Aplikace matematiky

Jozef Masarik

A thermodynamically motivated optimization algorithm: Circular wheel balance optimization

Aplikace matematiky, Vol. 30 (1985), No. 6, 413–424

Persistent URL: http://dml.cz/dmlcz/104171

A THERMODYNAMICALLY MOTIVATED OPTIMIZATION ALGORITHM: CIRCULAR WHEEL BALANCE OPTIMIZATION

JOZEF MASARIK

(Received March 26, 1984)

1. INTRODUCTION

Combinatorial optimization problems form a group of problems with a large number of practical applications. Recently a new thermodynamically motivated optimization method has been developed by Černý [1] and Kirkpatrick et. al. [2], which has been applied to find suboptimal solutions to the travelling salesman problem. In this paper we use a similar idea for solving the following optimization problem.

Let $(m(c_i))_1^N$ be a set of positive real numbers. Let $(c_i)_1^N$ be a permutation of the integers 1, 2, ..., N. Let us construct the sum

$$F[(c_i)] = \left| \sum_{k=1}^{N} m(c_k) \,.\, \exp\left\{ \mathrm{i} \frac{2\pi}{N} k \right\} \right| . \tag{1}$$

The problem is to find the permutation $(c_i)_1^N$ for which $F[(c_i)]$ is minimal.

One of the applications of this problem is the optimization of the static balance of a steam turbine circular wheel. This problem arises from unequal weights of the turbine paddles which are uniformly located at apriori determined places on the perimeter of the circular wheel. If the paddles are located in a random sequence, it could lead to a substantial unbalance of the wheel. If $m(c_i)$ represent the masses of the paddles then $F[(c_i)]$ represents the unbalance of the particular arrangement $(c_i)_1^N$ of the paddles.

The aim of our paper is to bring further evidence for the conjecture that the thermodynamically motivated optimization procedure can be effectively used to approach a wide class of complicated optimization problems.

The paper is organized as follows. In the next part we present the details of the algorithm and its application to the problem which was formulated abowe. The results of computations are presented in Sect 3, while Sect. 4 contains comments and conclusions. For the sake of completeness, we discuss the thermodynamical motivation of the algorithm in Appendix.

2. ALGORITHM

We begin with introducing the notations. We consider a circular wheel with N paddles on its perimeter. The paddles have masses $m(i)$ $i = 1, 2, \ldots, N$. Let $(s_i)_1^N$,

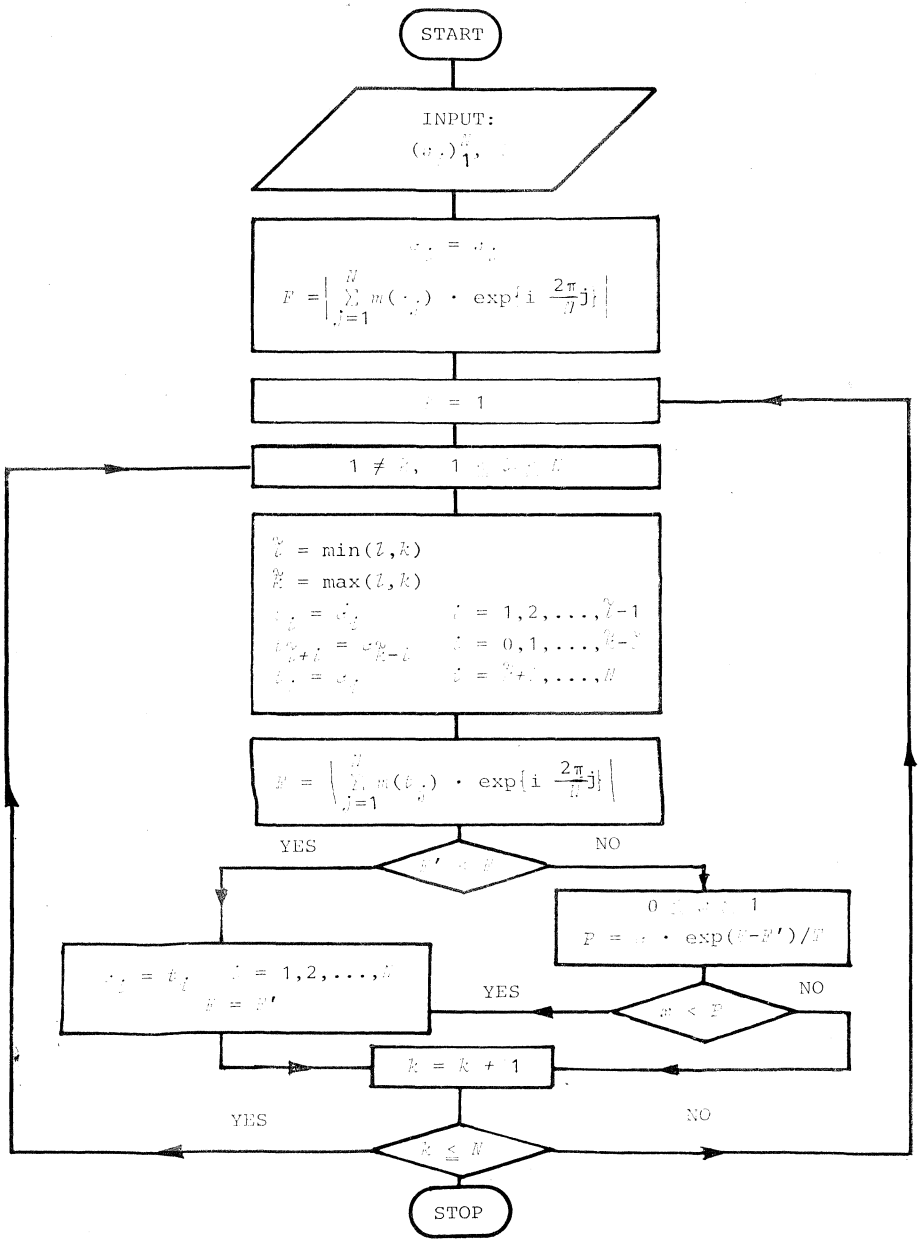

Fig. 1. Flow-chart of the algorithm discussed.

$(c_i)_1^N$, $(t_i)_1^N$ denote permutations of the integers 1, 2, ..., N. We shall call the permutations s, c and t the *starting*, *current* and *trial permutation*, respectively.

The algorithm (the corresponding flow-chart is presented in Fig. 1) which we shall discuss here produces a sequence (in principle infinite) of permutations $(c_i)_1^N$. The corresponding values $F[(c_i)]$ are expected to form a sequence which globally has decreasing character. Locally, however, fluctuations with increasing value of $F[(c_i)]$ will appear.

STEP 0: Choose an arbitrary starting permutation $(s_i)_1^N$. Choose a real number T.

Comments: Let in our case $N = 10$ and $m_i = i$. Then the starting permutation can be for example 1, 2, ..., 10 (Figure 2). T is called the temperature of the system. The meaning of this parameter will be explaned in Appendix.

STEP 1: Set $c_i = s_i$ for $i = 1, 2, \ldots, N$.
Calculate the corresponding unbalance

$$F = F[(c_i)] = \left| \sum_{j=1}^{N} m(c_j) \cdot \exp\left\{ \mathrm{i} \frac{2\pi}{N} j \right\} \right| .$$

Comments: In the general case $F[(c_i)]$ is called the cost function.

STEP 2: Set $k = 1$.

STEP 3: Generate randomly an integer l, $1 \leqq l \leqq n$, $l \neq k$.

STEP 4: Construct a trial permutation from the current permutation as follows
find

$$\tilde{l} = \min(l, k)$$
$$\tilde{k} = \max(l, k)$$

set

$$t_i = c_i \quad \text{for} \quad i = 1, 2 \ldots \tilde{l} - 1$$

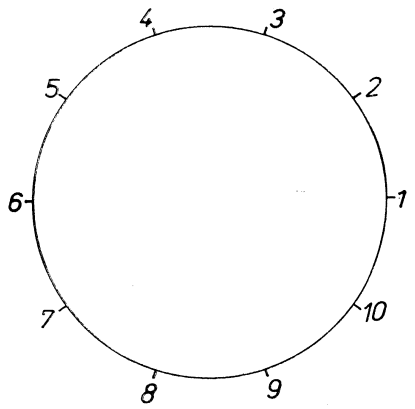

Fig. 2. Starting permutation.

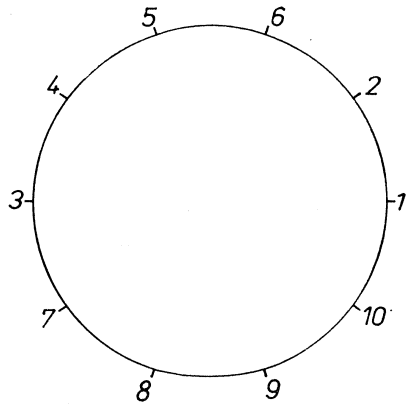

Fig. 3. Trial permutation.

$$t_{\tilde{l}+i} = c_{\tilde{k}-i} \quad \text{for} \quad i = 0, 1 \ \ldots \ \tilde{k} - \tilde{l}$$
$$t_i \quad = c_i \quad \text{for} \quad i = \tilde{k} + 1, \ldots, N$$

Comments: Let for example in our case $k = 3$ and $l = 6$; then $\tilde{l} = 3$ and $\tilde{k} = 6$ and the trial permutation $(t_i)_1^N$ is 1, 2, 6, 5, 4, 3, 7, 8, 9, 10 (Figure 3). This construction interchange the paddles in the k-th and l-th positions and the paddles between then are taken with reversed order.

STEP 5: Calculate the unbalance corresponding to the trial permutation

$$F' = F[(t_i)] = \left| \sum_{j=1}^{N} m(t_j) \cdot \exp\left\{ \mathrm{i} \frac{2\pi}{N} j \right\} \right| .$$

STEP 6: If $F' \leqq F$ go to Step 7.

Comments: In this case, the trial permutation is accepted as a new current permutation and is used as the starting point for the generation of the next trial permutation. However it could be dangerous to accept only rearrangements that lower the value of the cost function since the computation might end in local minimum. In order to increase the probability of getting out of the local minimum region also the rearrangements leading to an increase of the cost function value are considered for the next step.

STEP 6: If $F' > F$ generate a random number x, $0 \leqq x \leqq 1$, then if $x < \exp(F - F')/T$ go to Step 7 otherwise go to Step 8.

Comments: The case $F' > F$ is treated probabilistically, it is here where the analogy with thermodynamics is used. The trial permutation will be accepted with the probability given by the *Boltzmann-Gibbs factor* $\exp\{-\Delta F/T\}$, which means that with a certain probability we accept also the rearrangements that increase the value of the cost function. This is one of the most important properties of the algorithm.

STEP 7: Set $c_j = t_j$ for $j = 1, 2, \ldots, N$
Set $F = F'$

STEP 8: Increase k by one.
Then if $k \leqq N$ go to Step 3 otherwise go to Step 2.

The temperature is a formal parameter controlling the mobility of the system, i.e. the flexibility to accept a change increasing the cost function value. At the beginning of the minimization T is set at a reasonably high value so that the system is able to pass local minima near the starting permutation. One can make some educated guess in order to find an appropriate value of the temperature T, or one can find such value experimentally by trial and error method. Keeping the temperature T constant one shall reach, after a sufficient number of trial permutation generations, an equilibrium state, i.e. a state of the system in which the value of the cost function

fluctuates around a certain mean value. The reason for this is discussed in Appendix. Roughly speaking we simulate the thermal motion of some statistical system and such system spontaneously approach an equilibrium state. After reaching the equilibrium the algorithm would generate an infinite sequence of current permutations with the corresponding values $F[(c_i)]$ fluctuating around some mean value. Therefore one should stop the algorithm, decrease the temperature and start the calculation again till a new equilibrium state is reached. Since T was decreased, the mean value of $F[(c_i)]$ in the new equilibrium state should be lower than the previous one. The simulation must proceed at each temperature long enough in order the system could reach the equilibrium state. At some stage one has to decide to stop the process of lowering the temperature and thus to stop the whole calculation. The criterion might be the fact that values of $F[(c_i)]$ acceptable for practical purposes were reached. An other reason for stopping the calculation might be the computational time.

In order to increase the acceptance rate of the trial permutations one can try to repleace Step 3 by the following

STEP 3: Generate randomly an integer j such that $1 \leqq m \leqq n < N/2$ where n is a fixed number and $m = \min\{|j-1|, |j+i+N|, |j-i-N|\}$.

Comments: Rearrangements in the sequence of paddles are then more local. Experimentally it can be found that such rearrangements are useful in situations near the minimum.

3. COMPUTATIONAL EXPERIENCE

To investigate the possibilities of the above discussed algorithm we observed its behaviour in several problems, where the exact solutions were known because of symmetry reasons. However, it is clear that the fact of some symmetry in the data has no significance for the work of the algorithm. Thus if the algorithm finds good suboptimal solutions for such problems we can hope that it is able to find solutions for arbitrary other problems, as well.

Example 1. We consider a *circular wheel with* 10 paddles with masses $m_i = i$, $i = 1, 2, \ldots, 10$ uniformly located on the perimeter of the circular wheel. The optimal sequence e.g. 10, 7, 6, 3, 2, 9, 8, 5, 4, 2 give the unbalance $F[(c_i)] = 0$. *Unbalance calculated from the starting permutation* (Figure 4) *is* 7, 49. We applied the proposed algorithm to our optimization problem starting at the temperature $T = 1{\cdot}0$. The results of the optimization are presented in Figure 5. In this figure we plot the value of the unbalance corresponding to the current permutation after every 100 Monte Carlo trials. Quantitatively the picture has the character as expected by the thermodynamical analogy.

The starting tempetature $T = 1{\cdot}0$ seemed to be too high (large fluctuations), therefore we cooled the system after 200 Monte Carlo trials to the temperature $T = 0{\cdot}5$. At this temperature an equilibrium state seemed to be reached after 160

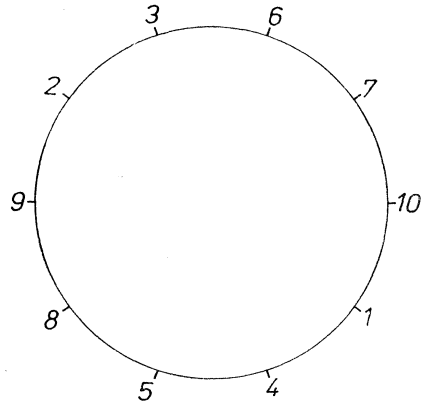

Fig. 4. Starting permutation — Example 1.

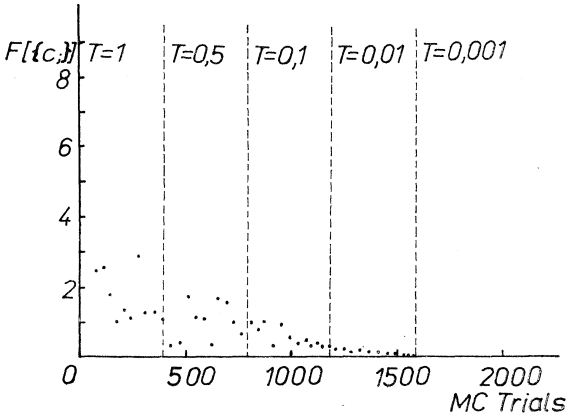

Fig. 5. Results of optimization for Example 1.

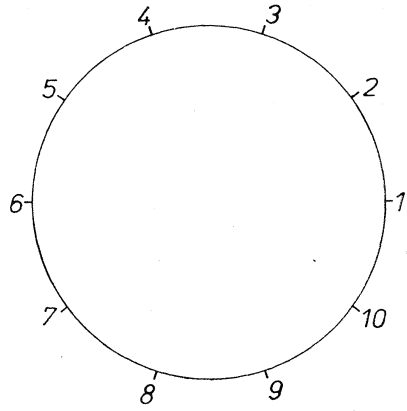

Fig. 6. Optimal solution for Example 1.

trials. We repeated this procedure again, choosing the temperature $T = 0{\cdot}1$, $0{\cdot}01$, $0{\cdot}001$. *After* 1600 *Monte Carlo trials the value of unbalance* 0·0 *was reached which means that the optimum was found.* The optimal permutation of the paddles is presented in Figure 6. This was quite a nice result, but the problem was rather simple.

Example 2. We consider the *circular wheel with* 40 *paddles*, with pairs of paddles having equal masses. In the starting permutation the paddles were located in such a way that paddles with equal masses were side by side (1·34, 1·34, 1·68, 1·68, ...). It is clear that the optimal permutation is that when the paddles with equal masses are put in the opposite positions. In this case the unbalance is 0·0. We began the optimization process at the temperature $T = 1{\cdot}0$. After 10 000 Monte Carlo trials we decreased the temperature to the value 0·5. We repeated the procedure choosing $T = 0{\cdot}1$, $0{\cdot}01$, $0{\cdot}001$, $0{\cdot}0001$. *The modified algorithm with Step* 3′ *was used at*

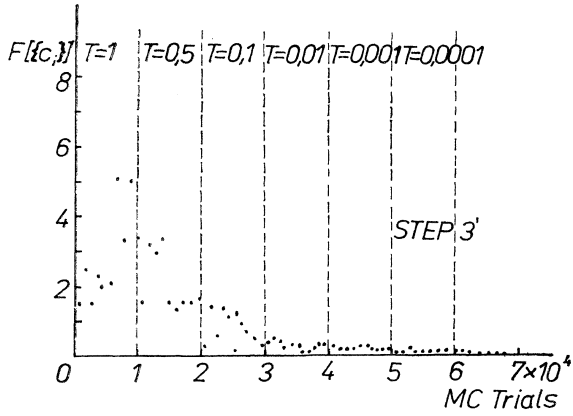

Fig. 7. Results of optimization for Example 2.

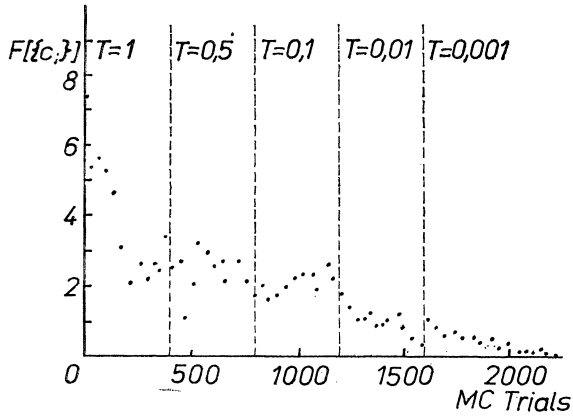

Fig. 8. Results of computation for Example 1, with Step 4′ instead Step 4.

temperature 0·0001. The results are presented in Figure 7. *In this case the optimum was found again after some* 67 000 *Monte Carlo trials.* The results of some other test runs are presented in Table I. In the last example from Table I the total minimum was unknown (the masses were choosen randomly).

In some of the cases presented in Table I we choose the masses of paddles to be almost equal, in other cases the masses were fairly different. The algorithm works equally well in all cases.

Table I

NT	TNTRY	Starting unbalance	The minimum found	Absolute minimum
10	1 600	7·49165	0·0	0
20	17 000	1·70161	0·0	0
40	67 000	4·66912	0·0	0
60	98 000	2·56456	2×10^{-6}	0
30	10 000	17·23736	$1{\cdot}1 \times 10^{-6}$	?

NT — number of paddles
TNTRY — number of trials

4. COMMENTS AND CONCLUSIONS

In the previous section we presented several examples indicating that by using the proposed algorithm one can get suboptimal or even optimal solutions of the given problem.

Our specific optimization problem belongs to a much larger group of combinatorial optimization problems [5]. These problems have many practical applications e.g. in backboard wiring [2], time scheduling or controlling a multiprocessor system [6]. The size of the problems arising in practice is in general too high to allow for exact solution methods. In practice (it is also our case) it is sufficient to find an approximate solution of the problem with using minimum of computation time. For the solution of problems of this type the proposed algorithm is very appropriate.

The algorithm was originally proposed [1], [2] for the *travelling salesman problem*: Given a list of N cities and a means of calculating the cost of travelling between any two cities one has to plan the salesman's route which will pass through each city once and finally return to the starting point, minimizing the total cost. In the algorithm the matrix structure of the travelling salesman problem is not important at all, so one can expect that the algorithm can be useful for other combinatorial problems as well. Our results suggest that it is indeed so. The slow increase of effort along with the increasing number of elements of the optimized system gives promise that the algorithm could serve as a widely applicable optimization method

not only for the travelling salesman problem and for our problem but also for a large group of problems characterized by a large combinatorial complexity. This belief is based also on the fact that the algorithm operates only with two objects: with the *configuration* of the system and with the *objective function* which is minimized.

One of the open problems is the universality of this algorithm with respect to the generation of the trial permutation from the current one. In Section 2 we discussed a modification of Step 3, in which the change of the permutation is more local. The substantial modification of algorithm can be performed in Step 4. If we have a configuration with N elements, so there are $N!$ possibilities how to generate a trial permutation from the current one. Some possibilities were investigated by Croes [8] and Lin [7]. The problem is not trivial since one can easily find rearrangements which empirically significantly lower the convergence rate of the method:

Instead of Step 4 we also tried to use

STEP 7: Construct the trial permutation as follows

$$t_i = c_i \quad i = 1, 2, \ldots, \tilde{l} - 1, \tilde{l} + 1, \ldots, \tilde{k} - 1, \tilde{k} + 1, \ldots, N$$
$$t_l = c_k$$
$$t_k = c_l$$

Here only the $\tilde{l}$th and $\tilde{k}$th positions are interchanged. Modifying the algorithm in this way we obtained, for the case described as Example 1, the results presented in Figure 8. We can see that the qualitative behaviour is the same as in Figure 5. Quantitatively, however, the results are substantially different (slower convergence), in agreement with what was found in [7].

To conclude: *Our results support the conjecture* [1], [2] *that the thermodynamically motivated optimization algorithm is a powerful and very general method to approach combinatorial optimization problems.* The results of applications to various problems so far do not even indicate the necessity to look for alternatives for "STEP 4" (a move from current to trial permutation). However, we feel that it is necessary to keep in mind that the universality of the algorithm need not be so strong in this respect and it is always good to try alternative moves e.g. so discussed by Lin and Kerninghan [7].

APPENDIX: THERMODYNAMICAL MOTIVATIONS

Statistical Thermodynamics [3] is one of the central disciplines of theoretical physics. It contains many tricks for extracting properties of a macroscopic system from the microscopic average. Because the number of atoms is of the order 10^{23} per cubic centimeter, only the most probable behaviour of the macroscopic system at a given temperature can be observed in practice. This can be characterized by an average and small fluctuations about the average behaviour of the system.

One of the main principles of statistical thermodynamics says that a large system spontaneously approaches the equilibrium state in most cases irresepectively of the

initial state. The probability to find the system in a particular microscopic configuration i is given by the *Boltzmann-Gibbs distribution*

$$F_i = C \,.\, \exp\{-E_i/T\}$$

where E_i is the energy of the configuration, T is the temperature and C is the normalization constant. The equilibrium state is characterized by a certain mean value of energy. This mean energy is a function of the temperature. It is clear from the Boltzmann-Gibbs formula that lower mean energies correspond to lower temperatures.

Let us imagine that we want to find experimentally the state of the system with the lowest possible energy, the so called ground state. According to the Boltzmann-Gibbs formula one has to freeze the system to the temperature $T = 0$ and the system would spontaneously tend to the ground state. But in practice a low temperature is not a sufficient condition for finding the ground state of a matter characterized by the minimal value of the energy. As an example one can imagine an experiment of growing a single crystal from a melt. To get a pellucid crystal one must first melt the substance, then lower the temperature slowly and spend a long time at temperatures in the vicinity of the freezing point. If we lower the temperature rapidly, the resulting crystal will have many defects, or the substance may form a glass with no crystalline structure (it is a metastable state, which means it is a local optimum).

Instead of doing this cooling procedure experimentally one can try to use numerical procedure simulating the spontaneous transition to the equilibrium state.

Metropolis et. al. [4] proposed a simple procedure that can be used for numerical simulation of a collection of atoms in equilibrium state at a given temperature. This procedure consists of a loop over a random displacement generator that produces changes in energy. The algorithm starts from an arbitrary state with energy E and temperature T. To this state we apply the random displacement and calculate the corresponding change dE of the energy of the system. If this change is less than zero, the displacement is accepted and the value of the energy is reduced by dE, the case when the change of energy is nonnegative is treated probabilistically according to the *Boltzmann-Gibbs law*

$$P(dE) = C \,.\, \exp\{-dE/T\} \,.$$

The probability $P(dE)$ is compared with the random variable x uniformly distributed in $\langle 0, 1\rangle$ and the displacement is accepted only if this random variable x is less than $P(dE)$.

Having a general optimization problem we do not usually speak about energy but abaut cost function and not about state but for instance about values of some parameters. But this is a mere difference in the wording. We can use the Metropolis algorithm for finding the ground state of a physical system. It is true, however, that the temperature in optimization problems is not defined in a natural way. It is rather arbitrary but a very important parameter for optimization problems. An algorithm which would accept only rearrangements that lower the value of the cost function

is equivalent to our algorithm at $T = 0$. One should not be surprised that computations would end in a local minimum. But at nonzero temperature the system fluctuates about the equilibrium state. Therefore if we accept with certain probability also rearrangements which increase the objective function value, the transitions out of a local minimum are possible. Of course one can never gurantee reaching the absolute minimum, but chanches are much higher.

Acknowledgement. The author is indebted to V. Černý for many useful discussions and comments which helped to clarify a lot of points of the argumentation. Suggestions by J. Pišút are also acknowledged.

References

[1] *V. Černý:* A Thermodynamical Approach to The Travelling Salesman Problem: An Efficient Simulation Algorithm. Report, Institute of Physics and Biophysics, Comenius University, Bratislava, 1982, to be published in Journal of Optimization Theory and Applications.

[2] *S. Kirkpatrick, S. D. Gelatt, M. J. Vecchi:* Optimization by Simulated Annealing: Science, 220 (1983), 671—680.

[3] *C. Kittel:* Thermal Physics. J. Wiley and Sons, New York, 1969.

[4] *N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller* and *E. Teller:* Equation of state Calculations by Fast Computing Machines. J. Chem. Phys., 21 (1953), 1087—1092.

[5] *R. E. Burkard* and *F. Rendl:* A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems. Report 83-12, Institut für Mathematik, Technische University, Graz, 1983.

[6] *V. Černý:* Multiprocessor System as a Statistical Ensemble: a Way Towards General-purpose Parallel Processing and MIND Computers?. Report, Institut of Physics and Biophysics, Comenius University, Bratislava, 1983.

[7] *S. Lin* and *B. W. Kernighan:* An Effective Heuristic Algorithm for The Travelling Salesman Problem. Opns. Res., 21 (1973), 498—516.

[8] *A. Croes:* A Method for Solving Travelling Salesman Problems. Opns. Res., 5 (1958), 791—812.

Súhrn

TERMODYNAMICKY MOTIVOVANÝ OPTIMALIZAČNÝ ALGORITMUS: OPTIMALIZÁCIA VYVÁŽENIA PARNÉHO KOLESA

JOZEF MASARIK

V článku uvádzame Monte Carlo algoritmus na hľadanie suboptimálneho alebo dokonca optimálneho riešenia pomerne širokej triedy komplikovaných optimalizačných úloh charakterizovaných veľkou kombinatorickou komplexnosťou. Tento algoritmus bol aplikovaný na jeden konkrétny problém — optimalizáciu statického vyváženia obežného kolesa parnej turbíny. Pomalé narastanie času potrebného na

optimalizáciu v závislosti so vzrastaním rozmerov optimalizačnej úlohy nás vedie k presvedčeniu, že termodynamicky motivovaný optimalizačný algoritmus môže byť pomerne univerzálny a efektívny optimalizačný algoritmus.

Author's address: Dr. *Jozef Masarik,* Katedra jadrovej fyziky UK, Mlynská dolina, 842 15 Bratislava.