

SIMULATED ANNEALING CONSTRUCTION OF SHORTEST PATHS ON INCOMPLETE GRAPHS

Jiří Pospíchal*, Vladimír Kvasnička†

Abstract: Simulated annealing construction of shortest (spanning/nonspanning and closed/open) paths on general connected graphs is discussed. A brief graph-theoretical analysis of the problem is given. A theorem has been proved that for connected graphs the shortest paths are semielementary, that is each edge on the path is visited at most twice in opposite directions. This observation considerably reduces the search space. Tasks may be further specified depending on whether the initial and terminal vertices are given or not. Similarly, in construction of shortest open paths a subtask is considered when the path must visit a prescribed subset of graph vertices. Illustrative calculations demonstrate that the proposed method results for incomplete graphs in the paths that are closely related to optimal solutions.

Key words: *Simulated annealing, traveling salesman problem, shortest paths on incomplete graphs*

Received: December 5, 2002

Revised and accepted: May 20, 2003

1. Introduction

The search for shortest Hamiltonian path in a complete graph with evaluated edges has become a standard task for operations research [2]. It is still in theory non-polynomial, but many useful heuristics have been suggested to find at least a sub-optimal solution. However, there are some drawbacks in applying most of these algorithms in practical applications. One of the shortcomings is the requirement for completeness of the graph, which is usually mended by evaluation of non-existent edges by sums of the evaluation of edges on the shortest path between two vertices. Another drawback is the appearance of constraints and multiple goals,

*Jiří Pospíchal

Department of Mathematics, Slovak Technical University, 812 37 Bratislava, Slovakia, E-mail: pospich@cvt.stuba.sk.

†Vladimír Kvasnička

Department of Mathematics, Slovak Technical University, 812 37 Bratislava, Slovakia, E-mail: kvasnic@cvt.stuba.sk.

such as creating shortest paths with necessity to visit some prescribed vertices. This can be attended to by a proper definition of an objective function; however, some algorithms do not handle the changed objective function very well.

Simulated annealing [1], [6], [8], [9], [10] has already been used many times for the search of Hamiltonian paths and occasionally for finding the shortest paths with some specified requirements. Simulated annealing does not need to introduce nonexistent edges to provide a complete graph and naturally handles an objective function with multiple goal evaluation. However, the extent of the search space of this kind of tasks is tremendous and whatever approach helps to curb it is welcome. The presented approach reduces the search space by looking only for semielementary paths (an edge is visited at most twice in opposite directions), which provides solutions that are of the same quality as is the quality of solutions without restrictions to semielementary paths. The shortest spanning paths, both closed and open, and other "shortest" paths with some additional requirements must be semielementary.

2. A brief outline of simulated annealing

The simulated annealing algorithm [1], [6], [10] is based on the analogy between the simulated annealing of solids and the problem of solving the large scale optimization problem. In physics annealing denotes a process in which a solid placed in a heat bath is heated up by increasing the temperature of the bath to the maximum value at which all particles of the solid are randomly arranged, followed by a cooling period with slowly decreasing temperature of the heat bath. All particles arrange themselves in the *low energy* state of the corresponding solid, assuming that the maximum temperature is sufficiently high and the cooling is carried out sufficiently slowly. Starting off at the maximum value of temperature, the cooling phase can be described as follows: At each temperature T , the solid is allowed to reach *thermal equilibrium*, described by the probability of being in a state i with energy E_i determined by the Boltzmann distribution

$$w_T(E_i) = \frac{1}{Q(T)} \exp\left(-\frac{E_i}{k_B T}\right) / \sum_i \exp\left(-\frac{E_i}{k_B T}\right) \quad (1)$$

where k_B is the Boltzmann constant, and the summation runs over all states i of the solid. As the temperature T decreases, the Boltzmann distribution concentrates on the state with the lowest energy, and finally, when temperature approaches zero, only the state with the minimal energy has a nonzero (unit) probability of occurrence.

In order to simulate the *evolution* to thermal equilibrium of a physical system (e.g. a many-particle solid or liquid) for a fixed value of the temperature T , Metropolis [10] suggested the *Monte Carlo method*, which generates sequences of the states of the system in the following way: Given the current state of system (determined by positions of particles) a small random perturbation is generated so that particles are displaced. If the difference $\Delta E = E_{\text{perturbed}} - E_{\text{current}}$ between the perturbed state and the current state is negative ($E_{\text{perturbed}} < E_{\text{current}}$), then

the current state is replaced by the new perturbed state and the process is continued. In the opposite case, if $\Delta E \geq 0$, then the probability of acceptance of the perturbed state, $\Pr(\text{perturbed} \leftarrow \text{current})$, is given by $\exp(-\Delta E/k_B T)$

$$\Pr(\text{perturbed} \leftarrow \text{current}) = \min(1, \exp(-\Delta E/k_B T)) \quad (2)$$

This acceptance rule of new states is called the Metropolis criterion. Following it, the system eventually evolves into thermal equilibrium, and after a large number of perturbations, using the acceptance criterion (2), the probability distribution of states approaches the Boltzmann distribution (1). This form of the Monte Carlo method is known in statistical mechanics as the *Metropolis algorithm* [10]. In order to formalize the Metropolis algorithm we introduce the following notation (useful also for our forthcoming discussion on applications of the method of simulated annealing to optimization of large scale problems): The state of a system is determined by a *state* - variable x (in general, a vector composed of many real entries) and an analogue of the energy $f(x)$ (treated as a function of x). The process of perturbation of the state x onto another state x' is represented by a stochastic function $x' = O_{\text{pertur}}(x)$. The stochastic character of this function consists in random changes of the entries of x onto entries of x' .

The Metropolis algorithm can be used for the computer simulation of the method of simulated annealing. It can be now viewed as a sequence of Metropolis algorithms performed for a sequence of properly decreasing values of the temperature and moreover, an output state from the Metropolis algorithm serves as an input state for the next Metropolis algorithm. Initially, the temperature is given by a high value T_{\max} and the Metropolis algorithm is applied until equilibrium is achieved (by r_{\max} times, where r_{\max} is the parameter of the Metropolis algorithm). The temperature is then lowered in steps (e.g. by $T \leftarrow \alpha T$, where $0 \ll \alpha < 1$), with the system being allowed to approach equilibrium for each step by generating a sequence of states in the previously described way. The algorithm is terminated for some small value of temperature T_{\min} , for which virtually no deterioration is accepted anymore. The final "frozen" state is then taken as the resulting solution.

The method of simulated annealing was formulated in a general way so that "physical ballast" was removed, though some physical terminology still used plays a role of fruitful heuristics useful for better intuitive understanding of the method. The main purpose of the simulated annealing is a search for a global solution of large-scale optimization problems of the type

$$x_{\text{opt}} = \arg \min_{x \in D} f(x) \quad (3)$$

where $f(x)$ is a real function determined over the domain D (usually discrete and finite), and x_{opt} is a value of variable corresponding to the global minimum of $f(x)$ over D . The variable x is considered as a state of hypothetical physical system and function $f(x)$ expresses its "energy". Then, after the above considerations, the optimization problem (3) may be successfully approached by the method of simulated annealing. The parameter "temperature" T plays now the role of a control parameter of the method.

3. Graph theory and construction of paths on graphs

Let $G = (V, E)$ be a connected (undirected) graph [5] without multiedges composed of vertices from the vertex set V and edges from the edge set E . An edge of G , incident with vertices $v, v' \in V$, is denoted by an unordered pair $[v, v'] \in E$. The cardinalities of these two sets $|V| = p$ and $|E| = q$ correspond to the numbers of vertices and edges, respectively, in G . In our forthcoming considerations we will consider only *connected graphs*, i.e. each vertex is reachable from another vertex by a sequence of edges. A *path* $P = (v_1, v_2, \dots, v_k)$ of length $|P| = k - 1$ on the graph G is an alternating sequence of $k - 1$ edges $[v_1, v_2], [v_2, v_3], \dots, [v_{k-1}, v_k]$ and k vertices $v_1, v_2, \dots, v_k (k > 0)$ so that each vertex (except the first and the last) is incident with the preceding edge, and with the succeeding edge. Vertices from path P form a vertex subset $V(P) \subseteq V(G)$, each vertex from $V(P)$ is visited by path P at least once. A *spanning path* visits each vertex of G , i.e. $V(P) = V(G)$. A path is called *closed* if the initial and terminal vertices coincide ($v_1 = v_k$), in the opposite case (i.e. the initial and terminal vertices are different, $v_1 \neq v_k$) it is called open. An *elementary path* is one in which no edge and no vertex occurs more than once, except in the case of starting and terminal vertices in closed paths. A *Hamiltonian path* is an elementary spanning path, i.e. it contains all vertices of the graph and no edge and no vertex occurs more than once. The concept of Hamiltonian path (closed or open) is of great importance in applications of graph algorithms, in particular in operations research [2]. Unfortunately, in graph theory [5] there does not exist a theorem that simultaneously specifies necessary and sufficient conditions for the existence of Hamiltonian paths on the graph. Until now the known theorems only separately specify sufficient or necessary conditions.

The following three different tasks of construction of shortest paths on graphs will be considered (see Fig. 1).

- (1) T_1 , construction of the shortest closed spanning path.
- (2) Construction of spanning shortest paths
 - (2a) $T_2^{(1)}$, initial and terminal vertices are not specified,
 - (2b) $T_2^{(2)}$, the initial vertex is specified and the terminal vertex is not specified, and
 - (2c) $T_2^{(3)}$, initial and terminal vertices are specified.
- (3) Construction of shortest paths, divided into two subtasks:
 - (3a) $T_3^{(1)}$, initial and terminal vertices are specified (standard task of algorithmic graph theory (Christofides, 1975),
 - (3b) $T_3^{(2)}$, same as above but path contains a set of prescribed vertices.

The first problem that will be considered is to suggest a method of construction of closed paths on connected graphs. Let us assign to the graph $G = (V, E)$ its directed counterpart denoted by \bar{G} so that its vertex and edge sets are determined as follows (see Fig. 2)

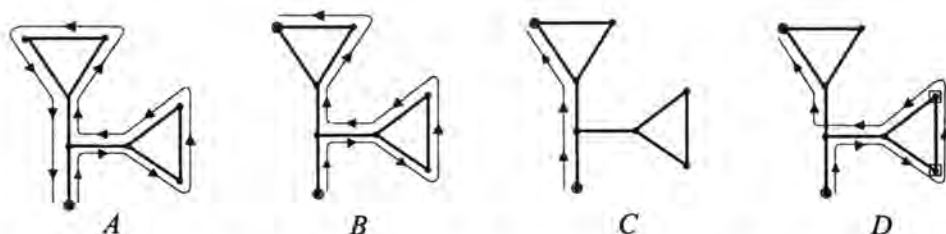


Fig. 1 Different types of paths on the given graph, heavy dots indicate initial/terminal vertices: (A) closed spanning path, (B) open spanning path, (C) open path, (D) open path visiting prescribed vertices labeled by encircled vertices.

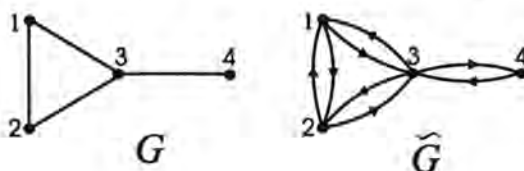


Fig. 2 A directed graph \tilde{G} may be assigned to a (undirected) graph G . Each edge $[v, v'] \in E(G)$ is substituted by a pair of directed edges $(v, v'), (v', v) \in E(\tilde{G})$.

$$v(\tilde{G}) = V(G) \quad (4a)$$

$$E(\tilde{G}) = \{(v, v'), (v', v); [v, v'] \in E(g)\} \quad (4b)$$

Each edge $[v, v'] \in E(G)$ is substituted by two directed edges $(v, v'), (v', v) \in E(\tilde{G})$. A directed path $P = (v_1, v_2, \dots, v_k)$ constructed on the directed graph \tilde{G} is considered as a sequence of $k - 1$ directed edges $(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)$. Path P on the directed graph \tilde{G} is called *semielementary* if none of its directed edges occurs more than once. Similarly, the same notion can be introduced also for the undirected graph G . Path P on G is called *semielementary* if its each edge is visited at most twice in opposite directions.

The directed graph \tilde{G} has always a closed spanning semielementary path. This property important for our forthcoming considerations can be simply proved as follows (see Fig. 3): Spanning tree of G can be always constructed by removing the so-called ring closure edges [5] (graph B in Fig. 1). An arbitrary vertex of this spanning tree may be classified as a root; it is employed as the initial (and terminal) vertex of the closed path to be constructed. By starting from the root and going successively from the left to right hand side through all branches in the tree all its vertices are subsequently visited (see graph D in Fig. 3). The thus constructed path may be interpreted with respect to the directed graph \tilde{G} as a semielementary closed path, where no directed edge was visited more than once. To summarize, we have proved that each connected graph has a closed spanning path, where edges are visited at most twice (in opposite directions).

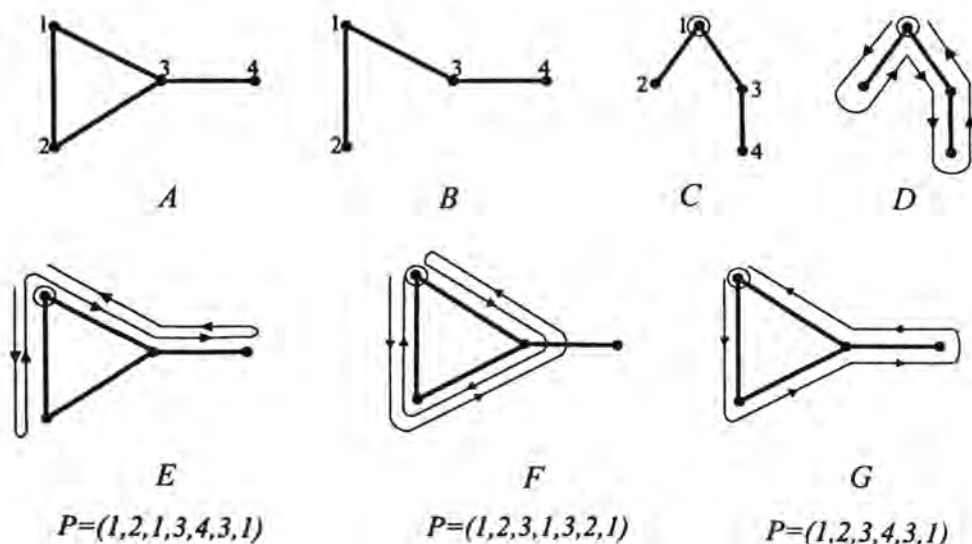


Fig. 3 A formal outline of the proof that each graph G (A) has a closed semielementary spanning path. Initially, by removing from the G ring closure edges we get its spanning tree (B), this tree is interpreted as the rooted tree (C) by selecting an arbitrary vertex as the root. Then, a closed spanning path (D) is easily constructed for the rooted tree so that all vertices are successively visited. The constructed closed spanning path on the rooted tree may be considered also as a closed spanning path of the original graph (E), its length being equal to six. If the closed path is randomly constructed, then it may happen that the path is coming to the initial vertex so that it cannot be extended further to a spanning closed path (F). Finally, the graph has a closed spanning path (G) of shorter length (five) than that one constructed previously by the spanning rooted tree.

Theorem 1. Each graph has a closed semielementary spanning path.

Similar considerations can also be used to prove that for an arbitrary pair of vertices of the connected graph G there exists an open semielementary spanning path so that the vertices of a chosen pair are initial and/or terminal.

Theorem 2. For arbitrary two vertices of a graph an open semielementary spanning path exists so that the chosen vertices are initial and/or terminal vertices of this path.

Both these theorems are of great importance for our forthcoming considerations as they ensure that for each connected graph semielementary spanning paths exist regardless whether they are closed or open and for arbitrary initial/terminal vertices.

The construction of closed semielementary paths on graph G is carried out by making use of the so-called *extension process* performed on the directed graph \vec{G} . Let $P = (v_1, v_2, \dots, v_k)$ be a semielementary path on graph \vec{G} , and \vec{E} be a subset

of $E(\tilde{G})$ composed of edges that are not contained in path \tilde{P}

$$\tilde{E} = E(\tilde{G}) \setminus \{(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)\} \quad (5)$$

Vertices that are candidates for an extension of P by one vertex so that the resulting path is again semielementary, form the subset

$$\tilde{\Gamma} = \{v; (v_k, v) \in \tilde{E}\} \quad (6)$$

composed of all vertices that are adjacent to the terminal vertex v_k of path P by edges outgoing from v_k and that are not used in path P . An extension of P is formally expressed by

$$P' = O_{ext}(P) \quad (7a)$$

where path P is extended to path P' by a vertex $v_{k+1} = v$ randomly selected from subset $\tilde{\Gamma}$,

$$P = (v_1, v_2, \dots, v_k) \rightarrow P' = (v_1, v_2, \dots, v_k, v_{k+1} = v) \quad (7b)$$

If subset $\tilde{\Gamma}$ is empty (i.e. $\tilde{\Gamma} = \emptyset$), then the process of extension is inapplicable and path \tilde{P} is called *nonextendible* and is closed.

Theorem 3. Let \tilde{P} be a semielementary path on graph \tilde{G} with set $\tilde{\Gamma}$ defined by (15). (1) If $\tilde{\Gamma} \neq \emptyset$, then P is *extendible* by $\tilde{P}' = O_{ext}(\tilde{P})$ or (2) if $\tilde{\Gamma} = \emptyset$, then \tilde{P} is *nonextendible* closed semielementary path (i.e. $v_1 = v_k$).

The proof may be simply done following the above discussion ensuring the existence of closed paths (spanning or not spanning) in graph \tilde{G} and by the fact that all vertices $v \in V(\tilde{G})$ have the same in- and out-valences, $val_{in}(v) = val_{out}(v)$. The only vertex, where one could end up without a possibility to go further through a new edge is then the starting vertex. The theorem says nothing about the fact whether the process of extension results in a spanning path or not. Fig. 3 (diagrams E and F) gives two illustrative examples of paths that are both nonextendible but only E is spanning while F is not spanning.

Theorem 4. Any path (closed or open) on a graph can be transformed into a semielementary path composed of the same vertices and of the same (or shorter) length as the original path.

This theorem can be deduced from model transformations displayed in Fig. 4. For instance, let us consider a path P that visits an edge e twice in the same direction (see the *left hand side* graph in diagram A , Fig. 4). Path P can be transformed into another path P' so that edge e is not visited (see the *right hand side* graph in diagram A , Fig. 4). This means that the length of P is shorter than the length of P' , $|P'| = |P| - 2d(e)$, where $d(e)$ is a "length" of edge e . Similar considerations are applicable also for more complicated cases when path P visits edge e more than twice in the same direction (see diagrams B and C in Fig. 4). These more complicated examples are transformed into paths that are semielementary and their lengths are shorter than the lengths of original paths.

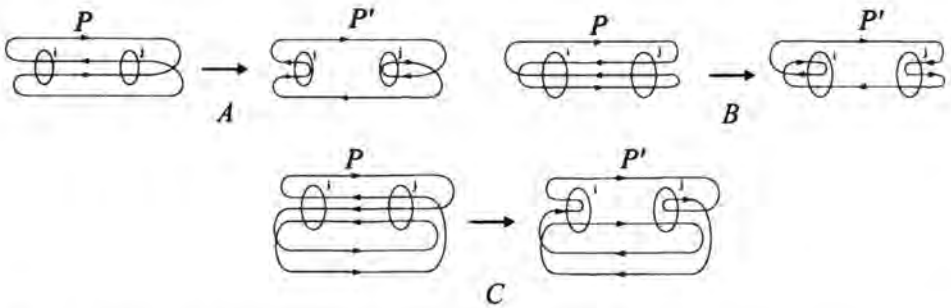


Fig. 4 Examples of transformations, which modify any nonsemelementary path going through a pair of adjacent vertices (represented by shaded ellipses) in the same direction to a shorter path containing all the vertices of the original path. Basic transformation is shown on diagram A, whereas diagrams B and C are only applications of the basic procedure to more complicated cases. The length of the new path is shorter than the original path by the length of two deleted edges.

Theorem 4 ensures that any shortest path (with some possible additional requirements like being Hamiltonian or going through specified vertices) must be semelementary. So when we are looking for a shortest path, we can restrict the search space to semelementary paths only. The use of this theorem considerably reduces the size of the search space and also the computational time.

According to the above discussion, a construction of the closed path on graph G may be carried out so that a closed path on graph \tilde{G} is constructed. Then, since the constructed path on \tilde{G} is semelementary (each directed edge on the path is visited at most once), the corresponding path on G is also semelementary, that is it may contain the same undirected edges at most twice (visited in opposite directions). The process of construction of a path is initiated by an arbitrarily selected vertex (formally considered a path). Let us assume that a subpath $P = (v_1, v_2, \dots, v_k)$ is already constructed, a new path is constructed from P by the above extension process. This successive extension of the constructed path is terminated either if the current path is nonextendible ($\tilde{\Gamma} = \emptyset$) or if the path is simultaneously closed and composed of all vertices of G . It may happen that the generated path is closed but not spanning, in this case the path should be penalized to an extent proportional to the number of vertices of G not participating in the path. The objective function assigned to a closed path $\tilde{P} = (v_1, v_2, \dots, v_k)$, which is used in our simulated annealing studies is determined as follows

$$f(P) = k + \omega (|V(G)| - |V(P)|) \quad (8)$$

where ω is a small positive constant used for the penalization of the difference between cardinalities of the vertex sets $V(G)$ and $V(P)$ (composed of vertices participating in the path P), in our illustrative calculation we used $\omega = 0.01$. In an ideal case, if graph G has a Hamiltonian closed path, then the minimal value of the objective function (17) is $f(P_{opt}) = |V|$.

A perturbation of a closed path $P = (v_1, v_2, \dots, v_k)$ into another closed path $P' = (v'_1, v'_2, \dots, v'_k)$ is formally expressed by

$$P' = O_{\text{pertur}}(P) \quad (9)$$

This transformation is carried out so that starting from a randomly selected index k path P is randomly extended until it gets closed. It is done by the extension process described in the text following Theorem 2, which is applied to a subpath $P = (v_1, v_2, \dots, v_k)$ in a way fully analogous to the method of construction of the closed path discussed above. In order to make the perturbation process more flexible we introduce the so-called inversion operator

$$\hat{P} = O_{\text{inv}}(P) \quad (10)$$

where the resulting path \hat{P} is determined by the inverse order of its entries-vertices with respect to the original path P

$$\hat{v}_i = v_{k-i+1} \quad (\text{for } i = 1, 2, \dots, k) \quad (11)$$

Applying the inverse operator O_{inv} we may say that two types of perturbations exist: The *forward perturbation*, $P' = O_{\text{pertur}}(P)$, and the *backward perturbation*, $P' = O_{\text{pertur}}(O_{\text{inv}}(P))$.

A few general remarks on incorporation of the above described approaches into the simulated annealing method. The expectation that the method of simulated annealing could provide a correct optimal solution (i.e. corresponding to the global minimum of objective function (8)), is based on an assumption that a closed spanning shortest path on a connected graph is semielementary, see Theorem 4. An important moment in our implementations of the simulated annealing construction of shortest paths (closed or open, and spanning or nonspanning) is the way to select a proper vertex from subset $\tilde{\Gamma}$, determined by (6), for the extension process of the current path P . In general this selection may be carried out randomly, but such strict randomization of the present method leads usually to implementations with a very slow convergence. Therefore, the pure random selection of the vertex from $\tilde{\Gamma}$ was changed to a quasirandom selection based on a priority scale of its vertices. The highest priority was assigned to those vertices that are not contained in the current path P , the lower priority to vertices that are already contained in the current path P but edges connecting these vertices with the terminal vertex have not been visited by the current path P , and finally, the lowest priority to all remaining vertices. The quasirandom selection of a vertex from set $\tilde{\Gamma}$ is realized so that vertices with higher priority are "randomly" preferred (an analogy of the so-called roulette wheel technique of selection of chromosomes for the reproduction process in genetic algorithms (Goldberg, 1989)).

The above theory was designed for the solution of task T_1 , where closed paths with randomly selected initial vertex are constructed. A similar approach can be used for the solution of tasks $T_2^{(1-3)}$ of construction of open paths where initial/terminal vertices are either specified or unspecified. Algorithms 1 and 2 should be then slightly modified.

4. Illustrative calculations

The construction of shortest paths of different types by the method of simulated annealing is illustrated by model calculations. Two kinds of calculations will be done. *First*, simple highly symmetric connected graph (see Fig. 5) is considered, for which it is possible to estimate the length of shortest spanning paths. *Second* type of model calculations will be carried out for solution of the well-known chess-game puzzle. The solution presents a sequence of knight moves on a chessboard so that all its fields are visited just once.

Highly symmetric graph

This graph (see Fig. 5) is composed of 76 vertices and 100 edges. The length of shortest spanning paths can be simply estimated as follows: First, let us consider the shortest spanning closed path, its form may be immediately deduced from diagrams *B* and *C* in Fig. 5. In particular, corner blocks of the path are represented by diagram *B*, while the central part of path is represented by diagram *C* (its oval edges correspond to the corner parts of the path). This means that the whole closed spanning paths are composed of five parts (four corner and one central), all of which have the length 20, therefore the length of a shortest spanning path is $100 = (4 + 1) \times 20$. Similarly, the shortest open spanning path is composed of four corner parts (diagram *B*) and one central part (diagram *D*), the latter also contains the initial and terminal vertices of the open path. Since the length of the central part is 17, the length of a shortest spanning path is $97 = 4 \times 20 + 17$. The length of shortest spanning paths with prescribed initial and/or terminal vertices can be deduced in an analogous way, it is necessary to consider all possible nonequivalent positions on the graph that are selected as initial and/or terminal vertices. Out of these we choose those of the shortest length. Since the number of all possible nonequivalent combinations is a few dozens, we do not present here a complete analysis of shortest spanning paths with prescribed initial and/or terminal vertices. We have observed that the lengths of all possible types of open spanning paths are always bounded by $97 \leq \text{length}(P) \leq 100$.

The simulated annealing calculations are performed for basic parameters specified as follows

$$r_{\max} = 10^4, T_{\max} = 50, T_{\min} = 0.5, \alpha = 0.95 \quad (12)$$

This means that the initial temperature T_{\max} is $T = 50$, its multiplicative decrease is specified by $\alpha = 0.95$ (i.e. the temperature is decreased by 5%), the lowest temperature T_{\min} being $T = 0.5$, and finally, for each fixed temperature T in the Metropolis algorithm 10^4 trials are done (r_{\max} is the number of trials). The Boltzmann constant is incorporated in the above value of the parameter "temperature".

Tab. 1 shows the best results from 10 independent calculations obtained for tasks T_1 and T_2 . The most complicated and time consuming is task $T_2^{(3)}$, where shortest open spanning paths with specified both initial and terminal vertices are constructed. Since in this task both vertices are specified in advance, the method of path perturbation described in the previous section is probably not very useful. The main difficulty here is caused by the fact that the perturbation process gen-

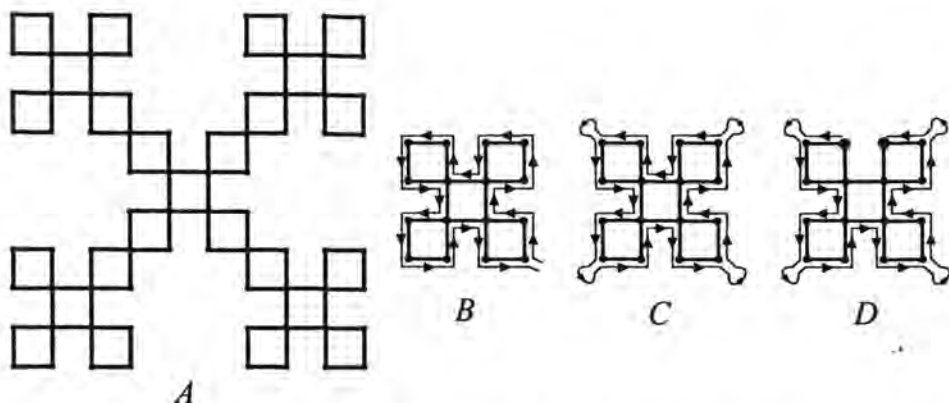


Fig. 5 Highly symmetric graph composed of 76 vertices and 100 edges (diagram A). Diagrams B to D are used for construction of shortest spanning paths that are either closed (diagrams B and C) or open (diagrams B and D).

No.	Task	Path length
1	$T_1^{(1)}$	100
2	$T_2^{(1)}$	97
3	$T_2^{(2)}$	97-100 ^a
4	$T_2^{(3)}$	97-100 ^a

^a The length of resulting spanning shortest paths depends on the selection of initial/terminal path vertices.

Tab. I Best results obtained for different tasks for graph Fig. 5.

erates most frequently paths that are not properly ended by the required terminal vertex, and therefore should be penalized. Only small ratio of r_{max} events in the Metropolis algorithm produces paths that are properly ended. This implies that this vexing feature of the suggested version of simulated annealing for the $T_2^{(3)}$ can be most likely removed, in general, by a substantial increase of parameter r_{max} , albeit at a price of slow a convergence of simulated annealing.

Knight piece moves on the chessboard

The second illustrative application of simulated annealing approach for construction of spanning elementary paths concerns the solution of the well-known chess game puzzle [3], [7], [11] - the task being to find elementary spanning paths (closed or open) of knight piece moves so that all chessboard fields are visited just once. This puzzle can be transformed to our problem of construction of spanning paths on connected graphs straightforwardly. Each chessboard field is represented by a graph vertex and two vertices are connected by an edge if the corresponding fields are achievable by single knight piece move.

The basic simulated annealing parameters in our calculations were selected as follows

$$r_{max} = 10^4, T_{max} = 30, T_{min} = 0.1, \alpha = 0.95 \quad (13)$$

The proposed simulated annealing approach has been successfully used for the construction of spanning (closed or open) elementary paths on chessboards of the 6×6 , 7×7 , and 8×8 type, respectively. We remember the well-known fact [3] that closed elementary paths exist only on chessboards of even dimension, that is for 6×6 and 8×8 , while on chessboard of odd dimension 7×7 there exist only open elementary spanning paths. Some representative results of our calculations are displayed in Fig. 6. We see that in all cases the present version of simulated annealing provides either correct results (i.e. elementary spanning paths are produced) or ones that are likely closely related to optimal solutions (in particular for paths constructed in the framework of tasks $T_1^{(1)}$ and $t_1^{(2)}$).

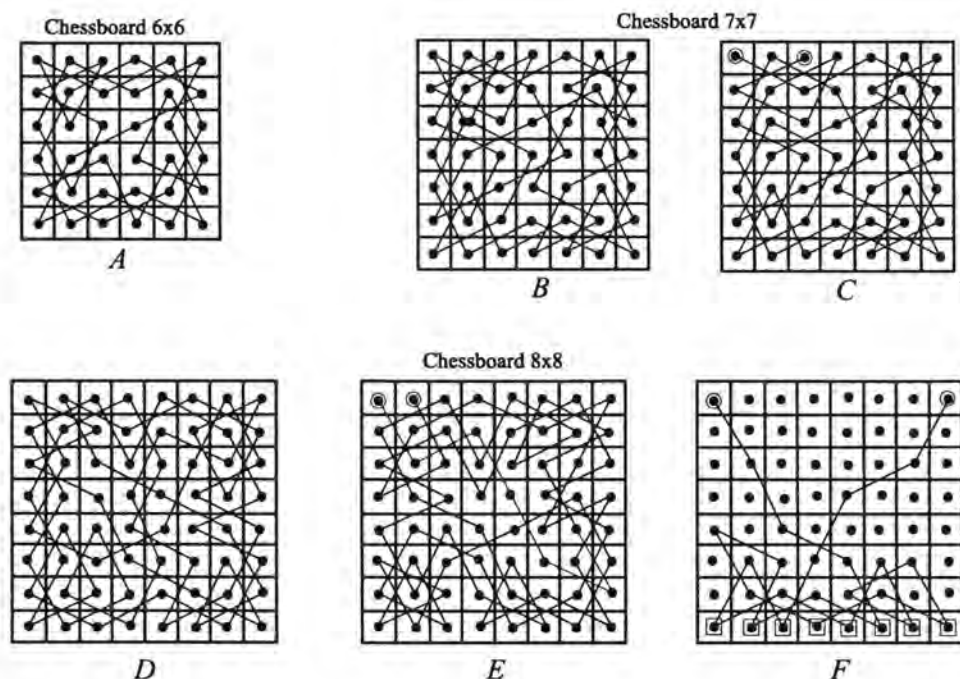


Fig. 6 Different paths constructed on chessboard of dimensions 6×6 (diagram A), 7×7 (diagrams B and C), and 8×8 (diagrams D to F). Closed elementary spanning paths exist on chessboard of even dimensions (see diagrams A and D), while for chessboard of odd dimension this closed spanning path exists only in nonelementary form (diagram B). Diagram F represents an open path with prescribed initial and terminal fields (encircled) that must visit eight prescribed fields (squared).

5. Discussion

The presented version of simulated annealing offers a simple and effective method for construction of spanning paths (closed or open) on connected graphs. The graph-theoretical analysis of the problem allows us to focus our attention on paths that are semielementary, that is edges of a path can be visited at most twice in the opposite directions. The formal construction of semielementary paths is considerably simplified by the extension process. The process automatically ensures that the produced paths are semielementary. A fulfillment of additional conditions that are required from the resulting paths is simply ensured by penalization included in the form of objective functions minimized by the simulated annealing approach.

We have to emphasize that in our two model calculations most difficulties have been encountered in paths constructed in the framework of task $T_2^{(3)}$, where open spanning paths with specified initial and terminal vertices were to be constructed. Likely, the used extension process is not very well equipped for the construction of open paths of this type, namely a path that should be simultaneously spanning and terminating at the prescribed vertices. In many cases these two requirements are "contradictory", the resulting path is either closed and spanning or open but not terminated at the prescribed vertex. The states formed in the process of simulated annealing most frequently do not satisfy simultaneously both conditions and are therefore penalized. This fact considerably decreases the effectiveness of the simulated annealing method; in most part of its performance it works, loosely speaking, as a "blind" search. The question is how to increase substantially the appearance of current states if they have to be simultaneously spanning and terminated at a required vertex? We are not able to give now a definite answer to this question. Most probably, its efficient solution requires another approach of construction of semielementary paths than the one based on the extension process.

Acknowledgement.

The work was supported by the grants 1/7336/20 and 1/8107/01 of the Scientific Grant Agency of the Slovak Republic.

References

- [1] Černý V.: Thermodynamic Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *J. Opt. Theory Appl.*, **45**, 1985, pp. 41-51.
- [2] Christofides N.: *Graph Theory. An Algorithmic Approach*. Academic Press, New York, 1975.
- [3] Gik E. Ya.: *Chess and Mathematics*. Nauka, Moscow, 1983 (in Russian).
- [4] Goldberg D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- [5] Harary F.: *Graph Theory*. Addison-Wesley, Reading, MA, 1969.
- [6] Kirkpatrick S., Gelat C., Vecchi M.: Optimization by Simulated Annealing. *Science*, **220**, 1983, pp. 671-680.
- [7] Kraitchik M.: *Le Probleme du Cavalier*. Gautiers-Villars et Cie, Paris, 1927.
- [8] Kvasnička V., Pospíchal J.: Simulated annealing. *Communications in Mathematical Chemistry (MATCH)*, **34**, 1996, pp. 7-49.

- [9] Kvasnička V., Pospíchal J., Hesek D.: Augmented Simulated Annealing Algorithm for the TSP. Central European Journal for Operations Research and Economics, **2**, 1993, pp. 307-317.
- [10] Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., Teller E.: Equation of State Calculations for Fast Computing Machines. J. Chem. Phys., **21**, 1953, pp. 1087-1092.
- [11] Parberry I.: An Efficient Algorithm for the Knight's Tour Problem. Discrete Applied Mathematics, **73**, 1997, pp. 251-260.