Paths and cycles in extended and decomposable digraphs

Jørgen Bang-Jensen
Gregory Gutin*
Department of Mathematics and Computer Science
Odense University, Denmark

Abstract

We consider digraphs – called extended locally semicomplete digraphs, or extended LSD's, for short – that can be obtained from locally semicomplete digraphs by substituting independent sets for vertices. We characterize Hamiltonian extended LSD's as well as extended LSD's containing Hamiltonian paths. These results as well as some additional ones imply polynomial algorithms for finding a longest path and a longest cycle in an extended LSD. Our characterization of Hamiltonian extended LSD's provides a partial solution to a problem posed by R. Häggkvist in [14]. Combining results from this paper with some general results derived for so-called totally Φ-decomposable digraphs in [3], we prove that the longest path problem is polynomially solvable for totally Φ_0-decomposable digraphs - a fairly wide family of digraphs which is a common generalization of acyclic digraphs, semicomplete multipartite digraphs, extended LSD's and quasi-transitive digraphs. Similar results are obtained for the longest cycle problem and other problems on cycles in subfamilies of totally Φ_0-decomposable digraphs. These polynomial algorithms are a natural and fairly deep generalization of algorithms obtained for quasi-transitive digraphs in [3] in order to solve a problem posed by N. Alon.

*This work was supported by the Danish Research Council under grant no. 11-0534-1. The support is gratefully acknowledged.

1 Introduction

The purpose of this paper is twofold. First, we introduce and investigate extended locally semicomplete digraphs[1] (extended LSD's, for short) - a common generalization of two well-studied families of digraphs, locally semicomplete digraphs (see e.g. [1, 2, 8, 16]) and extended semicomplete digraphs (see e.g. [4, 11, 12]). It is shown that extended LSD's inherit some useful properties of both "parents". Second, combining the results obtained for extended LSD's with some general results derived for totally Φ-decomposable digraphs in [3] we prove the longest path problem is polynomially solvable for totally Φ_0-decomposable digraphs, a fairly wide family of digraphs which is a common generalization of acyclic digraphs, semicomplete multipartite digraphs (see e.g. [4, 10, 11, 16]), extended LSD's and quasi-transitive digraphs (see e.g. [3, 5, 13]). Similar results are obtained for the longest cycle problem and other problems on cycles in subfamilies of totally Φ_0-decomposable digraphs. These polynomial algorithms are a natural and fairly deep generalization of algorithms obtained for quasi-transitive digraphs in [3] in order to solve a problem posed by N. Alon.

We list some results obtained for extended LSD's: In Sections 4 and 5 we show that an extended LSD has a Hamiltonian path (cycle) if and only if it has a path and a collection of cycles, all pairwise disjoint, which span the vertex set (is strong and has a spanning collection of disjoint cycles). These characterizations imply $O(n^3)$ algorithms for finding a Hamiltonian path and a Hamiltonian cycle in an extended LSD D with n vertices (if D contains one). R. Häggkvist [14] posed the problem of characterizing those digraph families for which a member is Hamiltonian if and only if it is strongly connected and contains a spanning collection of disjoint cycles. Our characterization of Hamiltonian extended LSD's provides a partial solution to this problem.

We point out that the algorithm for constructing a longest cycle in case of extended LSD's is much more difficult than that in case of extended semicomplete digraphs. Using the algorithms above as well as some additional results we construct polynomial algorithms for finding a longest path and a longest cycle in an extended LSD.

2 Terminology and preliminaries

We shall assume that the reader is familiar with the standard terminology on graphs and digraphs and refer the reader to [6],[7].

For any digraph D, the *underlying graph* of D is the graph obtained by ignoring the orientations of arcs in D and deleting parallel edges. We say that D is *connected* if its underlying graph is connected.

If $U \subset V(D)$ then we denote by $D < U >$ the subgraph of D induced by the vertices in U. We use n (m) to denote the number of vertices (arcs) of the actual digraph studied.

Let D be a digraph. If there is an arc from a vertex x to a vertex y in D we say that x *dominates* y and use the notation $x \to y$ to denote this. If A

[1] For definitions see the next section

and B are disjoint subsets of vertices of D such that there is no arc from B to A and $a{\rightarrow}b$ for every choice of $a \in A$ and $b \in B$, then we denote this by $A{\Rightarrow}B$. If $a{\rightarrow}b$ and $b{\rightarrow}a$ for all $a \in A$ and $b \in B$, then we write $A{\Leftrightarrow}B$. We let I_x (respectively, O_x) denote the set of vertices dominating x (respectively, dominated by) x in D. We call $|I_x|$ ($|O_x|$) the *in-degree* (*out-degree*) of x. Two vertices x and y are called *similar* if they are not adjacent and $I_x = I_y$, $O_x = O_y$.

A *path* (*cycle*) will always mean a directed path (cycle). If x and y are vertices of D and P is a path from x to y, we say that P is an $(x, y)-$path. If P is a path containing a subpath from x to y we let $P[x, y]$ denote that subpath. Similarly, if C is a cycle containing vertices x and y, $C[x, y]$ denotes the subpath of C from x to y.

A digraph D is said to be *semicomplete* if every pair of vertices of D is joined by an arc or by a pair of mutually opposite arcs. A *semicomplete multipartite digraph* is a digraph that can be obtained from a complete r-partite graph, for some $r \geq 2$, by giving each edge an orientation, or replacing it with two oppositely oriented arcs. A *locally semicomplete digraph* (LSD, for short) is a digraph for which the following holds: for every vertex x the vertices dominated by x induce a semicomplete digraph and the vertices that dominate x induce a semicomplete digraph. It was proved [1] that every connected LSD has a Hamiltonian path.

A digraph D is *strongly connected* (or just *strong*) if there exists an $(x, y)-$path and a $(y, x)-$path in D for every choice of distinct vertices x, y of D. It was shown [1] that every strong LSD has a Hamiltonian cycle. If a digraph is not strong then we can label its strong components $D_1, \ldots, D_s$, $s \geq 2$, such that there is no arc from D_j to D_i if $j > i$. In general this labelling is not unique, but it is so for LSD's [1, Theorem 3.1].

A *k-path-cycle subgraph* of a digraph D is a subgraph of D consisting a disjoint collection of k paths and some cycles. If the number of cycles is zero we call it a *k-path subgraph*. A *cycle subgraph* is a nonempty 0-path-cycle subgraph.

A *trivial digraph* is a digraph without arcs. Let R be a digraph on r vertices $v_1, ..., v_r$ and let $H_1, ..., H_r$ be a disjoint collection of digraphs. Then $D = R[H_1, ..., H_r]$ is the new digraph obtained from R by replacing each vertex v_i of R by H_i and adding an arc from every vertex of H_i to every vertex of H_j if and only if (v_i, v_j) is an arc of R ($1 \leq i \neq j \leq r$). If each of $H_1, ..., H_r$ is trivial, D is called an *extension* of R. In particular, a digraph D is an *extended locally semicomplete digraph* (extended LSD, for short) if there exists a LSD R such that $D = R[H_1, \ldots, H_r]$, where each H_i is an independent set of vertices (possibly of size 1).

Let Φ be a set of digraphs containing the trivial digraph with one vertex. A digraph D is called totally Φ-*decomposable* if either D has only one vertex, or there is a decomposition $D = R[H_1, ..., H_r]$, $r \geq 2$ so that $R \in \Phi$ and each of $H_1, ..., H_r$ is *totally* Φ-*decomposable.* In this case, the decomposition $D = R[H_1, ..., H_r]$, appropriate decompositions $H_i = R_i[H_{i1}, ..., H_{ir_i}]$ of all H_i except trivial ones on one vertex, appropriate decompositions of all H_{ij} except trivial ones of order 1, and so on, form a *total* Φ-*decomposition* of D.

A digraph D is called *quasi-transitive* if for any triple x, y, z of distinct vertices of D such that (x, y) and (y, z) are arcs of D there is at least one arc from x to z or from z to x. Let Ψ be the union of all acyclic and all semicomplete digraphs. The following is a weakening of a decomposition theorem from [5]: Every quasi-transitive digraph is totally Ψ-decomposable. One can find a total Ψ-decomposition of D in time $O(n^3)$.

The following claim was proved in [10, 11].

Proposition 2.1 *For every connected digraph D, a spanning cycle subgraph respectively, a spanning 1-path-cycle subgraph, can be found (if it exists) in time $O(n^{\frac{5}{2}})$. Furthermore a maximum order cycle subgraph respectively, a maximum order 1-path-cycle subgraph can be found in time $O(n^3)$.*

3 Basic properties of extended LSD's

The following two claims can be easily obtained from the definition of a LSD.

Lemma 3.1 *Let D be a connected extended LSD with decomposition $D = R[H_1, \dots, H_r]$, $r \geq 2$. If x and y are similar vertices of D, then $\{x, y\} \subset V(H_i)$ for some i.*

Proposition 3.2 *A connected extended LSD has a unique decomposition $D = R[H_1, \dots, H_r]$, $r \geq 2$, where R is a LSD.* □

Lemma 3.3 *Let D be a digraph and let $C_1 = x_1x_2 \dots x_rx_1$ and $C_2 = y_1 \dots y_sy_1$ be disjoint cycles in D. If x_i and y_j are similar vertices, then there exists a cycle C^* in D with $V(C^*) = V(C_1) \cup V(C_2)$.*

Proof: If x_i and y_j are similar, then $x_i \rightarrow y_{j+1}$ and $y_j \rightarrow x_{i+1}$, so we can take $C^* = C_1[x_{i+1}, x_i]C_2[y_{j+1}, y_j]x_{i+1}$. □

Lemma 3.4 *Let D be an extended LSD and let P_1 be an (x, y)-path and P_2 an (x, z)-path (possibly with $y = z$) which is internally disjoint from P_1. If no vertex of $V(P_1) \setminus V(P_2)$ is similar to a vertex of $V(P_2) \setminus V(P_1)$, then the following holds:*

1. *D contains a path P starting in x and ending in either y or z such that $V(P) = V(P_1) \cup V(P_2)$.*
2. *Furthermore, on P the relative order of vertices from P_i, $i = 1, 2$ is preserved.*
3. *P can be found in time $O(q)$, where q is the number of arcs between P_1 and P_2.*

Similarly, paths ending in the same vertex and otherwise disjoint can be merged, provided they have no pair of similar vertices.

Proof: Let $P_1 = x_1x_2 \dots x_k$ and $P_2 = y_1y_2 \dots y_r$ where $x_1 = y_1 = x$, $x_k = y$ and $y_r = z$. If $k = 1$, or $r = 1$, the claim is trivial, so we can assume $k \geq 2$ and $r \geq 2$. Note that x_2 and y_2 are adjacent, because they are not similar. Suppose $x_2 \rightarrow y_2$, then the claim follows by induction applied to the paths $P_1[x_2, x_k]$ and $x_2P_2[y_2, y_r]$. Similarly, if $y_2 \rightarrow x_2$. It is easy to see that the proof implies an $O(q)$ algorithm to merge the paths. □

Proposition 3.5 *Let D be a connected non-strong extended LSD.*

1. *If A and B are distinct strong components of D, then either $A \Rightarrow B$, or $B \Rightarrow A$, or there are no arcs between A and B.*
2. *There is a unique ordering $D_1, \dots, D_s$, $s \geq 2$ of the strong components of D so that there is no arc from D_j to D_i for $j > i$.*
3. *Furthermore with this ordering we have $D_1 \Rightarrow D_2 \Rightarrow \dots \Rightarrow D_s$.*

Proof: First note that no two distinct strong components A and B can contain vertices $a \in A$ and $b \in B$ such that a and b are similar. Now 1. follows from the fact that D is an extended LSD. From 1. it follows that the digraph obtained from D by contracting each strong component to one vertex is a non-strong connected LSD. By [1, Theorem 3.1(c)] it follows that the vertices $u'_1, \dots, u'_k$ of this LSD can be ordered in a unique way $u_1, \dots, u_k$ such that there is no arc from u_j to u_i for $j > i$ and $u_i \rightarrow u_{i+1}$ for $i = 1, 2, \dots, k-1$. Now 2. and 3. follow immediately. □

Corollary 3.6 *If D is a connected extended LSD which is not strong, then each strong component of D is an extended semicomplete digraph.* □

4 Longest cycles in extended LSD's

Lemma 4.1 *If C_1 and C_2 are disjoint cycles in an extended LSD such that no vertex on C_1 has a similar vertex on C_2 and $V(C_1) \cup V(C_2)$ induces a strong digraph, then D has a cycle C^* such that $V(C^*) = V(C_1) \cup V(C_2)$. Furthermore C^* can be found in time $O(|E_{C_1,C_2}|)$, where E_{C_1,C_2} is the set of arcs with one end-vertex in C_1 and the other in C_2 provided we are given two arcs e_{12}, e_{21} such that e_{12} goes from C_1 to C_2 and e_{21} from C_2 to C_1.*

Proof: Let e_{12} be an arc from C_1 to C_2 and e_{21} be an arc from C_2 to C_1. If e_{12} and e_{21} are not disjoint, then it is easy to see that, using the fact that there is no pair of similar vertices $x \in V(C_1)$, $y \in V(C_2)$, we can find a new disjoint pair e'_{12}, e'_{21} in time $O(|E_{C_1,C_2}|)$.

Let $C_1 = x_1x_2 \dots x_kx_1$ and $C_2 = y_1y_2 \dots y_ry_1$. The labelling is chosen such that $x_1 \rightarrow y_1$ and $y_i \rightarrow x_j$ for some $i > 1$, $j > 1$. It is not difficult to see that this can be done when D is an extended LSD. Applying Lemma 3.4 to the paths $C_1[x_1, x_j]$ and $x_1C_2[y_1, y_i]x_j$, we obtain an (x_1, x_j)-path P with $V(P) = \{x_1, x_2, \dots, x_j, y_1, y_2, \dots, y_i\}$. Furthermore, the vertices appear in the same order in P as they did on C_1 respectively C_2. Hence, $P[x_1, y_1]$ contains only the vertex y_1 from C_2 and $P[y_i, x_j]$ contains only y_i from C_2. This implies that the paths $C_2[y_i, y_1]$ and $P[y_i, x_j]C_1[x_j, x_1]P[x_1, y_1]$ contain no similar vertices u and v such that u and v belong to different paths. Thus applying Lemma 3.4 to these paths we obtain the desired cycle C^*.

Now the complexity claim follows from the proof and Lemma 3.4. □

The following characterization generalises the characterization of Hamiltonian extended semicomplete digraphs [11, 12] and is analogous to that of Hamiltonian semicomplete bipartite digraphs [9, 15].

Theorem 4.2 *An extended LSD is Hamiltonian if and only if it is strong and has a spanning cycle subgraph. Given a spanning cycle subgraph of a strong extended LSD D, one can find a Hamiltonian cycle in time $O(n^3)$.*

Proof: The necessity is clear. To prove the sufficiency we suppose that $F = C_1 \cup ... \cup C_k$ is a spanning cycle subgraph of D. By Lemma 4.1 we can assume that no two cycles of F induce a strong digraph. By Proposition 3.5, if two cycles C_i and C_j are adjacent, then either $C_i \Rightarrow C_j$, or $C_j \Rightarrow C_i$. Now it is easy to see that the digraph obtained by contracting each cycle C_i into one vertex c_i is a LSD D'. Since D' is strong it has a Hamiltonian cycle [1]. Let $c_1 c_2 \dots c_k c_1$ be such a cycle, where we have relabelled the cycles to allow the numbering. Now in D we have $C_1 \Rightarrow C_2 \Rightarrow \dots \Rightarrow C_k \Rightarrow C_1$ and it is easy to see that D is Hamiltonian.

It is not difficult to see that the proof above implies an $O(n^3)$ algorithm to find a Hamiltonian cycle, given a spanning cycle subgraph $C_1, ..., C_k$ of a strong extended LSD. □

We can prove that the complexity in Theorem 4.2 can be decreased to $O(n^2)$. However, we shall not give a proof of this result here since our proof is rather long, complicated and involves some advanced data structures.

Corollary 4.3 *There exists an $O(n^3)$-algorithm which, given any extended locally semicomplete digraph D, decides whether D is Hamiltonian and finds a Hamiltonian cycle if it exists.*

Proof: To check the existence of a spanning cycle subgraph we need $O(n^{\frac{5}{2}})$ time, by Proposition 2.1. Checking whether D is strong can be done in linear time $O(n + m)$. If D is not strong, or has no spanning cycle subgraph, then we stop. Otherwise use the algorithm of Theorem 4.2. □

Extended LSD's inherit the following property of extended semicomplete digraphs. Note that semicomplete bipartite digraphs, in general, do not satisfy this property [11].

Theorem 4.4 *If D is a strong extended LSD and $C_1, \dots, C_k$ is a collection of disjoint cycles of D, then D has a cycle C with $V(C_1) \cup \dots \cup V(C_k) \subset V(C)$.*

Proof: The proof is by induction on k. The case $k = 1$ is trivial, so assume $k \geq 2$. By Theorem 4.2, we can assume that $D' = D < V(C_1) \cup \ldots \cup V(C_k) >$ is not strong, and by induction we can assume that $C_1, \ldots, C_k$ form the strong components of D'.

Suppose first that D' is connected. Then, by Proposition 3.5, we can assume, by relabelling if necessary, that $C_1 \Rightarrow C_2 \Rightarrow \ldots \Rightarrow C_k$. Since D is strong, there exists a path P starting in a vertex x on C_k and ending in a vertex y on C_i, for some $i < k$ such that P has only x and y in common with $V(C_1) \cup \ldots \cup V(C_k)$. Then P together with $C_i \cup \ldots \cup C_k$ induce a Hamiltonian digraph and the claim follows by induction.

Now suppose that D' is not connected. Let $D'_1, \ldots, D'_r$, $r \geq 2$, be the connected components of D'. Again, we can relabel $C_1, \ldots, C_k$ such that $C_1, \ldots, C_{i_1}$ are in D'_1, $C_{i_1+1}, \ldots, C_{i_2}$ are in D'_2 and so on until $C_{i_{r-1}+1}, \ldots, C_{i_r} = C_k$ which are in D'_r. We can also assume, by Proposition 3.5, that if $i_j - i_{j-1} \geq 2$, then we have $C_{i_{j-1}+1} \Rightarrow \ldots \Rightarrow C_{i_j}$ for $j = 1, 2, \ldots r$, where $i_0 = 0$ and $i_r = k$.

Claim : If $P = xx' \ldots y'y$ (where possibly $x' = y'$) is any shortest path starting in a vertex $x \in V(D'_s)$ and ending in a vertex $y \in V(D'_t)$, $1 \leq s \neq t \leq r$, such that P has only x and y in common with $V(D')$, then $C_{i_s} \Rightarrow x'$ and $y' \Rightarrow C_{i_{t-1}+1}$. Hence we can replace x by a vertex in C_{i_s} and y by a vertex in $C_{i_{t-1}+1}$.

Proof of the claim: By the minimality of P, no vertex of $V(P) - \{x, y\}$ can be similar to a vertex in $V(D'_s) \cup V(D'_t)$. This implies that x' cannot dominate any vertex of $V(D'_s)$, because this would lead to a contradiction on either the minimality of P, or the fact that there is no arc between $V(D'_s)$ and $V(D'_t)$. Using this and Proposition 3.5 (1) , we conclude that $V(C_q) \Rightarrow x'$, where $x \in V(C_q)$. If $i_{s-1} + 1 \leq q < i_s$, then we use the fact that $V(C_q) \Rightarrow x'$ and $V(C_q) \Rightarrow V(C_{q+1})$ to conclude that $C_{q+1} \Rightarrow x'$ and, hence, by induction $C_{i_s} \Rightarrow x'$. The second part of the claim is proved similarly. □

Since D is strong, there exists an (x, y)-path P with $x \in V(C_{i_1})$ and $y \in V(C_{i_{t-1}+1})$, for some $t > 1$, such that P is shortest possible and has only x and y in common with $V(D')$. Using the claim and the fact that D is strong we can conclude that there is also a (u, v)-path P' with $u \in V(C_{i_t})$ and $v \in C_1$ with the following properties :

1. P' has only u and v in common with $V(D'_1) \cup V(D'_t)$,
2. if P' enters a D'_j $j \neq 1, t$, then it enters in $V(C_{i_{j-1}+1})$ and leaves in $V(C_{i_j})$ and contains all vertices of $V(D'_j)$,

If P and P' are disjoint, except for their endvertices, then D has a cycle C containing all vertices of $V(P) \cup V(P')$ and all vertices of those components D'_j that P' enters. Thus we can finish the proof by induction.

Suppose P and P' intersect in some vertex $z \notin V(D')$. Then it is easy to see that we can either replace some cycles from $C_1, \ldots, C_k$ by one and apply induction, or we can include some vertices of $V(D) - V(D')$ in a new collection of k cycles. Hence the claim follows by induction on the number of vertices in $V(D) - V(D')$. □

Corollary 4.5 *There exists an $O(n^3)$ algorithm to find the cycle C above if $C_1, \ldots, C_k$ are given.*

Proof: This follows from a close inspection of the proof above. □

Combining Proposition 2.1 and Corollary 4.5, we obtain.

Corollary 4.6 *There exists an $O(n^3)$ algorithm which finds a longest cycle in any extended LSD.* □

5 Longest paths in extended LSD's

Lemma 5.1 *Let D be a LSD. If P is a path in D and C a cycle in D disjoint from P such that there is an arc between P and C, then D contains a path P^* such that $V(P^*) = V(P) \cup V(C)$. If no vertex of P is similar to a vertex of C and we are given an arc between P and C, P^* can be found in time $O(q)$, where q is the number of arcs between P and C.*

Proof: Let $P = x_1x_2 \ldots x_k$ and $C = y_1y_2 \ldots y_ry_1$. It is easy to prove the claim if P and C contain similar vertices x_i and y_j. Suppose such vertices do not exist. If there is an arc $x_i \rightarrow y_j$, then our claim follows from Lemma 3.4 applied to the paths $P[x_i, x_k]$ and $x_iC[y_j, y_{j-1}]$. The proof when there is an arc $y_j \rightarrow x_i$ is analogous. The complexity claim follows from Lemma 3.4. □

Theorem 5.2 *An extended LSD D has a Hamiltonian path if and only if it is connected and has a spanning 1-path-cycle subgraph. Furthermore, given a spanning 1-path-cycle subgraph of D, one can find a Hamiltonian path in D in time $O(n^2)$.*

Proof: Let $P, C_1, \ldots, C_\ell$, $\ell \geq 1$, be a 1-path-cycle subgraph L of D. Mark all cycles of L containing vertices similar to vertices of P. Then, replace P and the marked cycles by one path P'(covering the vertices of P and the marked cycles). It is easy to see that P' can found in time $O(n^2)$. Now we can apply Lemma 5.1. Since we can attribute the cost of merging a path P and a cycle C to the arcs between P and C, the total cost will not be more than $O(n^2)$. □

Theorem 5.3 *Let D be a connected extended LSD. If $P_1, \ldots, P_k, C_1, \ldots, C_\ell$, $k, \ell \geq 1$, is a k-path-cycle subgraph L of D, then D contains a k-path subgraph F covering all the vertices of $V(P_1) \cup \ldots \cup V(P_k) \cup V(C_1) \cup \ldots \cup V(C_\ell)$. Moreover, one can find F in time $O(n^2)$.*

Proof: As in Theorem 5.2 we can mark all cycles of L containing vertices similar to vertices of $P_1 \cup ... \cup P_k$ and, then, add the vertices of the marked cycles to the corresponding paths. All this takes time $O(n^2)$. Now we can assume that no vertex of a cycle of L similar to a vertex of a path of L.

If the digraph D' induced by $V(P_1) \cup \ldots \cup V(P_k) \cup V(C_1) \cup \ldots \cup V(C_\ell)$ is connected, then the theorem follows easily from Lemma 5.1. In fact, if a vertex of a cycle C_i is adjacent to a vertex of a path P_j, then, by Lemma 5.1, we may replace these by a new path P'_j with $V(P'_j) = V(P_j) \cup V(C_i)$.

Now suppose that D' is not connected and that there is no arc between a path P_j and a cycle C_i. Since D is connected, there must exist i and j such that, in the underlying graph of D, there is a path P between $V(P_j)$ and $V(C_i)$ which does not contain any vertices from $V(P_1) \cup \ldots \cup V(P_k) \cup V(C_1) \cup \ldots \cup V(C_\ell)$, other than the two endvertices. We can assume that P is chosen shortest possible among all undirected paths with endvertices in $V(P_j)$ and $V(C_i)$. This implies that P is a directed path in D, because D is an extended LSD (there can be no similar vertices on P by the minimality). Now we can apply the same technique as we did in the proof of Lemma 5.1 to get a path P'_j with $V(P'_j) = V(P_j) \cup V(C_i) \cup V(P)$ (the minimality of P implies that there are no similar vertices x and y such that $x \in V(P_j) \cup V(C_i)$ and $y \in V(P) \setminus (V(P_j) \cup V(C_i)))$. Thus we have reduced ℓ by one and the result follows by induction.

To see that F can be found in time $O(n^2)$, it suffices to perform a breadth first search from the set $V(P_1 \cup ... \cup P_k)$. This will provide us with the arcs we need for the merging. Again we attribute the cost of the merging to different arcs. □

6 Recognition of extended LSD's

In order for our results in the previous sections to have some practical value, it is important to show that one can decide effectively, whether a given digraph is an extended LSD.

Call a pair of vertices x and y *bad* if x and y are not adjacent and there exist some $z \in V(D) - \{x, y\}$ such that $z \rightarrow x$ and $z \rightarrow y$, or $x \rightarrow z$ and $y \rightarrow z$.

Now we can state an easy characterization of extended LSD's in terms of bad pairs:

Proposition 6.1 *Let D be an arbitrary connected digraph and let $G(D)$ be the graph with vertex set $V(D)$ and edges all the bad pairs of D. Let $U_1, \ldots, U_k$, $k \geq 1$, be the vertex sets of the connected components of $G(D)$. D is an extended LSD if and only if*

1. *$D < U_i >$ is an independent set for each $i = 1, ..., k$ and*

2. *for each pair U_i, U_j, $i \neq j$, either there is no arc between U_i and U_j, or one of the following holds $U_i \Rightarrow U_j$, or $U_j \Rightarrow U_i$, or $U_i \Leftrightarrow U_j$.*

Proof: Suppose D is an extended LSD with decomposition $D = R[H_1, \ldots, H_r]$, where R is a LSD and each H_i is an independent set of vertices. It is clear that each H_i induces a connected component in $G(D)$ and that 2. is satisfied.

Conversely, suppose $G(D)$ satisfies 1. and 2. Then $D = R[U_1, \ldots, U_k]$ where R is the digraph obtained from D by contracting each U_i to one vertex u_i. H is locally semicomplete, because, by definition of U_i and U_j, there is no bad pair x and y such that $x \in U_i$ and $y \in U_j$ for $i \neq j$. Hence D is an extended LSD. □

Corollary 6.2 *Let D be an arbitrary connected digraph. It can be decided in time $O(nm)$ whether D is an extended LSD.*

Proof: This is a direct consequence of Theorem 6.1: We can find all bad pairs in time $O(nm)$ by considering I_x and O_x for all vertices $x \in V(D)$. We can find the connected components of $G(D)$ in time $O(n^2)$ and check if each induces an independent set in D in the same time. Checking whether 2. holds can easily be done in time $O(nm)$ (in fact faster, but we do not need this here, since we already used $O(nm)$ time above). □

7 Recognition of totally Φ_i-decomposable digraphs

In the next section we describe results on paths and cycles in totally Φ-decomposable digraphs for some special sets Φ: Φ_0 is the union of all semicomplete multipartite digraphs, all connected extended LSD's and all acyclic digraphs, Φ_1 be the union of all semicomplete bipartite digraphs, all connected extended locally semicomplete digraphs and all acyclic digraphs, and Φ_2 be the union of all connected extended LSD's and all acyclic digraphs. It is easy to check that, for every $i = 0, 1, 2$, the family of totally Φ_i-decomposable digraphs coincides with the family of totally Φ'_i-decomposable digraphs, where Φ'_i is defined similarly to Φ_i with the only difference that the former includes all extended LSD's while the later contains only connected extended LSD's. We consider Φ_i instead of Φ'_i for technical reasons (see the proof of the first part of Theorem 8.2).

In this section we propose a polynomial algorithm for checking if a given digraph D is Φ_i-decomposable for $i = 0, 1, 2$. Note that every Φ_i, $i = 1, 2, 3$ is a hereditary set in the following sense. A set Φ of digraphs is *hereditary* if $D \in \Phi$ implies that every induced subgraph of D is in Φ.

Lemma 7.1 *Let Φ be a hereditary set of digraphs closed with respect to the extension. If a given digraph D is totally Φ-decomposable, then every induced subgraph D' of D is totally Φ-decomposable. I.e. total Φ-decomposability is a hereditary property.*

Proof: We prove this by induction on the number of vertices of D. The claim is obviously true if D has less than 3 vertices.

If $D \in \Phi$, then our claim follows from the fact that Φ is hereditary. So we may assume that $D = R[H_1, \ldots H_r]$, $r \geq 2$, where $R \in \Phi$ and each of $H_1, \ldots, H_r$ is totally Φ-decomposable.

Let D' be an induced subgraph of D. If there is an index i so that $V(D') \subset V(H_i)$, then D' is totally Φ-decomposable by induction. Otherwise, $D' = R'[T_1, \dots, T_{r'}]$, where $r' \geq 2$ and $R' \in \Phi$ is the subgraph of H induced by those vertices i of R, whose H_i has a nonempty intersection with $V(D')$ and the T_j's are the corresponding S_i's restricted to the vertices of D'. Note that $R' \in \Phi$, since Φ is hereditary and closed with respect to the extension. Moreover, by induction, each T_j is totally Φ-decomposable, hence so is D'. □

Lemma 7.2 *There exists an $O(mn + n^2)$-algorithm for checking if a digraph D with n vertices and m arcs has a decomposition $D = R[H_1, ..., H_r]$, $r \geq 2$, where H_i is an arbitrary digraph and the digraph R is either acyclic or semicomplete multipartite or semicomplete bipartite or connected extended locally semicomplete.*

Proof: If D is not connected and D_1,...,D_c are its components, then $D = R[D_1, ..., D_c]$, where R is a trivial (i.e. acyclic) digraph. Hence, in the rest of the proof we assume that D is connected. Consider the different kinds of R, we are interested in, step by step.

Check if R can be acyclic:

First find the strong components $D_1, \dots, D_k$ of D. If $k = 1$ then R cannot be acyclic and we can stop verifying that possibility. So suppose $k \geq 2$.

If we find two strong components D_i and D_j such that there is an arc between them but there are nonadjacent vertices $x \in D_i$ and $y \in D_j$, then we replace D_i and D_j by their union. This is justified because D_i and D_j cannot be in different sets H_s and H_t in a possible decomposition. Repeat this step but now check also the possibility for a pair D' and D'' of new "components" to have arcs between D' and D'' in different directions. In last case we also replace D' and D'' by their union. Continue this procedure until all remaining sets satisfy that either there is no arc between them, or there are all possible arcs from one to the other. Let $V_1, \dots, V_r$, $r \geq 1$ denote the distinct vertex sets of the obtained "components". If $r = 1$, then we cannot find an acyclic graph as R. Otherwise $D = R[V_1, ..., V_r]$, $r \geq 2$ and we obtain R by contracting each V_i to a vertex.

Check if R can be a semicomplete multipartite digraph:

Find the connected components $\bar{G}_1, \dots, \bar{G}_c$, $c \geq 1$, of the complement of the underling graph $G(D)$ of D. If $c = 1$, then R cannot be semicomplete multipartite. So suppose $c \geq 2$ below. Let G_j be the subgraph of $G(D)$ induced by the vertices V_j of the j'th component $\bar{G}_j$ of the complement of $G(D)$. Furthermore, let $G_{j1}, \dots, G_{jn_j}$, $n_j \geq 1$, be the connected components of G_j. Denote $V_{jk} = V(G_{jk})$.

Starting with the collection $W = \{V_1, \dots, V_c\}$, we identify two of the sets V_i and V_j if there exist V_{ia} and V_{jb} $a \in \{1, ..., n_i\}$, $b \in \{1, ..., n_j\}$ such that we have none of the possibilities $V_{ia} \Rightarrow V_{jb}$, $V_{jb} \Rightarrow V_{ia}$ or $V_{ia} \Leftrightarrow V_{jb}$. Clearly the obtained set $V_i \cup V_j$ induces a connected subgraph of D. Let $Q_1, ..., Q_r$ denote the sets obtained, by repeating this process until no more changes occur. If $r = 1$, then R cannot be semicomplete multipartite. Otherwise, H is the semicomplete multipartite digraph obtained by contracting each connected component of Q_i into a vertex.

Check if R can be a semicomplete bipartite digraph:
First verify if R can be a semicomplete multipartite digraph. If not, then R can not be semicomplete bipartite either. Suppose that we have found semicomplete r-partite R such that $D = R[H_1, ..., H_r]$. If $r = 2$ we have the desired R. If $r > 2$, then still there is a possibility for another R which is semicomplete bipartite (and will be denoted by R'), but it is easy to see the last possibility means that the semicomplete bipartite digraph R' must be a semicomplete digraph of order two with either one or two arcs.

So, first we verify if there is semicomplete R such that $D = R[H_1, ..., H_r]$. Starting with the collection $V_1, \dots, V_c$, obtained as in the previous verification, identify two of the sets V_i and V_j if none of the following occurs $V_i \Rightarrow V_j$, $V_j \Rightarrow V_i$ or $V_i \Leftrightarrow V_j$. Let $H_1, \dots, H_r$, $r \geq 1$ denote the sets obtained in this process. If $r = 1$, then R cannot be semicomplete. Otherwise, let R be the semicomplete digraph obtained by contracting each H_i into a vertex. Suppose we find a semicomplete R with more than two vertices.

If R is not strong then, obviously one can find a decomposition $R = R'[M_1, M_2]$, where R' is the semicomplete digraph with two vertices and one arc. If R is strong, then we try to find a decomposition $R = R'[M_1, M_2]$, where R' is the semicomplete digraph with two vertices which induce a 2-cycle. To check the last possibility, construct the following graph G. The vertex set of G coincides with $V(R)$, and two vertices are adjacent in G if and only if they do not form a cycle of length two in R. If G is connected then obviously the last possibility cannot take place. On the other hand, if G is not connected, then we can easily find R' which is semicomplete and has two vertices and two arcs.

Check if R can be a connected extended LSD:
Find components of the graph $G(D)$ defined in Proposition 6.1. Let $V_1, ..., V_c$ be the vertex sets of these components. If $c = 1$, then R cannot be extended LSD. So suppose $c \geq 2$ bellow. Let $D_j = D < V_j >$, let $D_{j1}, ..., D_{jn_j}$ be connected components of D_j and $V_{jk} = V(D_{jk})$. Now we proceed as in the second paragraph of the semicomplete multipartite digraph case.

It is not difficult to see that, for all considered kinds of R, the procedures above can be realized as an $O(nm + n^2)$-algorithm. □

Theorem 7.3 *There exists an $O(n^2m + n^3)$-algorithm for checking if a digraph with n vertices and m arcs is totally Φ_i-decomposable for $i = 0, 1, 2$.*

Proof: We give a description of a recursive algorithm to check Φ_i-decomposability. We have shown in Lemma 7.2 how to verify if $D = R[H_1, \dots, H_r]$, $r \geq 2$, where H is acyclic, semicomplete multipartite, semicomplete bipartite or connected extended locally semicomplete. Whenever we find an R that could be used, the algorithm checks total Φ_i-decomposability of $H_1, \dots, H_h$ in recursive calls.

Notice how the algorithm exploits the fact that total Φ_i-decomposability is a hereditary property (see Lemma 7.1): if some R seems to work, then it is safe to proceed in that direction, because if D is totally Φ_i-decomposable, then each of $H_1, \ldots, H_r$ (being an induced subgraph of D) must also be totally Φ_i-decomposable. Since there are $O(n)$ recursive calls, the complexity of the algorithm is $O(n^2m + n^3)$. □

8 Paths and cycles in totally Φ_i-decomposable digraphs

In this section we generalize theorems on heaviest and longest paths and cycles in quasi-transitive digraphs obtained in [3]. To obtain the generalizations given in Theorem 8.2 we use both results shown in the previous sections of this paper and an approach suggested in [3]. Since the schemes of the proofs of the generalizations are the same as those of the original theorems on quasi-transitive digraphs, we shall give only a sketch for the first two claims of Theorem 8.2 and a few remarks on the rest of the claims.

From now on, assume that every digraph D we consider has non-negative weights $w(.)$ on the vertices. The weight $w(H)$ of a subgraph of D is the sum of the weights of its vertices. For a positive integer k, the symbol $w_k(D)$ denotes the weight of a heaviest k-path subgraph of D, i.e. one with the maximum weight among k-path subgraphs. For convenience we define $w_0(D) = 0$. We consider the following problem called *the HPS problem*: Given a digraph D on n vertices, find a heaviest k-path subgraph of D for every $k = 1, 2, ..., n$.

We need the following:

Theorem 8.1 *[3] Let Φ be a set of digraphs including the trivial digraph on one vertex. Suppose that $\Phi = \Phi^{ext}$ and, for every $D \in \Phi$ on n vertices,*

$$w_{k+1}(D) - w_k(D) \leq w_k(D) - w_{k-1}(D), \qquad (1)$$

where $k = 1, 2, ..., n-1$. If there is a constant $s \geq 2$ so that, for every $L \in \Phi$, the HPS problem can be solved in time $O(|V(L)|^s)$, then, for every totally Φ-decomposable digraph D, the HPS problem can be solved in time $O(|V(D)|^{s+1})$, provided we are given a total Φ-decomposition of D.

Theorem 8.2 *Let D be a digraph of order n with nonnegative integer weights on the vertices. One can check whether D is totally Φ_i-decomposable for $i = 0, 1, 2$ in time $O(n^4)$. Moreover,*

(1) If D is totally Φ_0-decomposable, then for all $k = 1, ..., n$, some maximum weight k-path subgraphs of D can be found in time $O(n^5)$;

(2) If D is totally Φ_0-decomposable and $X \subset V(D)$, then we can check if D has a path covering all the vertices of X and find one (if it exists) in time $O(n^5)$;

(3) If D is totally Φ_2-decomposable, then a maximum weight cycle of D can be found in time $O(n^5)$;

(4) If D is totally Φ_2-decomposable and $X \subset V(D)$, then a cycle of D containing all vertices of X can be found in time $O(n^5)$ (if it exists);

(5) If D is totally Φ_1-decomposable, then a longest cycle of D can be found in time $O(n^5)$.

Proof (sketch):

1. By Theorems 8.1 and 7.3, we can prove the first part of Theorem 8.2 by showing that a digraph D from Φ_0 satisfies the conditions of Theorem 8.1 with $s = 4$.

Using the algorithm on maximum cost flows in networks described in [3], find, for every $k = 1, ..., n$, a heaviest k-path-cycle subgraph L_k of D. This can be done in time $O(n^4)$ [3]. Since $D \in \Phi_0$, for every $k = 1, ..., n$, we can construct a k-path subgraph Q_k of D so that $V(L_k) = V(Q_k)$. Indeed, if D is acyclic, then just set $Q_k = L_k$. If D is an extended LSD, then we find Q_k using Theorem 5.3. If D is semicomplete multipartite, then we use an analog of Theorem 5.2 for semicomplete multipartite digraphs (every non-trivial induced subgraph of a semicomplete multipartite digraph is connected). Obviously, Q_k is a heaviest k-path subgraph of D. Note that $Q_1, ..., Q_n$ can be found in time $O(n^4)$. The inequality (1) now follows from the inequality (2) on flows given in [3].

2. Change the weights of the vertices of D as follows. $w(x) = 1$ if $x \in X$ and $w(x) = 0$, otherwise. Obviously, D has a path covering all the vertices of X if and only if a heaviest path of D has weight $|X|$.

3. The proof of this claim is the same as the proof of the second part of Theorem 3.1 in [3] except for the fact that in [3] we use the second part of Theorem 3.6 [3] and here we use a generalization of the last result given in Theorem 4.5 and Corollary 4.6 of this paper.

4. The proof is similar to that of the second claim of the theorem.

5. The proof of this claim differs from the proof of Claim 3 only in the case when we consider semicomplete bipartite digraphs. For the last family of graphs, the property given in Theorem 4.5 is not true, in general. Hence, we use its weakening [11]: Let D be a strong semicomplete bipartite digraph and let $C_1, ..., C_t$ be a maximum order cycle subgraph of D. Then D has a (longest) cycle C so that $|V(C)| = |V(C_1)| + ... + |V(C_t)|$ and C can be constructed in time $O(n^2)$ given the subgraph $C_1, ..., C_t$. □

References

[1] J. Bang-Jensen, Locally semicomplete digraphs: a generalization of tournaments, *J.Graph Theory* 14 (1990) 371-390.

[2] J.Bang-Jensen, On the structure of locally semicomplete digraphs, *Discrete Math.* 100 (1992) 243-265.

[3] J. Bang-Jensen and G. Gutin, Vertex heaviest paths and cycles in quasi-transitive digraphs, *Discrete Math.*, to appear.

[4] J. Bang-Jensen, G. Gutin and J. Huang, Weakly Hamiltonian-connected ordinary multipartite tournaments, *Discrete Math.* **138** (1995) 63-74.

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

[6] L.W. Beineke and R.J. Wilson, **Selected Topics in Graph Theory 3**, Academic Press, 1988.

[7] J. A. Bondy and U. S. R. Murty, **Graph Theory with applications**, MacMillan Press (1976).

[8] Y. Guo and L. Volkmann, Connectivity properties of locally semicomplete digraphs, *J. Graph Theory*, 18 (1994) 269-280.

[9] G. Gutin, A criterion for complete bipartite digraphs to be Hamiltonian. *Vestsi Acad. Navuk BSSR Ser. Fiz.-Mat. Navuk* No. 1 (1984) 99-100 (In Russian).

[10] G. Gutin, Finding a longest path in a complete multipartite digraph, *SIAM J. Disc. Math.* 6 (1993) 270-273.

[11] G. Gutin, Paths and cycles in directed graphs, *Ph.D Thesis*, Tel-Aviv University, 1993.

[12] G. Gutin, Characterizations of vertex pancyclic and pancyclic ordinary complete multipartite digraphs, *Discrete Math.* **141** (1995) 153-162.

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

[14] R. Häggkvist, Hamilton cycles in oriented graphs, *Combinatorics, Probability and Computing* **2** (1993) 25-32.

[15] R. Häggkvist and Y. Manoussakis, Cycles and paths in bipartite tournaments with spanning configuration, *Combinatorica* **9** (1989) 33-38.

[16] J. Huang, Tournament-like oriented graphs, *Ph.D thesis*, Simon Fraser University, 1992.