

# Automatic Tuning of Algorithms Through Sensitivity Minimization

Piero Conca<sup>1</sup>(✉), Giovanni Stracquadanio<sup>2</sup>, and Giuseppe Nicosia<sup>1</sup>

<sup>1</sup> Department of Mathematics and Computer Science,  
University of Catania, Catania, Italy  
{conca,nicosia}@dmi.unict.it

<sup>2</sup> Ludwig Institute for Cancer Research, University of Oxford, Oxford, UK  
giovanni.stracquadanio@ludwig.ox.ac.uk

**Abstract.** Parameters tuning is a crucial step in global optimization. In this work, we present a novel method, the *Sensitive Algorithmic Tuning*, which finds near-optimal parameter configurations through sensitivity minimization. The experimental results highlight the effectiveness and robustness of this novel approach.

## 1 Introduction

An algorithm is a formal description of a series of steps, which provides a solution for a given problem [1]. Obviously, the output of an algorithm depends on the input data and its parameters; an algorithm can be viewed as a black-box procedure, which receives in input a set of data and returns an output representing the results. The possibility of selecting different sets of parameters potentially allows to achieve satisfactory results for different instances of a problem, thus increasing the generality of an algorithm.

Parameters tuning represents a crucial part of any experimental protocol in global optimization; indeed, parameters setting heavily affects the quality of the solutions and the speed of convergence of an optimization algorithm [2, 3]. Finding good parameters values is pivotal when using stochastic algorithms [4]. Typically, parameters are set using commonly used values, or by a *trial-and-error* approach; interestingly, very few approaches have been proposed to systematically find optimal parameters settings [5–7]. Recently, Bartz-Beielstein introduced the *Sequential Parameter Optimization* (SPO) algorithm, a general framework for experimental analysis that accounts for iterative refinement of the parameters [8]. Successively, Hutter et al. proposed SPO<sup>+</sup>, an improved variant of SPO that uses log-transformations and the intensification criterion [9]. Hutter et al. also proposed PARAMILS, a method to find optimal parameters settings through local search in the parameters space [10]. An alternative approach consists of using a racing scheme where, starting from an initial set of techniques, an iterative selection is performed [11].

We address the problem of tuning an algorithm by introducing the *Sensitive Algorithmic Tuning* (SAT) method; our approach finds optimal parameters settings by minimizing the worst-case performance of the most sensitive parameters.

We evaluate this method by tuning the *Differential Evolution* (DE) optimization algorithm [12], although, in principle, our method can be applied to any algorithm.

The paper is organized as follows: in Sect. 2 we introduce the problem of tuning algorithms and the notion of *maximum success region*. Section 3 presents the concepts of robustness and sensitivity, and describes the Morris sensitivity analysis method and the Differential Evolution algorithm. In Sect. 4, we describe the SAT algorithm. Section 5 presents the experimental results. Finally, in Sect. 6 we discuss the conclusions and future works.

## 2 Algorithmic Tuning

*Algorithmic tuning* refers to the process of finding a set of values for the parameters of an algorithm, ensuring a satisfactory solution to the problem in the average case.

A trial-and-error approach is computationally expensive and could lead to poor performances if biased by a-priori knowledge. Without loss of generality, we assume that an algorithm is defined as follows:

$$Y = A(P, X) \tag{1}$$

where  $A$  is a procedure that takes in input a problem instance  $P$  and a vector of parameters  $X$ , and returns an output  $Y$ . W.l.o.g., we assume that each parameter  $x_i \in X \subset \mathbb{R}$  is constrained to an interval  $[x_-, x^+]$ .

Finding an optimal parameter setting is an intractable problem. Let  $X$  be a parameter setting for an algorithm  $A$ , where each parameter can take  $k$  different discrete values within the interval  $[x_-, x^+]$ ; it follows that the number of feasible parameters settings is  $k^{|X|}$ . This result makes an exhaustive search intractable for large instances.

Parameters tuning generally consists in running the algorithm on a testbed problem  $\bar{P}$ , which shares same characteristics with the original problem  $P$  (e.g. unimodality). In this context, it becomes crucial to identify a region of the parameters space that maximizes the probability of success, which we denote as *maximum success region*.

**Definition 1 (Maximum Success Region).** *Let  $X = [x_-, x^+] \subset \mathbb{R}$  be the range for the parameters of an algorithm  $A$ .  $X$  is called maximum efficiency region if the following condition holds:*

$$\forall x \in X : P_r(A(P, x) = S(P)) \approx 1$$

where  $P_r$  is a function that represents the probability of obtaining the exact solution  $S(P)$  of  $P$ .

The concept of maximum efficiency region fits particularly well the task of tuning the parameters of optimization algorithms; in this case, the maximum

efficiency region of an optimizer is the subspace of parameters, which ensures near-optimal solutions.

It should be noted that optimization methods are subject to *eager tuning*, typically by requiring a large number of objective function evaluations to ensure the convergence to an optimum. It is possible to overcome this limitation by systematically choosing the smallest parameter value ensuring a maximum success rate; this general principle is exploited by the SAT algorithm.

### 3 Methods

In this section, we describe a framework for robustness and sensitivity analysis that represents the basis of the *Sensitive Algorithmic Tuning* (SAT) algorithm.

#### 3.1 Robustness and Sensitivity Estimation

*Algorithmic robustness* is the probability of finding a satisfactory solution to a problem, even when the parameters are not optimally tuned. In general, there is a range of values for each parameter for which near-optimality is guaranteed. W.l.o.g., we assume that an algorithm is correct if a parameters setting ensuring an optimal or suboptimal solution exists. We define the *robustness condition*  $\rho$  and the *accuracy yield*  $\Gamma$  as follows:

**Definition 2 (Robustness Condition).** *Let  $X \in \mathbb{R}^n$  be a parameters setting for an algorithm  $A$ . Given a parameters set  $X_*$  obtained by perturbing  $X$ , the robustness condition  $\rho$  is defined as follows:*

$$\rho(X, X_*, A, P, \epsilon) = \begin{cases} 1 & \text{if } |A(P, X) - A(P, X_*)| \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where the robustness threshold  $\epsilon$  denotes the precision in the objective function value.

**Definition 3 (Yield).** *Let  $X \in \mathbb{R}^n$  be a parameters setting characterizing the behavior of a technique  $D$ . Given an ensemble  $T$  of parameters settings obtained by sampling the parameters space of  $X$ , the yield  $\Gamma$  is defined as follows:*

$$\Gamma(X, A, P, \epsilon, \rho, T) = \frac{\sum_{X_* \in T} \rho(X, X_*, A, P, \epsilon)}{|T|} \quad (3)$$

Since we consider subsets of the parameters space, we perform a Monte-Carlo sampling that generates trial settings in a specific parameters subspace. In our study, we set the robustness threshold  $\epsilon = 10^{-5}$  and the number of trials  $|T| = 100$ .

An ad-hoc algorithmic tuning requires knowledge of the effects of the parameters on the output. In this context, the Morris sensitivity analysis technique [13] represents an interesting approach; it ranks the parameters based on their effect on the output, and does not require information about the system being

analyzed [14]. In particular, a parameter is considered sensitive if variations of its value significantly affect the performance of the algorithm. In this context, the Morris technique can be used as an automated method for analyzing algorithms. We hypothesized that the identification of regions of low sensitivity could lead to an effective parameters tuning; this idea represents the basic principle of SAT.

### 3.2 Morris Method

*Sensitivity analysis* studies the influence of the input parameters on the output of a function (e.g. an algorithm), and identifies possible relations between parameters, e.g. linear and nonlinear.

The Morris method is a one-at-a-time (OAT) global sensitivity analysis technique. Given a set of parameters values, a parameter at time is modified and the variation of the output is recorded. This information is used to calculate the mean values  $\mu$  and the standard deviations  $\sigma$  associated with each parameter. Parameters with a high  $\mu$  have an important impact on the output, large values of  $\sigma$  indicate nonlinear relations with other parameters, whereas small mean values are associated with negligible effect.

### 3.3 Differential Evolution Algorithm

Differential Evolution (DE) is a stochastic *population-based* algorithm developed for global optimization in continuous spaces [12]. DE is used to solve multimodal, multi-objective, dynamic or constrained optimization problems; it finds application in several real-world problems, such as digital filter design, fermentation processes, dispatch optimization, and several others [15–18].

DE exploits a vector of differences that is used as a perturbation operator. Given an objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , DE starts by generating  $NP$  individuals at random, where the values of the variables are constrained in their respective lower and upper bounds. At each generation, each individual is modified according to a *crossover probability*  $C_r$ , using the following scheme:

$$x_{g+1}^i = x_g^i + F_w \times (y_g^i - z_g^i) \quad (4)$$

where  $x_{g+1}^i$  is the  $i$ -th variable of the new individual at generation  $g + 1$ ;  $y_g$  and  $z_g$  are two individuals of the population such that  $x \neq y \neq z$ ; and  $F_w$  is a weighting factor.

If  $f(x_{g+1}) < f(x_g)$ ,  $x_{g+1}$  replaces  $x_g$  in the population. Typically, the algorithm stops when a predetermined number of generations ( $G$ ) is reached.

The algorithm has few parameters;  $C_r$  controls the *exploring ability* of DE,  $F_w$  controls its *exploiting ability*, while  $NP$  determines the population size. In particular, for large-scale problems, DE requires large  $NP$  values and a sufficient number of generations to obtain satisfactory results.

## 4 Sensitive Algorithmic Tuning

The *Sensitive Algorithmic Tuning* (SAT) algorithm is a deterministic method that relies on *sensitivity analysis* and *worst-case screening* to identify maximum success regions within the parameters space.

Sensitive parameters are those that typically decrease the success or failure of an algorithm. The sensitivity of a parameter is strictly related to its uncertainty region; in general, a large parameter range makes difficult to find an optimal setting. When the value of a parameter is outside its maximum success region, we can observe an increase in sensitivity. Sensitivity minimization is a key principle in system design; it is necessary to obtain robust parameters setting, but not sufficient to guarantee near-optimal solutions. To overcome this limitation, we adopt a worst-case screening method. Given two parameters settings, SAT chooses the one providing the best solution in the worst case.

The SAT algorithm is depicted in Algorithm 1. At each step, SAT splits the parameters space of each parameter and performs Morris analysis in both regions; it then selects the subspace with the highest mean value, aiming at tuning the most sensitive parameter first. The splitting procedure can be any interval-cutting strategy [19]; in our experiments, we generate two regions by halving the range of sensitive parameters.

An example of the application of the SAT algorithm to a technique with two parameters is depicted in Fig. 1. The objective function values obtained by each parameters setting sampled during sensitivity analysis are also used for evaluating the worst case performance of the algorithm. We use two halting conditions;

---

**Algorithm 1.** Pseudo-code of the *Sensitive Algorithmic Tuning* (SAT) algorithm.

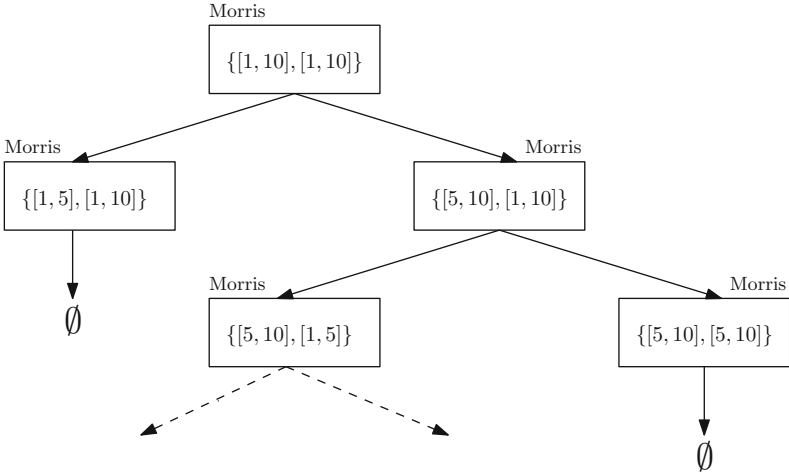
---

```

1: procedure SAT( $A, X_-, X^+$ )
2:    $k \leftarrow 0$ 
3:    $M_k \leftarrow \text{Morris}(A, X_-, X^+, r, p, \Delta)$ 
4:   while  $\neg \text{StopCondition}$  do
5:      $s_i \leftarrow \max(M_k)$ 
6:      $[lx_-, lx^+] \leftarrow \text{LowSplit}(X_-, X^+, s_i)$ 
7:      $[hx_-, hx^+] \leftarrow \text{HighSplit}(X_-, X^+, s_i)$ 
8:      $M_{lx} \leftarrow \text{Morris}(A, lx_-, lx^+, r, p, \Delta)$ 
9:      $M_{hx} \leftarrow \text{Morris}(A, hx_-, hx^+, r, p, \Delta)$ 
10:    if  $\max f(M_{lx}) > \max f(M_{hx})$  then
11:       $[X_-, X^+] \leftarrow [lx_-, lx^+]$ 
12:       $M_k \leftarrow M_{lx}$ 
13:    else
14:       $[X_-, X^+] \leftarrow [hx_-, hx^+]$ 
15:       $M_k \leftarrow M_{hx}$ 
16:    end if
17:     $k \leftarrow k + 1$ 
18:  end while
19: end procedure

```

---



**Fig. 1.** Iterations of the SAT algorithm on a bi-dimensional parameter space.  $\emptyset$  denotes a region that will not be furtherly splitted.

the attainment of an optimal solution in the worst case, or the impossibility of further splitting the parameters space. This strategy is useful to prevent a waste of computational effort when the parameter region is sufficiently small.

## 5 Experimental Results

We use the SAT algorithm for tuning DE, with the parameters reported in Table 1.

Since parameters settings are problem-dependent, we consider unimodal and multimodal numerical functions of dimensions  $d = 10$  and  $d = 20$  (see Table 2). Three unimodal functions are used, characterized by multiple quadratic terms. Multimodal functions take into account noise (see function  $f_4$ ), quadratic and quartic terms. It should be noted that the number of local minima of  $f_5$  increases exponentially with its dimension [20].

**Table 1.** Parameters of the Differential Evolution algorithm.  $n$  denotes the dimension of the problem.

Parameter	Description	$X_-$	$X^+$
$NP$	Population size	10	50
$G$	Number of generations	$n \times 25$	$n \times 250$
$F_w$	Weighting factor	0	1
$C_r$	Crossover rate	0	1

**Table 2.** Test problems.  $f^*$  represents the global minimum;  $(X_-)$  and  $(X^+)$  are the lower and upper bound, respectively,  $n$  denotes the dimension of the problem.

	Class	$f$	$f^*$	$X_-$	$X^+$
$f_1$	Unimodal	$\sum_{i=1}^n x_i^2$	0	-100	100
$f_2$	Unimodal	$\sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	0	-100	100
$f_3$	Unimodal	$\sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	0	-30	30
$f_4$	Multimodal	$\sum_{i=1}^n i * x_i^4 + random[0, 1)$	0	-1.28	1.28
$f_5$	Multimodal	$\sum_{i=1}^n x_i^2 - 10 \cos(2\pi x - i) + 10$	0	-5.12	5.12

The metric adopted for evaluating the DE performance is the average value of the best solution over 10 independent runs; moreover, we apply the SAT algorithm to find a setting that ensures a worst-case result of  $10^{-1}$ .

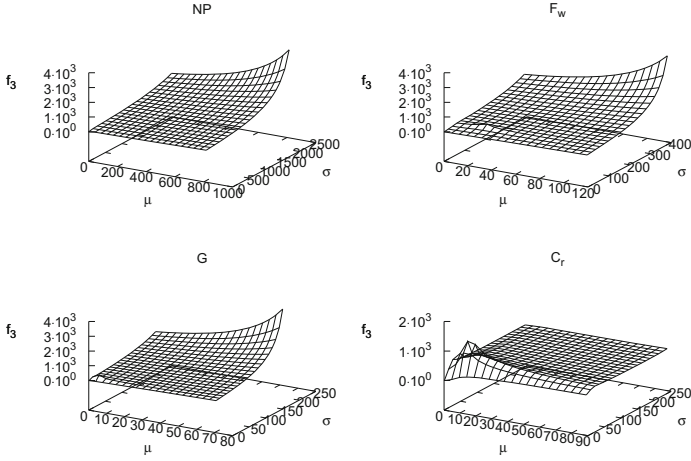
An extensive set of simulations is performed to achieve an optimal parameters setting for DE. Table 3 reports the results of SAT on the five testbed problems for  $d = 10$ . The experimental results show that SAT is able to find satisfactory settings for all the problems.

**Table 3.** Results for  $d = 10$ .  $f_w^*$  is the worst case objective function value;  $v(f)$  represents the variance of the output of all the sampled settings; the lower and the upper bound of each parameter found by SAT are within brackets.

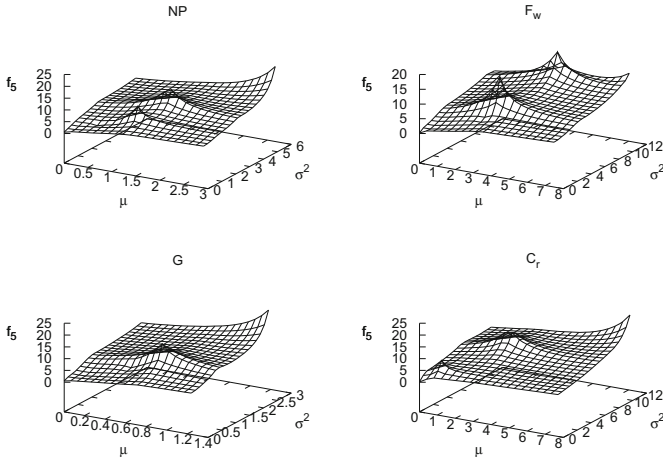
$f$	$f_w^*$	$v(f)$	$NP$	$G$	$F_w$	$C_r$
$f_1$	$1.434 \times 10^{-82}$	$1.690 \times 10^4$	[77, 100]	[2000, 2500]	[0.55, 0.66]	[0.21, 0.32]
$f_2$	$7.737 \times 10^{-96}$	$2.594 \times 10^6$	[55, 100]	[2375, 2500]	[0.55, 0.75]	[0.43, 0.55]
$f_3$	$8.586 \times 10^{-1}$	$2.095 \times 10^9$	[77, 88]	[1500, 1625]	[0.55, 1]	[0.1, 0.32]
$f_4$	1.482	$1.597 \times 10^{-1}$	[97, 100]	[4937, 5000]	[0.55, 0.66]	[0.43, 0.55]
$f_5$	$5.971 \times 10^{-1}$	5.9305	[83, 88]	[500, 1500]	[0.44, 0.55]	[0.1, 0.325]

On unimodal functions  $f_1$  and  $f_2$ , we are able to obtain an upper bound on the algorithm performance that is close to the global optimum. The  $f_3$  function represents an exception within the unimodal set: it is not surprising that the worst case result is many orders of magnitude worse than the others, due to the presence of several plateau regions that can trap the algorithm. The values of variance show that DE is highly influenced by its parameters, and the initial bounds contain many suboptimal parameters settings.

The multimodal functions are more complex, as the presence of noise and several local minima leads to a difficult tuning. Despite this complex scenario, SAT is able to find a setting that ensures near-optimal solutions in the worst case. The variance is several orders of magnitude lower with respect to unimodal functions; however, this is probably due to the smaller intervals on which the algorithm operates.



**Fig. 2.** Sensitivity landscape of  $f_3$  for  $d = 10$ .  $\mu, \sigma$  are reported on the  $x$  and  $y$  axis, respectively; the objective function values are reported on the  $z$  axis.



**Fig. 3.** Sensitivity landscape of  $f_5$  for  $d = 10$ .  $\mu, \sigma$  are reported on the  $x$  and  $y$  axis, respectively; the objective function values are reported on the  $z$  axis.

Figures 2 and 3 report the sensitivity landscapes of  $f_3$  and  $f_5$ . The lowest function value is achieved when the sensitivity of each parameter is close to zero. This result seems to support the underlying strategy implemented in SAT. The sensitivity landscapes of the parameters are different for the two problems; for the  $f_3$  function, the population size, the number of generations and the weighting factor show a convex shape, while the crossover probability has a spike near the maximum success region. By inspecting the sensitivity landscape of  $f_5$ , we note that the best results are obtained when the sensitivity is close



to zero, but the shape of the landscape is more rugged. For all parameters, we have a high number of peaks at different points, which remarks the complexity of tuning DE in this case. Despite this rugged parameter space, SAT is able to identify the best setting.

The results for the tuning of DE when  $d = 20$  are presented in Table 4.

**Table 4.** Results for  $d = 20$ .  $f_w^*$  is the worst case objective function value;  $v(f)$  represents the variance of the output of all the sampled settings; the lower and the upper bound of each parameter found by SAT are within brackets.

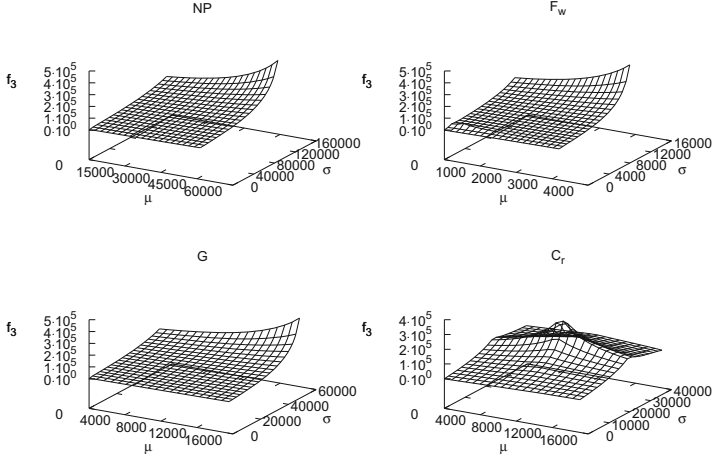
$f$	$f_w^*$	$v(f)$	$NP$	$G$	$F_w$	$C_r$
$f_1$	$2.322 \times 10^{-77}$	$3.601 \times 10^5$	[55, 77]	[4750, 5000]	[0.55, 1]	[0.21, 0.32]
$f_2$	$3.652 \times 10^{-90}$	$5.895 \times 10^8$	[55, 100]	[4750, 5000]	[0.55, 0.77]	[0.43, 0.55]
$f_3$	2.830	$2.193 \times 10^{11}$	[77, 88]	[4000, 5000]	[0.1, 0.32]	[0.1, 0.21]
$f_4$	5.402	0.795	[32, 55]	[1000, 2000]	[0.21, 0.32]	[0.88, 1]
$f_5$	2.985	60.316	[77, 88]	[4000, 5000]	[0.43, 0.55]	[0.32, 0.55]

On the unimodal functions  $f_1$  and  $f_2$ , the algorithm finds a setting with a satisfactory upper bound, however this result is not as good as the one found for the smaller dimension instances.  $f_3$  proves to be the most difficult unimodal problem of the entire set; in this case, lower accuracy than expected was obtained. The values of variance are dramatically increased, and the tuning becomes more difficult for increasing problem dimension. Although SAT was not able to find a parameters setting that matched the desired upper bound, the results required only limited computational effort.

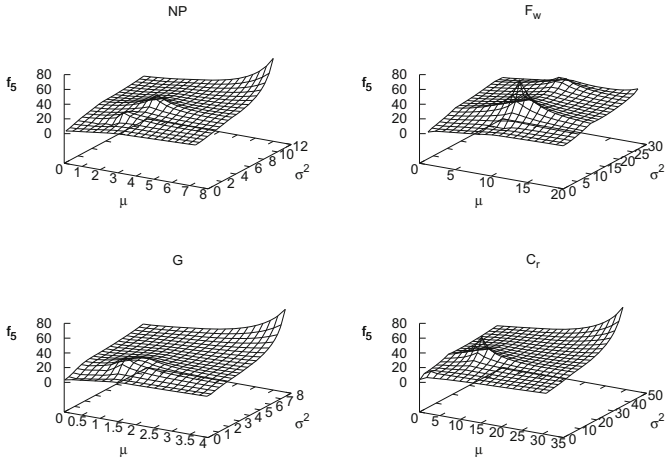
In Figs. 4 and 5, we report the sensitivity landscape of  $f_3$  and  $f_5$  for  $d = 20$ .

The plots confirm the results obtained for lower dimensional problems. The lowest function value is always achieved when the parameters sensitivity is close to zero. The landscape for the two problems are comparable with the previous, however we observe an increase in  $\mu$  and  $\sigma$ ; for example, for  $f_3$ , the crossover probability shows a peak in the central region that is far from the maximum success region. For the  $f_5$  function, a rugged landscape is also obtained, and we observe increased sensitivity values for all parameters. This landscape seems to remark the presence of sensitivity barriers, which makes difficult the tuning process.

Finally, we perform robustness analysis. Since parameters ranges are considered instead of single values, we have to evaluate the probability of finding satisfactory results using any setting belonging to a given parameter space. Table 5 shows the yield values of the settings for each problem. High yield values are obtained in all testbeds, except for the  $f_3$  and  $f_4$  functions. We conclude that the proposed settings are robust and effective, and guarantee near-optimal performances of the DE algorithm.



**Fig. 4.** Sensitivity landscape of  $f_3$  for  $d = 20$ .  $\mu, \sigma$  are reported on the  $x$  and  $y$  axis, respectively; the objective function values are reported on the  $z$  axis.



**Fig. 5.** Sensitivity landscape of  $f_5$  for  $d = 20$ .  $\mu, \sigma$  are reported on the  $x$  and  $y$  axis, respectively; the objective function values are reported on the  $z$  axis.

**Table 5.** Robustness analysis of SAT parameters.  $d$  represents the dimension,  $\Gamma$  is the yield.

$f$	$d$	$\Gamma$ (%)	$f$	$d$	$\Gamma$ (%)
$f_1$	10	100	$f_1$	20	100
$f_2$	10	100	$f_2$	20	100
$f_3$	10	52	$f_3$	20	64
$f_4$	10	54	$f_4$	20	42
$f_5$	10	92	$f_5$	20	92

## 6 Conclusions

Algorithm design is a complex process that should pursue three general properties; efficiency, effectiveness and generality. These properties are strictly dependent on how algorithms parameters are configured. Parameters tuning is therefore required to ensure efficiency and effectiveness when solving a problem.

In this work, we propose the *Sensitive Algorithmic Tuning* (SAT), a new method for the automatic tuning of algorithms. SAT uses the Morris technique to determine the parameters to optimize to obtain satisfactory upper bounds on the expected performances. We evaluate the performance of SAT on the problem of tuning the *Differential Evolution* (DE) algorithm. The experimental results show the effectiveness of our approach; moreover, the discovered settings present a high degree of robustness. Interestingly, these results confirm that optimal settings are associated with a low parameters sensitivity.

We believe the parameters tuning problem should be exploited more deeply by extending the cutting strategy; for instance, an alternative *branch-and-bound* [21] approach could represent an efficient sampling strategy, where sensitivity information could be used to perform expansion and pruning of the exploring tree. We suggest the enhanced SAT algorithm could represent a valid candidate for tuning one of the most important algorithms of the Internet era: PageRank [22].

**Acknowledgments.** The authors would like to acknowledge Professor Angelo Marcello Anile for the useful discussions on the seminal idea of automatic algorithms tuning. Professor Anile was a continuous source of inspiration during our research work.

## References

1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. The MIT Press, Cambridge (2001)
2. Horst, R., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization. Springer, USA (2000)
3. Floudas, C.A.: Deterministic Global Optimization. Kluwer, New York (2000)
4. Motwani, R., Raghavan, P.: Randomized algorithms. ACM Comput. Surv. (CSUR) **28**(1), 37 (1996)
5. Back, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.* **1**(1), 1–23 (1993)
6. Audet, C., Orban, D.: Finding optimal algorithmic parameters using derivative-free optimization. *SIAM J. Optim.* **17**(3), 642–664 (2006)
7. Nannen, V., Eiben, A.E.: Relevance estimation and value calibration of evolutionary algorithm parameters. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp. 975–980. Morgan Kaufmann Publishers Inc. (2007)
8. Bartz-Beielstein, T.: Sequential parameter optimization - sampling-based optimization in the presence of uncertainty (2009)
9. Hutter, F., Hoos, H.H., Leyton-Brown, K., Murphy, K.P.: An experimental investigation of model-based parameter optimisation: Spo and beyond. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, GECCO 2009, pp. 271–278. ACM, New York (2009)

10. Hutter, F., Hoos, H.H., Stützle, T.: Automatic algorithm configuration based on local search. In: AAAI, vol. 7, pp. 1152–1157 (2007)
11. Birattari, M., Yuan, Z., Balaprakash, P., Stützle, T.: F-race and iterated f-race: an overview. In: Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M. (eds.) *Experimental Methods for the Analysis of Optimization Algorithms*, pp. 311–336. Springer, Berlin (2010)
12. Storn, R., Price, K.V.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**(4), 341–359 (1997)
13. Morris, M.D.: Factorial sampling plans for preliminary computational experiments. *Technometrics* **33**(2), 161–174 (1991)
14. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S.: *Global Sensitivity Analysis: The Primer*. Wiley-Interscience, Hoboken (2008)
15. Storn, R.: Differential evolution design of an iir-filter. In: *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 268–273 (1996)
16. dos Santos Coelho, L., Mariani, V.C.: Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints. *Energy Convers. Manage.* **48**(5), 1631–1639 (2007)
17. Chiou, J.-P., Wang, F.-S.: Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process. *Comput. Chem. Eng.* **23**(9), 1277–1291 (1999)
18. Price, K.V., Storn, R.M., Lampinen, J.A.: *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Heidelberg (2005)
19. Fusiello, A., Benedetti, A., Farenzena, M., Busti, A.: Globally convergent auto-calibration using interval analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 1633–1638 (2004)
20. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *Evol. Comput.* **3**(2), 82–102 (1999)
21. Fuchs, M., Neumaier, A.: A splitting technique for discrete search based on convex relaxation. *J. Uncertain Syst.* **4**(1), 14–21 (2010)
22. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab (1999)