ELSEVIER

Chemometrics and Intelligent Laboratory Systems 39 (1997) 161–173

Chemometrics and intelligent laboratory systems

A hybrid of simplex method and simulated annealing

Vladimír Kvasnička *, Jiří Pospíchal

Department of Mathematics, Slovak Technical University, 81237 Bratislava, Slovakia

Received 15 July 1995; accepted 22 August 1997

Abstract

One of basic concepts of the well-known simplex optimization method is that from the current simplex set of points (solutions) a new point – reflection is constructed. The reflection point is used for a conditional updating of the simplex set. This simple and efficient idea is applied in the simulated annealing to suggest a new version of this stochastic optimization method. As a forerunner of the presented simulated annealing is the controlled random search invented by Price in the middle of seventies. He proposed the very important idea that a population of points is considered and from this population the simplex set is randomly selected. Reflection points update the population so that they conditionally substitute points with highest values of objective function. The simplex simulated annealing enhances further stronger stochastic and evolution character of this method. The construction of reflection points is randomized and their returning to the population is solved by the Metropolis criterion. A parallel version of simplex simulated annealing uses a decomposition of the whole population into disjoint subpopulations for which independent simulated annealings are done. The subpopulations randomly interact so that between two subpopulations their best points are exchanged and worst ones are eliminated. © 1997 Elsevier Science B.V.

Keywords: Simplex optimization; Simulated annealing; Evolutionary optimization

1. Introduction

The standard simplex method [1,2] belongs to the well-known nongradient optimization techniques often used in chemistry. Its generalization towards its randomization was done by Price [3] (cf. also Ref. [4]). His so-called control random search (CRS) method is able to look for, to some limited extent, global minima, and may be now considered as one of forerunners of modern stochastic optimization techniques (genetic algorithms [5], simulated annealing [6], evolution strategy [7], etc.). The applications of optimization algorithms in chemistry can be found, e.g., in Refs. [8–13]. The purpose of this communication is to give a brief outline of our recent efforts to elaborate a hybrid of the simplex method and the simulated annealing, which would be more effective and robust than their single predecessors. The proposed hybrid represents a simple approach to evolutionary optimization methods with substantially increased effectiveness and robustness in comparison with its single constituents alone.

* Corresponding author. E-mail: kvasnic@theochem.chtf.stuba.sk.

PII S0169-7439(97)00071-3

2. Theory

Let us consider an objective function

$$f: D \subset R^n \rightarrow R, \tag{1}$$

where $D = \Pi[a_i, b_i]$ is a domain of f. If $x = (x_1, x_2, \ldots, x_n) \in D$, then $\forall i$: $a_i \leq x_i \leq b_i$. The following optimization problem is considered

$$x_{\text{opt}} = \arg \min_{x \in D} f(x). \tag{2}$$

Its solution $x_{\text{opt}} \in D$ corresponds to the so-called optimal point for which the objective function f over the domain D has the minimal values, $\forall x \in D$: $f(x) \geq f(x_{\text{opt}})$. If a point $x \in R^n$ is outside the domain D, then a repair process returns this point to the domain D. Loosely speaking, this process is a 'mirroring' of entries x_i by the bounds a_i and b_i

$$x_i < a_i \Rightarrow x_i \leftarrow 2a_i - x_i, \tag{3a}$$

$$x_i > b_i \Rightarrow x_i \leftarrow 2b_i - x_i, \tag{3b}$$

for $i = 1, 2, \ldots, n$. These elementary mirroring steps of the repair process are repeated until all entries x_i belong to closed intervals $[a_i, b_i]$.

2.1. Basic principles of simplex method [1,2]

A simplex set S is composed of $n + 1$ points from the domain D

$$S = \{x_1, x_2, \ldots, x_{n+1}\} \subset D. \tag{4}$$

Its three points are especially distinguished

$$x_{\text{H}} = \arg \max_{x \in S} f(x), \tag{5a}$$

$$x_{\text{L}} = \arg \min_{x \in S} f(x), \tag{5b}$$

$$\bar{x} = \frac{1}{n}\left(\sum_{x \in S} x - x_{\text{H}} \right). \tag{5c}$$

At point x_{H} (x_{L}) the objective function f has highest (lowest) functional value on the simplex S, and $\bar{x}$ corresponds to the center of gravity of simplex points but the point x_{H}. A *reflection* (see Fig. 1) of x_{H} with respect to the center of gravity $\bar{x}$ is

$$x^* = x_{\text{H}} + 2(\bar{x} - x_{\text{H}}) = 2\bar{x} - x_{\text{H}}. \tag{6}$$

The reflection point x^* is used for the updating of simplex S

$$f(x^*) < f(x_{\text{H}}) \Rightarrow x_{\text{H}} \leftarrow x^*. \tag{7}$$

In the opposite case, $f(x^*) \geq f(x_{\text{H}})$, the so-called simplex *reduction* (see Fig. 1) is done, where simplex points are updated by

$$x_i \leftarrow \tfrac{1}{2}(x_i + x_{\text{L}}). \tag{8}$$

The simplex method is stopped if the current simplex is sufficiently small, e.g., the L_1 distance between x_{H} and x_{L} is smaller than a prescribed precision ε (small positive number)

$$|x_{\text{H}} - x_{\text{L}}| < \varepsilon. \tag{9}$$

An algorithmic implementation of simplex method is very simple and its pseudocode is outlined in algorithm 1.

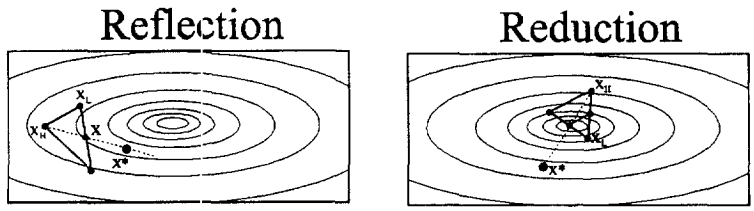

Fig. 1. Schematic outline of reflection and reduction operations. Encircled dots correspond to reflection points x^* constructed by the 'reflection' operation of x_H point through the center of gravity $\bar{x}$. If the functional value $f(x^*)$ is smaller than $f(x_H)$, then the current simplex is 'reduced' so that the center of reduction is the point x_L.

Algorithm 1. Implementation of simplex method. The algorithm is initialized by a random generation of $n+1$ simplex points from the domain D. For the current simplex S a reflection point x^* is constructed by a procedure `reflection (S)` (see Eq. (6)). We assume that this procedure contains also a repair process (see Eqs. (3a) and (3b)) applied if the reflection x^* is outside the domain D, the repair process returns the reflection x^* inside the domain D. The simplex methods is stopped if the L_1 distance between points x_H and x_L (that correspond to simplex points evaluated by the highest and lowest value, respectively, of the objective function) is smaller than a small positive number ε. The updating of the current simplex S is done so that if x^* provides better solution than x_H (i.e. $f(x^*) < f(x_H)$), then the point x_H is eliminated from the simplex S by the reflection x^*; in the opposite case (i.e., $f(x^*) \geq f(x_H)$) a reduction of S is done, this is realized by the procedure `reduction (S)`, see Eq. (8).

```
Procedure Simplex;
begin S:=set of (n+1) randomly generated points of D;
      while |xH-xL|>ε do
      begin x*:=reflection(S);
            if f(x*)<f(xH) then xH:=x* else reduction(S);
      end;
end;
```

2.2. Controlled random search (CRS)

The method of controlled random search, introduced by Price [3], is a generalization of simplex algorithm and may be considered as an 'archetype' of modern evolution algorithms.

Let P be a population of p randomly selected points from D ($p \gg n$)

$$P = \{x_1, x_2, \ldots, x_p\}. \tag{10}$$

The simplex subset S is composed of $(n+1)$ points that are randomly selected from the population P

$$S = \{x_{\alpha_1}, x_{\alpha_2}, \ldots, x_{\alpha_{n+1}}\} \subset P. \tag{11}$$

For the given simplex S a reflection point x^* is constructed in the same way as in the standard simplex method, see Eq. (6), but now the point x_H is randomly selected and not determined as the simplex point with highest value of objective function. The reflection point x^* is used for an updating of the population P, the point

$$x_{\max} = \arg \max_{x \in P} f(x) \tag{12}$$

is competitively changed by the reflection x^* (see Fig. 2)

$$f(x^*) < f(x_{\max}) \Rightarrow x_{\max} \leftarrow x^*. \tag{13}$$

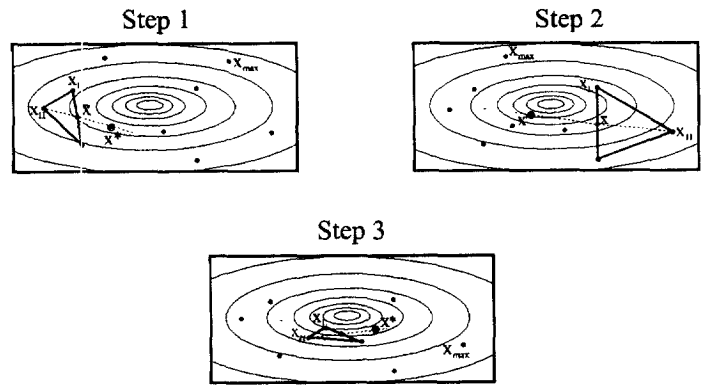

Fig. 2. Outline of the first three steps in CRS method. In step 1 a simplex is randomly generated, its reflection x^* (encircled dot) eliminates the point $x_{\max}$. Formally the same procedure is applied in the next steps of CRS method. Since the 'marginal' points $x_{\max}$ are conditionally removed from the population, all points are in turn 'compressed' to a close neighbourhood of the potential final solution.

(while in the simplex method the point x_H is substituted by the reflection). In the opposite case, $f(x^*) \geq f(x_{\max})$, the population remains unchanged. The way of updating (see Eq. (13)) of the population P corresponds to the most principal deviation from the simplex method. Since the points $x_{\max}$ are successively removed in each step of CRS algorithm, the 'diameter' of population P is monotonously decreasing. At the final stage of the algorithm all points of P are situated in a close neighbourhood of the resulting 'optimal' solution. This means that the process of simplex reduction (see Eq. (8)) may be omitted in the present algorithm. A 'concentration' of points of P is now caused by the way of updating (see Eq. (13)), where 'marginal' points $x_{\max}$ are substituted by reflection points x^* (see Fig. 3). The algorithm is finished, e.g., if L_1 distance between points corresponding to the maximal and minimal functional values from the whole population is smaller than a prescribed precision, $|x_{\max} - x_{\min}| < \varepsilon$. A pseudocode of CRS is outlined in algorithm 2.

Algorithm 2. Implementation of controlled random search (CRS) method. The algorithm is initialized by a random generation of p ($p \gg n$) points of the population P. Simplex points from the subset S (composed of $n+1$ points) are randomly selected from the current population P. The reflection point x^*, created from simplex points

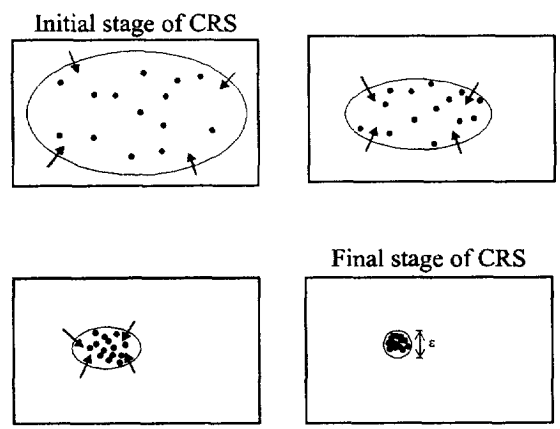

Fig. 3. 'Compression' of population points in CRS method, discussed in Fig. 2. A diameter of population P is decreasing successively as number of iterative steps increases. The method is stopped if the diameter of P is smaller than a prescribed small positive number ε (precision).

of S, is returned to the population P if its functional value of f is smaller than the maximal functional value of the population P (represented by x_{max}), if so, then x^* eliminates the point x_{max}. The method is stopped if L_1 distance between points x_{max} and x_{min} is smaller than a prescribed ε.

```
procedure CRS;
begin P:=set of p randomly generated points of D;
      while |xmax-xmin|>ε do
      begin S:=set of n+1 randomly selected points of P;
            x*:=reflection(S);
            if f(x*)<f(xmax) then xmax:=x*;
      end;
end;
```

2.3. Simplex simulated annealing (SSA)

CRS method is modified to be more competitive and stochastic. The following two modifications are introduced. First, the construction of reflection point x^* is randomized (cf. Ref. [4], see Fig. 4)

$$x^* = x_H + r(\alpha_0, \sigma)(\bar{x} - x_H), \tag{14}$$

where $r(\alpha_0, \sigma)$ is a random (e.g. Gaussian) number with a mean α_0 and a 'standard deviation' σ, in all our illustrative calculations we used $\alpha_0 = 2$ and $\sigma = 0.5$. Second, the updating of population P is modified so that instead of the point x_{max} the simplex point x_H is considered, introduction of the new point x^* to the population P is solved by the Metropolis criterion [7] with probability

$$\Pr = \min\left\{1, \exp\left(-\frac{f(x^*) - f(x_H)}{T}\right)\right\}, \tag{15}$$

where T is a 'temperature' playing a role of basic simulated annealing parameter. If T tends to zero, then the acceptance probability Pr of reflection point x^* with higher functional value than $f(x_H)$ (i.e., $f(x^*) > f(x_H)$) is vanishing. Simple pseudocode of SSA is outlined in algorithm 3.

Algorithm 3. Implementation of simplex simulated annealing method. Algorithm is initialized by a random generation of p points of the domain D, these points create an initial population P. The initial (maximal) temperature T is set equal to T_{max}. Outer cycle is repeated while $T > T_{min}$, where the temperature is subsequently multiplicatively decreased by $T := \alpha * T$. The inner cycle is repeated by k_{max} times, where k_{max} is sufficiently great number (usually from 10^3 to 10^5). Real variable `random` corresponds to a uniform random number generator from the semiopen interval [0,1).

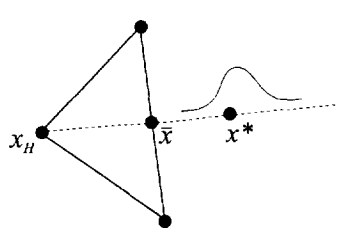

Fig. 4. An outline of stochastic reflection process determined by Eq. (14). The unimodal function above dotted reflection line represents a distribution of random number generator with a mean $\alpha_0 = 2$ and with width determined by the 'deviation' σ.

```
procedure SSA;
begin P:=set of p randomly selected points of D;
      T:=Tmax;
      while T>Tmin do
      begin for k:=1 to kmax do
            begin S:=set of n+1 randomly
                     selected points of P;
                  x*:=reflection(S);
                  Pr:=min(1,exp(-(f(x*)-f(xH))/T));
                  if random<Pr then xH:=x*;
            end;
            T:=α*T;
      end;
end;
```

2.4. Parallel simplex simulated annealing (PSSA)

In order to introduce a parallelization of SSA the population P should be divided onto r disjoint subpopulations composed of p points from D (i.e., the whole population P is composed of $|P| = r \times p$ points)

$$P = \bigcup_{i=1}^{r} P_i \quad (\text{if } i \neq j, \text{ then } P_i \cap P_j = \varnothing). \tag{16}$$

The method of SSA is applied independently for these subpopulations. The simulated annealing is synchronized for all subpopulations so that all have the same temperature T, which is simultaneously decreased for all of them, see Fig. 5. An interaction between subpopulations is realized by a random selection (with very low level of probability, e.g., $P_{\text{exch}} = 10^{-3}$) of two subpopulations and then copies of best solutions (points) are mutually exchanged, and simultaneously worst ones are eliminated. Let I and J be two randomly selected indices, $1 \leq I < J \leq p$, of subpopulations, we find the best and worst solutions for these two subpopulations

$$x_{\min}^{(i)} = \arg \min_{x \in P_i} f(x), \tag{17a}$$

$$x_{\max}^{(i)} = \arg \max_{x \in P_i} f(x), \tag{17b}$$

for $i = I, J$. Then, an updating of the selected subpopulations (called *exchange*) is carried out by

$$P_I \leftarrow \left(P_I \setminus \left\{x_{\max}^{(I)}\right\}\right) \cup \left\{x_{\min}^{(J)}\right\}, \tag{18a}$$

$$P_J \leftarrow \left(P_J \setminus \left\{x_{\max}^{(J)}\right\}\right) \cup \left\{x_{\min}^{(I)}\right\}. \tag{18b}$$

A pseudocode of PSSA is outlined in algorithm 4 in a form useful for an implementation for computers with parallel architecture of processors.

Algorithm 4. Implementation of parallel simplex simulated annealing (PSSA) method. Procedure `PSSA` corresponds to the master part of PSSA algorithm. In the framework of this procedure the temperature T is controlled (from $T_{\max}$ to $T_{\min}$) for all subpopulations $P_1, P_2, \ldots, P_r$. A simulated annealing for a fixed temperature T (i.e., in fact the Metropolis algorithm [7]) is performed by the procedure `Single_step_SSA` independently for all r subpopulations. This procedure is an inner-cycle part of the procedure `SSA` presented above in algorithm 3. Procedure `Exchange`, activated in the procedure `PSSA` after finishing Metropolis algorithms for all separate subpopulations, updates two randomly selected subpopulations so that their best solutions eliminate worst ones. A structure of the procedure `PSSA` is suggested in such a way that it allows to assign activations of `Single_step_SSA` for different subpopulations (determined by index `i`) to different processors of a parallel computer.

```
procedure Single_step_SSA(counter,P,T,k_max);
begin if counter=1 then
      P:=set of q randomly generated points of D;
      for k:=1 to k_max do
      begin S:=set of n+1 randomly selected
                points of P;
            x*:=reflection(S);
            Pr:=min(1,exp(-(f(x*)-f(x_H))/T));
            if random<Pr then x_H:=x*;
      end;
end;

procedure Exchange;
begin I:=1+random(p);
      J:=1+random(p);
      P_I:=(P_I\{x^(I)_max})∪{x^(J)_min};
      P_J:=(P_J\{x^(J)_max})∪{x^(I)_min};
end;


procedure PSSA;
begin counter:=0;
      T:=T_max;
      while T>T_min do
      begin counter:=counter+1;
            for i:=1 to r do
            Single_step SSA(counter;P_i,T,k_max);
            if random<P_exch then Exchange;
            T:=α*T;
      end;
end;
```

3. Illustrative calculations

An effectiveness and robustness of the presented simplex simulated annealing methods (SSA and PSSA) are tested either by numerical examples known in literature or by our model highly multimodal objective functions.

3.1. Regression analysis

In recent publication of Křivý and Tvrdík [4] a modified version of CRS has been discussed, its effectiveness was demonstrated by many examples of regression analysis [4,14,15] that are known as notoriously difficult. We have recalculated all these examples by SSA and PSSA, in all cases we have obtained the same numerical values of parameters and objective functions as those ones listed in [4]. As illustrative example we present a regression problem which is most time consuming with very slow convergence. The model function is determined by

$$g(x;\beta) = \beta_1 x^{\beta_3} + \beta_2 x^{\beta_4}, \tag{19}$$

where the parameters are restricted by

$$0 \le \beta_1 \le 1, \quad 1 \le \beta_2 \le 8, \quad 1 \le \beta_3 \le 5, \quad 0 \le \beta_4 \le 1. \tag{20}$$

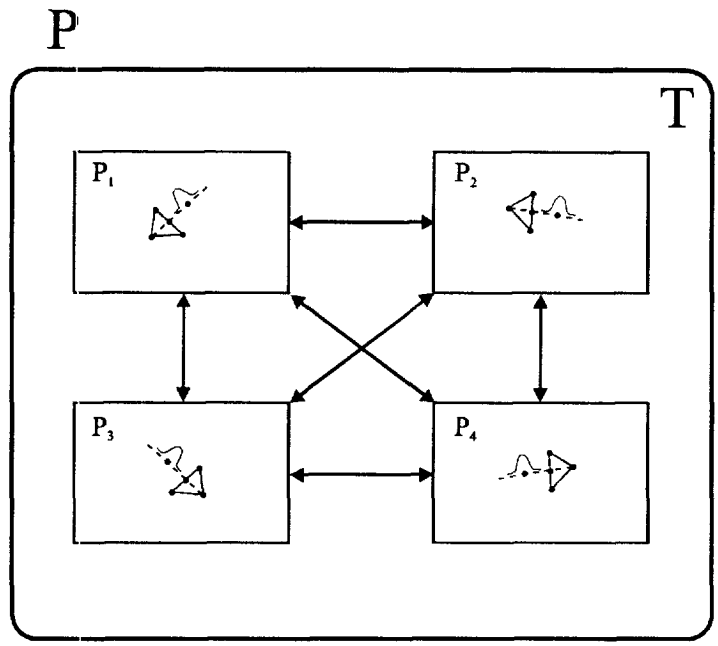

Fig. 5. Schematic outline of parallel simulated annealing method. Whole population P is divided onto subpopulations P_1, P_2, ... Single-step SSA processes are running independently for all subpopulations and for the same temperature T. Arrows between subpopulations denote an exchange of 'information' between them, this is realized so that for randomly selected pair of subpopulations P_I and P_J the copies of best solutions are exchanged and their worst ones are eliminated.

The objective function is expressed as a logarithm of sum of squares of differences between required and calculated values,

$$f(\beta) = \ln\left(\sum_{i=1}^{N} [g(x_i, \beta) - y_i]^2 \right), \tag{21}$$

where the 'training-set points' (x_i, y_i), $i = 1, 2, \ldots, N$, are listed in Table 1. An application of the logarithm in the objective function is the well-known numerical trick for simple mapping of a function which is varying in a range of many orders onto a smaller interval. Numerical results (average CPU time from 20 runs obtained for all the mentioned methods by using the same compiler and the same PC with 486DX2 processor, clock 66 MHz) are listed in Table 2. The CRS algorithm did not converge, it finished at the values of objective function $f = 10^{-3}$, that is about two orders higher than its optimal value. The modified version of CRS, suggested by Křivý and Tvrdík [4], already converges though relatively slowly. The results of present versions of simplex simulated annealing for basic parameters

$$k_{\max} = 1000, \quad T_{\max} = 10^{-3}, \quad T_{\min} = 10^{-5}, \quad \alpha = 0.99, \quad p = 10 \times n, \quad r = 10, \tag{22}$$

where p is the number of points in the population and r is the number of subpopulations in the parallel SSA, are summarized in Table 2. We see that both present simulated annealing methods are more effective than the randomized version of CRS [4], obtained CPU times are almost five times smaller than for the randomized CRS. We have to note that similar CPU times are not obtained for all examples studied in [4], but in worst cases the acceleration was twofold with respect to the modified CRS.

We have tried to solve these regression analysis examples also by pure simulated annealing approaches using either binary or real representation of variables (with the similar parameters as in Eq. (22)). We have achieved

Table 1
Training set of regression analysis specified by the model function Eq. (19)

No.	x	y
1	12	7.31
2	13	7.55
3	14	7.80
4	15	8.05
5	16	8.31
6	17	8.57
7	18	8.84
8	19	9.12
9	20	9.40
10	21	9.69
11	22	9.99
12	23	10.30

the correct solutions only for a few examples. In general, the method of simulated annealing in the investigated form was unable to solve correctly all regression analysis examples. The main obstacle for the simulated annealing is that values of objective functions are varied in ranges of many orders, they are very sensitive to small perturbations of variables. Consequently, the Metropolis criterion may (may not) accept new perturbed states that are not (that are) very promising for 'evolution' of current states towards correct solutions. Since in the hybrid of simplex method and simulated annealing a 'blind' construction of new states is substituted by a more sophisticated approach based on the construction of reflection points, where a kind of knowledge of promising directions is used, suggested methods successfully overcome the above mentioned numerical difficulties of pure simulated annealing approaches.

3.2. Rosenbrock's function

This function of two independent variables is unimodal and bi-quadratic

$$g(x_1,x_2) = 100 \cdot \left(x_1^2 - x_2\right)^2 + (1 - x_1)^2. \tag{23a}$$

It is a standard test function in optimization and was proposed by Rosenbrock [16]. Its difficulty arises from the fact that the minimum $x_{opt} = (x_1, x_2) = (1, 1)$, $g(x_{opt}) = 0$, is localized in a steepest parabolic valley with a flat bottom (see Fig. 6), approximately determined by $x_2 = x_1^2$. Most optimization methods at the initial stage of the search quickly locate the valley and reach its bottom. Due to a curved gradient path with relatively steep walls of the valley, all nongradient optimization algorithms are very slow, the direction of search must be changed permanently to reach the minimum. In other words, changes of variables are strongly correlated. For people working with genetic algorithm or simulated annealing, using either binary or real representation of variables,

Table 2
CPU times of regression model calculations

Optimal parameters: $f_{opt} = 2.981 \times 10^{-5}$, $\beta_1 = 0.000414$, $\beta_2 = 3.80180$, $\beta_3 = 2.06087$, $\beta_4 = 0.22289$	
Method	Time (s)
CRS [3]	—
CRS_{mod} [4]	140
SSA	32
PSSA	37

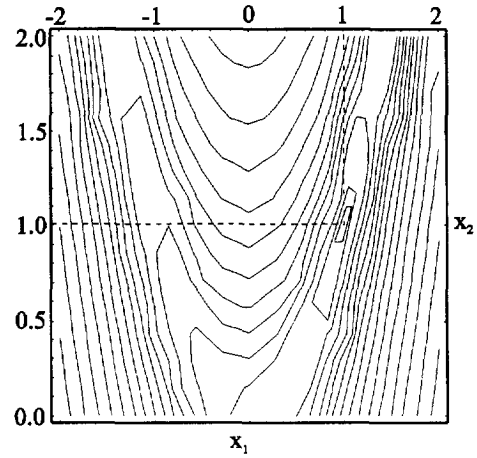

Fig. 6. Contour plot of Rosenbrock's function determined by Eqs. (23a) and (23b). It is unimodal function with minimum (global) at $x_{opt} = (1,1)$, where $f(x_{opt}) = 0$.

this function is almost a 'nightmare', since changes in these methods are independent and uncorrelated. Simple generalization of the function for more than two variables is

$$g(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n-1} \left[100 \cdot \left(x_i^2 - x_{i+1} \right)^2 + (1 - x_i)^2 \right], \tag{23b}$$

where single variables are bounded by $-10 \leq x_i \leq 10$, for $i = 1, 2, \ldots, n$. This function of n continuous variables is unimodal and has the only one minimum, at $x_{opt} = (1, 1, \ldots, 1)$, $g(x_{opt}) = 0$. Then the objective function is determined as a logarithm of n-dimension Rosenbrock's function

$$f(x_1, x_2, \ldots, x_n) = \ln[g(x_1, x_2, \ldots, x_n)]. \tag{24}$$

Simple simplex method and CRS versions [3] failed in finding the correct minimum for $n \geq 3$. They are terminated at the valley bottom relatively far from the point x_{opt}, where the L_1 distance between simplex (or population) points x_{min} and x_{max} is smaller than a prescribed precision ε. On the other hand, the present simulated annealing methods (SSA and PSSA) surprisingly provide after a few hundreds iterations the correct solution $x_{opt} = (1, 1, \ldots, 1)$, for $2 \leq n \leq 10$. Similarly, as for the above examples of regression analysis tasks, the suggested methods SSA and PSSA are able to overcome the numerical convergence difficulties of simple simulated annealing methods. This is true due to the fact that entirely random perturbations of variables are substituted by reflection points that are placed in promising directions for which the objective function given by Eq. (24) with very high level of probability is decreasing.

3.3. Model highly multimodal function

Let us define the following trigonometric function damped by an exponential term

$$g(x) = 0.940249612 + e^{-0.1x^2} \cdot \sin(10x) \cdot \cos(8x). \tag{25}$$

Its plot is displayed in Fig. 7. We see that it is highly multimodal function with many minima, the first three lowest minima are

$x^{(1)} = -0.7844416$, $g(x^{(1)}) = 1.75 \times 10^{-10}$,
$x^{(2)} = -0.4397995$, $g(x^{(2)}) = 0.072921$,
$x^{(3)} = -1.1293051$, $g(x^{(3)}) = 0.161959$.

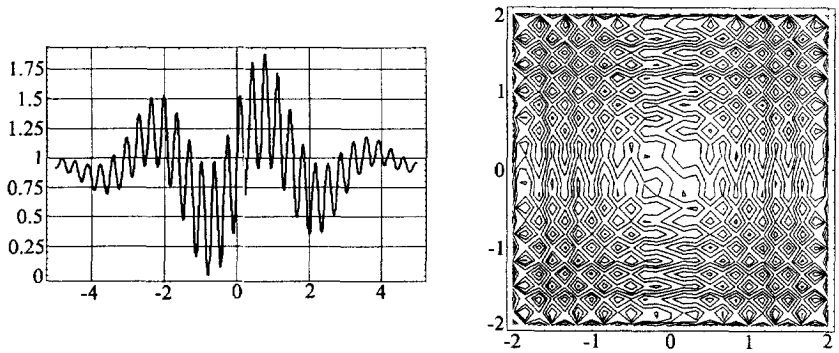

Fig. 7. Plot of multimodal function defined by Eq. (25). Right contour plot corresponds to 2-dimensional function determined by Eq. (26).

An n-dimension highly multimodal objective function is determined as follows

$$f(x_1, x_2, \ldots, x_n) = \ln\left(\sum_{i=1}^{n} g(x_i)\right), \tag{26}$$

where single variables are bounded by $-10 \leq x_i \leq 10$, for $i = 1, 2, \ldots, n$. This function has the global minimum at x_{opt} composed of all entries equal to $x^{(1)}$, its functional value at the point x_{opt} is $f(x_{opt}) = n \times 1.75 \times 10^{-10}$. A similar function has been recently successfully used by the author for demonstration of effectiveness and robustness of the so-called messy simulated annealing [17]. We have tried without success to use the standard version of simplex method (see algorithm 1) for the correct solution of the optimization task with objective function given by Eq. (26) for $n \geq 2$. Most often the simplex method has achieved only suboptimal points and moreover with very poor convergence properties. In particular the size of simplex has been quickly reduced and then it rolled to the closest local solution. Slightly better behaviour is manifested by CRS method [3] and its randomized version [4]. In particular, for $2 \leq n \leq 3$, these methods provide with high level of probability (80–90%) the correct solutions. Unfortunately, this correctness substantially decreases if $n \geq 4$, e.g. for $n = 4$ about 20–30% resulting solutions are correct. An effectiveness of CRS for $n \geq 4$ may be partially increased by enlarging the population dimension p, but then method is plagued by a slower convergence. The present hybrid methods of simulated annealing successfully overcome convergence difficulties of simple simplex method and CRS. The following set of basic control parameters of simulated annealing has been used

$$k_{max} = 1000, \quad T_{max} = 0.1, \quad T_{min} = 0.001, \quad \alpha = 0.99, \quad p = 10 \times n, \quad r = 10. \tag{27}$$

The obtained results from 20 runs are summarized in Table 3. We see that PSSA provides for all $2 \leq n \leq 6$ correct global minimum, while its simpler version SSA provides correct solutions only for $2 \leq n \leq 4$, for $n \geq 5$ its effectiveness is decreasing.

Table 3
Correctness (in %) of SSA and PSSA for optimization of multimodal function specified by Eq. (26)

n	SSA	PSSA
2	100%	100%
3	100%	100%
4	100%	100%
5	95%	100%
6	90%	100%

4. Discussion and summary

The proposed hybrid of simplex method and simulated annealing represents effective and robust stochastic optimization method, which is able to solve correctly many tasks that are notoriously difficult for standard (gradient as well as nongradient) optimization methods. As a prototype of our approach Price's controlled random search [3] (see also Ref. [4]) is used. This pioneering work may be now considered as one of forerunners of modern evolutionary approaches like the genetic algorithms [5], simulated annealing [6], evolution strategy [7], etc. Moreover, interesting resemblance between CRS and GA may be outlined. Both methods use the concepts of population and reproduction. In GA the population is composed of binary vectors whereas in CRS the population is composed of points (vectors with real entries). This difference, on the first sight principal, is removed in current versions of GA [18], there are successfully used chromosomes represented by vectors with real entries. The process of reproduction in GA is performed by the so-called selection, crossover, and mutation, whereas in CRS it corresponds to a construction of reflection points from the randomly selected simplex points. Similarly as in the previous case, current versions of GA with chromosomes represented by real vectors [18], use simple algebraic operations (e.g., arithmetic mean) over them to create offspring. In the framework of CRS we may speak about a *simplex mating*, the pair reproduction (GA) is changed by the $(n+1)$-tuple reproduction (reflection). The above considerations on the resemblance of GA and CRS may turn our attention to a modification of CRS in a way that common similarities are enforced. In particular, the modifications of CRS should cover the selection of simplex points from the population and the returning of reflection points to the population. Both these requirements may be done by a quasirandom manner reflecting the quality of single points (i.e., fitness of chromosomes in GA).

An effectiveness and robustness of the proposed hybrid methods SSA and PSSA is numerically tested by three examples that are notoriously difficult for single simplex method and simulated annealing. We have demonstrated that these methods are sufficiently effective and robust to produce correct solutions. The proposed method contains a few parameters that should be properly tuned to get its best performance. In particular, the performance is fairly sensitive on parameters $k_{\max}$, $T_{\max}$, $T_{\min}$, and α, this means that their proper numerical values must result from some preliminary 'trial-and-error' calculations. The parameter $k_{\max}$ should be sufficiently large so that the Metropolis algorithm will produce states (reflection points that are accepted by the Metropolis criterion) from search space with probability distribution closely related to the Boltzmann distribution. In all our examples we have used $k_{\max} = 10^3$. The starting temperature $T_{\max}$ is roughly determined [7] so that about 50% reflection points are accepted by the Metropolis criterion. If $T_{\max}$ is increased to greater values, then almost all new reflection points are accepted by the Metropolis criterion and none new information is incorporated by the process of simulated annealing. According to a physical analogy, we may say that the system is melted or even evaporated, and it is impossible to look for a formation of new structures with lower energy. On the other hand, if $T_{\max}$ is very small, then the system is 'frozen' to a current state, Metropolis criterion accepts almost entirely reflection points with values of objective function smaller than that one assigned to the current state. In this region of smaller temperatures the method of simulated annealing starts becoming to behave like a local search algorithm (i.e., only better solutions are accepted for further extension of local search). Finally, the parameter α used for multiplicative decreasing of temperature (i.e., $T \leftarrow \alpha * T$) should be selected so that the temperature is sufficiently slowly decreased avoiding large temperature 'jumps', we set this parameters in all our calculations to $\alpha = 0.99$, i.e., the temperature T is decreased by 1% relative steps. The size p of population P represents also a very important control parameter of SSA. In general, it should be greater than the simplex dimension $n+1$, i.e., $p > n+1$, where n is the number of independent variables of objective function f. Its setting to great values increases the diversity of search space (i.e., population) and this increases the chance to get correct solutions, but is accompanied with slower convergence manifested by the fact that a population 'diameter' expressed by $|x_{\min} - x_{\max}|$ (see Fig. 3) is very slowly vanishing in the course of simulated annealing. In all our calculations a proper value of the dimension p was set to $p = 10 \times n$, that is the size of population is roughly ten times greater than the size of simplex. In the framework of PSSA the population P is divided into r subpopula-

tions of the same size, it means that the size (cardinality) of P is determined by $|P| = r \times p$, where p is the size of subpopulations. We have used $r = 10$ and the size p was determined by the above approach $p = 10 \times n$ used in SSA. It means that ten quasi-independent simulated annealings are performed over subpopulations composed of $p = 10 \times n$ points, and subpopulations randomly interact so that the best solutions are exchanged with simultaneous elimination of the worst ones. The probability of this exchange P_{exch} should be very small, we set $P_{\text{exch}} = 0.001$, in the opposite case the subpopulations are quickly equalized (i.e., all subpopulations are composed of almost the same points) and the simulated annealing is stopped at an incorrect solution. An implementation of SSA and PSSA is very simple and does not require any special 'tricks' and approaches. The algorithmization of PSSA (see algorithm 4) is done so that it can be performed on computers with parallel architecture of processors, an activation of the procedure `single_step_SSA` may be mapped by compiler for different subpopulations to different processors. This means that the time consuming of PSSA is roughly the same as for SSA, but with greater effectiveness than a simple SSA.

References

[1] J.A. Nelder, R. Mead, Simplex method for function minimization, Computer J. 7 (1965) 308–313.
[2] S.L.S. Jacoby, J.S. Kowalik, J.T. Pizzo, Iterative Methods for Nonlinear Optimization Problems, Prentice Hall, Englewood Cliffs, 1972.
[3] W.L. Price, A controlled random search procedure for global optimization, Computer J. 7 (1965) 347–370.
[4] I. Křivý, J. Tvrdík, The controlled random search algorithm in optimizing regression model, Computational Statistics and Data Analysis 20 (1995) 229–234.
[5] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA, 1989.
[6] P.J.M. van Laarhoven and E.H.L Aarts, Simulated Annealing: Theory and Applications, Reidel, Dordrecht, 1987.
[7] H.-P. Schwefel, Numerical Optimization of Computer Models, Wiley, New York, 1981.
[8] S.D. Brown, S.T. Sum, F. Despagne, B.K. Lavine, Chemometrics, Analytical Chemistry 68 (1996) R21–62.
[9] I.E. Frank, Modern nonlinear regression methods, Chemometrics and Intelligent Laboratory Systems 27 (1995) 1–19.
[10] J. Acevedo, E.N. Pistikopoulos, Computational studies of stochastic optimization algorithms for process synthesis under uncertainty, Computers and Chemical Engineering, Supplement A 20 (1996) S1–S6.
[11] J.H. Kalivas (Ed.), Adaption of Simulated Annealing to Chemical Optimization Problems, Series Data Handling in Science and Technology 15, Elsevier Science B.V., Amsterdam, 1995.
[12] A. San Vicente, A. Arranz, J.M. Moreda, J.F. Arranz, Simplex and classical methods for the selection of parameters for the adsorptive stripping voltammetric determination of nitralin, A comparative study, Analytica Chimica Acta 298 (1994) 87–90.
[13] U.H.E. Hansmann, Y. Okamoto, Comparative study of multicanonical and simulated annealing algorithms in the protein folding problem, Physica A 212 (1994) 415–437.
[14] R.R. Mayer, P.M. Roth, Modified damped least squares. An algorithm for non-linear estimation, J. Institute Mathematics and Applications 9 (1972) 218–233.
[15] J. Militký, M. Meloun, Modus operandi of the least square algorithm MINOPT, Talanta 40 (1994) 269–277.
[16] H.H. Rosenbrock, An automatic method for finding the greatest or least value of a function, Computer J. 3 (1960) 175–184.
[17] V. Kvasnička, J. Pospíchal, Messy simulated annealing, J. Chemometrics 9 (1995) 309–322.
[18] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin 1992.