

# An $O(n)$ Time Algorithm for Scheduling UET-UCT of Bipartite Digraphs of Depth One on Two Processors

Ruzayn Quaddoura

Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa, Jordan  
Email: ruzayn@zu.edu.jo

Received 17 November 2015; accepted 24 January 2016; published 28 January 2016

Copyright © 2016 by author and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY).  
<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Given  $n$  unit execution time (UET) tasks whose precedence constraints form a directed acyclic graph, the arcs are associated with unit communication time (UCT) delays. The problem is to schedule the tasks on two identical processors in order to minimize the makespan. Several polynomial algorithms in the literature are proposed for special classes of digraphs, but the complexity of solving this problem in general case is still a challenging open question. We present in this paper an  $O(n)$  time algorithm to compute an optimal schedule for the class of bipartite digraphs of depth one.

## Keywords

Scheduling, Makespan, Precedence Constraints, Bipartite Graph, Optimal Algorithm

---

## 1. Introduction

The problem of scheduling a set of tasks on a set of identical processors under a precedence relation has been studied for a long time. A general description of the problem is the following. There are  $n$  tasks that have to be executed by  $m$  identical processors subject to precedence constraints and (may be without) communication delays. The objective is to schedule all the tasks on the processors such that the makespan is the minimum. Generally, this problem can be represented by a directed acyclic graph  $G=(V, E)$  called a task graph. The set of vertices  $V$  corresponds to the set of tasks and the set of edges  $E$  corresponds to the set of precedence constraints. With every vertex  $i$ , a weight  $p_i$  is associated that represents the execution time of the task  $i$ , and with every edge  $(i, j)$ , a weight  $c_{ij}$  is associated that represents the communication time between the tasks  $i$  and  $j$ . If  $(i, j) \in E$  and the task  $i$  starts its execution at time  $t$  on a processor  $P$ , then either  $j$  starts its execution on  $P$  at time greater than or equal to  $t + p_j$ , or  $j$  starts its execution on some other processor at time greater than or equal to  $t + p_j + c_{ij}$ .

**How to cite this paper:** Quaddoura, R. (2016) An  $O(n)$  Time Algorithm for Scheduling UET-UCT of Bipartite Digraphs of Depth One on Two Processors. *American Journal of Operations Research*, 6, 75-80.  
<http://dx.doi.org/10.4236/ajor.2016.61010>

According to the three field notation scheme introduced in [1] and extended in [2] for scheduling problems with communication delays, this problem is denoted as  $P_m | prec, p_i, c_{ij} | C_{\max}$ .

A large amount of work in the literature studies this problem with a restriction on its structure: the time of execution of every task is one unit execution time (UET), the number of processors  $m$  is fixed, the communication delays are neglected, constant or one unit (UCT), or special classes of task graph are considered. We find In this context, the problem  $P_2 | prec, p_i = 1 | C_{\max}$ , is polynomial [3] [4], *i.e.* when the communication delays are not taken into account. On the contrary, the problem  $P_3 | prec, p_i = 1 | C_{\max}$  remains an open question [5].

The problem of two processors scheduling with communication delays is extensively studied [6] [7]. In particular, it is proven in [8] that the problem  $P_2 | prec = \text{binary tree}, p_i = 1, c_{ij} = c | C_{\max}$  is NP-hard where  $c$  is a large integer, whereas this problem is polynomial when the task graph is a complete binary tree.

A challenging open problem is the two processors scheduling with UET-UCT, *i.e.* the problem  $P_2 | prec, p_i = 1, c_{ij} = 1 | C_{\max}$  for which the complexity is unknown. However, several polynomial algorithms have been shown for special classes of task graphs, especially for trees [9] [10], interval orders [11] and a subclass of series parallel digraphs [12]. In this paper we present an  $O(n)$  time algorithm to compute an optimal algorithm for the class of bipartite digraphs of depth one, that is the digraphs for which every vertex is either a source (without predecessors) or a sink (without successors).

## 2. Scheduling UET-UCT for a Bipartite Digraph of Depth One on Two Processors

### 2.1. Preliminaries

A schedule UET-UCT on two processors for a general directed acyclic digraph  $G = (V, E)$  is defined by a function  $\sigma: V \rightarrow \mathbb{N}^+ \times \{P_1, P_2\}$ ,  $\sigma(v) = (t_v, P_i)$ ,  $i = 1, 2$  where  $t_v$  is the time for which the task  $v$  is executed and  $P_i$  the processor on which the task  $v$  is scheduled. A schedule  $\sigma$  is feasible if:

- a)  $\forall u, v \in V$ , if  $u \neq v$  then  $\sigma(u) \neq \sigma(v)$
- b) If  $(u, v) \in E$  then  $t_u + 1 \leq t_v$ , if  $u$  and  $v$  are scheduled on the same processor, and  $t_u + 2 \leq t_v$  if  $u$  and  $v$  are scheduled on distinct processors.

A time  $t$  of a schedule  $\sigma$  is said to be idle if one of the processors is idle during this time. The makespan  $C_{\max}$  or the length of a schedule  $\sigma$  is the last non-idle time of  $\sigma$ , that is:

$$C_{\max} = \max \{t : \exists v \in V \sigma(v) = (t, P_i), i = 1 \text{ or } 2\}$$

A schedule  $\sigma$  is optimal if  $C_{\max}$  is the minimum among all feasible schedules.

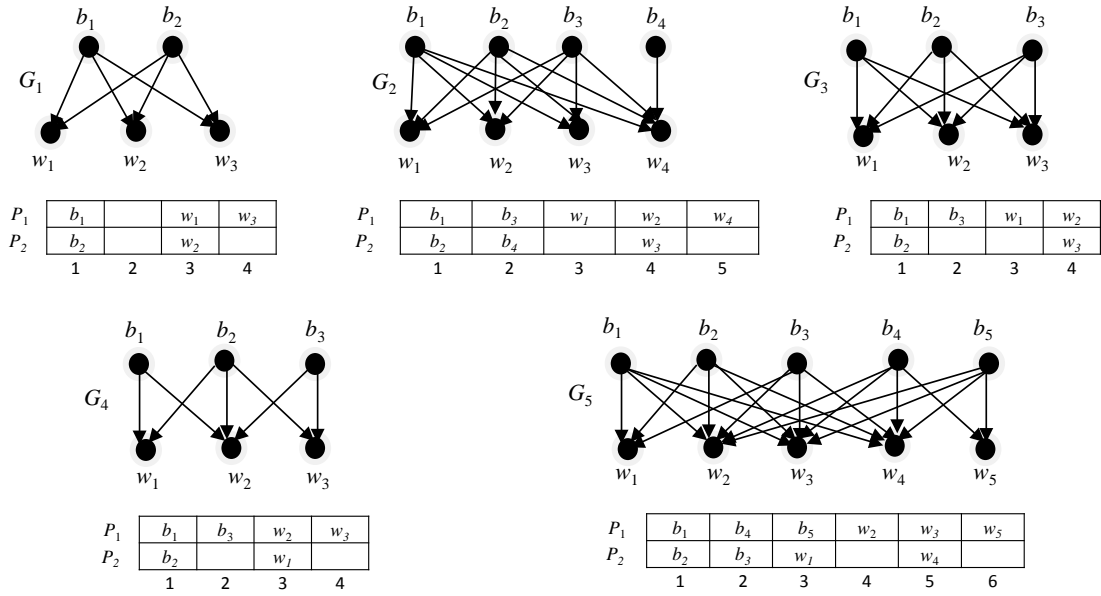
Let  $G = (B \cup W, E)$  be a bipartite digraph of depth one. Since every vertex of  $G$  is either a source or a sink, there exists always a feasible schedule such that the sources  $B$  are executed before executing the sinks  $W$ . Our algorithm for solving the problem under consideration produces an optimal schedule satisfies this condition and that we called a natural schedule defined as follows.

**Definition 1** Let  $G = (B \cup W, E)$  be a bipartite digraph of depth one. A natural schedule of  $G$  is obtained by scheduling first the sources  $B$  then the sinks  $W$  starting from the processor  $P_1$  and alternating between  $P_1$  and  $P_2$  such that the resulting schedule is optimal.

The definition of a natural schedule  $\sigma$  of a bipartite digraph of depth one  $G = (B \cup W, E)$  implies the following properties:

- 1) The number of sources executed on  $P_1$  is  $\lceil |B|/2 \rceil$  and the number of sources executed on  $P_2$  is  $\lfloor |B|/2 \rfloor$ .
- 2) If  $|B|$  is even then:
  - a)  $\sigma$  contains at most 2 idle times, the first is at time  $|B|/2 + 1$ , and the second is at time  $C_{\max}$ .
  - b) If  $|B|/2 + 1$  is an idle time then  $P_2$  is idle at this time (may be  $P_1$  also).
- 3) If  $|B|$  is odd then:
  - a)  $\sigma$  contains at most 3 idle times, the first is at time  $\lceil |B|/2 \rceil$ , the second is at time  $\lceil |B|/2 \rceil + 1$ , and the third is at time  $C_{\max}$ .
  - b) If  $\lceil |B|/2 \rceil$  or  $\lceil |B|/2 \rceil + 1$  is an idle time then  $P_2$  is idle at this time and  $P_1$  is non.
- 4)  $\lceil |B \cup W|/2 \rceil \leq C_{\max} \leq \lceil |B \cup W|/2 \rceil + 1$ .

Without loss of generality, we can suppose that both idle times  $C_{\max}$  and  $\lceil |B \cup W|/2 \rceil + 1$ , if exist, are distinct. In this supposition,  $C_{\max} = \lceil |B \cup W|/2 \rceil$  if and only if  $\sigma$  has at most the idle time  $C_{\max}$  otherwise.



**Figure 1.** Bipartite digraphs of depth one and their corresponding natural schedules.

$C_{\max} = \lceil |B \cup W|/2 \rceil + 1$ . **Figure 1** illustrates some bipartite digraphs of depth one and their corresponding natural schedules.

## 2.2. Scheduling Algorithm

The idea of solving the problem  $P_2 \mid prec = \text{bipartite of depth one}, p_j = 1, c_{ij} = 1 \mid C_{\max}$  is to determine the necessary and sufficient conditions to exist idle times in a natural schedule of the task graph. In the following, we consider  $G = (B \cup W, E)$  is a bipartite digraph of depth one where  $B$  is the set of sources and  $W$  is the set of sinks, and  $\sigma$  is a natural schedule for  $G$ . A vertex  $b \in B$  ( $w \in W$ ) is called universal if  $d(b) = |W|$  ( $d(w) = |B|$ ). We distinguish two cases,  $|B|$  is even and  $|B|$  is odd.

**Lemma 2** Assume that  $|B|$  is even.

- 1) The two processors  $P_1$  and  $P_2$  are idle at time  $|B|/2 + 1$  if and only if  $G$  is a bipartite complete.
- 2) The processor  $P_2$  only is idle at time  $|B|/2 + 1$  if and only if one of the following holds:
  - a) Every vertex of  $B$  is universal except exactly one.
  - b) Every vertex of  $W$  is universal except exactly one.

**Proof.** 1) If  $G$  is a bipartite complete then obviously  $P_1$  and  $P_2$  must be idle at time  $|B|/2 + 1$ . The inverse, let  $b$  and  $w$  be a source and a sink of  $G$ . If  $b$  is not adjacent to  $w$ , then we can schedule  $b$  on  $P_2$  at time  $|B|/2$  and  $w$  on  $P_1$  at time  $|B|/2 + 1$ , a contradiction. So  $b$  must be adjacent to  $w$ , therefore  $G$  is a bipartite complete.

2) Assume that  $P_2$  only is idle at time  $|B|/2 + 1$  and the conditions a and b are not hold. Then, there exist  $b_1, b_2 \in B$  and  $w_1, w_2 \in W$  such that  $b_1, b_2, w_1$  and  $w_2$  are all not universal. If  $\{b_1, b_2, w_1, w_2\}$  is a stable set, i.e. no two vertices are adjacent, we can schedule  $b_1, b_2$  on  $P_1, P_2$  at time  $|B|/2$  and  $w_1, w_2$  on  $P_1, P_2$  at time  $|B|/2 + 1$ , a contradiction. If  $\{b_1, b_2, w_1, w_2\}$  is not a stable set then we can suppose that  $b_1 w_1 \in E$ . Since  $b_1$  and  $w_1$  are not universal, there exist  $w \in W$  and  $b \in B$  such that  $b_1 w \notin E$  and  $b w_1 \notin E$ . But now, we can schedule  $b_1, b$  on  $P_1, P_2$  respectively at time  $|B|/2$ , and  $w_1, w$  on  $P_1, P_2$  respectively at time  $|B|/2 + 1$ , a contradiction.

The inverse, suppose that every vertex of  $B$  is universal except exactly one. Since  $|B|$  is even, there exist  $b_1, b_2 \in B$  scheduled on  $P_1, P_2$  at time  $|B|/2$ . By 1, the two processors can't be idle at time  $|B|/2 + 1$ . If  $P_2$  is not idle at time  $|B|/2 + 1$ , then both  $b_1$  and  $b_2$  are not universal, a contradiction. In a similar way we prove the case b.  $\square$

Notice that if  $B$  (or  $W$ ) contains exactly one non-universal vertex  $b$  then the vertex of  $W$  which is independent of  $b$  is also non-universal but it is not necessary unique (see **Figure 1**). Algorithm `Schedule_` $|B|$ `_is_even` ( $G$ ) constructs a natural schedule for  $G = (B \cup W, E)$  if  $|B|$  is even, Lemma 2 proves its correctness.

**Algorithm Schedule\_ $|B|$ \_is\_even ( $G$ )**

$$B_1 = \{b \in B : d(b) = |W|\}$$

$$B_2 = \{b \in B : d(b) < |W|\}$$

$$W_1 = \{w \in W : d(w) = |B|\}$$

$$W_2 = \{w \in W : d(w) < |B|\}$$

If  $|B_1| = |B|$  and  $|W_1| = W$  then  
 Schedule  $B_1$  alternately on  $P_1$  and  $P_2$  at times  $1, 2, \dots, |B|/2$   
 Schedule  $W_1$  alternately on  $P_1$  and  $P_2$  at times  $|B|/2 + 1, \dots, \lceil |B \cup W|/2 \rceil + 1$   
 Else if  $|B_2| = 1$  or  $|W_2| = 1$  then  
 Let  $b \in B_2$  and  $w \in W_2$  such that  $bw \notin E$   
 Schedule  $b$  on  $P_2$  at time  $|B|/2$  and  $w$  on  $P_1$  at time  $|B|/2 + 1$   
 Schedule  $B - \{b\}$  alternately on  $P_1$  and  $P_2$  at times  $1, 2, \dots, |B|/2$   
 Schedule  $W - \{w\}$  alternately on  $P_1$  and  $P_2$  at times  $|B|/2 + 1, \dots, \lceil |B \cup W|/2 \rceil + 1$   
 Else let  $b_1, b_2 \in B_2$  and  $w_1, w_2 \in W_2$  such that  $b_1 w_1 \notin E, b_2 w_2 \notin E$   
 Schedule  $b_1$  on  $P_1$  and  $b_2$  on  $P_2$  at time  $|B|/2$   
 Schedule  $w_2$  on  $P_1$  and  $w_1$  on  $P_2$  at time  $|B|/2 + 1$   
 Schedule  $B - \{b_1, b_2\}$  alternately on  $P_1$  and  $P_2$  at times  $1, 2, \dots, |B|/2 - 1$   
 Schedule  $W - \{w_1, w_2\}$  alternately on  $P_1$  and  $P_2$  at times  $|B|/2 + 2, \dots, \lceil |B \cup W|/2 \rceil$

**Figure 1** shows the construction of natural schedules of the graphs  $G_1$  and  $G_2$  resulting from the algorithm Schedule\_ $|B|$ \_is\_even ( $G$ ).

**Lemma 3** Assume that  $|B|$  is odd.

- 1) The processor  $P_2$  is idle at times  $\lceil |B|/2 \rceil$  and  $\lceil |B|/2 \rceil + 1$  if and only if  $G$  is a bipartite complete.
- 2) The processor  $P_2$  is idle at time  $\lceil |B|/2 \rceil$  and not idle at time  $\lceil |B|/2 \rceil + 1$  if and only if
  - a) There is  $w \in W$  such that  $d(w) = |B| - 1$
  - b) For every  $w \in W$ ,  $d(w) = |B|$  or  $|B| - 1$ .
- 3) The processor  $P_2$  is idle at time  $\lceil |B|/2 \rceil + 1$  and not idle at time  $\lceil |B|/2 \rceil$  if and only if
  - a) There is  $w \in W$  such that  $d(w) \leq |B| - 2$
  - b) For every  $w \in W$  for which  $d(w) \leq |B| - 2$  and for every  $b \in \bar{N}(w)$ ,  $d(b) = |W| - 1$  where  $\bar{N}(w)$  is the set of non-neighbors of  $w$ .

**Proof.** 1) If  $G$  is a bipartite complete then obviously  $P_2$  is idle at times  $\lceil |B|/2 \rceil$  and  $\lceil |B|/2 \rceil + 1$ . The inverse, let  $b$  and  $w$  be a source and a sink of  $G$ . If  $b$  is not adjacent to  $w$ , then we can schedule  $b$  on  $P_1$  at time  $\lceil |B|/2 \rceil$  and  $w$  on  $P_2$  at time  $\lceil |B|/2 \rceil$  or at time  $\lceil |B|/2 \rceil + 1$  according to the adjacency relation between the source scheduled on  $P_1$  at time  $\lceil |B|/2 \rceil - 1$  and  $w$ , a contradiction. So  $b$  must be adjacent to  $w$ , therefore  $G$  is a bipartite complete.

2) Assume that  $P_2$  is idle at time  $\lceil |B|/2 \rceil$  and not idle at time  $\lceil |B|/2 \rceil + 1$ . The vertex  $w \in W$  scheduled on  $P_2$  at time  $\lceil |B|/2 \rceil + 1$  is of degree less than or equal to  $|B| - 1$ , since it must be independent of the vertex  $b \in B$  scheduled on  $P_1$  at time  $\lceil |B|/2 \rceil$ . If there is a vertex  $w \in W$  such that  $d(w) \leq |B| - 2$ , then we can schedule  $w$  on  $P_2$  at time  $\lceil |B|/2 \rceil$  and two vertices from  $B$  independent of  $w$  on  $P_1$  at times  $\lceil |B|/2 \rceil - 1$  and  $\lceil |B|/2 \rceil$ , a contradiction.

The inverse, by 1, the processor  $P_2$  is not idle at time  $\lceil |B|/2 \rceil$  or at time  $\lceil |B|/2 \rceil + 1$ . If  $P_2$  is not idle at time  $\lceil |B|/2 \rceil$  then the vertex scheduled at this time would be of degree less than or equal to  $|B| - 2$ , a contradiction.

3) Assume that  $P_2$  is idle at time  $\lceil |B|/2 \rceil + 1$  and not idle at time  $\lceil |B|/2 \rceil$ . The vertex  $w \in W$  scheduled on  $P_2$  at time  $\lceil |B|/2 \rceil$  is of degree less than or equal to  $|B| - 2$ , since it must be independent of the two vertices  $b_1, b_2 \in B$  scheduled on  $P_1$  at times  $\lceil |B|/2 \rceil - 1$  and  $\lceil |B|/2 \rceil$ . Let  $w \in W$  such that  $d(w) \leq |B| - 2$  and let  $b \in \bar{N}(w)$  such that  $d(b) \leq |W| - 2$ . Now, we can schedule  $w$  on  $P_2$  at time  $\lceil |B|/2 \rceil$ ,  $b$  and another vertex  $b'$  independent of  $w$  on  $P_1$  at time  $\lceil |B|/2 \rceil$  and  $\lceil |B|/2 \rceil - 1$  respectively, and schedule a vertex  $w'$  independent of  $b$  on  $P_2$  at time  $\lceil |B|/2 \rceil + 1$ , a contradiction. The inverse, by 1 and 2, the processor  $P_2$  can't be idle at time  $\lceil |B|/2 \rceil$  and any vertex scheduled on  $P_2$  at time  $\lceil |B|/2 \rceil$  must be of degree less than or equal to  $|B| - 2$ . The processor  $P_2$  is idle at time  $\lceil |B|/2 \rceil + 1$ , otherwise the vertex scheduled on  $P_1$  at time  $\lceil |B|/2 \rceil$  is of degree less than or equal to  $|W| - 2$ , a contradiction.  $\square$

To construct a natural schedule for  $G$  when  $|B|$  is odd, we need to the Procedure Two\_Vertices ( $G$ ). This procedure return 1 if the condition 3.b of Lemma 3 holds, and return two vertices  $w, b$  such that  $d(w) \leq |B| - 2$ ,  $b \in \bar{N}(w)$  and  $d(b) \leq |W| - 2$  if this condition is not hold.

**Procedure Two\_Vertices (G)**

$$W_3 = \{w \in W : d(w) \leq |B| - 2\} = \{w_1, \dots, w_{|W_3|}\}$$

For  $i=1$  to  $|W_3|$

$$\bar{N}(w_i) = \{b_1^i, b_2^i, \dots, b_k^i\}$$

For  $j=1$  to  $k$

If  $d(b_j^i) \leq |W| - 2$  then Return  $(w_i, b_j^i)$

Return 1.

Algorithm Schedule\_ $|B|$ \_is\_odd (G) constructs a natural schedule for  $G = (B \cup W, E)$  if  $|B|$  is odd. Lemma 3 proves its correctness.

**Algorithm Schedule\_ $|B|$ \_is\_odd (G)**

$$B_1 = \{b \in B : d(b) = |W|\}$$

$$B_2 = \{b \in B : d(b) < |W|\}$$

$$W_1 = \{w \in W : d(w) = |B|\}$$

$$W_2 = \{w \in W : d(w) = |B| - 1\}$$

$$W_3 = \{w \in W : d(w) \leq |B| - 2\}$$

If  $|B_1| = |B|$  and  $|W_1| = |W|$  then

Schedule a vertex  $b \in B$  on  $P_1$  at time  $\lceil |B|/2 \rceil$

Schedule a vertex  $w \in W$  on  $P_1$  at time  $\lceil |B|/2 \rceil + 1$

Schedule  $B - \{b\}$  alternately on  $P_1$  and  $P_2$  at times  $1, \dots, \lceil |B|/2 \rceil - 1$

Schedule  $W - \{w\}$  alternately on  $P_1$  and  $P_2$  at times  $\lceil |B|/2 \rceil + 2, \dots, \lceil |B \cup W|/2 \rceil + 1$

Else if  $|W_3| = 0$  then

Let  $w \in W_2$  and  $b \in B_2$  such that  $bw \notin E$

Schedule  $b$  on  $P_1$  at time  $\lceil |B|/2 \rceil$

Schedule  $w$  on  $P_2$  at time  $\lceil |B|/2 \rceil + 1$

Schedule  $B - \{b\}$  alternately on  $P_1$  and  $P_2$  at times  $1, \dots, \lceil |B|/2 \rceil - 1$

Schedule  $W - \{w\}$  alternately on  $P_1$  and  $P_2$  at times  $\lceil |B|/2 \rceil + 1, \dots, \lceil |B \cup W|/2 \rceil + 1$

Else If Two\_Vertices (G) = 1 then

Let  $w \in W_3$  and  $b_1, b_2 \in B_2$  such that  $b_1 w, b_2 w \notin E$

Schedule  $b_1, b_2$  on  $P_1$  at times  $\lceil |B|/2 \rceil - 1$  and  $\lceil |B|/2 \rceil$

Schedule  $w$  on  $P_2$  at time  $\lceil |B|/2 \rceil$

Schedule  $B - \{b_1, b_2\}$  alternately on  $P_1$  and  $P_2$  at times  $1, \dots, \lceil |B|/2 \rceil - 1$

Schedule a vertex  $w' \in W - \{w\}$  on  $P_1$  at time  $\lceil |B|/2 \rceil + 1$

Schedule  $W - \{w, w'\}$  alternately on  $P_1$  and  $P_2$  at times  $\lceil |B|/2 \rceil + 2, \dots, \lceil |B \cup W|/2 \rceil + 1$

Else let  $(w, b) = \text{Two\_Vertices}(G)$

Let  $b_1 \in B_2 - \{b\}$  and  $w_1 \in W_2 \cup W_3 - \{w\}$  such that  $b_1 w, b w_1 \notin E$

Schedule  $b_1, b$  on  $P_1$  at times  $\lceil |B|/2 \rceil - 1$  and  $\lceil |B|/2 \rceil$  respectively

Schedule  $w, w_1$  on  $P_2$  at times  $\lceil |B|/2 \rceil$  and  $\lceil |B|/2 \rceil + 1$  respectively

Schedule  $B - \{b_1, b\}$  alternately on  $P_1$  and  $P_2$  at times  $1, \dots, \lceil |B|/2 \rceil - 1$

Schedule  $W - \{w, w_1\}$  alternately on  $P_1$  and  $P_2$  at times  $\lceil |B|/2 \rceil + 1, \dots, \lceil |B \cup W|/2 \rceil$

**Figure 1** shows the construction of natural schedules of the graphs  $G_3, G_4$  and  $G_5$  resulting from the algorithm Schedule\_ $|B|$ \_is\_odd (G).

**2.3. Complexity**

We assume that  $G = (B \cup W, E)$  is represented by its adjacency lists, so the set of neighbors and the set of non neighbors of every vertex of  $G$  are known already. In this supposition, we can check easily that any step (except the step Two\_Vertices (G)) of the two algorithms Schedule\_ $|B|$ \_is\_even (G) and Schedule\_ $|B|$ \_is\_odd (G) can be executed either within a constant time or within an  $O(n)$  time where  $|B \cup W| = n$ . Let's prove that the Procedure Two\_Vertices (G) runs within  $O(|B|)$  time.

The worst case of this Procedure occurs when its result is 1. In this case, for any  $w_1, w_2 \in W_3$ ,  $\bar{N}(w_1) \cap \bar{N}(w_2) = \emptyset$ , otherwise, a vertex  $b$  independent of  $w_1$  and  $w_2$  would be of degree less than or equal to  $|W| - 2$ . So the number of comparisons of if statement in this procedure is equal to at most  $|\bar{N}(w_1)| + \dots + |\bar{N}(w_{|W_3|})| \leq |B|$ . Therefore, the procedure Two\_Vertices (G) runs within  $O(|B|)$  time and the total

time of our scheduling algorithm is  $O(n)$ .

### 3. Conclusion

We have presented an  $O(n)$  time algorithm for the optimal schedule of bipartite digraphs of depth one with UET-UCT on two processors. The complexity of this problem for general directed acyclic graphs is still an open question. We believe that our algorithm can be used to solve this problem in general as follow: Consider a topological sort of a directed acyclic graph  $G$ . The linear ordering defined by this topological sort decomposes  $G$  into consecutives bipartite digraphs of depth one. The schedule obtained by the concatenation of the schedules of these bipartite digraphs is a feasible schedule or may be modified to a feasible schedule of  $G$ . Now, if we can determine the necessary and sufficient conditions to exist idle times in this feasible schedule then we can determine the complexity of this problem. This is a useful guide and foundation for future research.

### Acknowledgements

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University/Jordan. The author is grateful to anonymous referee's suggestion and improvement of the presentation of this paper.

### References

- [1] Graham, R.L., Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G. (1979) Optimization and Approximation in Deterministic Scheduling: A Survey. *Annals of Discrete Mathematics*, **5**, 287-326. [http://dx.doi.org/10.1016/S0167-5060\(08\)70356-X](http://dx.doi.org/10.1016/S0167-5060(08)70356-X)
- [2] Veltman, B., Lageweg, B.J. and Lenstra, L.K. (1990) Multiprocessor Scheduling with Communication Delays. *Parallel Computing*, **16**, 173-182. [http://dx.doi.org/10.1016/0167-8191\(90\)90056-F](http://dx.doi.org/10.1016/0167-8191(90)90056-F)
- [3] Coffman Jr., E.G. and Graham, R.L. (1972) Optimal Scheduling for Two-Processor Systems. *Acta Informatica*, **1**, 200-213. <http://dx.doi.org/10.1007/BF00288685>
- [4] Fujii, M., Kasami, T. and Ninomiya, K. (1969) Optimal Sequencing of Two Equivalent Processors. *SIAM Journal on Applied Mathematics*, **17**, 784-789. <http://dx.doi.org/10.1137/0117070>
- [5] Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman.
- [6] Chrétienne, P. and Picouleau, C. (1995) Scheduling with Communication Delays: A Survey. In: *Scheduling Theory and Its Applications*, John Wiley & Sons.
- [7] Norman, M.G., Pelagatti, S. and Thanisch, P. (1995) On the Complexity of Scheduling with Communication Delay and Contention. *Parallel Processing Letters*, **5**, 331-341. <http://dx.doi.org/10.1142/S012962649500031X>
- [8] Afrati, F., Bampis, E., Finta, L. and Mili, I. (2005) Scheduling Trees with Large Communication Delays on Two Identical Processors. *Journal of Scheduling*, **8**, 179-190. <http://dx.doi.org/10.1007/s10951-005-6366-3>
- [9] Varvarigou, T., Roychowdhury, V.P., Kailath, T. and Lawler, E. (1996) Scheduling in and out Forests in the Presence of Communication Delays. *IEEE Transactions on Parallel and Distributed Systems*, **7**, 1065-1074. <http://dx.doi.org/10.1109/71.539738>
- [10] Veldhorst, M. (1993) A Linear Time Algorithm to Schedule Trees with Communication Delays Optimally on Two Machines. Technical Report COSOR 93-07, Department of Math, and Computer Science, Eindhoven University of Technology, Eindhoven.
- [11] Ali, H. and El-Rewini, H. (1993) The Time Complexity of Scheduling Interval Orders with Communication Is Polynomial. *Parallel Processing Letters*, **3**, 53-58. <http://dx.doi.org/10.1142/S0129626493000083>
- [12] Finta, L., Liu, Z., Mills, I. and Bampis, E. (1996) Scheduling UET-UCT Series Parallel Graphs on Two Processors. *Theoretical Computer Science*, **162**, 323-340. [http://dx.doi.org/10.1016/0304-3975\(96\)00035-7](http://dx.doi.org/10.1016/0304-3975(96)00035-7)