J Sched (2010) 13: 123–129
DOI 10.1007/s10951-009-0151-7

Minimizing the total weighted completion time in the relocation problem

Alexander V. Kononov · Bertrand M.T. Lin

Published online: 17 December 2009

Abstract This paper studies the minimization of total weighted completion time in the relocation problem on a single machine. The relocation problem, formulated from an area redevelopment project, can be treated as a resource-constrained scheduling problem. In this paper, we show four special cases to be NP-hard in the strong sense. Problem equivalence between the unit-weighted case and the UET (unit-execution-time) case is established. For two further restricted special cases, we present a polynomial time approximation algorithm and show its performance ratio to be 2.

Keywords Relocation problem · Resource-constrained scheduling · NP-hardness · Approximation algorithm

1 Introduction

Resource constraints are one of the most commonly considered factors in project management and scheduling. In this paper, we study a variant of the relocation problem that involves scheduling with generalized resource constraints. Formally, there is a set of jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ available for processing on a single machine. A pool of V_0 units of a common single-type resource is given for processing the jobs. A job $J_i \in \mathcal{J}$ requires and consumes α_i units of the resource from the resource pool. That is, when the job J_i is to be processed, there must be at least α_i units of the resource in the pool. Upon its completion, the job J_i will immediately return β_i units back to the resource pool. The job J_i has a processing length (time) p_i and a weight w_i. A schedule is said to be feasible if all jobs can be successfully processed. As no idle time is assumed, throughout this paper, schedule and sequence are used interchangeably if no confusion would arise. Let C_i be the completion time of the job J_i in a particular schedule. The studied problem is to determine a schedule that is feasible with respect to V_0 and the total weighted completion time $\sum_{i=1}^{n} w_i C_i$ is minimum. Let us denote the problem by RPWT.

The relocation problem was first proposed and formulated from a redevelopment project in Boston (Kaplan 1986; Kaplan and Berman 1988; PHRG 1986). There were several buildings to be demolished and rebuilt. Before each building was redeveloped, its tenants had to be temporarily housed until new capacities were available for relocation. Tenants were not subjected to reside at the old site. Given a fixed budget for temporary housing during the redevelopment, the municipal government needed to determine a reconstruction sequence of the buildings such that all tenants could be successfully evacuated and housed during the course of the project. The significance of the relocation problem could be attributed to its potential applications in database management (Amir and Kaplan 1988) and financial planning (Xie 1997). Moreover, the relocation problem provides a generalization of conventional resource constraints (Blazewicz et al. 1983; Hammer 1986; Brucker et al. 1999) by allowing that the amount of resource β_i returned by a job can be less than, equal to or greater than α_i, the amount the job has required.

A.V. Kononov
Sobolev Institute of Mathematics, Novosibirsk, Russia
e-mail: alvenko@math.nsc.ru

A.V. Kononov
Novosibirsk State University, Novosibirsk, Russia

B.M.T. Lin (✉)
Institute of Information Management, Department of Information and Finance Management, National Chiao Tung University, Hsinchu 300, Taiwan
e-mail: bmtlin@mail.nctu.edu.tw

In a basic model of the relocation problem, the temporal parameter p_i is not included, i.e., given a fixed amount of the resource, the problem is to plan a feasible redevelopment sequence of the buildings. The minimization counterpart of the feasibility problem is to determine the minimum initial budget required for the existence of a feasible sequence. Kaplan and Amir (1988) showed that this minimization problem is equivalent to the well-known makespan minimization in a two-machine flowshop (Johnson 1954). Kaplan (1986) and Amir and Kaplan (1988) addressed the deployment of multiple working crews. If there are available crews and sufficient resource, the development of several buildings can overlap. Kononov and Lin (2006) showed that minimizing the makespan is strongly NP-hard even when there are only two working crews, all buildings have the same processing time and the new capacity of each building is no less than its original capacity. They also proposed approximation algorithms and analyzed the associated performance ratios. Sevastyanov et al. (2009) investigated the relocation problem of makespan minimization subject to release dates. They analyzed the complexities of several cases and also developed a pseudo-polynomial time algorithm, based on a multi-parametric dynamic programming technique, for the case where the number of different release dates is constant.

To the best of our knowledge, the problem of minimizing the total weighted completion time studied in this paper is new. In Sect. 2, we study the computational complexities of several special cases. The equivalence between special cases will be given to extend the complexity results. Section 3 is devoted the development and analysis of an approximation algorithm for two further restricted cases. We analyze the performance ratio of the proposed algorithm. An instance is given to establish the tightness of the ratio. Concluding remarks will be presented in Sect. 4.

2 Complexity results and problem equivalence

In this section, we give the complexity results of the RPWT problem. Auxiliary notations for the following discussion are required. The contribution of the job J_i is defined by $\delta_i = \beta_i - \alpha_i$. The resource level at time t is denoted by V_t. If at time t some job J_i completes and another job J_j starts being processed, V_t gives the resource level of the moment after J_i deposits the resource it produces and before J_j requires the resource for processing. Let σ be a particular sequence or schedule of jobs. The job in position i, $1 \le i \le n$, is denoted by $\sigma(i)$. The resource level at time t subject to schedule σ is specified by $V_t(\sigma)$. If no confusion arises, V_t will be used for simplicity. The weighted sum of job completion times subject to a feasible schedule σ is represented by $Z(\sigma)$.

Although the WSPT (Weighted Shortest Processing Time First) rule (Smith 1956) optimally solves the $1||\sum w_i C_i$ problem, it cannot deal with the RPWT problem. Consider the following numerical example of 4 jobs and $V_0 = 0$.

Jobs	J_1	J_2	J_3	J_4
p_i	1	1	1	1
w_i	1	1	1	5
α_i	0	0	0	10
β_i	2	3	5	10

By the WSPT rule, the job J_4 should be processed first. In RPWT, the lack of resource, however, defers the processing of J_4. The example indicates the difficulty in composing an optimal schedule as well as suggests the design of NP-hardness proofs.

The NP-hardness results start with the case where each job makes non-positive contributions and has a unit execution time (UET), i.e., $\delta_i \le 0$ and $p_i = 1$ for all jobs $J_i \in \mathcal{J}$. The reduction is based upon the following *Ordered Numerical 3-Dimensional Matching* problem, which can be easily transformed from the well-known *Numerical 3-Dimensional Matching* problem (Garey and Johnson 1979).

Ordered Numerical 3-Dimensional Matching (ON3M problem)

Instance. An integer bound $B \in Z^+$, the sets of indices $\mathcal{A}_1 = \{1, \dots, m\}$, $\mathcal{A}_2 = \{m+1, \dots, 2m\}$, $\mathcal{A}_3 = \{2m+1, \dots, 3m\}$, a positive size x_i of each element i, $1 \le i \le 3m$, with $\sum_{i=1}^{3m} x_i = mB$ and $x_1 \ge x_2 \ge \cdots \ge x_m \ge x_{m+1} \ge \cdots \ge x_{2m} \ge x_{2m+1} \ge \cdots \ge x_{3m}$.

Question. Can $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3$ be partitioned into m disjoint sets $A_1, A_2, \dots, A_m$ such that each A_j, $1 \le j \le m$, contains exactly one element from each of $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and $\sum_{i \in A_j} x_i = B$?

Theorem 1 *The RPWT problem is strongly NP-hard, even if $\delta_i \le 0$ and $p_i = 1$ for all jobs $J_i \in \mathcal{J}$.*

Proof Given $3m$, B, and $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ as specified for ON3M, we let $\theta_k = \sum_{i \in \mathcal{A}_k} x_i$ for $1 \le k \le 3$. Let $\eta = 3m^2 B$. An instance I of $6m - 3$ jobs is constructed as follows:

- Basic jobs J_i, $1 \le i \le 3m$,
  $\alpha_i = mB + x_i$, $\beta_i = 0$, $w_i = \eta + x_i$.
- Connecting jobs $J_{3(m+l-1)+k}$, $1 \le l \le m-1$, $1 \le k \le 3$,
  $\alpha_{3(m+l-1)+k} = \beta_{3(m+l-1)+k} = (3mB + B)(m - l)$,
  $w_{3(m+l-1)+k} = 0$.

The initial resource level is $V_0 = (3mB + B)m$. Throughout the proof, we use Z^* to denote $3m\eta(3m-1) + 3mB(m-1) + \sum_{i \in \mathcal{A}_1} x_i + 2\sum_{i \in \mathcal{A}_2} x_i + 3\sum_{i \in \mathcal{A}_3} x_i$. Note that $Z^* < 3m\eta(3m-1) + 3mB(m-1) + 3(\sum_{i \in \mathcal{A}_1} x_i + \sum_{i \in \mathcal{A}_2} x_i + \sum_{i \in \mathcal{A}_3} x_i) = 3m\eta(3m-1) + 3m^2 B$. It can be showed that a feasible schedule of I with $\sum w_i C_i \le Z^*$ exists if and only if the ON3M problem has a partition as specified. Please refer to Appendix A for the details of the proof. □

Next we investigate the case where all jobs have non-negative contributions and a unit execution time.

Theorem 2 *The RPWT problem is strongly NP-hard, even if $\delta_i \ge 0$ and $p_i = 1$ for all jobs $J_i \in \mathcal{J}$.*

Proof Similarly, from an instance of the ON3M problem, we construct an instance I of $6m-3$ jobs as follows:

- Basic jobs J_i, $1 \le i \le 3m$,
  $\alpha_i = 0, \beta_i = mB + x_i, w_i = B - x_i$.
- Connecting jobs $J_{3(m+i-1)+k}$, $1 \le i \le m-1$, $1 \le k \le 3$,
  $\alpha_{3(m+i-1)+k} = \beta_{3(m+i-1)+k} = (3mB+B)i$, $w_{3(m+i-1)+k} = \eta$.

Because all jobs have non-negative contributions, we set the initial resource level as $V_0 = 0$. Using a similar line of reasoning as in the proof of Theorem 1, we can show that there exists a feasible schedule of I with a weighted completion time of no more than $3\eta(3m-7)(m-1) + 12m^2B - 3\sum_{i\in\mathcal{A}_1} x_i - 2\sum_{i\in\mathcal{A}_2} x_i - \sum_{i\in\mathcal{A}_3} x_i$ if and only if the ON3M problem has the required partition. □

Following the above two theorems, we subsequently want to study the complexity status of the following two problems with arbitrary processing times and a unit weight: (1) $\delta_i \ge 0, w_i = 1$; and (2) $\delta_i \le 0, w_i = 1$. In the following, we show the strong connection between the UET case ($p_i = 1$) and the unit-weighted case ($w_i = 1$). We adopt the definition on p. 118 from Korte and Vygen textbook (Korte and Vygen 2006) for our case.

Let us consider two minimization problems $\mathcal{P}$ and $\mathcal{Q}$ with objective functions $\Phi_{\mathcal{P}}$ and $\Phi_{\mathcal{Q}}$, correspondingly. We say that the problem $\mathcal{P}$ totally reduces to the problem $\mathcal{Q}$ if there are functions f and g, each computable in linear time, such that f transforms an instance I of $\mathcal{P}$ to an instance $\bar{I}$ of $\mathcal{Q}$, and g transforms a solution $\bar{\sigma}$ of $\bar{I}$ to a solution σ of I and $\Phi_{\mathcal{P}}(I,\sigma) = \Phi_{\mathcal{Q}}(\bar{I},\bar{\sigma})$. If $\mathcal{P}$ totally reduces to $\mathcal{Q}$ and $\mathcal{Q}$ totally reduces to $\mathcal{P}$, then both problems are called *totally equivalent*.

Theorem 3 *The UET case ($p_i = 1$) and the unit-weighted case ($w_i = 1$) are totally equivalent.*

Proof Let I be an instance of the UET case containing n jobs $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$ with α_i, β_i and w_i given for each job J_i and an initial resource level V_0. We construct an instance $\bar{I}$ of the unit-weighted case having n jobs $\bar{\mathcal{J}} = \{\bar{J}_1, \ldots, \bar{J}_n\}$ with $\bar{p}_i = w_i, \bar{\alpha}_i = \beta_i$, and $\bar{\beta}_i = \alpha_i$. Set the initial resource level $\bar{V}_0 = V_0 + \sum_{i=1}^n (\beta_i - \alpha_i)$.

Let $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(n))$ be a feasible permutation of jobs in the UET case. We show that $\bar{\sigma} = (\sigma(n), \sigma(n-1), \ldots, \sigma(1))$ is feasible for $\bar{I}$. Indeed, let $\bar{V}_k$ denote the amount of resource after the completion of the jobs $J_{\sigma(n)}, \ldots, J_{\sigma(k+1)}$ in $\bar{\sigma}$. We have $\bar{V}_k - \bar{\alpha}_{\sigma(k)} = \bar{V}_0 + \sum_{i=k+1}^n (\bar{\beta}_{\sigma(i)} - \bar{\alpha}_{\sigma(i)}) - \bar{\alpha}_{\sigma(k)} = V_0 + \sum_{i=1}^k (\beta_{\sigma(i)} - \alpha_{\sigma(i)}) - \beta_{\sigma(k)} = V_0 + \sum_{i=1}^{k-1} (\beta_{\sigma(i)} - \alpha_{\sigma(i)}) - \alpha_{\sigma(k)} \ge 0$. The last inequality follows from the feasibility of schedule σ. Thus, we get $\bar{V}_k \ge \bar{\alpha}_{\sigma(k)}$, and the feasibility of $\bar{\sigma}$ is guaranteed.

Let $C_i(\sigma)$ and $C_i(\bar{\sigma})$ denote the completion times of the job J_i in schedules σ and $\bar{\sigma}$, respectively. Then, $\sum_{i=1}^n C_i(\bar{\sigma}) = \sum_{k=1}^n k\bar{p}_{\bar{\sigma}(n-k+1)} = \sum_{k=1}^n k w_{\sigma(k)} = \sum_{i=1}^n w_i C_i(\sigma)$. It follows that if we can get an optimal schedule of the UET case, then we can easily construct an optimal one of the unit-weighted case, and vice versa. □

Note that an instance of the UET case with $\delta_i \le 0$ is totally equivalent to an instance of the unit-weighted case with $\delta_i \ge 0$, and that an instance of the UET case with $\delta_i \ge 0$ is totally equivalent to an instance of the unit-weighted case with $\delta_i \le 0$. Therefore, two results follow from Theorem 1 and Theorem 2.

Corollary 1 *The RPWT problem is strongly NP-hard, even if $\delta_i \ge 0$ and $w_i = 1$ for all jobs $J_i \in \mathcal{J}$.*

Corollary 2 *The RPWT problem is strongly NP-hard, even if $\delta_i \le 0$ and $w_i = 1$ for all jobs $J_i \in \mathcal{J}$.*

3 2-Approximation algorithm

In this section, we present a 2-approximation algorithm for the UET case with $\delta_i \ge 0$ for all jobs J_i. The algorithm dispatches jobs in a greedy way. In the relocation problem, intuition suggests that a job is preferred if it is more important (w_i is larger) or produces more resource (δ_i is larger). Taking into account both attributes, we therefore create a sequence π_1 of all jobs in non-increasing order of weights w_i and a sequence π_2 of all jobs in non-increasing order of contributions δ_i. In the course of execution of the algorithm, a job J_i is called *available* at time t if its resource requirement $\alpha_i \le V_t$. The algorithm starts by locating the first available job of sequence π_1. The job, if it exists, is assigned to the first position of our schedule. If no job is available, then infeasibility arises. The same logic is applied to sequence π_2 for the second position of our schedule. The dispatching process is continued, by exploiting π_1 and π_2 alternatively, until either all jobs are dispatched or infeasibility is encountered.

Algorithm W

Input. An initial resource level V_0 and a job set $\mathcal{J}$ with $p_i = 1, \alpha_i \le \beta_i$ for all $J_i \in \mathcal{J}$;

Output. A feasible schedule σ or "No feasible schedule".

```
1.  For i = 0, 1, ..., ⌊n/2⌋ do
2.      If no job is available then stop and report
        "no feasible schedule".
3.      Let J_k be the first available job in π_1.
4.      Set σ(2i+1) = k and delete J_k from π_1 and π_2.
5.      Set V_{2i+1} = V_{2i} + δ_k.
6.      If all jobs are scheduled then stop and output σ.
7.      If no job is available then stop and report
        "no feasible schedule".
8.      Let J_k be the first available job in π_2.
9.      Set σ(2i+2) = k and delete J_k from π_1 and π_2.
10.     Set V_{2i+2} = V_{2i+1} + δ_k.
11. End For
12. Stop and output σ
```

The running time of Algorithm W is analyzed as follows. The execution consists of $O(n)$ iterations. Step 3 and Step 8 require $O(n)$ time to locate the first available element in the sequence π_1 and in the sequence π_2. Therefore, the overall running time is $O(n^2)$. The following lemma is concerned with the feasibility of produced schedules.

Lemma 1 *If Algorithm* W *cannot generate a feasible schedule of the given instance, then there exists no feasible schedule of the instance.*

Proof Assume Algorithm W terminates without a feasible schedule after successfully scheduling k jobs, $1 \le k < n$. Let σ' be some feasible schedule and the index i, $1 \le i < k$, be the first position where $\sigma(i) \neq \sigma'(i)$. We locate the job $J_{\sigma(i)}$ in the schedule σ' and insert it into the position i of the schedule σ'. The new schedule remains feasible because the job $J_{\sigma(i)}$ has non-negative contributions. Repeating the process, we can come up with a feasible schedule whose first k jobs are the same as those in σ, which is, however, infeasible. A contradiction arises. Therefore, no feasible schedule can exist if Algorithm W stops without a feasible schedule constructed. □

To analyze the performance ratio, we let σ be the schedule obtained by Algorithm W and let σ^* be an optimal schedule.

Lemma 2 $V_t(\sigma^*) \le V_{2t}(\sigma)$ *for all* $t = 1, \ldots, \lfloor \frac{n}{2} \rfloor$.

Proof Note that for a specific t, $V_t(\sigma^*) = V_0 + \sum_{i=1}^{t} \delta_{\sigma^*(i)}$ and $V_{2t}(\sigma) = V_0 + \sum_{i=1}^{2t} \delta_{\sigma(i)}$. To show $\sum_{i=1}^{t} \delta_{\sigma^*(i)} \le \sum_{i=1}^{2t} \delta_{\sigma(i)}$ for any $t = 1, \ldots, \lfloor \frac{n}{2} \rfloor$, we prove by induction on t that for any $t = 1, \ldots, \lfloor \frac{n}{2} \rfloor$ there is a one-to-one mapping $f_t : \{\delta_{\sigma^*(1)}, \ldots, \delta_{\sigma^*(t)}\} \to \{\delta_{\sigma(1)}, \ldots, \delta_{\sigma(2t)}\}$ such that $\delta_{\sigma^*(i)} \le f_t(\delta_{\sigma^*(i)})$ for all $1 \le i \le t$.

For $t = 1$, define $f_1(\delta_{\sigma^*(1)}) = \max\{\delta_{\sigma(1)}, \delta_{\sigma(2)}\} \ge \delta_{\sigma^*(1)}$. Assume there exists a mapping f_t as specified for some t, $1 \le t < \lfloor \frac{n}{2} \rfloor$. It follows that $V_t(\sigma^*) \le V_{2t}(\sigma)$ for this t. Consider the case for $t+1$. Define $f_{t+1}(\delta_{\sigma^*(i)}) = f_t(\delta_{\sigma^*(i)})$ for all $1 \le i \le t$. If $\delta_{\sigma^*(t+1)} \le \delta_{\sigma(2t+2)}$, then define $f_{t+1}(\delta_{\sigma^*(t+1)}) = \delta_{\sigma(2t+2)}$, and the proof is complete. On the other hand, if $\delta_{\sigma^*(t+1)} > \delta_{\sigma(2t+2)}$, then by the logic of Algorithm W, in the schedule σ the job $J_{\sigma^*(t+1)}$ should have been scheduled earlier than the job $J_{\sigma(2t+2)}$, in other words, $\delta_{\sigma^*(t+1)} = \delta_{\sigma(j)}$ for some $j, 1 \le j \le 2t+1$. Let job $J_{\sigma^*(k)}$ be the job satisfying $f_{t+1}(\delta_{\sigma^*(k)}) = \delta_{\sigma(j)}$. If $\delta_{\sigma^*(k)} \le \delta_{\sigma(2t+2)}$, then redefine $f_{t+1}(\delta_{\sigma^*(k)}) = \delta_{\sigma(2t+2)}$, and define $f_{t+1}(\delta_{\sigma^*(t+1)}) = \delta_{\sigma(j)}$ $(\delta_{\sigma(j)} = \delta_{\sigma^*(t+1)})$. The mapping f_{t+1} satisfies the required criterion, and thus the proof is complete. If, however, $\delta_{\sigma^*(k)} > \delta_{\sigma(2t+2)}$, then repeat the above process to redefine the mapping f_{t+1}. Because $|\{\delta_{\sigma^*(1)}, \delta_{\sigma^*(2)}, \ldots, \delta_{\sigma^*(t+1)}\}| < |\{\delta_{\sigma(1)}, \delta_{\sigma(2)}, \ldots, \delta_{\sigma(2t+2)}\}|$ and we never use any element of σ^* twice, the process will terminate with a mapping f_{t+1} as specified. □

Theorem 4 *Algorithm* W *for the RPWT problem with* $\delta_i \ge 0$ *and* $p_i = 1$ *has a performance ratio of* 2.

Proof Given the job weights $w_{\sigma^*(1)}, w_{\sigma^*(2)}, \ldots, w_{\sigma^*(n)}$ along the positions in an optimal schedule σ^*, we have the weighted completion time $Z(\sigma^*) = \sum_{k=1}^{n} k w_{\sigma^*(k)}$. The following discussion will construct a sequence σ' out of σ such that $Z(\sigma) \le Z(\sigma')$ and $Z(\sigma') \le 2Z(\sigma^*)$. The second inequality will be established by confirming that the coefficient of any job weight $w_{\sigma^*(k)}$, $1 \le k \le \lfloor \frac{n}{2} \rfloor$, is no greater than the coefficient of the job weight $w_{\sigma'(2k-1)}$. In the proof, we focus on the coefficients of job weights and ignore the feasibility issue of the constructed intermediate sequences.

Initially, let $\sigma' = \sigma$. For each k from $\lfloor \frac{n}{2} \rfloor$ down to 1, we consider two cases:

Case 1. The job $J_{\sigma^*(k)}$ with the weight $w_{\sigma^*(k)}$ occupies a position from $\{1, 2, \ldots, 2k-1\}$ in σ'.

In this case, the coefficient of weight $w_{\sigma^*(k)}$ is less than $2k$ in σ'.

Case 2. The job $J_{\sigma^*(k)}$ with the weight $w_{\sigma^*(k)}$ occupies a position from $\{2k, 2k+1, \ldots, n\}$ in σ'.

From Lemma 2 and the fact that all jobs have non-negative contributions, the inequality

$$V_{k-1}(\sigma^*) \le V_{2k-2}(\sigma') \le V_{2k-1}(\sigma')$$

will hold. Therefore, in the execution of Algorithm W, the job $J_{\sigma^*(k)}$ is in the candidate list for position $2k-1$. Due to the selection logic of Algorithm W, we know that $w_{\sigma^*(k)} \le w_{\sigma'(2k-1)}$. Swapping the positions of job weights $w_{\sigma^*(k)}$ and $w_{\sigma'(2k-1)}$ in σ' will not decrease the total weighted completion time of σ'. Moreover, the coefficient of $w_{\sigma^*(k)}$ is now $2k-1$.

The above iterative process will result in a sequence σ' such that

$$Z(\sigma') \le \sum_{k=1}^{n} 2k w_{\sigma^*(k)} = 2Z(\sigma^*).$$

Moreover, $Z(\sigma') \ge Z(\sigma)$ is maintained in the iterative process. Therefore, we have $Z(\sigma)/Z(\sigma^*) \le 2$ and the proof is complete. □

To examine the tightness of the performance ratio, we consider an instance with $V_0 = 0$ and $\mathcal{J} = \{J_1, J_2, \ldots, J_n\}$, where n is even. The jobs are defined as

$$\alpha_i = 0, \beta_i = 0, w_i = 1, \quad \text{for } i = 1, 2, \ldots, \frac{n}{2};$$

$$\alpha_i = 0, \beta_i = 1, w_i = 0, \quad \text{for } i = \frac{n}{2} + 1, \frac{n}{2} + 2, \ldots, n.$$

Applied to this instance, Algorithm W produces the schedule

$$\sigma = \left(1, \frac{n}{2} + 1, 2, \frac{n}{2} + 2, \ldots, \frac{n}{2} - 1, n\right)$$

with $Z(\sigma) = 1 + 3 + \cdots + (n-1) = \frac{n^2 - n}{4}$. The optimal schedule of the instance is

$$\sigma^* = \left(1, 2, \ldots, \frac{n}{2}, \frac{n}{2} + 1, \ldots, n\right)$$

with $Z(\sigma^*) = 1 + 2 + \cdots + \frac{n}{2} = \frac{n^2 + 2n}{8}$. Therefore, $\lim_{n\to\infty} \frac{Z(\sigma)}{Z(\sigma^*)} = 2$.

By the problem equivalence given in Theorem 3, Algorithm W can be applied to generate an approximate solution for the problem with $\delta_i \le 0$ and $w_i = 1$.

4 Conclusion

In this paper, we have considered the minimization of the weighted sum of completion times in the relocation problem, which is a generalized resource-constrained scheduling problem. Four restricted cases were shown to be strongly NP-hard. In the proof, we have also introduced the equivalence between the UET case and the unit-weighted case. A polynomial-time approximation algorithm with a performance ratio of 2 was presented for the special case with $\delta_i \ge 0$ and $p_i = 1$. We gave an instance to establish the tightness of the ratio. By the equivalence conveyed by Theorem 3, the approximation result can be applied to the special case with $\delta_i \le 0$ and $w_i = 1$. An intriguing phenomenon is that the proof techniques used to establish the performance ratio cannot be modified to the two other cases: $\delta_i \le 0$ and $p_i = 1$, and $\delta_i \ge 0$ and $w_i = 1$. It could be interesting to develop approximation algorithms for these two special cases. Moreover, investigating the approximability or inapproximability of the general version of the RPWT problem could be another research topic.

Acknowledgements This research was partially supported by a Russia–Taiwan joint grant under contract numbers NSC95-2416-H-009-013 (Taiwan) and RFBR08-06-92000-HHCa (Russia).

The authors are grateful for the reviewers' comments which improved the presentation of an earlier version of this paper.

Appendix A: Detailed proof of Theorem 1

Assume that the connecting jobs start in increasing order of their indices. If it is not the case, we let J_i and J_j be the two connecting jobs with $i < j$ and J_j preceding J_i in some optimal schedule σ. We swap their positions. Since their net contributions and weights are equal to zero, the swap will not change the amount of resource over time and the value of the objective function. So we just need to be sure that J_i and J_j are still available in new positions. Since all jobs have non-positive contributions, the amount of resource will not increase over time. Further, the inequality $\alpha_i \ge \alpha_j$ holds and J_i is available in σ. It follows that both jobs are available after the swap.

IF Let the sets $A_1, A_2, \ldots, A_m$ be a partition as specified in the ON3M problem. Let π_l be a permutation of integers of A_l in increasing order. Define the sequence

$$\begin{aligned}\sigma_0 = \big(&\pi_1, 3m+1, 3m+2, 3m+3, \pi_2, \ldots, \pi_l, 3(m+l-1)\\ &+1, 3(m+l-1)+2, 3(m+l-1)\\ &+3, \pi_{l+1}, \ldots, \pi_m\big).\end{aligned}$$

It is easy to verify that the schedule defined by the integers in σ_0 as job indices is feasible. Now we calculate the total weighted completion time, $Z(\sigma_0)$, of the schedule. Hereafter, subscripts enclosed by brackets are positional indices for a particular schedule. Because the jobs indexed by the elements of set A_l are processed from time $6(l-1)$ to time $6l-3$ and the weights of all connecting jobs are 0, we have

$$\begin{aligned}Z(\sigma_0) &= \sum_{l=1}^{m}\sum_{k=1}^{3}\big(6(l-1)+k\big)(\eta + x_{[6(l-1)+k]})\\ &= \sum_{l=1}^{m}(18l-12)\eta + \sum_{l=1}^{m}\sum_{k=1}^{3} 6(l-1)x_{[6(l-1)+k]}\\ &\quad + \sum_{l=1}^{m}\sum_{k=1}^{3} k x_{[6(l-1)+k]}.\end{aligned}$$

Note that the indices $6(l-1)+1, 6(l-1)+2$ and $6(l-1)+3$ correspond to the integers of the set A_l. Thus, we have $\sum_{k=1}^{3} x_{[6(l-1)+k]} = B$ for all $l = 1, \ldots, m$. It follows that

$$Z(\sigma_0) = 9m(m+1)\eta - 12m\eta + 6\sum_{l=1}^{m}(l-1)B + \sum_{l=1}^{m}\sum_{k=1}^{3} kx_{[6(l-1)+k]}$$
$$= 3m\eta(3m-1) + 3mB(m-1) + \sum_{i=1}^{m}\sum_{k=1}^{3} kx_{[6(l-1)+k]}.$$

When σ_0 was constructed, each element of $\mathcal{A}_k$, $k = 1, 2$, or 3, corresponds to some position $6(l-1)+k$, $l = 1, 2, \ldots, m$ in σ_0. It follows that for a specific k, $\sum_{l=1}^{m} kx_{[6(l-1)+k]} = k\sum_{i\in\mathcal{A}_k} x_i$, and we obtain $Z(\sigma) = Z^*$.

ONLY IF Let σ be a feasible schedule of the instance I such that $Z(\sigma) \le Z^*$. Firstly, we show that connecting $J_{3(m+l-1)+k}$ does not start after time $6l+k-4$, $1 \le l \le m-1$, $1 \le k \le 3$. Let $J_{3(m+l-1)+k}$ be some connecting job that starts at time $\tau > 6l+k-4$. In the schedule σ, exactly $3(l-1)+k-1$ connecting jobs will complete before $J_{3(m+l-1)+k}$. It follows that at least $3l+1$ basic jobs complete before τ. Then,

$$V_\tau \le (3mB+B)m - (3l+1)mB$$
$$= (3mB+B)(m-l) - B(m-l)$$
$$< (3mB+B)(m-l).$$

The last strict inequality follows from $l \le m-1$. Thus, $\alpha_{3(m+l-1)+k} = (3mB+B)(m-l) > V_\tau$, and we get a contradiction to the feasibility of schedule σ. Consequently, for all l and k, $1 \le l \le m-1$, $1 \le k \le 3$, the completion time of job $J_{3(m+l-1)+k}$ must be less than or equal to $6l+k-4$. Let $J_{3(m+l-1)+k}$ be some connecting job which starts at time $\tau < 6l+k-4$ in the schedule σ. We move $J_{3(m+l-1)+k}$ to time $6l+k-4$ and shift all jobs between $\tau+1$ and $6l+k-4$ one unit earlier. Because $\alpha_{3(m+l-1)+k} = \beta_{3(m+l-1)+k}$ and $w_{3(m+l-1)+k} = 0$, the move will not increase the objective function value.

Thus, we proved that the connecting jobs must occupy time intervals $[6l-3, 6l)$, $l = 1, \ldots, m-1$ and, correspondingly, the basic jobs must occupy time intervals $[6(l-1), 6l-3)$, $l = 1, \ldots, m$. It is clear that each interval $[6l-6, 6l-3)$ contains exactly three jobs $J_{[6l-6]}$, $J_{[6l-5]}$, and $J_{[6l-4]}$. Let $B_l = \sum_{k=1}^{3} x_{[6(l-1)+k]}$. Next we show that

$$\sum_{l=1}^{s} B_l = \sum_{l=1}^{s}\sum_{k=1}^{3} x_{[6(l-1)+k]} \le sB \tag{1}$$

for all $s = 1, \ldots, m$. Let s' be some index such that (1) does not hold. Then,

$$V_{6s'-3} = V_0 - \sum_{l=1}^{s'}\sum_{k=1}^{3} \alpha_{[6(l-1)+k]}$$
$$= (3mB+B)m - \left[3s'mB + \sum_{l=1}^{s'}\sum_{k=1}^{3} x_{[6(l-1)+k]}\right]$$
$$< (3mB+B)(m-s').$$

It is a contradiction to the feasibility of the schedule σ because the connecting job $J_{3(m+s'-1)+1}$ requires $(3mB+B)(m-s')$ units of the resource.

Finally, we have

$$Z(\sigma) = \sum_{l=1}^{m}\sum_{k=1}^{3}\big(6(l-1)+k\big)\big(\eta + x_{[6(l-1)+k]}\big)$$
$$= 9m(m+1)\eta - 12m\eta + \sum_{l=1}^{m}\sum_{k=1}^{3} 6(l-1)x_{[6(l-1)+k]} + \sum_{l=1}^{m}\sum_{k=1}^{3} kx_{[6(l-1)+k]}.$$

The second term $\sum_{l=1}^{m}\sum_{k=1}^{3} 6(l-1)x_{[6(l-1)+k]}$ can be further elaborated as follows:

$$\sum_{l=1}^{m}\sum_{k=1}^{3} 6(l-1)x_{[6(l-1)+k]}$$
$$= 6\sum_{l=1}^{m}(l-1)B_l$$
$$= 6\left(\sum_{l=1}^{m} mB_l - \sum_{l=1}^{m}(m-l+1)B_l\right)$$
$$= 6\big(m^2B - \big(mB_1 + (m-1)B_2 + \cdots + B_m\big)\big)$$
$$= 6\left(m^2B - \left(\sum_{l=1}^{m} B_l + \sum_{l=1}^{m-1} B_l + \cdots + \sum_{l=1}^{1} B_l\right)\right)$$
$$\ge 6m^2B - 6(mB + (m-1)B + \cdots + B) \quad \text{(by (1))}$$
$$= 3mB(m-1).$$

Incorporating the inequality into the equation of $Z(\sigma)$, we get

$$Z(\sigma) \ge 3m\eta(3m-1) + 3mB(m-1) + \sum_{l=1}^{m}\sum_{k=1}^{3} kx_{[6(l-1)+k]}. \tag{2}$$

The last term in (2) is at least $\sum_{i\in\mathcal{A}_1} x_i + 2\sum_{i\in\mathcal{A}_2} x_i + 3\sum_{i\in\mathcal{A}_3} x_i$. Indeed, the equality

$$\sum_{l=1}^{m}\sum_{k=1}^{3} kx_{[6(l-1)+k]} = \sum_{i\in\mathcal{A}_1} x_i + 2\sum_{i\in\mathcal{A}_2} x_i + 3\sum_{i\in\mathcal{A}_3} x_i$$

holds if for all l, the job $J_{[6(l-1)+1]}$ corresponds to some element from the set $\mathcal{A}_1$, the job $J_{[6(l-1)+2]}$ corresponds to some element from the set $\mathcal{A}_2$, and the job $J_{[6(l-1)+3]}$ corresponds to some element from the set $\mathcal{A}_3$. It follows that

$$Z \geq 3m\eta(3m-1) + 3mB(m-1) + \sum_{i\in\mathcal{A}_1} x_i + 2\sum_{i\in\mathcal{A}_2} x_i + 3\sum_{i\in\mathcal{A}_3} x_i = Z^*.$$

Equations (1) and (2) together imply that $Z = Z^*$ if and only if $\sum_{l=1}^{s} B_l = sB$ for all s, $1 \leq s \leq m$. It follows that $B_l = B$, $1 \leq l \leq m$, and the corresponding instance of the Ordered Numerical 3-Dimensional Matching problem has the required partition. The proof is complete.

References

Amir, A., & Kaplan, E. H. (1988). Relocation problems are hard. *International Journal of Computer Mathematics*, *25*, 101–110.

Blazewicz, J., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, *5*, 11–24.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., & Pesch, E. (1999). Resource-constrained project scheduling: notation, classification, models and methods. *European Journal of Operational Research*, *112*(1), 3–41.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. Freedman: San Francisco.

Hammer, P. L. (1986). *Scheduling under resource constraints—deterministic models*. *Annals of operations research: Vol. 7*. Basel: Baltzer.

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, *1*, 61–67.

Kaplan, E. H. (1986). Relocation models for public housing redevelopment programs. *Planning and Design*, *13*(1), 5–19.

Kaplan, E. H., & Amir, A. (1988). A fast feasibility test for relocation problems. *European Journal of Operational Research*, *35*, 201–205.

Kaplan, E. H., & Berman, O. (1988). Orient Heights housing projects. *Interfaces*, *18*(6), 14–22.

Kononov, A. V., & Lin, B. M. T. (2006). On the relocation problems with multiple identical working crews. *Discrete Optimization*, *3*(4), 368–381.

Korte, B., & Vygen, J. (2006). *Combinatorial Optimization: Theory and Algorithms: Vol. 21*. Berlin: Springer.

Sevastyanov, S. V., Lin, B. M. T., & Huang, H. L. (2009, in revision). Tight complexity analysis of the relocation problem with arbitrary release dates. *Discrete Optimization*.

Smith, W. E. (1956). Various optimizers for single stage production. *Naval Research Logistics Quarterly*, *3*, 59–66.

Xie, J.-X. (1997). Polynomial algorithms for single machine scheduling problems with financial constraints. *Operations Research Letters*, *21*(1), 39–42.

PHRG (1986). *New lives for old buildings: revitalizing public housing project*. Public Housing Group, Department of Urban Studies and Planning, MIT Press, Cambridge