FINDING A LONGEST PATH IN A COMPLETE MULTIPARTITE DIGRAPH

G. Gutin

Raymond and Beverly Sackler
Faculty of Exact Sciences
School of Mathematical Sciences
Tel Aviv University
Ramat-Aviv 69978, Israel

Abstract. A digraph obtained by replacing each edge of a complete m-partite graph with an arc or a pair of mutually opposite arcs with the same end vertices is called a complete m-partite digraph. We describe an $O(n^3)$ algorithm for finding a longest path in a complete m-partite ($m \geq 2$) digraph with n vertices. The algorithm requires time $O(n^{2.5})$ in case of testing only the existence of a Hamiltonian path and finding it if one exists. It is simpler than the algorithm of Manoussakis and Tuza [4], which works only for $m = 2$. Our algorithm implies a simple characterization of complete m-partite digraphs having Hamiltonian paths which was obtained for the first time in [1] (for $m = 2$) and in [2] (for $m \geq 2$).

Key words. Digraph, longest path, polynomial algorithm

AMS (MOS) subject classifications. 05C38, 05C45, 68R10

1. Introduction and Terminology

In this note we consider only digraphs without loops, unless otherwise specified. A digraph D on m disjoint vertex classes is called a complete m-partite (multipartite) digraph (abbreviated to CMD) if for any two vertices u, v in different classes either (u, v) or (v, u) (or both) is an arc of D.

In [1] a characterization of complete bipartite digraphs (abbreviated to CBM) containing Hamiltonian paths was given. This characterization was generalized to CMD in [2]. Using another approach, Häggkvist and Manoussakis gave in [3] analogous characterization of CBD having a Hamiltonian path. The results in [1], [2] supply an $O(n^{2.5})$ algorithm for checking if a given CMD with n vertices has a Hamiltonian path.

Manoussakis and Tuza obtained in [4] an $O(n^{2.5})$ algorithm for finding a Hamiltonian path in a complete oriented bipartite graph B (if B has a Hamiltonian path). In this work we describe an $O(n^3)$ algorithm for finding a longest path in a CMD. This algorithm requires time $O(n^{2.5})$ in the case of testing only the existence of a Hamiltonian path and finding it, if one exists. It is simpler than the algorithm of Manoussakis and Tuza [4] (in case $m = 2$ particularly, see section 3), and does not require an algorithm for finding a Hamiltonian cycle (as in [4]). Our algorithm implies a simple characterization of CMD, having Hamiltonian paths [2].

$V(D)$, $A(D)$ are the sets of vertices and arcs of a digraph D. A digraph D is called 1-diregular if $d^+(x) = d^-(x) = 1$ for any $x \in V(D)$. A digraph D is called almost 1-diregular if there exists vertices x, y (possibly, $x = y$) such that $d^+(x) = d^-(y) = 0$, and $d^+(z) = 1$ for $z \in V(D)\backslash x$, $d^-(v) = 1$ for $v \in V(D)\backslash y$. It is easy to see that a 1-diregular digraph F represents a collection of vertex disjoint cycles $C_1, C_2, \dots, C_t$ $(t \geq 1)$, i.e. $F = C_1 \cup C_2 \cup \cdots \cup C_t$. Similarly, an almost 1-diregular digraph $S = C_0 \cup C_1 \cup C_2 \cup \cdots \cup C_q$, where C_0 is a path, (which may have only 1 vertex), $C_1, C_2, \dots, C_q$ are cycles, $V(C_i) \cap V(C_j) = \emptyset$ for $0 \leq i \neq j \leq q$, $q \geq 0$.

If $C = (x_1, x_2, \ldots, x_p, x_1)$ is a cycle and $P = (y_1, y_2, \ldots, y_q)$ is a path, then

$$PT(y_i, y_j, P) \quad \text{is the path} \quad (y_i, y_i + 1, \ldots, y_j) \quad (i \leq j) \ ,$$

$$PT(x_i, C) = (x_i, x_{i+1}, \ldots, x_p, x_1, \ldots, x_{i-1}) \ ,$$

$$PT(C, x_i) = PT(x_{i+1}, C), PT(x_i, x_j, C) = PT(x_i, x_j, PT(x_i, C)) \ .$$

Let D be a digraph, and let x be a vertex of D, then

$$\Gamma^+(x) = \{y \in V(D) : (x, y) \in A(D)\} \ , \quad \Gamma^-(x) = \{z \in V(D) : (z, x) \in A(D)\} \ .$$

A "digraph" containing loops is called a general digraph.

2. Main Results

At first, we consider a construction (due to N. Alon) which allows one to find a 1-diregular subgraph with maximum order of a given digraph D. We add to D a loop in each vertex, associated with any loop a weight equals 2, and with any other arc of D a weight equals 1. We obtain a weighted general digraph L. Let $B = B(D)$ be a complete bipartite undirected graph, such that (X, X') is the partition of B, where $X = V(L)$, $X = \{x : x \in X\}$, $xy' \in E(B)$, if and only if $(x, y) \in A(L)$ (including the case $x = y$) and the weight of an edge xy' of B equals the weight of the arc (x, y). Obviously, a minimum weight 1-factor of B corresponds to a minimum weight 1-diregular spanning general subdigraph Q of L (i.e. a union of disjoint cycles and loops covering $V(L)$). It is easy to see that removing all loops from Q we obtain some 1-diregular subgraph F of D of maximum order. Since Q can be found by solving an assignment problem we may find a 1-diregular subgraph of D of maximum order in time $O(n^3)$, (cf. [5]) where, here and below, $n = |V(D)|$. Now we are ready to consider the main algorithm.

Algorithm ALP.

Input. A complete multipartite digraph D.

Output. A longest path H of D.

Step 1. Construct the digraph D' with

$$V(D') = \{x\} \cup V(D) \quad (x \notin V(D)) \ ,$$

$$A(D') = A(D) \cup \{(x, y),\ (y, x) : y \in V(D)\} \ .$$

Find a 1-diregular subgraph F' of D' of maximum order. Let $C_0, C_1, \ldots, C_t (t \geq 0)$ be the cycles of F', and suppose $x \in V(C_0)$. (It is easy to see that $x \in F'$). Find $P = C_0 - x$, and put

$$F := P \cup C_1 \cup \cdots \cup C_t \ .$$

Note that F is almost a 1-diregular subgraph of D of maximum order. We will construct a path on all the vertices of F – this will clearly be a longest path.

Step 2. If $t = 0$, then $H := P$, and we have finished. Otherwise put $C := C_t,\ t := t-1$. Let

$$P = (x_1, x_2, \ldots, x_m),\ C = (y_1, y_2, \ldots, y_k, y_1) \ .$$

Step 3. If $\Gamma^-(x_1) \cap V(C) \neq \emptyset$, then pick any $x \in \Gamma^-(x_1) \cap V(C)$, put $P := (PT(C, x), P)$, and go back to Step 2. Analogously, if there exists $y \in \Gamma^+(x_m) \cap V(C)$ put $P := (P,\ PT(y, C))$, and go back to Step 2.

Step 4. For $i = 1, 2, \ldots, m-1;\ j = 1, 2, \ldots, k$ if $(y_j,\ x_{i+1}),\ (x_i,\ y_{j+1})\ \in A(D)$, then

$$P := (PT(x_1, x_i, P),\ PT(y_{j+1},\ C),\ PT(x_{i+1}, x_m, P)) \ ,$$

and go to Step 2.

If none of Steps 2,3,4 can be applied, we go to Step 5 below.

Step 5. For $j = 1, 2, \ldots, k;\ i = 1, 2, \ldots, m-1$ if i is minimal such that there exists $j = j(i)$ for which

$$(y_j,\ x_{i+1}),\ (y_{j+1}, x_i) \in A(D) \ , \tag{1}$$

then let P be a directed path containing

$$PT\,(x_1,\ x_{i-1}, P)\ ,\quad y_{j+1},\ x_i\ ,\quad PT\,(y_{j+2},\ y_j,\ C),\ PT\,(x_{i+1},\ x_m,\ P)\ , \tag{2}$$

and three additional arcs and go to Step 2. (We prove below that such a directed path indeed exists.)

Lemma 1. *Algorithm ALP finds a longest path in a CMD D in time $O(n^3)$.*

Proof: We claim that during the algorithm ALP P is always a path in D. It is obvious that this is the case after each execution of step 1,2,3 or 4 (provided this was the case before starting such a step). Hence we consider only Step 5. When algorithm ALP executes Step 5, none of the conditions of Steps 3,4 hold for the current P and C. Hence

$$\Gamma^-(x_1) \cap V(C) = \Gamma^+(x_m) \cap V(C) = \emptyset \ , \tag{3}$$

and there are no indices $i \in \{1, 2, \ldots, m-1\}$, $j \in \{1, 2, \ldots, k\}$ such that both $(y_j,\ x_{i+1})$ and $(x_i,\ y_{j+1})$ belong to D, i.e.

$$\{(y_j,\ x_{i+1})\ ,\ (x_i,\ y_{j+1})\} \not\subseteq A(D)\ . \tag{4}$$

We must prove that if algorithm ALP is at Step 5, then there exist arcs satisfying (1), and in this case there exists the path (2).

At first, assume that there are no arcs satisfying (1). By (3) $(y_s, x_m) \in A(D)$ for some s. Then x_{m-1} and y_{s+1} are non-adjacent. Indeed, by (4) $(x_{m-1},\ y_{s+1}) \notin A(D)$, and by the assumption $(y_{s+1},\ x_{m-1}) \notin A(D)$. Since y_{s+1} is not adjacent with x_{m-1} it is adjacent with x_m. Therefore, $(y_{s+1},\ x_m) \in A(D)$. Hence x_{m-1} and y_{s+2} are non-adjacent, and x_{m-1} is not adjacent with any of y_{s+1} and y_{s+2} . Since y_{s+1} and y_{s+2} are adjacent (and hence do not belong to the same part) this is a contradiction. We conclude that there exist arcs satisfying (1). Let i be the minimum possible index in (1) and put $j = j(i)$.

Now we prove that D has the path (2). By (3) $i > 1$. By the minimality of i and by (4) the vertices $x_{i-1},\ y_{j+2}$ are non-adjacent. If $(y_{j+2},\ x_i) \in A(D)$, then, again, by the minimality of i (and by (4)) the vertices $x_{i-1},\ y_{j+3}$ are non-adjacent but this is impossible. Hence

$$(x_i,\ y_{j+2}) \in A(D)\ . \tag{5}$$

If $i = 2$ we have (by (3)) that $(x_{i-1},\ y_{j+1}) \in A(D)$. If $i > 2$ and $(y_{j+1},\ x_{i-1}) \in A(D)$, it follows that $x_{i-2},\ y_{j+2}$ are non-adjacent; that as impossible because x_{i-1} and y_{j+2} are non-adjacent. Hence

$$(x_{i-1},\ y_{j+1}) \in A(D)\ , \tag{6}$$

in any case. Therefore, using the arcs from (5), (6), we may form path (2).

Note that the number of operations we need for executing Steps 3-5 is $O(|V(P)| \cdot |V(C)|)$ for the current pair P, C. Hence the total number of operations at Steps 2-5 is

$$O(|V(P)| \cdot |V(C_1)|) + \sum_{j=1}^{t-1} (|V(P)| + \cdots + |V(C_j)|)(|V(C_{j+1})|) = O(n^2) \ .$$

At last note that the execution of Step 1 takes time $O(n^3)$. ◻

Algorithm ALP and the proof of Lemma 1 imply immediately the following result.

Theorem. *Let D be a CMD. Then for any almost 1-diregular subgraph F of D there is a path P of D satisfying $V(P) = V(F)$. If F is a maximum 1-diregular subgraph each such path is a longest path of D. There exists an algorithm for finding a longest path in D in time $O(n^3)$.*

3. Modifications of the Main Results

Using any maximum matching algorithm (see [5], [6]), one can test whether a digraph contains a 1-diregular spanning subgraph F' and find some F' in time $O(n^{2.5})$. Note that $F = F' - x(x \in V(F))$ is an almost 1-diregular spanning subgraph. Hence after a trivial modification of Step 1 in algorithm ALP we obtain an $O(n^{2.5})$ algorithm allowing to test whether a CMD D has a Hamiltonian path (and to construct one of them in case it exists). This implies

Corollary 1. *A CMD D has a Hamiltonian path, if and only if it has an almost 1-diregular spanning subgraph. Testing whether D has a Hamiltonian path (and finding one of them) requires at most time $O(n^{2.5})$.*

Let D be a CBD. Then we can remove Step 5 from the algorithm, since the algorithm does not use Step 5 in this case. To prove this we must show that the algorithm never goes to Step 5 (from Step 4), i.e. it always constructs a new path P in Step 3 or Step 4. If the algorithm reaches Step 4 after executing Step 3 for the current P and C, then (3) holds. Therefore, there exists $i \in \{1, 2, \ldots, m-1\}$ such that $\Gamma^-(x_i) \cap V(C) = \emptyset$, but $\Gamma^-(x_{i+1}) \cap V(C) \neq \emptyset$; $(y_j,\ x_{i+1}) \in A(D)$. Since D is bipartite the vertices $x_i,\ y_{j+1}$ are adjacent. Hence $(x_i,\ y_{j+1}) \in A(D)$, and the algorithm can construct a new path P.

4. Acknowledgment

This paper forms part of a Ph.D. thesis written by the author under the supervision of Professor N. Alon.

I would like to thank Professor Alon for fruitful comments and suggestions.

References

1. Gutin G., Effective characterization of complete bipartite digraphs that have a Hamiltonian path. Kibernetica (Kiev), 1985, No4, P.124-125. (In Russian).
2. Gutin G., A characterization of complete n-partite digraphs that have a Hamiltonian path. Kibernetica (Kiev). 1988, No1, P.107-108. (In Russian).
3. Häggkvist R., Manoussakis Y., Cycles and paths in bipartite tournaments with spanning configurations. Combinatorica. 1989, Vol.9, N1, P.33-38.
4. Manoussakis Y., Tura Z., Polynomial algorithms for finding cycles and paths in bipartite tournaments. SIAM J. Disc. Math. 1990, Vol.3, N4, pp.537-543.
5. Papadimitriou H., Steiglitz K., Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall Inc., New Jersey, 1982.
6. Hopcroft J., Karp R., A $n^{5/2}$ algorithm for maximum matching in bipartite graphs. J. SIAM Comp. 1973, Vol.2, pp.225-231.