Discussiones Mathematicae
Graph Theory 23 (2003) 5–21

BALANCED PROBLEMS ON GRAPHS WITH CATEGORIZATION OF EDGES

Štefan Berežný

Air Force Academy of M.R. Štefánik, Košice
Rampová 7, 041 21 Košice, Slovakia

and

Vladimír Lacko

Institute of Mathematics
University of P.J. Šafárik, Košice
Jesenná 5, 041 54 Košice, Slovakia

e-mail: lacko@science.upjs.sk

Abstract

Suppose a graph $G = (V, E)$ with edge weights $w(e)$ and edges partitioned into disjoint categories $S_1, \ldots, S_p$ is given. We consider optimization problems $\mathcal{P}$ on G defined by a family of feasible sets $\mathcal{D}(G)$ and the following objective function:

$$L_5(D) \quad = \quad \max_{1 \le i \le p} \left(\max_{e \in S_i \cap D} w(e) - \min_{e \in S_i \cap D} w(e) \right)$$

For an arbitrary number of categories we show that the L_5-perfect matching, L_5-a-b path, L_5-spanning tree problems and L_5-Hamilton cycle (on a Halin graph) problem are NP-complete.

We also summarize polynomiality results concerning above objective functions for arbitrary and for fixed number of categories.

Keywords: algorithms on graphs, categorization of edges, NP-completeness.

2000 Mathematics Subject Classification: 05C85, 90C27, 68Q17.

1. Introduction

An optimization problem is given by a family of feasible sets $\mathcal{D}$ and an objective function which is to be minimized (or maximized) over $\mathcal{D}$. The most common objective functions are the sum of weights of the elements and their maximum weight. The former type leads to the *sum problems*, the latter to the *bottleneck optimization problems.* Also, if some kind of equity of used elements is sought, the range of used weights is optimized, which leads to *balanced problems.* Balanced optimization problems are a relatively new topic, see Duin and Volgenant [7] and Martello et al [11], where a polynomial algorithm has been derived for such cases of balanced problems, where the feasibility check can be done polynomially, including e.g. the spanning tree, matching and path problems. Another area of balancing objective function application is the location theory, where e.g. Gavalec and Hudec [9] consider the problem of balanced location of service facilities with respect to a given set of demands on a graph.

Recently, new types of objective functions were considered, based on dividing the set of edges into disjoint categories, taking a sum, minimum or maximum within each category and then combining the obtained values by means of the second function. More formally, suppose a graph $G = (V, E)$ with nonnegative edge weights $w(e)$ and edges partitioned into disjoint categories $S_1, \ldots, S_p$ is given. An optimization problem $\mathcal{P}$ on a graph is given by a family of feasible sets $\mathcal{D} = \mathcal{D}(G)$ and an objective function $L : 2^E \longrightarrow \mathbb{R}_0^+$. In this paper, we consider as feasible sets $\mathcal{D}(G)$ the set of all spanning trees, perfect matchings, a-b paths in a graph or Hamiltonian circuits in a Halin graph G. As an objective function we consider the function L_5 defined as follows:

$$L_5(D) \quad = \quad \max_{1 \le i \le p} \left(\max_{e \in S_i \cap D} w(e) - \min_{e \in S_i \cap D} w(e) \right).$$

A pair $(\mathcal{D}(G), L_5)$ determines a particular optimization problem

$$\begin{gathered} L_5(D) \longrightarrow \min \\ D \in \mathcal{D}(G) \end{gathered} \tag{1}$$

The objective function defined above is called L_5 since our work extends papers by Averbakh and Berman [1], Berežný, Cechlárová, Lacko [2, 3] and Punnen and Richie [12, 13], where other problems with objective functions

$$
\begin{aligned}
L_1(D) &= \sum_{i=1}^{p}\left(\max_{e\in S_i\cap D} w(e)\right),\\
L_2(D) &= \max_{1\le i\le p}\sum_{e\in S_i\cap D} w(e),\\
L_3(D) &= \max_{1\le i\le p}\sum_{e\in S_i\cap D} w(e)-\min_{1\le i\le p}\sum_{e\in S_i\cap D} w(e),\\
L_4(D) &= \sum_{i=1}^{p}\left(\max_{e\in S_i\cap D} w(e)-\min_{e\in S_i\cap D} w(e)\right)
\end{aligned}
$$

and others using the operators $\min, \max$ and $\sum$ together with edge categorization were studied. The problem (1) belongs to the class of balanced problems studied in [7, 9, 11].

The objective functions L_1 and L_2 were considered by Richey and Punnen [13] for perfect matchings and spanning trees, by Punnen [12] for the travelling salesman problem on Halin graphs and by Averbakh and Berman [1] for the path problem. It was shown that the above L_1-problems are strongly NP-complete in a general case and polynomial for a fixed number of categories and L_2-problems are NP-complete even for two categories. In [2] Berežný, Cechlárová and Lacko showed that the spanning tree, matching and path problems considered with the L_3 objective function are NP complete already on bipartite outerplanar graphs even for two categories, similarly the L_3-travelling salesman problem is NP complete on Halin graphs even for two categories. The above problems considered with the objective function L_4 were also shown to be NP complete. Here, however, the situation for a fixed number of categories is different: there is a polynomial algorithm for any L_4 problem for which a polynomial feasibility checking algorithm exists, see [2]. The following objective functions were considered in [5] (in addition to L_5):

$$
\begin{aligned}
L_6(D) &= \max_{1\le i\le p}\left(\max_{e\in S_i\cap D} w(e)\right)-\min_{1\le i\le p}\left(\max_{e\in S_i\cap D} w(e)\right),\\
L_7(D) &= \max_{1\le i\le p}\left(\max_{e\in S_i\cap D} w(e)\right)-\max_{1\le i\le p}\left(\min_{e\in S_i\cap D} w(e)\right),\\
L_8(D) &= \min_{1\le i\le p}\left(\max_{e\in S_i\cap D} w(e)-\min_{e\in S_i\cap D} w(e)\right).
\end{aligned}
$$

It was shown there that if the number of categories p is fixed, then problems with the L_5 objective function are polynomially solvable provided that for the given system $\mathcal{D}(G)$ there exists a polynomial feasibility test (decision about the existence of a feasible solution on the given set) and problems with the L_6 objective function are polynomial provided that for the system $\mathcal{D}(G)$ we are able to polynomially enlarge the set of prescribed edges using only allowed edges to a feasible solution (extension test). In particular, if $p = 1$ then problems with L_5 and L_6 objective functions correspond to classical balanced problems.

On the other hand, if the number of categories p is not fixed, then problems with the L_7 objective function are polynomial if there exists a polynomial extension test and problems with the L_8 objective function are polynomial if there exists a polynomial feasibility test (see [4]).

A polynomial solution to the L_6 spanning tree problem for the general number of categories was also derived in [5].

In this paper, we show that the L_5-perfect matching, L_5-spanning tree, L_5-a-b path problems and L_5-Hamilton cycle (in a Halin graph) problem are NP-complete in a general case (i.e., when the number of categories p is not fixed).

2. Application in Management

A manufacturing plant produces various commodities. There are several production lines in the plant, each of them is capable of producing some subset of commodities. While producing a given commodity, the production line must maintain a specific optimal temperature which may differ from commodity to commodity. The heating and perhaps the cooling process of the production lines is extremely slow. Thus, to minimize the time delays while changing the temperature when switching production from one commodity to another, the production process for each production line is organized as follows: in the idle time (during nights) the line is heated to the lowest optimal temperature of the commodity it produces. The production then proceeds in increasing order of optimal temperatures — starting with the commodity of the lowest and finishing with the commodity of the highest optimal temperature.

The total time the production line is delayed due to temperature changes depends entirely on the difference between the highest and the lowest optimal temperature of the commodity it produces. The total delay of the manufacturing plant is now the maximum of delays of production lines. Here

we may assume that all those dependences are linear. A goal of production manager is to minimize the total delay of the manufacturing plant.
The model: Let a graph G be a union $G_1 \cup G_2 \cup \cdots \cup G_n$, where n is the total number of commodities and each G_i is a graph depicted in Figure 1. Let p be the number of production lines. Each graph G_i contains two vertices v_i and v_i' corresponding to the i-th article and $2 * i_k$ vertices $\{m_{i,j}, m_{i,j}';\ j \in PL_i\}$ where PL_i is the set of i_k production lines which can produce the i-th commodity. Let us assign weight equal to the optimum temperature of the i-th commodity to edges $v_i m_{i,j},\ j \in PL_i$ and put each edge $v_i m_{i,j}$ into the category S_j. All remaining edges are of weight 0 and belong to one extra category S_o.

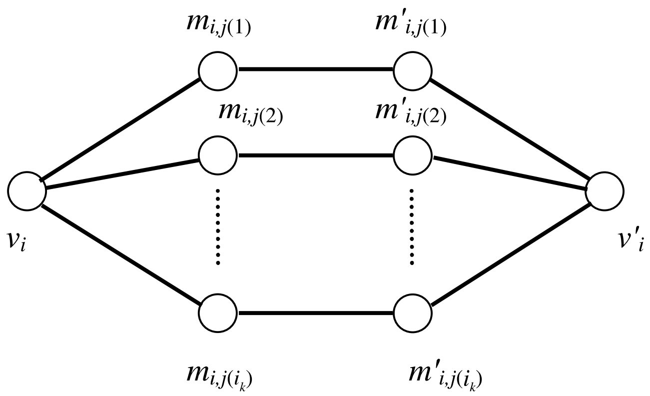

Figure 1

M is a perfect matching in G if and only if in each graph G_i exactly one of the edges $v_i m_{i,j}$ is matched. (At the same time edges $v_i' m_{i,j}'$ and $m_{i,k} m_{i,k}',\ k \in PL_i - \{j\}$ are matched, too). Think of the fact that edge $v_i m_{i,j}$ is matched to mean that commodity i is produced on production line j. Then it is easy to see one-to-one correspondence between feasible commodity-line assignments and perfect matchings in G. Moreover, the objective function of the production manager is now exactly the L_5-objective function.

3. L_5 Problems with Arbitrary Number of Categories

In this section, we show NP-completeness of recognition versions of the L_5-perfect matching, L_5-a-b path, L_5-spanning tree problems on graphs and the L_5-Hamiltionian cycle problem on Halin graphs.

It is easy to see that the above problems are in class NP. We prove NP-completeness by showing that 3-SATISFIABILITY problem [8] polynomially transforms to recognition versions of the L_5 problems in question. In our proofs of NP-completeness below we assume that a Boolean formula F consisting of m clauses $C_1, \ldots, C_m$ and involving n variables $x_1, \ldots, x_n$ is given. The key question of each proof is to construct a graph $G = (V, E)$, assign weights $w(e)$ to its edges and partition them into categories $S_1, \ldots, S_p$ in such a way that

$$\begin{array}{c} G \text{ has a feasible solution } D \in \mathcal{D}(G) \\ \text{(the perfect matching, a-b path, spanning tree, Hamilton cycle)} \\ \text{with } L_5(D) \leq K \text{ if and only if } F \text{ is satisfiable.} \end{array} \tag{2}$$

Thus, in all proofs we start with the construction of a graph G and then show *only if* and *if* parts of (2). It suffices to show polynomial transformations to recognition versions with $K = 0$.

We begin with the L_5-perfect matching problem:

Theorem 1. *Given a graph $G = (V, E)$ and a constant K, it is NP-complete to decide whether there exists a perfect matching $\tilde{M}$ on a graph G with $L_5(\tilde{M}) \leq K$.*

Proof. In our construction we use the folowing special-purpose graphs: *Graph A* (shown in Figure 2a) and *Graph B* (shown in Figure 2b).

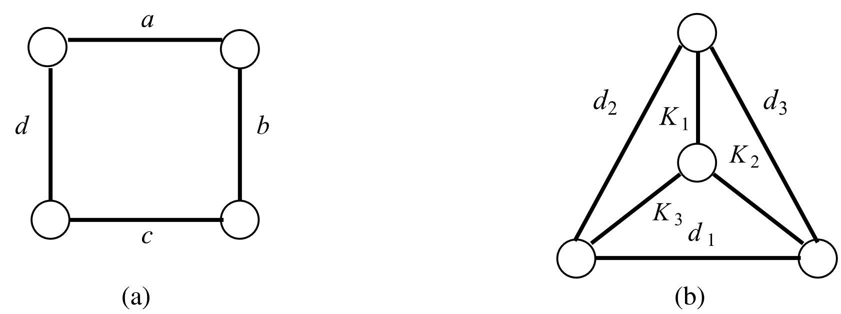

Figure 2

Supose that the graph A is an isolated component of another graph G and let $\tilde{M}$ be a perfect matching of G. Then either $\{b, d\} \subseteq \tilde{M}$ or $\{a, c\} \subseteq \tilde{M}$.

The graph B has a similar property. If B is a subgraph of G such that no other edges of G are incident upon any node of B, then each perfect matching $\tilde{M}$ of G contains exactly one of the edges K_1, K_2, K_3 (the only possible matchings of graph B are $\{K_1, d_1\}, \{K_2, d_2\}$ and $\{K_3, d_3\}$).

A graph $G = (V, E)$ will consist of n copies $A_1, \ldots, A_n$ of the graph A corresponding to variables $x_1, \ldots, x_n$, of further $3m$ copies A_{jr}, $j = 1, \ldots, m, r = 1, 2, 3$ of the graph A and m copies $B_1, \ldots, B_m$ of the graph B corresponding to clauses $C_1, \ldots, C_m$. We create n categories $S_1, \ldots, S_n$ corresponding to variables, $3m$ categories $S_{jr}, j = 1, \ldots, m, r = 1, 2, 3$ corresponding to clauses and one extra category S_o.

In each copy A_i of the graph A two opposite edges have weight 1 (corresponding to the literal x_i) and the remaining two edges have weight 0 (corresponding to the literal $\bar{x}_i$), all edges belong to the category S_i (see Figure 3 for an illustration of this construction).

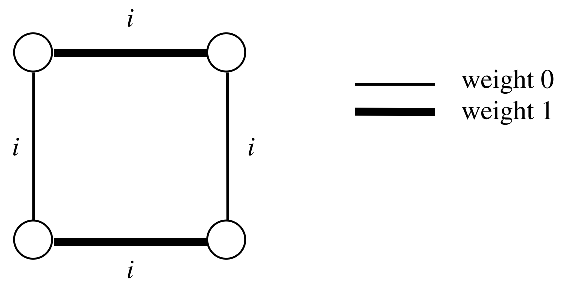

Figure 3

The situation for m clauses $C_1, \ldots, C_m$ is slightly more complicated. For each clause $C_j = (l_1 + l_2 + l_3)$, where each l_i is a literal — either x_k or $\bar{x}_k$ for some $k \in \{1, \ldots, n\}$, the r-th copy, $r = 1, 2, 3$, A_{jr} of the graph A consists of one edge of weight either 1 if the corresponding literal l_r is x_k or 0 if the literal l_r is $\bar{x}_k$. This edge then belongs to the category S_k. The other three edges are of weights 0, 1 and 0 and all belong to the category S_{jr}. We finally assign weights and categories to edges of the subgraph B_j: the three central edges K_i have weights 1 and belong to categories S_{j1}, S_{j2}, S_{j3}, respectively. The remaining three edges are of weight 0 and belong to the category S_o. (See Figure 4 and Figure 5 for an illustration of subgraphs constructed for clauses).

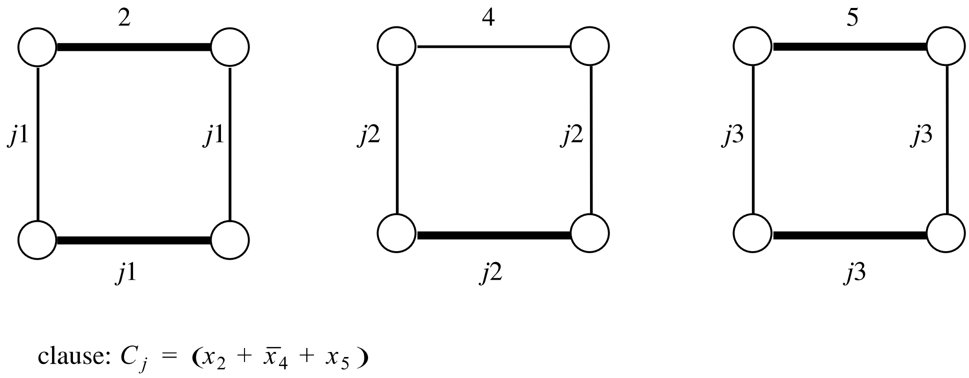

Figure 4

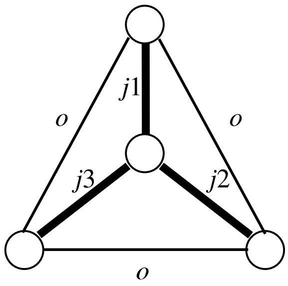

Figure 5

We shall now argue that our construction of the graph G from a Boolean formula F is correct, i.e., G has a perfect matching $\tilde{M}$ with $L_5(\tilde{M}) \leq 0$ if and only if F is satisfiable.

For the *only if* direction, suppose that G has a perfect matching $\tilde{M}$ with $L_5(\tilde{M}) \leq 0$. In each of the subgraphs B_j exactly one central edge K_r of weight 1 is matched. $K_r \in S_{jr}$, and since $L_5(\tilde{M}) = 0$ only edges of weight 1 in the category S_{jr} can be matched. It follows that an edge of the subgraph A_{jr} corresponding to the r-th literal of clause j of weight 1, which belongs to the category S_{jr} is matched, too. Denote this edge by f. Then from the above mentioned property of the graph A the other matched edge in this subgraph A_{jr} must be an edge $g \in S_k$ for some $k \in \{1, \dots, n\}$. But a similar argument shows that in the subgraph A_k corresponding to the variable x_k both edges of weight equal to $w(g)$ must be matched (the other two of weight $1 - w(g)$ are of course unmatched).

Think of the fact that edges with weight 1 in the category S_k are matched to mean that x_k takes the value *true*, otherwise, if edges with weight 0 in the category S_k are matched, we say that x_i is *false*. The reader should be able to convince himself that the resulting function is a valid truth assignment: each x_i is assigned exactly one of the two truth assignments (due to the subgraph A_i corresponding to x_i) and due to the above mentioned property of the graph B, each clause is satisfied by one of its three literals.

For the *if* part, suppose that the Boolean formula F is satisfiable by some truth assignment τ. The perfect matching $\tilde{M}$ of G can be constructed by the following rules stated in the *only if* part: If x_i is *true*, both edges of the subgraph A_i having weight 1 are included in $\tilde{M}$, otherwise both edges of weight 0 are included. Let l_r be the first literal which satisfies clause $C_j = (l_1 + l_2 + l_3)$. Then edges K_r and d_r of the subgraph B_j are included in $\tilde{M}$. From the subgraph A_{jr} we include in $\tilde{M}$ an edge $e \in S_{jr}$ having weight 1 and its opposite edge and from subgraphs $A_{jt}, t \in \{1, 2, 3\} - \{r\}$ we include two edges from the category S_{jt} having weight 0. It is then clear that $L_5(\tilde{M}) = 0$. ■

Next we prove that the L_5-a-b path problem is NP-complete, too.

Theorem 2. *Given a graph $G = (V, E)$, vertices $a, b \in V$ and a constant K, it is NP-complete to decide whether there exists an a-b path P in G with $L_5(P) \leq K$.*

Proof. In our construction we use the graph A shown in Figure 6. Suppose that A is a subgraph of some other graph G such that

1. no other edges of G (edges not shown in Figure 6) are incident upon any node of A except for C^j and D^j
2. C^j and D^j are articulations of G.

Then each $C^j - D^j$ path in G traverses through edges $c_i a_i c_i'$ for some $i = 1, 2, 3$.

A graph $G = (V, E)$ will consist of a chain $E^0 - F^0$ of parallel edges of length n corresponding to variables $x_1, \ldots, x_n$ and of m copies $A_1, \ldots, A_m$ of the graph A corresponding to clauses $C_1, \ldots, C_m$. We create n categories $S_1, \ldots, S_n$ corresponding to variables and one extra category S_o.

For each variable x_i we have two parallel edges: one of weight 1 (corresponding to the literal x_i) and the other of weight 0 (corresponding to the literal $\bar{x}_i$), and both belonging to the category S_i. We arrange these edges to form a chain (see Figure 7 for an illustration of this construction).

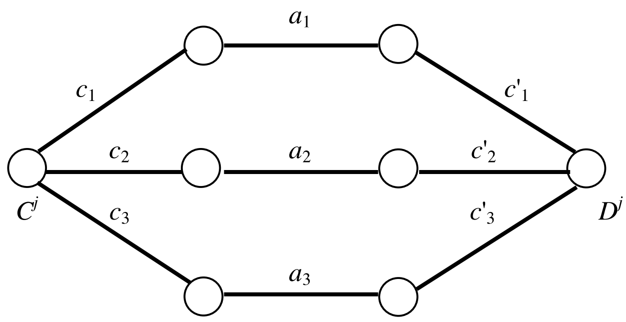

Figure 6

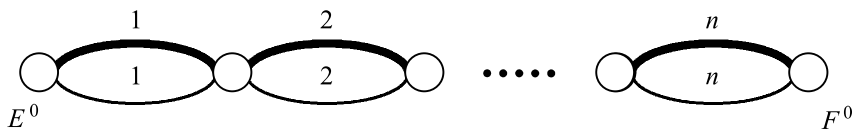

Figure 7

For each clause $C_j = (l_1 + l_2 + l_3)$, where each l_i is a literal — either x_{k_i} or $\bar{x}_{k_i}$ for some $k_i \in \{1, \ldots, n\}$, we do the following construction: In the j-th copy A_j of the graph A we assign weight 1 to edges a_i for $i = 1, 2, 3$ if the literal $l_i = x_{k_i}$ and 0 otherwise and put them in the category S_{k_i}. All other edges of the subgraph A_j are of 0 weight and are included in the category S_o. (See Figure 8 for an illustration of constructing the subgraph A_j).

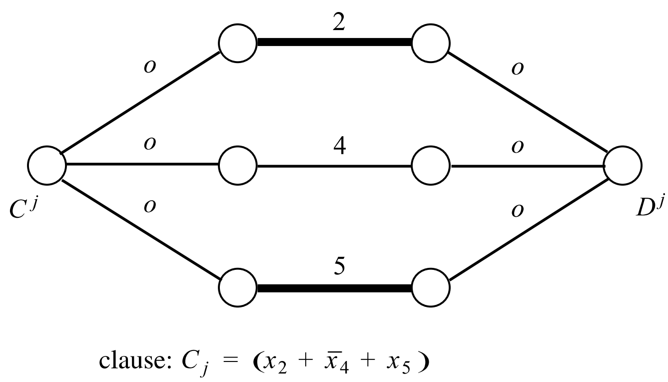

Figure 8

All subgraphs A_j are joined in vertices C^j and D^j to form a chain-like graph to which we adjoin chain $E^0 - F^0$ and let $a = C^1$ and $b = F^0$. A scheme of the resulting graph $G = (V, E)$ is depicted in Figure 9.

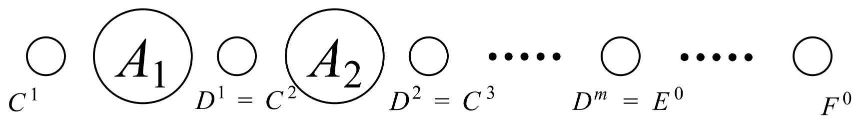

Figure 9

We shall now argue that our construction of the graph G from a Boolean formula F is correct, i.e., G has an a-b path P with $L_5(P) \leq 0$ if and only if F is satisfiable.

For the *only if* direction, suppose that G has an a-b path P with $L_5(P) \leq 0$. If we take into account the already mentioned property of the subgraph A, a path P must traverse through the edge a_r for some $r \in \{1, 2, 3\}$. Let x_{k_r} be the variable corresponding to the r-th literal l_r of clause C_j. Then P cannot traverse edges of the category S_{k_r} with weight not equal to $w(a_r)$ and therefore, from the pair of edges corresponding to the variable x_{k_r} in chain $E^0 - F^0$, path P must traverse the edge of weight $w(a_r)$.

Think of the fact that edges with weight 1 in the category S_k are traversed by P to mean that x_k takes the value *true*, otherwise, if edges with weight 0 in the category S_k are traversed by P, we say that x_i is *false*. It is not difficult now to verify that we have just defined a valid truth assignment: each x_i is uniquely assigned either *true* or *false* and each clause is satisfied by one of its three literals.

For the *if* part, suppose that a Boolean formula F is satisfiable by some truth assignment τ. The a-b path P with $L_5(P) = 0$ can be constructed as follows: If x_i is *true*, P will traverse from the pair of parallel edges corresponding to the i-th variable the edge having weight 1, otherwise the one with weight 0. Let l_r be the first literal which satisfies the clause $C_j = (l_1 + l_2 + l_3)$. Then P will traverse edges c_r, a_r and c'_r of the subgraph B_j. It is then clear that $L_5(P) = 0$. ■

Almost the same arguments can be used to prove the next theorem.

Theorem 3. *Given a graph $G = (V, E)$ and a constant K, it is NP-complete to decide whether there exists a spanning tree T in a graph G with $L_5(T) \leq K$.*

Proof. We use the same construction of a graph $G = (V, E)$ as in the proof of Theorem 2. We shall now argue that G has a spanning tree T with $L_5(T) \leq 0$ if and only if F is satisfiable.

For the *only if* direction, suppose that G has a spanning tree T with $L_5(T) \leq 0$. Then T must contain an a-b path as a subgraph and therefore exactly the same arguments as in the proof of Theorem 2 apply here.

For the *if* part, suppose that a Boolean formula F is satisfiable by some truth assignment τ. We know that there exists an a-b path P with $L_5(P) \leq 0$. If we extend it by remaining edges of the category S_o, the resulting subgraph T is clearly a spanning tree of G with $L_5(T) = 0$. ∎

In the following theorem we state a result about the L_5-Hamilton cycle problem, which is closely related to the Traveling Salesman Problem (TSP). TSP is NP-complete in a general case. Thus, we direct our attention to a special type of graphs — so called Halin graphs, where TSP and Hamilton cycle problems can be solved in polynomial time [6].

A *Halin graph* $G = T \cup C$ is obtained by embedding a tree T without nodes of degree 2 in the plane and adding a cycle C (outer cycle) joining the leaves of T in such a way that the resulting graph is planar. Given a Halin graph $G = T \cup C$, the question to decide is whether there exists a cycle O in a graph H passing through each vertex exactly once.

Theorem 4. *Given a Halin graph $G = (V, E)$ and a constant K, it is NP-complete to decide whether there exists a Hamilton cycle O in a Halin graph G with $L_5(O) \leq K$.*

Proof. In our construction we use the folowing special-purpose graphs: *Graph A* (shown in Figure 10a) and *Graph B* (shown in Figure 10b).

Suppose that A is a subgraph of some other graph G such that no other edges of G (edges not shown in Figure 10) are incident upon any vertex of A except for C^i and D^i. Then there are only two possible ways in which a Hamilton path can cross the graph A traversing vertices C^i and D^i: either using edges $b_1b_2a_2$ or edges $a_1b_2b_3$.

Similarly, if there are no edges incident upon any vertex of the graph B except for E^j and F^j, there are 5 possible ways in which a Hamilton path can cross the graph B traversing vertices E^j and F^j. Such a Hamilton path must then contain exactly two edges $b_{pq},\ p, q = 1, \ldots, 3$. Moreover, for every $p = 1, \ldots, 3$ there exists an $E^j - F^j$ path traversing through b_{p1} and b_{p2} but not through $b_{p'1}, b_{p'2},\ p' \in \{1, 2, 3\} - \{p\}$.

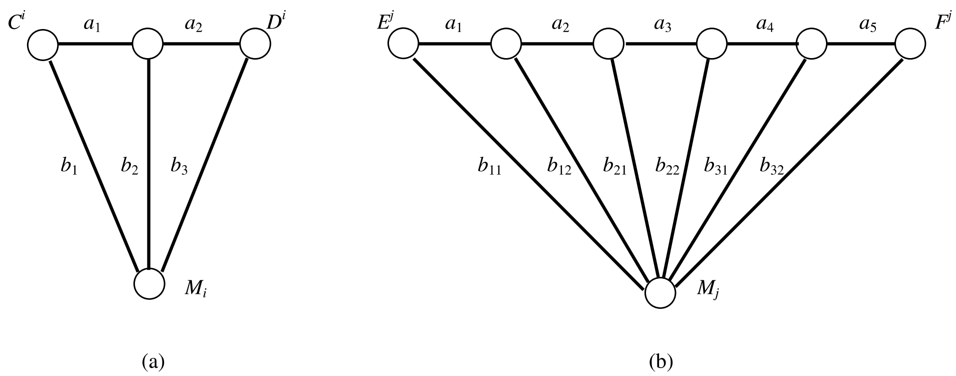

Figure 10

A Halin graph $G = (V, E)$ will consist of n copies $A_1, \dots, A_n$ of the graph A corresponding to variables $x_1, \dots, x_n$ and of m copies $B_1, \dots, B_m$ of the graph B corresponding to clauses $C_1, \dots, C_m$. We also create n categories $S_1, \dots, S_n$ corresponding to variables and one extra category S_o.

In each copy A_i of the graph A an edge b_1 (corresponding to the literal x_i) has weight 1 and belongs together with edge b_3 of weight 0 (corresponding to the literal $\bar{x}_i$) to the category S_i. All remaining edges have weight 0 and are included in the category S_o. (See Figure 11 for an illustration of this construction).

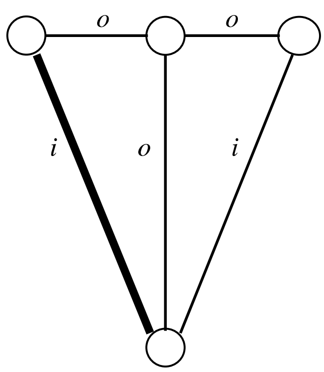

Figure 11

For each clause $C_j = (l_1 + l_2 + l_3)$ we have one copy B_j of the graph B. We assign weight 1 to edges $b_{i1}, b_{i2},\ i = 1, 2, 3$ if the literal $l_i = x_{k_i}$ and 0 otherwise and put them in the category S_{k_i}. All other edges of the subgraph B_j are of 0 weight and are included in the category S_o. (See Figure 12 for an illustration of constructing a subgraph B_j).

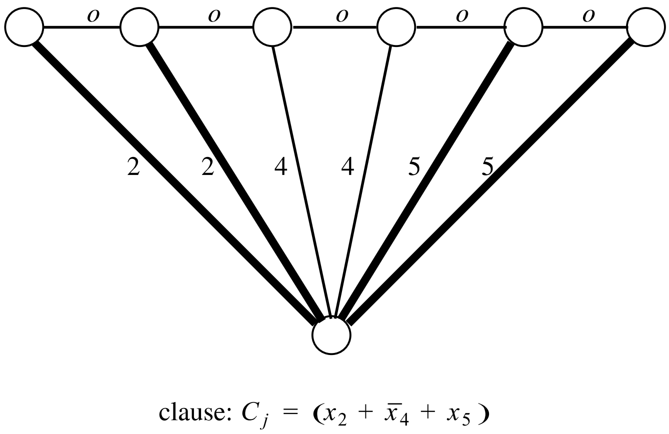

Figure 12

We join vertices D^i and C^{i+1} and vertices F^j and E^{j+1} with edges of weight 0. We also join all vertices M_i and M_j with one extra vertex M with edges of weight 1. We finally add two extra vertices K and L and edges C^1K, KM, ML, LF^m and D^nE^1 of weight 0 and an edge KL of weight 1. All these extra edges belong to the category S_o. See Figure 13 for an illustration of the final construction of a Halin graph G.

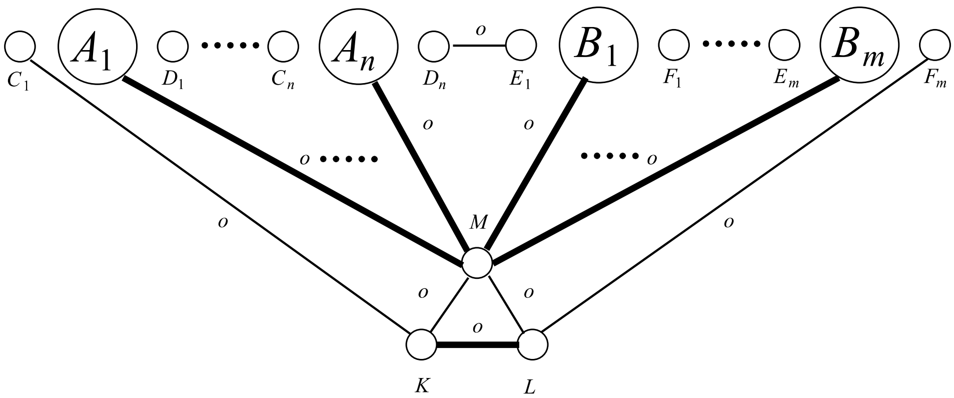

Figure 13

We shall now argue that our construction of a Halin graph G from a Boolean formula F is correct, i.e., G has a Hamilton cycle O with $L_5(O) \leq 0$ if and only if F is satisfiable.

For the *only if* direction, suppose that G has a Hamilton cycle O with $L_5(O) \leq 0$. A Hamilton cycle O clearly traverses an edge $e \in S_o$ of weight 0 and since $L_5(O) = 0$, O cannot traverse edges of S_o of weight 1. If we take into account the already mentioned property of the graph B, we see that Hamilton cycle O must traverse in each subgraph B_j at least one edge b_{ip}, $i = 1, 2, 3$, $p = 1, 2$ corresponding to the i-th literal of clause j — either x_{k_i} or $\bar{x}_{k_i}$. It follows that an edge f of the subgraph A_{k_i} of weight $w(f) = w(b_{ip})$, which belongs to the same category as an edge b_{ip} is traversed by Hamilton cycle, too.

Think of the fact that edges with weight 1 in the category S_k, $k = 1, \ldots, n$ are traversed by O to mean that x_k takes the value *true*, otherwise, if edges with weight 0 in the category S_k are traversed by O, we say that x_i is *false*. It is not difficult now to verify that we have just defined a valid truth assignment: each x_i is uniquely assigned either *true* or *false* and each clause is satisfied by at least one of its three literals.

For the *if* part, suppose that a Boolean formula F is satisfiable by some truth assignment τ. The construction of a Hamilton cycle O in a Halin graph G is as follows: If x_i is *true*, O traverses edges b_1, b_2, a_2 (in that order) of the subgraph A_i, otherwise it traverses edges a_1, b_2, b_3. Let l_r be the first literal which satisfies clause $C_j = (l_1 + l_2 + l_3)$. Then O traverses edges $b_{r1}, b_{r2}, a_s, s \in \{1, \ldots, 5\} - \{2r - 1\}$ (in proper order) of the subgraph B_j. Clearly $L_5(O) = 0$. ■

Since all four constructions of a graph G and assignments of edge weights and categories described above can clearly be carried out in polynomial time (polynomial in size of a Boolean formula F), this completes the proofs of NP-completeness.

Acknowledgement

The authors are grateful to an anonymous referee for a very careful reading of the paper and for numerous constructive suggestions that improved the presentation tremendously, as well as for providing an alternative simpler proof of Theorem 1:

The transformation will be from the following variant of the satisfiability problem (see Garey-Johnson [8], problem LOG4):

A Boolean formula F consists of clauses with exactly three literals each and with no negated variables. Does there exist such a Boolean valuation of variables that each clause contains exactly one true literal?

We shall have one graph A for each variable, one modified graph B for each clause (see Figure 14) and only one category S_i for each variable x_i and one extra category S_0. In the figure of the graph B for clause $C = x_1+x_2+x_3$ the label i attached to an edge denotes its category. The edges of weight 1 are drawn thick and edges of weight 0 are thin.

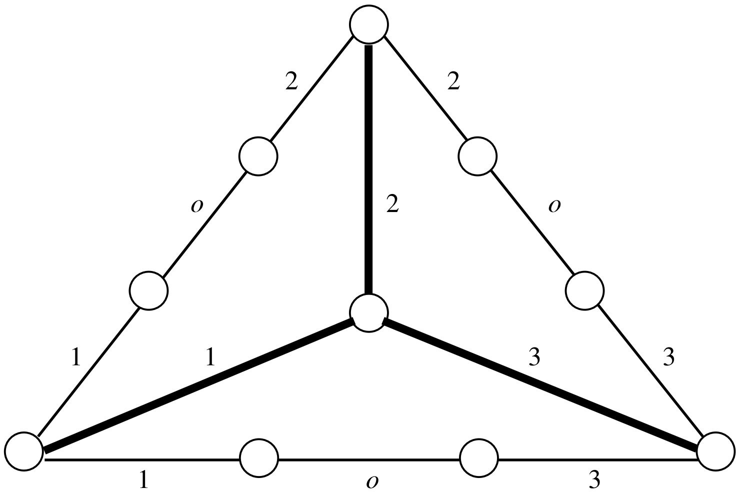

Figure 14

The reader may verify that the Boolean formula is satisfiable if and only if there exists a perfect matching M for the constructed graph with $L_5(M) = 0$.

A support of the Slovak VEGA grant 1/7465/20 is acknowledged.

References

[1] I. Averbakh and O. Berman, *Categorized bottleneck-minisum path problems on networks*, Oper. Res. Letters **16** (1994) 291–297.

[2] Š. Berežný, K. Cechlárová and V. Lacko, *Optimization problems on graphs with categorization of edges*, in: Proc. SOR 2001, eds. V. Rupnik, L. Zadnik-stirn, S. Drobne (Preddvor, Slovenia, 2001) 171–176.

[3] Š. Berežný, K. Cechlárová and V. Lacko, *A polynomially solvable case of optimization problems on graphs with categorization of edges*, in: Proc. of MME'1999 (Jindřichův Hradec, 1999) 25–31.

[4] Š. Berežný and V. Lacko, *Special problems on graphs with categorization*, in: Proc. of MME'2000 (Praha, 2000) 7–13.

[5] Š. Berežný and V. Lacko, *Easy (polynomial) problems on graphs with categorization*, in: Proc. of New trends of aviation development (Air Force Academy of gen. M.R. Štefánik, Košice, 2000) 36–46.

[6] G. Cornuéjols, D. Naddef and W.R. Pulleyblank, *Halin graphs and the Travelling salesman problem*, Mathematical Programming **26** (1983) 287–294.

[7] C.W. Duin and A. Volgenant, *Minimum deviation and balanced optimization: A unified aproach*, Operation Research Letters **10** (1991) 43–48.

[8] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness (Freeman, New York, 1979).

[9] M. Gavalec and O. Hudec, *Balanced Location on a Graph*, Optimization, **35** (1995) 367–372.

[10] V. Lacko, *Persistency in Traveling Salesman Problem on Halin graphs*, Discuss. Math. Graph Theory **20** (2000) 231–242.

[11] S. Martello, W.R. Pulleyblank, P. Toth and D. de Werra, *Balanced optimization problems*, Oper. Res. Lett. **3** (1984) 275–278.

[12] A.P. Punnen, *Traveling salesman problem under categorization*, Oper. Res. Lett. **12** (1992) 89–95.

[13] M.B. Richey and A.P. Punnen, *Minimum weight perfect bipartite matchings and spanning trees under categorizations*, Discrete Appl. Math. **39** (1992) 147–153.

Received 18 December 2000
Revised 6 May 2002