	2.84E+01	6.99E+02	4.15E+01	3.34E+00	8.63E-01	9.37E+00	1.29E+00	1.43E-01	0.00E+00	2.52E+00	1.0E-5
	STS	2.69E+02	9.20E+01	7.05E+02	2.26E+02	2.66E+01	1.39E+01	3.91E+01	2.97E+01	2.63E-01	1.64E-01	3.51E-01	2.93E-01
200	DE	3.70E+01	3.49E+01	3.95E+01	3.70E+01	4.73E-4	2.94E-4	6.07E-4	4.7E-4	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	CHC	1.75E+05	2.64E+02	4.37E+06	1.75E+05	2.12E+02	1.63E+02	2.61E+02	2.12E+02	2.06E+03	1.47E+01	2.11E+04	2.06E+03
	G-CMA-ES	3.36E+03	3.07E+03	3.84E+03	3.33E+03	6.89E+02	6.42E+02	7.35E+02	6.88E+02	7.52E+02	3.36E+01	3.08E+03	5.74E+02
	EvoPR	1.56E+02	6.46E+01	6.99E+02	6.81E+01	8.85E+00	3.78E+00	9.37E+00	5.96E+00	2.15E+00	1.73E-03	2.52E+00	3.40E-01
	STS	5.19E+02	3.45E+02	1.16E+03	3.91E+02	7.29E+01	5.56E+01	9.67E+01	6.74E+01	3.72E+00	2.94E+00	4.55E+00	3.48E+00
500	DE	1.11E+02	1.10E+02	1.13E+02	1.11E+02	1.22E-03	9.19E-4	1.74E-03	1.22E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	CHC	8.40E+05	2.96E+02	1.21E7	8.40E+05	7.32E+02	6.32E+02	8.13E+02	7.32E+02	1.76E+03	4.82E+01	1.17E+04	1.76E+03
	G-CMA-ES	9.59E+03	8.60E+03	1.06E+04	9.50E+03	2.05E+03	1.92E+03	2.17E+03	2.06E+03	2.44E+06	3.48E+05	6.00E+06	2.50E+06
	EvoPR	3.94E+02	2.11E+02	8.01E+02	3.68E+02	3.28E+01	1.33E+01	1.72E+02	2.62E+01	5.00E+01	3.03E+01	8.30E+01	4.71E+01
	STS	1.79E+03	1.25E+03	4.13E+03	1.54E+03	2.36E+02	1.77E+02	3.83E+02	2.37E+02	1.29E+01	1.06E+01	1.49E+01	1.26E+01
1000	DE	2.36E+02	2.34E+02	2.51E+02	2.35E+02	2.37E-03	2.03E-03	3.23E-03	2.37E-03	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	CHC	2.04E7	1.91E+03	3.15E8	2.04E7	1.72E+03	1.59E+03	2.02E+03	1.72E+03	4.20E+03	1.19E+02	1.73E+04	4.20E+03
	G-CMA-ES												
	EvoPR	6.75E+02	4.76E+02	9.41E+02	7.86E+02	1.95E+02	1.61E+02	3.16E+02	1.75E+02	2.03E+02	1.53E+02	2.81E+02	1.74E+02
	STS	3.43E+03	2.97E+03	4.54E+03	3.43E+03	5.57E+02	4.26E+02	7.11E+02	4.36E+02	2.62E+01	2.27E+01	3.18E+01	2.63E+01

Table 16. Results for functions F7*-F19*