Complexity of Finding a Join of Maximum Weight ⋆

A. A. Ageev [1]

Sobolev Institute of Mathematics, 630090, Novosibirsk, Russia

Abstract

A subset of edges $J \subseteq E(G)$ in a undirected graph G is called a join if at most half the edges of each cycle of G are contained in J. In this paper we consider the problem of finding a join of maximum weight: given a graph G and an edge weighting $c : E(G) \to \mathbf{R}$, find a join of maximum weight. We show that the problem is NP-hard even in the case of $0, 1$-weights, which answers in the negative a question of A. Frank. We also show that in the case of series-parallel graphs and arbitrary weights the problem can be solved in time $O(n^3)$, where n is the number of vertices in G.

Key words: Graph; Join; NP-hard problem; Series-parallel graph; Polynomial-time algorithm

1 Introduction

In this paper a graph stands for an undirected multigraph without loops.

Let G be a graph. A subset of edges $J \subseteq E(G)$ is called a *join* of G, if at most half the edges of each cycle in G belong to J. We consider the problem of finding a join of maximum weight that can be stated as follows: given a graph G and an edge weighting $c : E(G) \to \mathbf{R}$, find a join of G with maximum weight. Frank [4] proves that the problem is polynomial-time solvable in the case of unit weights. In Section 2 of this paper we show that the problem is NP-hard in the case of $0, 1$-weights, which answers in the negative a question

⋆ Translated from *Discrete Analysis and Operations Research*, Novosibirsk, 4, no. 3 (1997), 3–8.

[1] Supported in part by DIMANET/PECO, contract ERBCIPDCT 94–0623, and by the Russian Foundation for Basic Research, grant 96–01–01614.

E-mail address: ageev@math.nsc.ru

in [4]. In Section 3 we show that in the case of series-parallel graphs and arbitrary weights the problem can be solved in time $O(n^3)$, where n is the number of vertices in the graph.

2 NP-hardness

Theorem 1 *The problem of finding a join of maximum weight in a graph G is* NP*-hard even when restricted to the case $c(e) \in \{0, 1\}$ for all $e \in E(G)$.*

PROOF. To show that the $0, 1$-weights case of the maximum weight join problem is NP-hard it suffices to show that the following recognition problem is NP-complete: given a graph G, a positive integer k, and an edge subset $F \subseteq E(G)$, decide whether there exists a join J such that $J \subseteq F$ and $|J| \geq k$. We prove this by constructing a polynomial time reduction from the independent set problem (which is well known to be NP-complete, see [1]). Let a graph G and an integer k form an instance of the independence set problem. Consider the graph H that is obtained from G by adding a new vertex u and edges connecting u with all vertices of G. Let F denote the set of these edges. For any subset $Y \subseteq V(G)$, denote by $\varphi(Y)$ the set of edges in F that connect the vertices in Y with u. Now let $X \subseteq V(G)$. If X is independent in G, then each triangle of H contains at most one edge in $\varphi(X)$, whereas the cycles of larger length can have at most two such edges. Thus $\varphi(X)$ is a join in H. If $\varphi(X)$ is a join of H, then the vertices in X are pairwise non-adjacent, since otherwise H would contain a triangle with two edges in $\varphi(X)$. It follows that G contains an independent set of cardinality k if and only if H contains a join $J \subseteq F$ of size k. This proves the theorem. $\square$

3 The case of series-parallel graphs

A graph is called *series-parallel* if it can be obtained from K_2 (K_n here and hereafter stands for the complete graph on n vertices) by a successive application of the following operations:

(1) adding a parallel edge to an edge;
(2) replacing some edge with a simple path.

The following classical result of Dirac [2] and Duffin [3] establishes a characterization of series-parallel graphs.

Theorem 2 ([2,3]) *A 2-connected graph G is a series-parallel graph if and only if G does not contain a subgraph homeomorphic to K_4.*

Notice that in the literature series-parallel graphs are often defined as those that do not contain subgraphs isomorphic to K_4. Theorem 2 shows that the distinction between thus produced classes of graphs is immaterial from the viewpoint of algorithmic complexity of the maximum weight join problem. On the other hand, when constructing algorithms for some optimization problems, the classical definition turns out to be more appropriate (see, e. g. [5]). Moreover, in our treatment, as well as in [5], the well-known equivalent recursive definition of series-parallel graphs in terms of linkages of two-terminal graphs proves to be useful.

By a *two-terminal graph* (G, s, t) we shall mean a graph G with two specified vertices denoted by s and t, and called *terminals* .

Let (G_1, s_1, t_1) and (G_2, s_2, t_2) be two-terminal graphs. A graph (H, u, v) is called a *series linkage* of (G_1, s_1, t_1) and (G_2, s_2, t_2) if H is obtained from G_1 and G_2 by identifying the terminals t_1 and s_2, and moreover, $E(H) = E(G_1) \cup E(G_2)$, $u = s_1$, and $v = t_2$. A graph (H, u, v) is called a *parallel linkage* of (G_1, s_1, t_1) and (G_2, s_2, t_2) if H is obtained from G_1 and G_2 by identifying the terminal s_1 with the terminal s_2, and the terminal t_1 with the terminal t_2; and moreover, $E(H) = E(G_1) \cup E(G_2)$, the vertex u is the result of identifying s_1 with s_2, and the vertex v is the result of identifying t_1 with t_2.

In the above terms series-parallel graphs can be equivalently defined in the following way: a two-terminal graph (G, s, t) is *series-parallel* if it satisfies either of the following conditions:

(1) it consists only of the edge st;
(2) it is either a series or a parallel linkage of two two-terminal series-parallel graphs.

Let J be a join of a graph G. Define a function $w_J : E \to \{-1, +1\}$ by the equality

$$\begin{cases} w_J(e) = -1 & \text{if } e \in J, \\ +1 & \text{otherwise.} \end{cases}$$

It follows from the definition of join that every cycle of G has nonnegative weight with respect to w_J. An arbitrary function from $E(G)$ to $\{-1, +1\}$, having this property, uniquely defines a join in G and is called a *conservative weighting* of the graph G [6]. For any two vertices x and y in G, denote by $\lambda_J(G, x, y)$ the length of a minimum weighted (with respect to w_J) path connecting x and y.

Lemma 3 *Let (G, s, t) be a two-terminal graph and x be the length of a shortest (with respect to the number of edges) path between s and t. Then for any join J of G,*

$$-x \leq \lambda_J(G, s, t) \leq x.$$

PROOF. The second inequality holds, since by definition $\lambda_J(G, s, t)$ does not exceed the length of every (s, t)-path. We now prove the first inequality. Let J be a join of (G, s, t). Let P' and P'' denote the sets of edges in a shortest (with respect to the number of edges) and a minimum weighted (with respect to w_J) (s, t)-paths, respectively. Denote by P the symmetric difference of P' and P''. It suffices to prove that the total weight of edges in $P \cap P''$ is at least $-|P \cap P'|$. It is easy to see that P induces an Eulerian subgraph of G. Denote it by H. Since H is Eulerian, it splits into edge disjoint cycles $C_1, \ldots, C_r$. For $i = 1, \ldots, r$, let $E(C_i)$ denote the set of edges of the cycle C_i. Then, for each $i = 1, \ldots, r$, $E(C_i) = L'_i \cup L''_i$ where $L'_i \subseteq P'$ and $L''_i \subseteq P''$. Since w_J is a conservative weighting,

$$0 \leq \sum_{e \in L'_i} w_J(e) + \sum_{e \in L''_i} w_J(e) \leq |L'_i| + \sum_{e \in L''_i} w_J(e),$$

that is $\sum_{e \in L''_i} w_J(e) \geq -|L'_i|$. By adding up the last inequalities over all i from 1 to r, we arrive at the desired conclusion. $\Box$

Let (G, s, t) be a two-terminal series-parallel graph in which each edge e has a weight $c(e)$. For any fixed integer k, let $Q(G, s, t, k)$ denote the maximum value of $c(J) = \sum_{e \in J} c(e)$ over all joins J of G such that $\lambda_J(G, s, t) = k$; if the set of such joins is empty, we assume that $Q(G, s, t, k) = -\infty$.

Theorem 4 . *Let (G_i, s_i, t_i) $(i = 1, 2)$ be two-terminal series-parallel graphs. Then*

(a) *if (G, s, t) is a series linkage of these graphs, then*

$$Q(G, s, t, k) = \max\{Q(G_1, s_1, t_1, k_1) + Q(G_2, s_2, t_2, k_2) : k_1 + k_2 = k\}; \quad (1)$$

(b) *if (G, s, t) is a parallel linkage of these graphs, then*

$$\begin{aligned} Q(G, s, t, k) = \max(&\max\{Q(G_1, s_1, t_1, k) + Q(G_2, s_2, t_2, k_2) : k + k_2 \geq 0\}, \\ &\max\{Q(G_1, s_1, t_1, k_1) + Q(G_2, s_2, t_2, k) : k + k_1 \geq 0\}). \quad (2) \end{aligned}$$

PROOF. Note that in both cases the join J in G is the union of two disjoint joins J_1 and J_2 in the subgraphs G_1 and G_2, respectively. Consequently, $c(J) = c(J_1)+c(J_2)$. In the case when G is a series linkage of G_1 and G_2, the converse is also true: the union J of any two joins J_1 and J_2 in G_1 and G_2, respectively, is a join in G; moreover, $\lambda_J(G,s,t) = \lambda_{J_1}(G_1,s,t)+\lambda_{J_2}(G_2,s,t)$. This implies (a). Consider now the case when G is a parallel linkage of G_1 and G_2. Assume that $J = J_1 \cup J_2$ is a join in G. But then the joins J_1 and J_2 satisfy the inequality $\lambda_{J_1}(G_1,s,t) + \lambda_{J_2}(G_2,s,t) \geq 0$. And conversely, the union of J_1 and J_2 is a join in G whenever J_1 and J_2 are joins in G_1 and G_2, respectively, and satisfy the above inequality. This proves (b). $\square$

The process of constructing a series-parallel graph G can be represented by a rooted binary tree. Each vertex of this tree is associated with a subgraph of G; moreover, the root is associated with G itself, and the leaves are associated with the subgraphs of G isomorphic to K_2. If subgraphs H_1 and H_2 are associated with the descendants of a vertex corresponding to a subgraph H, then H is either a series or a parallel linkage of H_1 and H_2. A linear algorithm for constructing such a tree is described in [7]. By using these tree, the recursive relations (1) and (2), and Lemma 3, one can successively compute the values $Q(H,s,t,k)$ for all the subgraphs H of G. Since, by Lemma 3, the total number of these quantities is equal to $O(n^2)$, the computations can be implemented in $O(n^3)$ time. Finally, by using the standard technique of dynamic programming, given the values of $Q(H,s,t,k)$, one can find a join of maximum weight. The overall running time of the described algorithm is $O(n^3)$.

References

[1] M. Garey and D. Johnson,*Computers and Intractability, A guide to the theory of NP-completeness* (Freeman, San Francisco, 1979).

[2] G. A. Dirac, A property of 4-chromatic graphs and some remarks on critical graphs, *J. London Math. Soc.* **17** (1952) 85–92.

[3] R. J. Duffin, Topology of series parallel networks,*J. Math. Anal. Appl.* **10** (1965) 303–318.

[4] A. Frank, Conservative weightings and ear decompositions of graphs, *Combinatorica* **13** (1993) 65–82.

[5] R. Hassin, A. Tamir, Efficient algorithms for optimization and selection on series-parallel graphs, *SIAM J. Algebraic Discrete Methods* **7** (1986) 379–389.

[6] A. Sebő, Undirected distances and the postman structure of graphs, *J. Combin. Theory* **49**(B) (1990) 10–39.

[7] J. Valdes, R. E. Tarjan, E. L. Lawler, The recognition of series parallel digraphs, *SIAM J. Comput.* **11** (1982) 298–313.