



Single machine parallel batch scheduling with unbounded capacity

Yuan Jinjiang

Department of mathematics, Zhengzhou University
Zhengzhou, Henan 450052

yuanjj@zzu.edu.cn



Introduction and ...
The existing ...
NP-hardness proof

[Home Page](#)

[Title Page](#)



Page 1 of 30

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

1 Introduction and Problem Formulation

- Let n jobs J_1, J_2, \dots, J_n and a single machine that can handle batch jobs at the same time be given, where a batch is a subset of jobs and distinct batches cannot have a job in common.
- We assume in this report that each batch can contain arbitrary many jobs, i.e., the capacity of batches is unbounded. There is also another model called bounded batch scheduling in which each batch contains at most c jobs. But in this report we only consider the unbounded version.
- Each job J_j has a processing time $p_j > 0$.
- The processing time of a batch B is defined by

$$p_B = \max\{p_j : J_j \in B\}.$$

That is, the jobs in a common batch are processed in a parallel form simultaneously.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 2 of 30

Go Back

Full Screen

Close

Quit



Introduction and ...
The existing ...
NP-hardness proof

- Hence, the studied scheduling model is called parallel batch scheduling. (If the processing time of a batch B is defined by $p_B = \sum_{J_j \in B} p_j$, then the corresponding model is called serial batch scheduling.)
- Each job J_j has a release date r_j . This means that job J_j cannot start being processing before the time moment r_j .
- The release date of a batch B is defined by

$$r_B = \max\{r_j : J_j \in B\}.$$

Clearly, batch B cannot be processed before r_B .

- There are precedence relations \prec between the jobs, i.e., $J_i \prec J_j$ means that J_i must be completed before J_j starts; furthermore, $J_i \prec J_j$ and $J_j \prec J_k$ imply $J_i \prec J_k$.

Home Page

Title Page



Page 3 of 30

Go Back

Full Screen

Close

Quit



- Especially, we say the precedence relations \prec between the jobs are chains if the set of all jobs can be partitioned into m job subsets $\{J_{(i,j)} : 1 \leq j \leq n_i\}$, $1 \leq i \leq m$, such that the precedence relations are given by m chains (total orders):

$$J_{(i,1)} \prec J_{(i,2)} \prec \dots \prec J_{(i,n_i)}, 1 \leq i \leq m.$$

- If J_i and J_j are two jobs such that $J_i \prec J_j$, we require that J_j is processed at or after the completion time of J_i , and so J_i and J_j cannot be processed in the same batch. In fact, if $J_i \in B_x$ and $J_j \in B_y$, then batch B_x must be processed before batch B_y .
- Following Brucker (2001) and Lenstra et al. (1977), the parallel batch scheduling problem is denoted by

$$1|r_j; \text{prec}; p\text{-batch}|f,$$

where f is the objective function to be minimized, and “p-batch” means “parallel batch”.

- A feasible schedule is given by a batch sequence

$$BS = (B_1, B_2, \dots, B_N),$$

such that, for any two jobs J_i and J_j with $J_i \prec J_j$, if $J_i \in B_x$ and $J_j \in B_y$, then $x < y$.

- Whence a feasible batch sequence $BS = (B_1, B_2, \dots, B_N)$ is given, we assume that each batch starts at the earliest possible starting time such that every two distinct batches do not overlap each other.
- Hence, the starting time S_{B_j} and completion time C_{B_i} can be defined recursively in the following way:

$$S_{B_1} = r_{B_1} \text{ and } C_{B_1} = S_{B_1} + p_{B_1},$$

and for $i \geq 2$,

$$S_{B_i} = \max\{r_{B_i}, C_{B_{i-1}}\} \text{ and } C_{B_i} = S_{B_i} + p_{B_i}.$$



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 5 of 30

Go Back

Full Screen

Close

Quit



- If $r_j = 0$ for all jobs, then the completion time of any batch B_x is naturally

$$C_{B_x} = \sum_{i=1}^x \max_{J_j \in B_i} p_j.$$

- Suppose that each job J_j has a due date d_j .
- $L_j(\pi) = C_j(\pi) - d_j$: the lateness of job J_j under schedule π .
- $T_j(\pi) = \max\{0, L_j(\pi)\}$: the tardiness of job J_j under schedule π .
- $U_j(\pi) = 1$ if $T_j(\pi) > 0$ and $U_j(\pi) = 0$ if $T_j(\pi) \leq 0$.

Home Page

Title Page



Page 6 of 30

Go Back

Full Screen

Close

Quit

- The common objective functions has the following form:

C_{\max} : $C_{\max}(\pi) = \max\{C_j(\pi) : 1 \leq j \leq n\}$, the makespan;

L_{\max} : $L_{\max}(\pi) = \max\{L_j(\pi) : 1 \leq j \leq n\}$, the maximum lateness;

$\sum C_j$: $\sum_{1 \leq i \leq n} C_j(\pi)$, the total completion time;

$\sum T_j$: $\sum_{1 \leq i \leq n} T_j(\pi)$, the total tardiness;

$\sum U_j$: $\sum_{1 \leq i \leq n} U_j(\pi)$, the number of tardy jobs;

$\sum w_j C_j$: $\sum_{1 \leq i \leq n} w_j C_j(\pi)$, the total completion time;

$\sum w_j T_j$: $\sum_{1 \leq i \leq n} w_j T_j(\pi)$, the total tardiness;

$\sum w_j U_j$: $\sum_{1 \leq i \leq n} w_j U_j(\pi)$, the weighted number of tardy jobs.

- All of the above objective functions are regular, where a scheduling objective function $f(C_1, C_2, \dots, C_n)$ is called regular if it is nondecreasing for each job's completion time C_j . Occasionally, some results or algorithms may be applied for all regular functions.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 7 of 30

Go Back

Full Screen

Close

Quit

2 The existing complexity results and open problems

• Polynomially solved problems:

1 p-batch C_{\max} ,	$O(n)$,	Brucker et al. (JOS, 1998)
1 r_j ; p-batch C_{\max} ,	$O(n^2)$,	Lee and Uzsoy (IJPR, 1999)
1 p-batch f_{\max} ,	polynomial,	Brucker et al. (JOS, 1998)
1 p-batch L_{\max} ,	$O(n^2)$,	Brucker et al. (JOS, 1998)
1 p-batch $\sum C_j$,	$O(n \log n)$,	Brucker et al. (JOS, 1998)
1 p-batch $\sum w_j C_j$,	$O(n \log n)$,	Brucker et al. (JOS, 1998)
1 p-batch $\sum U_j$,	$O(n^3)$,	Brucker et al. (JOS, 1998)
1 r_j ; $p_j = p$; p-batch f ,	polynomial,	Baptiste (MMOR, 2000)
1 $prec$; $p_j = p$; p-batch f ,	$O(n^2)$,	Brucker and Knust (2000)
1 r_j ; $prec$; $p_j = p$; p-batch C_{\max} ,	$O(n^2)$,	Cheng, Yuan and Yang (CAOR, 2005), posed as open by Brucker and Knust (2000)



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 8 of 30

Go Back

Full Screen

Close

Quit

● NP-hard problems:

- 1| p-batch| $\sum w_j T_j$, ordinary NP-hard, $O(n^2 P)$,
Brucker et al. (JOS, 1998)
- 1| p-batch| $\sum T_j$, ordinary NP-hard,
Liu, Yuan and Cheng (ORL, 2003),
Posed as open by Brucker et al. (JOS, 1998)
- 1| p-batch| $\sum w_j U_j$, ordinary NP-hard, $O(n^2 P)$,
Brucker et al. (JOS, 1998)
- 1| r_j ; p-batch| L_{\max} , ordinary NP-hard, $O(n^2 P)$,
Cheng, Liu and Yu (IIE Trans. 2001)
- 1| r_j ; p-batch| $\sum w_j C_j$, ordinary NP-hard, $O(n^2 P)$,
Deng and Zhang (LNCS. 1999)
- 1| r_j ; p-batch| f , ordinary NP-hard, $O(n^4 P^3)$,
Liu, Yuan and Cheng (ORL, 2003)
- 1| *chains*; p-batch| C_{\max} , strongly NP-hard,
Cheng, Ng, Yuan and Liu (NRL, 2004)
- 1| *chains*; p-batch| $\sum C_j$, strongly NP-hard,
Cheng, Ng, Yuan and Liu (NRL, 2004)



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 9 of 30

Go Back

Full Screen

Close

Quit



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 10 of 30

Go Back

Full Screen

Close

Quit

The last two results implies that

$1|chains; p\text{-batch}|f$

is strongly NP-hard for every

$$f \in \{C_{\max}, L_{\max}, \sum C_j, \sum T_j, \sum U_j, \sum w_j C_j, \sum w_j T_j, \sum w_j U_j\}.$$

The exact complexity of these problems are posed as open (Brucker and Knust, 2000) in the form

$1|prec; p\text{-batch}|f.$



Introduction and ...
The existing ...
NP-hardness proof

• Open problems:

(1) $1|r_j, \text{p-batch}| \sum C_j$. See Liu, Yuan and Cheng (ORL, 2003).

(2) $1|prec; r_j; p_j = p; \text{p-batch}|f$ for every

$$f \in \{L_{\max}, \sum C_j, \sum T_j, \sum U_j, \sum w_j C_j, \sum w_j T_j, \sum w_j U_j\}.$$

(3) $1|chains; r_j; p_j = p; \text{p-batch}|f$ for every

$$f \in \{L_{\max}, \sum C_j, \sum T_j, \sum U_j, \sum w_j C_j, \sum w_j T_j, \sum w_j U_j\}.$$

Home Page

Title Page

◀ ▶

◀ ▶

Page 11 of 30

Go Back

Full Screen

Close

Quit

3 NP-hardness proof

- We will show that $1|chains; p\text{-batch}|C_{\max}$ is strongly NP-hard.
- The reduction will use the NP-complete vertex cover problem of graphs.
- For a graph G , $V = V(G)$ and $E = E(G)$ denote its sets of vertices and edges, respectively.
- An edge e with end vertices u and v will be denoted by $e = uv = vu$.
- For $e = uv \in E$, we say that e is incident to u and u is incident to e .
- A vertex subset $S \subseteq V(G)$ is said to be a vertex cover of G , if for every edge $e \in E(G)$ there is $u \in S$ such that e is incident to u .
- A path P of G is a sequence of vertices (v_1, v_2, \dots, v_k) such that $v_i \neq v_j$ for $i \neq j$ and $v_i v_{i+1} \in E(G)$ for $1 \leq i \leq k - 1$. A path P of G can be regarded as a subgraph of G .



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 12 of 30

Go Back

Full Screen

Close

Quit



Introduction and ...
The existing ...
NP-hardness proof

- A Hamiltonian path of a graph G is a path P of G such that $V(P) = V(G)$.
- For two graphs G and H , the joint of G and H , denoted by $G + H$, is obtained from G and H by joining each vertex in G with each vertex in H with edges.
- **Vertex cover problem:** For a given graph G and a positive integer k with $k \leq |V(G)| - 1$, is there a vertex cover S of G such that $|S| \leq k$?
- By Garey (1976 and 1979), it is known that the vertex cover problem is NP-complete in the strong sense.
- For the vertex cover problem, if a Hamiltonian path of the considered graph is given as input, the special vertex cover problem is called “restricted vertex cover problem” in this paper.

Home Page

Title Page



Page 13 of 30

Go Back

Full Screen

Close

Quit

Lemma 1 The restricted vertex cover problem of graphs is NP-complete in the strong sense.

Proof The restricted vertex cover problem of graphs is a subproblem of the vertex cover problem, and so in the class NP. To prove the NP-completeness, we establish a polynomial reduction from the vertex cover problem of graphs to the restricted one.

Let an instance of the vertex cover problem be given, which inputs a graph G with $V(G) = \{v_1, v_2, \dots, v_n\}$ and an integer k with $1 \leq k \leq n - 1$ and asks whether or not there is a vertex cover S of G such that $|S| \leq k$. We construct an instance of the restricted vertex cover problem (*i.e.*, a graph H together with a Hamiltonian path P and an integer k' with $1 \leq k' \leq |V(H)| - 1$) as follows.

Set $H = G + K_n$, where K_n is the complete graph with vertex set $V(K_n) = \{u_1, u_2, \dots, u_n\}$ such that $V(K_n) \cap V(G) = \emptyset$. Set $P = (u_1, v_1, u_2, v_2, \dots, u_n, v_n)$ and $k' = n + k$.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 14 of 30

Go Back

Full Screen

Close

Quit



Introduction and ...
The existing ...
NP-hardness proof

The above construction takes a polynomial time. Moreover, P is clearly a Hamiltonian path of H .

It is routine to check that S is a vertex cover of G with $|S| \leq k$ if and only if $S' = S \cup \{u_1, u_2, \dots, u_n\}$ is a vertex cover of H with $|S'| \leq k' = n + k$. This means that the vertex cover problem can be polynomially reduced to the restricted vertex cover problem. By the fact that the vertex cover problem is strongly NP-complete, we conclude that the restricted vertex cover problem is strongly NP-complete. \square

[Home Page](#)

[Title Page](#)

[◀◀](#) [▶▶](#)

[◀](#) [▶](#)

Page 15 of 30

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Theorem 2 The scheduling problem $1|chains; p\text{-batch}|C_{\max}$ is strongly NP-hard.

Proof The decision version of the considered scheduling problem asks, for a given instance of the problem and a positive integer Y , whether there is a feasible batch sequence BS such that $C_{\max}(BS) \leq Y$. It can be easily seen that the decision problem is in NP. To prove the strong NP-completeness, we use the strongly NP-complete restricted vertex cover problem for the reduction.

Let an instance of the restricted vertex cover problem be given, which inputs a graph G with $V(G) = \{v_1, v_2, \dots, v_n\}$, a Hamiltonian path $P = (v_1, v_2, \dots, v_n)$ of G and an integer k with $1 \leq k \leq n - 1$ and asks whether or not there is a vertex cover S of G such that $|S| \leq k$. Write $m = |E(G)|$. We construct an instance of the decision version of the scheduling problem $1|chains; p\text{-batch}|C_{\max}$ as follows.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 16 of 30

Go Back

Full Screen

Close

Quit



We have $m(n-1) + n$ jobs. Each vertex $v_i \in V(G)$ corresponds to a vertex job J_i , $1 \leq i \leq n$. Each edge $v_i v_j \in E(G)$ with $i < j$ corresponds to $n-1$ edge jobs

$$J_{(1;i,j)}, J_{(2;i,j)}, J_{(3;i,j)}, \dots, J_{(n-1;i,j)}.$$

The processing time of each job is defined in the following way. For $1 \leq i \leq n$, the processing time p_i of the vertex job J_i is n . For $1 \leq x \leq n-1$ and $v_i v_j \in E(G)$ with $i < j$, the processing time $p_{(x;i,j)}$ of the edge job $J_{(x;i,j)}$ is

$$p_{(x;i,j)} = \begin{cases} n+1, & \text{if either } i = x \text{ or } j = x+1; \\ n, & \text{otherwise.} \end{cases}$$

(We should note that, for every edge $v_i v_j \in E(G)$ with $i < j$, among the jobs $J_{(x;i,j)}$, $1 \leq x \leq n-1$, $J_{(i;i,j)}$ and $J_{(j-1;i,j)}$ are the only jobs with processing time $n+1$.)



Introduction and ...
The existing ...
NP-hardness proof

The immediate precedence relations between the jobs are defined by the following $m + 1$ chains:

$$J_1 \prec J_2 \prec \dots \prec J_n;$$

$$J_{(1;i,j)} \prec J_{(2;i,j)} \prec \dots \prec J_{(n-1;i,j)},$$

for $v_i v_j \in E(G)$ with $i < j$. The threshold value Y is defined as $Y = n^2 + k$. We ask whether there is a feasible batch sequence BS such that the makespan under BS is at most Y .

The above reduction takes a polynomial time. In the following, we will prove that the instance of the restricted vertex cover problem has a vertex cover $S \subseteq V(G)$ such that $|S| \leq k$ if and only if the instance of the problem $1|chains; p\text{-batch}|C_{\max}$ has a feasible batch sequence with makespan at most Y .

Home Page

Title Page

◀▶

◀▶

Page 18 of 30

Go Back

Full Screen

Close

Quit

If the instance of the restricted vertex cover problem has a vertex cover $S \subseteq V(G)$ such that $|S| \leq k$, then for each edge $uv \in E(G)$ at least one of u and v is in S . For the reason that $P = (v_1, v_2, \dots, v_n)$ is a Hamiltonian path of G , for each i with $1 \leq i \leq n - 1$, either $v_i \in S$ or $v_{i+1} \in S$. We define the batch sequence

$$BS = (B_1, B_2, \dots, B_n)$$

in the following way. For $1 \leq i \leq n$, we set $J_i \in B_i$. For $v_i v_j \in E(G)$ with $i < j$, if $\{v_i, v_j\} \subseteq S$, we set

$$\begin{aligned} J_{(x;i,j)} &\in B_x, & \text{when } 1 \leq x \leq j - 2, \\ J_{(x;i,j)} &\in B_{x+1}, & \text{when } j - 1 \leq x \leq n - 1; \end{aligned}$$

if $v_i \in S$ and $v_j \notin S$, we set

$$J_{(x;i,j)} \in B_x, \text{ for } 1 \leq x \leq n - 1;$$

if $v_i \notin S$ and $v_j \in S$, we set

$$J_{(x;i,j)} \in B_{x+1}, \text{ for } 1 \leq x \leq n - 1.$$

It is not hard to see that $BS = (B_1, B_2, \dots, B_n)$ is a feasible schedule, and that the processing time of each batch is either n or $n + 1$.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 19 of 30

Go Back

Full Screen

Close

Quit

Now for a job J , let $\sigma(J)$ be the vertex v_x such that $J \in B_x$.

Claim 1 For every job J with processing time $n + 1$, $\sigma(J) \in S$.

Let J be a job with processing time $n + 1$. Then there must be an edge $v_i v_j \in E(G)$ with $i < j$, such that either $J = J_{(i;i,j)}$ or $J = J_{(j-1;i,j)}$. By the definition of vertex cover, at least one of v_i and v_j is in S . We distinguish the following three cases.

Case 1 $J = J_{(i;i,i+1)}$.

In this case, by the definition of the batch sequence BS , if $v_{i+1} \in S$, then $\sigma(J) = v_{i+1} \in S$; if $v_i \in S$ and $v_{i+1} \notin S$, then $\sigma(J) = v_i \in S$.

Case 2 $J = J_{(i;i,j)}$ and $i \leq j - 2$.

In this case, by the definition of the batch sequence BS , if $v_i \in S$, then $\sigma(J) = v_i \in S$; if $v_i \notin S$, then by the fact $v_i v_{i+1} \in E(G)$, $\sigma(J) = v_{i+1} \in S$.

Case 3 $J = J_{(j-1;i,j)}$ and $i \leq j - 2$.

In this case, again by the definition of the batch sequence BS , if $v_j \in S$, then $\sigma(J) = v_j \in S$; if $v_j \notin S$, then by the fact $v_{j-1} v_j \in E(G)$, $\sigma(J) = v_{j-1} \in S$. This completes the proof of Claim 1.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 20 of 30

Go Back

Full Screen

Close

Quit



Now by the result of Claim 1, for each $v_x \notin S$, the processing time of B_x is n under the batch sequence BS ; for each $v_y \in S$, the processing time of B_y is at most $n + 1$ under the batch sequence BS . Hence, the makespan under BS is at most $n(n - |S|) + (n + 1)|S| = n^2 + |S| \leq n^2 + k = Y$.

On the other hand, suppose that $BS^* = (A_1, A_2, \dots, A_N)$ is a feasible batch sequence such that the makespan under BS^* is at most $Y = n^2 + k$. By the fact that the processing time of each job is either n or $n + 1$, we know that the processing time of each batch A_i ($1 \leq i \leq N$) is either n or $n + 1$. This means that $N \leq n$. Because $J_1 \prec J_2 \prec \dots \prec J_n$, there are at least n different batches under any feasible batch sequence. Hence, we must have $N = n$. Set

$$S^* = \{v_x : \text{the processing time of the batch } A_x \text{ is } n + 1\}.$$

By the fact that the makespan under BS^* is at most $Y = n^2 + k$, $|S^*| \leq k$. Hence, we only need to show that S^* is a vertex cover of G in the following.



Let $v_i v_j$ with $i < j$ be an edge of G . Then $J_{(i;i,j)}$ and $J_{(j-1;i,j)}$ are the only jobs with processing time $n + 1$ among the jobs $J_{(x;i,j)}$, $1 \leq i \leq n - 1$. By the fact that the chain

$$J_{(1;i,j)} \prec J_{(2;i,j)} \prec J_{(3;i,j)} \prec \dots \prec J_{(n-1;i,j)}$$

contains $n - 1$ jobs, one of the following cases must occur: either $J_{(i;i,j)} \in A_i$ and $J_{(j-1;i,j)} \in A_{j-1}$, or $J_{(i;i,j)} \in A_i$ and $J_{(j-1;i,j)} \in A_j$, or $J_{(i;i,j)} \in A_{i+1}$ and $J_{(j-1;i,j)} \in A_j$. In the first case, $v_i \in S^*$; in the second case, $\{v_i, v_j\} \subseteq S^*$; and in the third case, $v_j \in A_j$. Hence, at least one of v_i and v_j is in S^* . This means that S^* is a vertex cover of G with $|S^*| \leq k$. The proof is completed. \square

Theorem 3 The scheduling problem $1|chains; p\text{-batch}|\sum C_j$ is strongly NP-hard.

Proof The decision version of the considered scheduling problem asks, for a given instance of the problem and a positive integer Y , whether there is a feasible batch sequence BS such that $\sum C_j(BS) \leq Y$. It can easily be seen that the decision problem is in NP. To prove the strong NP-completeness, we again use the strongly NP-complete restricted vertex cover problem for the reduction.

Let an instance of the restricted vertex cover problem be given, which inputs a graph G with $V(G) = \{v_1, v_2, \dots, v_n\}$, a Hamiltonian path $P = (v_1, v_2, \dots, v_n)$ of G and an integer k with $1 \leq k \leq n - 1$ and asks whether or not there is a vertex cover S of G such that $|S| \leq k$. Write $m = |E(G)|$. We construct an instance of the decision version of the scheduling problem $1|chains; p\text{-batch}|\sum C_j$ as follows.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 23 of 30

Go Back

Full Screen

Close

Quit



We have $m(n - 1) + n + M$ jobs, where $M = (mn - m + n)(n^2 + k)$. Each vertex $v_i \in V(G)$ corresponds to a vertex job J_i , $1 \leq i \leq n$. Each edge $v_i v_j \in E(G)$ with $i < j$ corresponds to $n - 1$ edge jobs

$$J_{(1;i,j)}, J_{(2;i,j)}, J_{(3;i,j)}, \dots, J_{(n-1;i,j)}.$$

In addition, we have M large jobs $J_{n+1}, J_{n+2}, \dots, J_{n+M}$. The processing time of each job is defined in the following way. For $1 \leq i \leq n$, the processing time p_i of the vertex job J_i is n . For $1 \leq i \leq M$, the processing time p_{n+i} of the large job J_{n+i} is nM . For $1 \leq x \leq n - 1$ and $v_i v_j \in E(G)$ with $i < j$, the processing time $p_{(x;i,j)}$ of the edge job $J_{(x;i,j)}$ is

$$p_{(x;i,j)} = \begin{cases} n + 1, & \text{if either } i = x \text{ or } j = x + 1; \\ n, & \text{otherwise.} \end{cases}$$

The immediate precedence relations between jobs are defined by the following $m + 1$ chains:

$$J_1 \prec J_2 \prec \dots \prec J_n \prec J_{n+1} \prec J_{n+2} \prec \dots \prec J_{n+M};$$

$$J_{(1;i,j)} \prec J_{(2;i,j)} \prec \dots \prec J_{(n-1;i,j)},$$

for $v_i v_j \in E(G)$ with $i < j$. The threshold value Y is defined as

$$Y = (n^2 + k + 1)M + \frac{1}{2}nM^2(M + 1).$$

We ask whether or not there is a feasible batch sequence BS such that $\sum C_j(BS) \leq Y$.

The above reduction takes a polynomial time. In the following, we will prove that the instance of the restricted vertex cover problem has a vertex cover $S \subseteq V(G)$ such that $|S| \leq k$ if and only if the instance of the problem $1|chains; p\text{-batch}| \sum C_j$ has a feasible batch sequence BS such that $\sum C_j(BS) \leq Y$.



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 25 of 30

Go Back

Full Screen

Close

Quit

If the instance of the restricted vertex cover problem has a vertex cover $S \subseteq V(G)$ such that $|S| \leq k$, we define the batch sequence

$$BS = (B_1, B_2, \dots, B_{n+M})$$

in the following way. For $1 \leq i \leq n + M$, we set $J_i \in B_i$. For $v_i v_j \in E(G)$ with $i < j$, if $\{v_i, v_j\} \subseteq S$, we set

$$J_{(x;i,j)} \in B_x, \quad \text{when } 1 \leq x \leq j - 2,$$

$$J_{(x;i,j)} \in B_{x+1}, \quad \text{when } j - 1 \leq x \leq n - 1;$$

if $v_i \in S$ and $v_j \notin S$, we set

$$J_{(x;i,j)} \in B_x, \quad \text{for } 1 \leq x \leq n - 1;$$

if $v_i \notin S$ and $v_j \in S$, we set

$$J_{(x;i,j)} \in B_{x+1}, \quad \text{for } 1 \leq x \leq n - 1.$$



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 26 of 30

Go Back

Full Screen

Close

Quit



It is not hard to see that $BS = (B_1, B_2, \dots, B_{n+M})$ is a feasible schedule. By the discussion of the *if* part of Theorem 2, the completion time of the batch B_i , $1 \leq i \leq n$, is at most $n^2 + k$. Then the completion time of the batch $B_{n+i} = \{J_{n+i}\}$, $1 \leq i \leq M$, is at most $(n^2 + k) + inM$. For the reason that the first n batches contain $mn - m + n$ jobs, the total completion time of jobs can be roughly estimated as

$$\sum C_j(BS) \leq (mn - m + n)(n^2 + k) + \sum_{i=1}^M (n^2 + k + inM) = Y.$$

This implies that the instance of the scheduling problem has a required batch sequence.

On the other hand, suppose that $BS^* = (A_1, A_2, \dots, A_N)$ is a feasible batch sequence such that the total completion time of jobs under BS^* is at most Y .

By the fact that

$$J_1 \prec J_2 \prec \dots \prec J_n \prec J_{n+1} \prec J_{n+2} \prec \dots \prec J_{n+M},$$

the n vertex jobs must be processed before the large jobs with each vertex job being processed in one batch. Because each vertex job has processing time n , the starting time of any batch that contains a large job must be at least n^2 . If there is an edge job J that is processed either in the same batch as a large job or after a large job, then by the fact that the large job has processing time nM , the completion time of job J is at least $n^2 + nM$. Because the completion time of the large job J_{n+i} is at least $n^2 + inM$, the total completion time of the jobs under BS^* is greater than

$$n^2 + nM + \sum_{i=1}^M (n^2 + inM) = Y + (n - k - 1)M + n^2 > Y.$$



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 28 of 30

Go Back

Full Screen

Close

Quit



This contradicts our assumption. Hence, each edge job must be processed before every large job. Denote by Δ the maximum completion time of vertex jobs and edges jobs. If $\Delta \geq n^2 + k + 1$, then the completion time of the large job J_{n+i} is at least $n^2 + k + 1 + inM$. It follows that the total completion time of the jobs under BS^* is greater than

$$\sum_{i=1}^M (n^2 + k + 1 + inM) = Y.$$

This contradicts our assumption again. Hence, we must have $\Delta \leq n^2 + k$.

The above discussion means that there is a feasible batch sequence BS' for the vertex jobs and edge jobs such that the makespan Δ is at most $n^2 + k$. By the discussion of the *only if* part of Theorem 2, there is a vertex cover S^* of G such that $|S^*| \leq k$. This completes the proof. \square

Thank You!



Introduction and ...
The existing ...
NP-hardness proof

Home Page

Title Page



Page 30 of 30

Go Back

Full Screen

Close

Quit