Finding cheapest cycles in vertex-weighted quasi-transitive and extended semicomplete digraphs

Jørgen Bang-Jensen
Department of Mathematics and Computer Science
University of Southern Denmark, Odense M
DK-5230, Denmark, jbj@imada.sdu.dk

Gregory Gutin
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK, gutin@cs.rhul.ac.uk

Anders Yeo
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK, anders@cs.rhul.ac.uk

Abstract

We consider the problem of finding a minimum cost cycle in a digraph with real-valued costs on the vertices. This problem generalizes the problem of finding a longest cycle and hence is NP-hard for general digraphs. We prove that the problem is solvable in polynomial time for extended semicomplete digraphs and for quasi-transitive digraphs, thereby generalizing a number of previous results on these classes. As a byproduct of our method we develop polynomial algorithms for the following problem: Given a quasi-transitive digraph D with real-valued vertex costs, find, for each $j = 1, 2, \ldots, |V(D)|$, j disjoint paths $P_1, P_2, \ldots, P_j$ such that the total cost of these paths is minimum among all collections of j disjoint paths in D.

Keywords: vertex-weighted digraphs, quasi-transitive digraphs, minimum cost cycles.

1 Introduction, Terminology and Notation

For basic terminology and notation on digraphs, see [3]. We consider (non-empty) finite digraphs with no loops or parallel arcs. For a compact introduction to flows as well as applications of flows to find certain substructures in digraphs, some of which are used extensively in this paper, see Chapter 3 in [3]. (For a more comprehensive account on flows in networks and their applications, see [1].)

For a digraph $D = (V, A)$, the *order* (*size*) of D is the cardinality of V (A). We will denote the order (size) of a digraph under consideration by n (m). If a digraph $D = (V, A)$ has an arc from x to y, we will denote it by $xy \in A$ or $x{\rightarrow}y$. We write $R{\rightarrow}S$ for disjoint subsets or digraphs R, S if $r{\rightarrow}s$ for every choice of vertices $r \in R, s \in S$.

In this paper, by a *cycle* (*path*) we mean simple directed cycle (simple directed path); we often call vertex-disjoint cycles (paths) *disjoint cycles (paths)*. A digraph D is *strongly connected* (or, shortly *strong*) if for every pair x, y of its vertices, there are paths from x to y and from y to x in D.

For each $x \in V(D)$, $N^+(x)$ ($N^-(x)$) denotes the set of those vertices $y \in V(D)$ for which $x{\rightarrow}y$ ($y{\rightarrow}x$). Two vertices x, y in a digraph D are *similar* if $N^+(x) = N^+(y)$ and $N^-(x) = N^-(y)$, that is, they have the same in- and out-neighbours. For a digraph $D = (V, A)$ and a set $X \subseteq V$, $D\langle X\rangle$ is the subdigraph induced by X. When we are considering a vertex x on some cycle C we denote by x^- (x^+) the predecessor (successor) of x on C. Notice that we do not use the subscript C as it will always be clear from the context which cycle we are considering. For a pair of distinct vertices x, y on a cycle C, $C[x, y]$ is a subpath of C from x to y.

For a digraph R with vertex set $V(R) = \{u_1, u_2, \ldots, u_r\}$, and digraphs $H_1, H_2, \ldots, H_r$, let $D = R[H_1, H_2, \ldots, H_r]$ be a digraph with vertex set $V(D) = V(H_1) \cup \cdots \cup V(H_r)$, in which $xy \in A(D)$ if and only if $x \in V(H_i), y \in V(H_j)$ and $u_iu_j \in A(R)$, where $i \neq j$, or $xy \in A(H_i)$ and $x, y \in V(H_i)$. In other words, D is obtained from R by substituting the vertex u_i with H_i for $i = 1, 2, \ldots, r$.

A *k-path-cycle subdigraph* of a digraph D is a collection of k paths and some cycles, all disjoint. A 0-path-cycle subdigraph is a *cycle subdigraph*. Thus, a cycle subdigraph is a collection of vertex-disjoint cycles. A k-path-cycle subdigraph with no cycles is a *k-path subdigraph*. Let $X \subseteq V(D)$ be non-empty. We say that a subdigraph D' of D *covers* X if $X \subseteq V(D')$.

We will often assign real-valued *costs* to vertices of digraphs. These costs will always be finite. The cost of a subset of vertices is the sum of the costs of its vertices and the cost of a subdigraph is the sum of the costs of its vertices. For $i = 1, 2, \ldots, n$ we define $mp_i(D)$ ($mpc_i(D)$) to be the minimum cost of an i-path (i-path-cycle) subdigraph of D. By definition $mp_0(D) = 0$ and $mpc_0(D)$ is zero if D has no negative cycle and otherwise it is the minimum cost of a cycle subdigraph in D. Note that these numbers always exist as we may take single vertices as paths and we always have $mpc_i(D) \leq mp_i(D)$. For any digraph D with at least one cycle we denote by $mc(D)$ the minimum cost of a cycle in D.

Let $D = (V, A)$ be a digraph and let X be a non-empty subset of V. We say that a cycle C in D is an X-*cycle* if C contains all vertices of X. In this paper we consider the following problems for a digraph $D = (V, A)$ with n vertices and real-valued costs on the vertices:

(P1) Determine $mp_i(D)$ for all $i = 1, 2, \ldots, n$.

(P2) Find a cheapest cycle in D or determine that D has no cycle.

Clearly, problems (P1) and (P2) are NP-hard as determining the numbers $mp_1(D)$ and $mc(D)$ generalize the hamiltonian path and cycle problems (assign cost -1 to each vertex of D). The problem (P2) can be solved in time $O(n^3)$ when all costs are non-negative using an all pairs shortest path calculation.

In this paper, we develop polynomial algorithms for both problems for some special classes of digraphs, quasi-transitive digraphs and extended semicomplete digraphs, defined below. These classes have been extensively studied in the literature, see, e.g., [3] and references therein, and more recent papers [5, 9]. Since the costs are arbitrary real numbers, we can also find most expensive cycles and path subdigraphs for these classes of digraphs in polynomial time.

Notice that (P1) and (P2) for the special case when all costs are non-negative were solved in [4]. However, the approach of [4] cannot be used or modified to work with negative costs. Through the use of a more efficient minimum cost flow algorithm as a subroutine, we obtain a better complexity (than in [4]) for the problem of finding a most expensive path in a quasi-transitive digraph.

The maximization version of problem (P2) (which is equivalent to (P2) itself) is of interest as a special case of the Prize Collecting Travelling Salesman Problem (PCTSP) [2]. In the PCTSP, a salesman wishes to visit a set V of cities and he gets price p_i if he visits city i. However, he pays penalty q_i if he fails to visit city i. Given distances d_{ij} between the cities, the salesman wants to maximize

$$\sum_{i\in U} p_i - \sum_{i\in W} q_i - \sum_{(i,j)\in C} d_{ij},$$

where U is the set of visited cities, $W = V - U$ and C is a cycle with vertex set U. The problem is NP-hard.

In our special case we assume that the entries of matrix $[d_{ij}]$ are 0 and ∞, corresponding to the case when the salesman pays for the travel in some directions nothing or very little, but some other directions of travel may be too expensive or forbidden. This means that we have a digraph D with vertex set V and want to maximize $-\sum_{i\in W} q_i + \sum_{i\in U} p_i$ provided U is the vertex set of a cycle in D. By adding the constant $\sum_{i\in V} q_i$ to the last objective function, we see that we can reduce the last problem into the maximum cost cycle problem by assigning new cost $q_i + p_i$ to every city i.

A digraph $D = (V, A)$ is *semicomplete* if there is an arc between any pair of vertices of D. Notice that, for some $x \neq y \in V$, we may have both arcs $xy, yx \in A$. Semicomplete digraphs generalize tournaments. A digraph D is an *extended semicomplete* if there is a semicomplete digraph R and digraphs $E_{n_1}, \dots, E_{n_r}$ with no arcs such that $D = R[E_{n_1}, \dots, E_{n_r}]$. In other words, an extended semicomplete digraph is obtained from a semicomplete digraph by substituting vertices with sets of independent vertices. A digraph D is *transitive* (*quasi-transitive*) if $x{\rightarrow}y$ and $y{\rightarrow}z$, for distinct vertices x, y, z, implies that $x{\rightarrow}z$ (either $x{\rightarrow}z$ or $z{\rightarrow}x$ or both). Notice that extended semicomplete digraph with no cycles of length two form a special class of quasi-transitive digraphs.

A digraph $D = (V, A)$ is *semicomplete multipartite* if V can be partitioned into $V_1 \cup \cdots \cup V_p$ such that, for a pair $x \in V_i, y \in V_j$ of distinct vertices, there is an arc between x and y if and only if $i \neq j$. Clearly, extended semicomplete digraphs form a special class of semicomplete multipartite digraphs. We call the sets $V_1, \dots, V_p$ the *partite sets* of D. Note that if D is extended semicomplete and x, y belong to the same partite set of D, then x and y are similar.

We finish this introduction by pointing out that for semicomplete multipartite digraphs D the problem (P1) can be readily solved in polynomial time (the polynomial time complexity follows by combining Corollary 1.2 and Lemma 2.4(a)); see Theorem 3.1. The following result was proved by the second author.

Theorem 1.1 *[7] A semicomplete multipartite digraph D has a hamiltonian path if and only if it has a spanning 1-path-cycle subdigraph F. Given a* *spanning 1-path-cycle subdigraph F in D, a hamiltonian path of D can be found in time $O(n^2)$.*

By Theorem 1.1, in a semicomplete multipartite digraph D all cycles of a k-path-cycle subdigraph with $k \geq 1$ can be merged with one of the paths to form a new path. This easily implies the following corollary which plays an important role in our algorithms.

Corollary 1.2 *Let D be a semicomplete multipartite digraph. Then for every $i = 1, 2, \ldots, n$ we have $mp_i(D) = mpc_i(D)$.*

2 Minimum cost k-path-cycle subdigraphs

In this section we recall some results from [3, Section 3.11] which will be used later. We say that a flow f in a network N is *integer-valued* if the value of f on any arc is an integer. Given a network $N = (V, A, l, u)$ with lower bound 0 and capacity $u(a) \geq 0$ on each arc $a \in A$ we say that a flow f is a *feasible* flow in N if $0 \leq f(a) \leq u(a)$ holds for every $a \in A$. Below we will always assume that the flows we consider are feasible. Let N be a network with two designated vertices s and t (called the *source* and the *sink*). An (s, t)-flow in N is a feasible flow f which satisfies the following for some k. (The number k is called the *value* of the flow f.)

$$\sum_{w:vw \in A} f(vw) - \sum_{w:wv \in A} f(wv) = \begin{cases} k & \text{if } v = s \\ -k & \text{if } v = t \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

Below we also allow capacities and costs on the vertices in our networks. This makes it easier to model certain problems for digraphs and it is easy to transform such a network into one where all capacities and costs are on the arcs (see [3, Section 3.2.4] for details). With these remarks in mind, the following lemma follows directly from [3, Lemma 3.2.4 and Proposition 3.10.7].

Lemma 2.1 *Let $N = (V, A)$ be a network with source s and sink t, capacities on arcs and vertices and a real-valued cost $c(v)$ for each vertex $v \in V$. For all integer i such that there exists a feasible (s, t)-flow of value i in N, let f_i be a minimum cost (s, t)-flow in N of value i and let $c(f_i)$ be the cost of f_i. Then, for all i where all of f_{i-1}, f_i, f_{i+1} exist, we have*

$$c(f_{i+1}) - c(f_i) \geq c(f_i) - c(f_{i-1}). \qquad (2)$$

The following lemma is a consequence of Lemma 3.3.2 in [3].

Lemma 2.2 *Given an arbitrary feasible integer-valued (s,t)-flow f in a network N of value k one can find in time $O(nm)$ a collection of k (s,t)-paths, $P_1,\dots,P_k$, and zero or more cycles $C_1,\dots,C_r$, $r \geq 0$ such that the (s,t)-flow one obtains by sending one unit of flow along each of $P_1,\dots,P_k,C_1,\dots,C_r$ is precisely* [1] *the flow f. Furthermore, if each arc (vertex, except for s and t) of N has capacity one then the paths and cycles above are all arc-disjoint (vertex-disjoint, except for s and t).*

Recall that a cycle subdigraph of a digraph D is a collection of vertex-disjoint cycles of D.

Lemma 2.3 *Let $D=(V,A)$ be a digraph with real-valued cost function c on the vertices. In time $O(n(m+n\log n))$ we can determine the number $mpc_0(D)$ and find a cycle subdigraph of cost $mpc_0(D)$ if $mpc_0(D)<0$.*

Proof: Let $H(w)$ be the digraph on 4 vertices w_1,w_2,w_3,w_4 and the following arcs $w_1w_2, w_2w_1, w_2w_3, w_3w_4, w_4w_3$. Let $D^*=(V^*,A^*)$ be obtained from D as follows: replace every vertex v by the digraph $H(v)$. Furthermore, for every original arc $uv\in A$, D^* contains the arc u_4v_1 . There are no costs on the vertices and all arcs have cost 0 except the arcs of the form v_2v_3 which have cost $c(v)$. Observe that $mpc_0(D)$ is precisely the minimum cost of a spanning cycle subdigraph in D^*. Let $V^*=\{x_1,x_2,\dots,x_{4n}\}$. Construct a bipartite graph B with partite sets $L=\{\ell_1,\dots,\ell_{4n}\}$ and $R=\{r_1,\dots,r_{4n}\}$, in which ℓ_ir_j is an edge if and only if $x_ix_j\in A^*$. Moreover, the cost of ℓ_ir_j is equal to the cost of x_ix_j. Observe that a minimum cost perfect matching in B corresponds to a minimum cost cycle subdigraph in D^*. We can find a minimum cost perfect matching in B in time $O(n(m+n\log n))$, see the remark after the proof of Theorem 11.1 in [10]. Using the transformations from B to D^*, we can compute the minimum cost of a spanning cycle subdigraph F in D^* in time $O(n(m+n\log n))$. If this cost is negative, we can find a minimum cost cycle subdigraph of D within the same time. □

Lemma 2.4 *Let $D=(V,A)$ be a digraph in which every vertex has a real-valued cost.*

(a) In total time $O(n^2m+n^3)$ we can determine the numbers

[1] That is, the sum of the flows on the arc a is $f(a)$ for every arc a.

$\{mpc_1(D), mpc_2(D), \ldots, mpc_n(D)\}$ *and find* j*-path-cycle subdigraphs* F_j*,* $j = 1, 2, \ldots, n$*, where* F_j *has cost* $mpc_j(D)$*.*

(b) *The costs* $mpc_i(D)$ *satisfy the following inequality for* $i = 1, 2, \ldots, n-1$*:*

$$mpc_{i+1}(D) - mpc_i(D) \geq mpc_i(D) - mpc_{i-1}(D) \qquad (3)$$

Proof: Form a network $N(D)$ from D by adding a pair s, t of new vertices along with arcs $\{(s, v), (v, t) : \; v \in V\}$. Let all vertices and all arcs of D have lower bound 0 and capacity 1. Let $c(s) = c(t) = 0$, let each other vertex of $N(D)$ inherit its cost from D and let all arcs have cost 0.

Suppose F_j is a j-path-cycle subdigraph of D. Using F_j we can obtain a feasible flow f_j of value j in $N(D)$ if we assign $f_j(a) = 1$ to all arcs a in F_j and those arcs a of $N(D)$ that start (terminate) at s (t) and terminate (start) at the initial (terminal) vertex of a path in F_j, and $f_j(a) = 0$ for all other arcs of $N(D)$. Similarly, by Lemma 2.2, we can transform a feasible integer-valued (s, t)-flow of value j in $N(D)$ into a j-path-cycle subdigraph of D.

Notice that $N(D)$ has a feasible integer-valued (s, t)-flow of value k for any integer $k = 0, 1, \ldots, n$. Thus it follows from the observations above that for every $j = 0, 1, \ldots, n$ the value $mpc_j(D)$ is exactly the minimum cost of a flow of value j in $N(D)$. Now (2) implies that the inequality (3) is valid.

It remains to prove (a). It follows from Lemma 2.3 that we can find a minimum cost flow f of value 0 in time $O(n^3)$. Now we can use the so-called Buildup algorithm (see e.g. [3, Section 3.10]) starting from f. Using the Buildup algorithm we can find feasible integer-valued flows f_j, $j = 1, 2, \ldots, n$, such that f_j is a minimum cost feasible (s, t)-flow of value j in $N(D)$, in total time $O(n^2 m)$ (the complexity of obtaining f_{j+1} starting from f_j is $O(nm)$). This proves (a). $\square$

3 Minimum cost i-path subdigraphs and cycles in semicomplete multipartite digraphs

By Corollary 1.2 to determine the value $mp_i(D)$ for some $i > 0$ in a semicomplete multipartite digraph we just have to find the minimum cost of an i-path cycle subdigraph in D. Now Lemma 2.4 implies the following:

Theorem 3.1 *Let* D *be a semicomplete multipartite digraph with real-valued costs on the vertices. In total time* $O(n^2 m + n^3)$ *we can determine the* *numbers $mp_i(D)$, $i = 1, 2, \ldots, n$ and find corresponding cheapest path subdigraphs.*

The following lemma was proved by the second author, see e.g. [3, Section 5.7].

Lemma 3.2 *Let D be an extended semicomplete digraph, and let F be a cycle subdigraph in D. If $D\langle V(F)\rangle$ is strong, then there exists a cycle, C, in D, with $V(C) = V(F)$. In particular, if there is some partite set V_i in D, such that every cycle in F contains a vertex from V_i, then there also exists a cycle, C, in D, with $V(C) = V(F)$.*

Theorem 3.3 *Let $D = (V, A)$ be an extended semicomplete digraph with a cycle and real-valued costs on V. We can find a cheapest cycle in D in time $O(n^3m + n^4 \log n)$.*

Proof: Running an all pairs shortest path algorithm in $O(n^3)$ time we can either find a shortest cycle of D or determine that D has a cycle of negative cost (see [3, Section 2.3.5]). So assume that there exists a negative cost cycle in D, and let $W \subseteq V$ be the set of vertices in D with negative cost. Let $S_1, S_2, \ldots, S_t$ be a partition of W, such that t is maximum ($t \geq 1$) and $S_i \rightarrow S_j$ (recall that it means that every vertex of S_i dominates every vertex of S_j) for each $j > i$. It follows from the definition of an extended semicomplete digraph that S_i either induces a strong component in $D\langle W\rangle$ or is a maximal set of independent vertices in $D\langle W\rangle$.

We consider the cases when $t = 1$ and $t \geq 2$ separately.

Assume that $t = 1$, which implies that either $D\langle W\rangle$ is strong or W is independent. By Lemma 2.3 we can find a minimum cost cycle subdigraph, F, in D in time $O(n(m + n \log n))$. Since the cost of F is negative, we may assume (by discarding cycles of cost zero if necessary) that every cycle of F contains a vertex of negative weight. If W is independent, then Lemma 3.2 implies that we can obtain a cycle in D with the same cost as F, which therefore is optimal. So assume that $D\langle W\rangle$ is strong. By Lemma 3.2, we may assume that we can order the cycles $C_1, C_2, \ldots, C_s$ of F such that $C_i \rightarrow C_j$, whenever $i < j$ (otherwise we can merge some cycles). Assume that $s > 1$. Since $D\langle W\rangle$ is strong there must be some path totally within $D\langle W\rangle$, from a cycle C_j to C_i, with $j > i$, such that the path only has its end-vertices in common with $V(F)$. Clearly this path together with C_i and C_j can be merged into one cycle of cost less than $c(C_i) + c(C_j)$ (use the path plus the appropriate arc from C_i to C_j). This contradicts the optimality of F. Hence $s = 1$ and C_1 is the desired minimum cost cycle.

Assume that $t \geq 2$. For all $1 \leq i \leq j \leq t$ we define $D_{i,j}$ as follows:

$$D_{i,j} = D\left\langle V - (S_1 \cup S_2 \cup \ldots \cup S_{i-1}) - (S_{j+1} \cup S_{j+2} \cup \ldots \cup S_t)\right\rangle.$$

We will now show how to find a cheapest cycle $C_{i,j}$ in $D_{i,j}$, which contains both a vertex from S_i and a vertex from S_j (possibly the same vertex when $i = j$). By taking the cheapest cycle of all cycles $C_{i,j}$ we can clearly get an optimal cycle in D.

If $i = j$, then we proceed as above when $t = 1$, so assume that $i < j$. Let $M = S_{i+1} \cup S_{i+2} \cup \cdots \cup S_{j-1}$ ($M = \emptyset$ is possible), and define the new digraph $D'_{i,j}$ by adding a new vertex a and new arcs to $D_{i,j}$ such that $S_i \rightarrow a \rightarrow S_j$. Let $X_{i,j}$ contain a minimum cost vertex from each partite set in M (i.e. $X_{i,j}$ contains exactly one vertex from each partite set in M, and it is a vertex of minimum cost). Now let all costs in $D'_{i,j}$ be the same as in $D_{i,j}$ except for the vertices in $X_{i,j}$ which are assigned cost zero and the cost of a which is a negative number large enough to force a minimum cost cycle subdigraph in $D'_{i,j}$ to use it (if there is any cycle subdigraph using it). Let c denote the costs in $D_{i,j}$ and let c' denote the costs in $D'_{i,j}$. We now find a minimum cost cycle subdigraph, $F'_{i,j}$, in $D'_{i,j}$ in time $O(n(m + n\log n))$. If $F'_{i,j}$ does not contain a, then there is no path from S_j to S_i in $D_{i,j}$, as such a path together with a would produce a cycle and the cost assignment to a would force $F'_{i,j}$ to contain a. So in this case $C_{i,j}$ does not exist. Thus, we may assume that $a \in V(F'_{i,j})$. We will now show that the cost of $C_{i,j}$ is exactly $c'(F'_{i,j} - \{a\}) + c(X_{i,j})$.

We first show how to transform an optimal cycle $C_{i,j}$ into a cycle subdigraph containing a in $D'_{i,j}$.

Consider $C_{i,j}$ and a pair of vertices $s_i \in S_i,\ s_j \in S_j$ such that no vertex from $S_i \cup S_j$, apart from s_i, s_j themselves, is in the subpath $C_{i,j}[s_i, s_j]$. Suppose that $C_{i,j}[s_i, s_j^-]$ has a pair x, y of vertices from the same partite set such that x appears earlier than y in $C_{i,j}[s_i, s_j^-]$. Then $y \rightarrow x^+$ and $x \rightarrow y^+$. Hence, the arc from y to x^+ together with $C_{i,j}[x^+, y]$ forms a cycle $Q^{(1)}$ and the arc from x to y^+ together with $C_{i,j}[y^+, x]$ forms a cycle $C_{i,j}^{(1)}$ which contains s_i, s_j. Considering $C_{i,j}^{(1)}$ instead of $C_{i,j}$ and continuing in the manner above, after a number k of steps, we will arrive to the situation when the current substitute $C_{i,j}^{(k)}$ of $C_{i,j}$ will not have any pair of vertices from the same partite set in $C_{i,j}^{(k)}[s_i, s_j^-]$.

If there is a vertex $x \in C_{i,j}^{(k)}[s_i^+, s_j^-]$, such that $c'(x) < 0$, then by the minimality of $c(C_{i,j})$, the minimum cost vertex (w.r.t. c), from the same partite set as x, must belong to some cycle $Q^{(p)}$, $p \leq k$. Now swap this vertex (which belongs to $X_{i,j}$) and x, which can be done as they are similar. Continuing like this we get $c'(C^{(k)}[s_i^+, s_j^-]) \geq 0$. Remove from $C^{(k)}[s_i^+, s_j^-]$ all vertices of $C^{(k)}[s_i^+, s_j^-]$, and add the path $s_i \rightarrow a \rightarrow s_j$ instead. This gives us the cycle $C_{i,j}^{(k+1)}$. Let $F' = Q^{(1)} \cup \cdots \cup Q^{(k)} \cup C_{i,j}^{(k+1)}$ denote the resulting cycle subdigraph, and note that $c(C_{i,j}) \geq c'(C_{i,j}) + c(X_{i,j}) \geq c'(F' - \{a\}) + c(X_{i,j})$.

We now show how to transform $F'_{i,j}$ into the desired cycle in $D_{i,j}$. Delete a from $F'_{i,j}$, and note that this results in a path from S_j to S_i, and a number of cycles in $D_{i,j}$. Assume that the path starts in $s_j \in S_j$ and ends in $s_i \in S_i$. If $X_{i,j} \subseteq V(F'_{i,j})$, then add the arc from s_i to s_j in order to obtain a cycle subdigraph $F^*_{i,j}$. If $X_{i,j} \not\subseteq V(F'_{i,j})$ then we obtain $F^*_{i,j}$, by inserting a hamiltonian path in $D\left\langle X_{i,j} - V(F'_{i,j})\right\rangle$ between s_i and s_j (this is possible since $D\langle X_{i,j}\rangle$ is a tournament, and $s_i \rightarrow X_{i,j} \rightarrow s_j$). As we only insert vertices from $X_{i,j}$, we note that $c'(F^*_{i,j}) = c'(F'_{i,j} - a)$. Since $F'_{i,j}$ is a minimum cost cycle subdigraph we may assume that every cycle of $F^*_{i,j}$ contains a vertex from W. Now use Lemma 3.2 to merge cycles as long as we can, and let $F''_{i,j}$ be the resulting cycle subdigraph.

Suppose that $F''_{i,j}$ is not a cycle. Let $C_1, C_2, \ldots, C_s$ be an ordering of the cycles in $F''_{i,j}$ such that $C_u \rightarrow C_w$, for all $u < w$. This ordering exists since we could not merge more cycles above using Lemma 3.2. Let C_r be the cycle containing both s_i and s_j. If C_q is some cycle different from C_r, then we must have that $W \cap V(C_q)$ belongs to S_i or S_j but not both, since otherwise $D\langle V(C_r) \cup V(C_q)\rangle$ is strong (this follows from the fact that $S_i \rightarrow S_j$ and C_r contains both s_i and s_j) and we could apply Lemma 3.2. Furthermore, if $W \cap V(C_q) \subseteq S_i$, then $C_q \rightarrow C_r$ and if $W \cap V(C_q) \subseteq S_j$, then $C_r \rightarrow C_q$.

Thus, if $r > 1$, then $V(C_k) \cap S_i \neq \emptyset$ for every $k = 1, 2, \ldots, r-1$. If S_i were an independent set, then we could merge $C_1, C_2, \ldots, C_{r-1}$ with C_r, which is impossible. Hence, S_i is a strong component in $D\langle W\rangle$. So if $r > 1$, then we take a shortest path from C_r to $C_1 \cup C_2 \cup \ldots \cup C_{r-1}$ in S_i, which together with the two cycles it touches can be merged into one cycle, contradicting the minimality of $c'(F'_{i,j})$. If $r < s$, then we can also merge some cycles (looking at S_j instead of S_i). So $F''_{i,j}$ is a cycle, and we see that $c(F''_{i,j}) = c'(F''_{i,j}) + c(X_{i,j}) = c'(F^*_{i,j}) + c(X_{i,j}) = c'(F'_{i,j} - \{a\}) + c(X_{i,j})$. So the cost of an optimal cycle $C_{i,j}$ in $D_{i,j}$ must be less than or equal to $c'(F'_{i,j} - \{a\}) + c(X_{i,j})$.

We have now shown that $c(C_{i,j}) = c'(F'_{i,j} - \{a\}) + c(X_{i,j})$, as desired. And, furthermore, the argument above indicates how to obtain the cycle $C_{i,j}$, given the cycle subdigraph $F'_{i,j}$. Therefore, as we construct $O(n^2)$ different cycle subdigraphs $F'_{i,j}$ we can find the desired cycle in $O(n^3 m + n^4 \log n)$

time as stated in the theorem. □

The third author [11] proved that, in time $O(n^5)$, one can verify whether a semicomplete multipartite digraph has a cycle covering a prescribed vertex set X and find one, if it exists. He conjectured that a longest cycle covering a prescribed set of vertices in a semicomplete multipartite digraph can be found in polynomial time. We conjecture the following generalization.

Conjecture 3.4 *A cheapest cycle in a semicomplete multipartite digraph with real-valued costs on the vertices can be found in polynomial time.*

4 Cheapest i-path subdigraphs in quasi-transitive digraphs

The following theorem which is a slight weakening of a result from [6] shows that quasi-transitive digraphs have a recursive structure with semicomplete digraphs and acyclic transitive digraphs as building blocks. We will make extensive use of this decomposition theorem below.

Theorem 4.1 *[6] Let Q be a quasi-transitive digraph.*

If Q is strong, then there exists an integer t, a semicomplete digraph T on t vertices, and digraphs $Q_1, Q_1, \dots, Q_t$ each of which is either a single vertex or a non-strong quasi-transitive digraph, such that $Q = T[Q_1, Q_2, \dots, Q_t]$.

If Q is non-strong, then there exists an integer t, an acyclic graph T on t vertices[2], and strong quasi-transitive digraphs $Q_1, Q_2, \dots, Q_t$, such that $Q = T[Q_1, Q_2, \dots, Q_t]$.

Furthermore one can find the above decompositions in $O(n^2)$ time.

The next theorem shows that (P1) is polynomially solvable for quasi-transitive digraphs.

Theorem 4.2 *Let $D = (V, A)$ be a quasi-transitive digraph, with real-valued costs on its vertices. Then the following holds:*

(a): *In total time $O(n^2m + n^3)$ we can find for every $i = 1, 2, \dots, n$, the value of $mp_i(D)$ and an i-path subdigraph F_i of cost $mp_i(D)$.*

[2] In fact, T is transitive, but for our purposes in this paper it suffices to say that T is acyclic.

(b) *For all* $i = 1, 2, \ldots, n-1$ *we have*

$$mp_{i+1}(D) - mp_i(D) \geq mp_i(D) - mp_{i-1}(D) \tag{4}$$

Proof: We prove (b) by induction on n. The statement vacuously holds for $n = 1$ and is easy to verify for $n = 2$ (recall that, by definition, $mp_0(D) = 0$). This proves the basis of induction and we now assume that $n \geq 3$.

By Theorem 4.1, D has a decomposition $D = T[Q_1, \ldots, Q_t]$, $t = |T| \geq 2$, where T is an acyclic digraph or a semicomplete digraph. Assume that (b) holds for each Q_k, $k = 1, 2, \ldots, t$. Let $D' = T[E_{n_1}, \ldots, E_{n_t}]$ be obtained from D by deleting all arcs inside each Q_i, $i = 1, 2, \ldots, t$. Assign costs to the vertices $v_1^k, \ldots, v_{n_k}^k$ of E_{n_k}, as follows:

$$c'(v_j^k) = mp_j(Q_k) - mp_{j-1}(Q_k) \tag{5}$$

By the induction hypothesis (b) holds for Q_k implying that we have

$$c'(v_j^k) \leq c'(v_{j+1}^k) \text{ for every } j \geq 1 \tag{6}$$

Let F_i' be an i-path-cycle subdigraph of D'. If T is acyclic then D' is acyclic and, thus, F_i' is an i-path subdigraph of D'. If T is semicomplete, then D' is extended semicomplete and, thus, by Theorem 1.1 and Corollary 1.2, we may assume that F_i' is an i-path subdigraph of D'. Hence, $mp_i(D') = mpc_i(D')$ and it follows from Lemma 2.4(b) that (4) holds for D'. Thus it suffices to prove that $mp_i(D) = mp_i(D')$.

Let F_i' by an i-path subdigraph of D' and let p_k denote the number of vertices from E_{n_k} which are covered by F_i'. Since all vertices of E_{n_k} are similar it follows from (6) that we may assume (by making the proper replacements if necessary) that F_i' includes $v_1^k, \ldots, v_{p_k}^k$. For each k, replace the vertices $v_1^k, \ldots, v_{p_k}^k$ in F_i' by a p_k-path subdigraph of Q_k with cost $mp_{p_k}(Q_k) = \sum_{i=1}^{p_k} c(v_i^k)$. As a result, we obtain, from F_i', an i-path subdigraph F_i of D for which we have $c'(F_i') = \sum_{k=1}^{t} mp_{p_k}(Q_k) = c(F_i)$ and, thus, $c(F_i) = c'(F_i')$. Reversing the process above it is easy to get, from an i-path subdigraph of D, an i-path subdigraph F_i' of D' such that $c(F_i) = c'(F_i')$. This shows that $mp_i(D) = mp_i(D')$ and hence (4) holds for D by the remark above.

We prove the complexity by induction on n. Let m' be the number of arcs in D' and recall that all these arcs are also in D. Clearly when a digraph H has $|V(H)| \leq 2$ we can chose a constant c_1 so that we can determine the numbers $mp_i(H)$, $i = 1, 2, \ldots, |V(H)|$ in time at most $c_1|V(H)|^2(|A(H)| +$

$|V(H)|$). Now assume by induction that for each Q_i we can determine the desired numbers inside Q_i in time at most $c_1 n_i^2(m_i + n_i)$. This means that we can find all the numbers $mp_i(Q_j)$, $j = 1, 2, \ldots, t$, $i = 1, 2, \ldots, n_j$ in total time

$$\sum_{j=1}^{t} c_1 n_j^2 (m_j + n_j) \leq c_1 n^2 \sum_{j=1}^{t} (m_j + n_j) = c_1 n^2 (m - m' + n).$$

By Lemma 2.4 (a), Theorem 1.1 and Corollary 1.2, there is a constant c_2 such that in total time at most $c_2 n^2(m' + n)$ we can find, for every $j = 1, 2, \ldots, n$, a j-path-cycle subdigraph of cost $mp_j(D')$ in D'. It follows from the way we construct F_i above from F_i' that if we are given for each $k = 1, \ldots, t$ and each $1 \leq j \leq n_k$ a j-path subdigraph in Q_k of cost $mp_j(Q_k)$, then we can construct all the path subdigraphs F_r, $1 \leq r \leq n$ in time at most $c_3 n^3$ for some constant c_3. Hence the total time needed by the algorithm is at most

$$c_1 n^2 (m - m' + n) + c_2 n^2 (m' + n) + c_3 n^3 = c_1 n^2 (m + n) + (c_2 - c_1) n^2 m' + (c_2 + c_3) n^3,$$

which is at most $c_1 n^2(m + n)$ for c_1 sufficiently large. □

The next theorem which is an easy consequence of Theorem 4.2 (give all vertices cost -1) improves the complexity $O(n^5)$ of the algorithm from [4].

Theorem 4.3 *One can find a longest path in any quasi-transitive digraph in time* $O(n^2 m + n^3)$.

Sometimes, one is interested in finding path subdigraphs that include maximum number of vertices from a given set X or avoid as many vertices of X as possible. We consider a minimum cost extension of this problem in the next result.

Theorem 4.4 *Let* $D = (V, A)$ *be a quasi-transitive digraph with real-valued costs on the vertices and let* $X \subseteq V$ *be non-empty. Let* p_j *be the maximum possible number of vertices from* X *in a* j*-path subdigraph and let* q_j *be maximum possible number of vertices from* X *not in a* j*-path subdigraph. In total time* $O(n^2 m + n^3)$ *we can find, for all* $j = 1, 2, \ldots, n$*, a cheapest* j*-path-subdigraph which includes* p_j *(avoids* q_j*, respectively) vertices of* X*.*

Proof: Let $C = \sum_{v \in V} |c(v)|$ and subtract $C + 1$ from the cost of every vertex in X. Now, for each $j = 1, 2, \ldots, n$, every cheapest j-path subdigraph F_j must cover as many vertices from X as possible, i.e., p_j vertices. Furthermore, since the new cost of F_j is exactly the original one minus $p_j(C + 1)$, cheapest j-path subdigraphs covering p_j vertices from X are preserved under this transformation. Now the 'including' part of the claim follows from Theorem 4.2(a). The 'avoiding' part can be proved similarly, by adding $C + 1$ to every vertex of X. □

5 Finding a cheapest cycle in a quasi-transitive digraph

Using the solution of (P2) for extended semicomplete digraphs we can now give a short proof that (P2) is polynomial for quasi-transitive digraphs.

Theorem 5.1 *For quasi-transitive digraphs with real-valued costs on the vertices the minimum cost cycle problem can be solved in time* $O(n^5 \log n)$.

Proof: Let D be a quasi-transitive digraph. If D is not strong then we simply look at the strong components, so assume that D is strong. By Theorem 4.1, $D = T[Q_1, \ldots, Q_t]$, where T is a strong semicomplete digraph, and each Q_i is either a single vertex or a non-strong quasi-transitive digraph.

Suppose we have found a minimum cost cycle C_i in each Q_i which contains a cycle. Then clearly the minimum cost of a cycle in D is the minimum cost cycle among those cycles C_i that exist and the minimum cost of a cycle C which intersects at least two Q_i's. Hence it follows that applying this approach recursively we can find the minimum cost cycle in D. Now we show how to compute a minimum cost cycle C as above.

Let D' be defined as in the proof of Theorem 4.2 including the vertex-costs. It is easy to show using the same approach as when we converted between i-path subdigraphs of D' and D in the proof of Theorem 4.2, that the cost of C is precisely $mc(D')$. Now it follows from Theorem 3.3 that we can find the cycle C in time $O(n^3 m + n^4 \log n)$.

Since we can construct D' including finding the costs for all the vertices in time $O(n^2 m + n^3)$ by Theorem 4.2 and there are at most $O(n)$ recursive calls the approach above will lead to a minimum cost cycle of D in time $O(n^4 m + n^5 \log n)$. In fact, we can bound the first term as we did in the proof of Theorem 4.2 and obtain $O(n^3 m + n^5 \log n) = O(n^5 \log n)$ rather than $O(n^4 m + n^5 \log n)$. This completes the proof. □

It is not difficult to formulate and prove a 'cycle' analog of Theorem 4.4; we leave it to the reader.

Acknowledgement Part of the work was carried out while the first and the third author were visiting Department of Mathematics and Statistics, University of Victoria, B.C, Canada. They thank the department for its hospitality and NSERC for financial support. Part of the work was carried out while the second author was visiting University of Haifa, Israel. He thanks the HIACS Research Center, the Caesarea Edmond Benjamin de Rothschild Foundation Institute for Interdisciplinary Applications of Computer Science (Israel) and Leverhulme Trust (UK) for financial support. All three authors thank the Danish Natural Science Research Council for financial support.

References

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows*, Prentice Hall, Upper Saddle River, 1993.

[2] E. Balas, The price collecting traveling salesman problem and its applications. In *The Traveling Salesman Problem and its Variations* (G. Gutin and A.P. Punnen, eds.), Kluwer, 2002.

[3] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, Springer-Verlag, London, 2000.

[4] J. Bang-Jensen and G. Gutin, Vertex heaviest paths and cycles in quasi-transitive digraphs. *Discrete Math.* **163** (1997) 217-223.

[5] J. Bang-Jensen, G. Gutin and A. Yeo, Steiner type problems for digraphs that are locally semicomplete or extended semicomplete. To appear in *Discrete Appl. Math.*

[6] J. Bang-Jensen and J. Huang, Quasi-transitive digraphs. *J. Graph Theory* **20** (1995) 141-161.

[7] G. Gutin, A characterization of complete n-partite digraphs that have a Hamiltonian path. *Kibernetika* no. 1 (1988) 107-108. [In Russian]

[8] G. Gutin, Polynomial algorithms for finding hamiltonian paths and cycles in quasi-transitive digraphs, *Australasian J. Combinatorics* **10** (1994) 231-236.

[9] G. Gutin and A. Yeo, Orientations of digraphs almost preserving diameter. *Discrete Appl. Math.* **121** (2002) 129–138.

[10] B. Korte and J. Vygen, *Combinatorial Optimization*, Springer, Berlin, 2000.

[11] A. Yeo, A polynomial algorithm for finding a cycle covering a given set of vertices in a semicomplete multipartite digraph, *J. Algorithms* **33** (1999) 124-139.