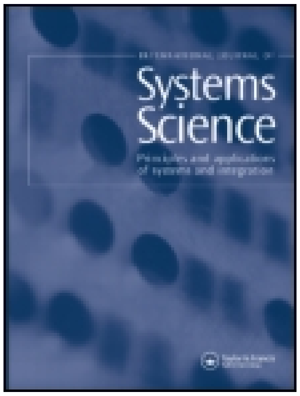

International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

On the relocation problem with a second working crew for resource recycling

B. M. T. Lin [a] & H. L. Huang [b]

[a] Department of Information and Finance Management , Institute of Information Management, National Chiao Tung University , Hsinchu, Taiwan 300

[b] Department of Information Management , National Chi Nan University , Nantou, Taiwan 545

Published online: 24 Feb 2007.

To cite this article: B. M. T. Lin & H. L. Huang (2006) On the relocation problem with a second working crew for resource recycling, International Journal of Systems Science, 37:1, 27-34, DOI: 10.1080/10611860500411100

To link to this article: http://dx.doi.org/10.1080/10611860500411100

International Journal of Systems Science
Vol. 37, No. 1, 15 January 2006, 27–34

Taylor & Francis
Taylor & Francis Group

On the relocation problem with a second working crew for resource recycling

B. M. T. LIN*† and H. L. HUANG‡

†Department of Information and Finance Management, Institute of Information Management, National Chiao Tung University, Hsinchu, Taiwan 300
‡Department of Information Management, National Chi Nan University, Nantou, Taiwan 545

(*Received 21 October 2004; in final form 15 April 2005*)

In this paper, we introduce a variant of the relocation problem, which was formulated from a public house redevelopment project in Boston. In the problem of interest, given some initial resources in a common pool there is a set of jobs to be processed on a two-machine flowshop. Each job acquires a specific number of resources to start its processing and will return a number of resources to the pool at its completion. The resource consumption and resource recycle processes are performed on machine one and machine two, respectively, in a two-machine flowshop style. Abiding by the resource constraints, the problem seeks to find a feasible schedule whose makespan is minimized. In this paper, we first present NP-hardness proofs for some special cases. Three heuristic algorithms are designed to compose approximate schedules. Two lower bounds are developed and then used to test the performance of our proposed heuristics. Numerical results from computational experiments suggest that the proposed heuristics can produce quality solutions in a reasonable time.

Keywords: Relocation problem; Flowshop; Makespan; NP-hard; Heuristic algorithms; Lower bound

1. Introduction

The study on the relocation problem originated from the redevelopment of buildings in a housing project in Boston (Kaplan 1986, PHRG 1986). In the project, a number of buildings were to be torn down and erected for regional development. Before the project started, the authority encountered the situation that temporary housing units were required by the tenants who would be evacuated during the rehabilitation process. The authority needed to determine a minimum budget required for temporarily housing the evacuated tenants. Kaplan (1986) first formulated, through an analytical study, the problem of determining a construction sequence of the buildings subject to the initial budget allocated. This problem can also be studied from an optimization point of view, that is, to determine the minimum initial budget guaranteeing a feasible redevelopment sequence of the buildings. Theoretical significance of the relocation problem lies in its mathematical equivalence to the classical Johnson's two-machine flowshop problem of makespan minimization (Johnson 1954, Kaplan and Amir 1988). The financial constraint problem (Xie 1997) can be also treated as a special case of the relocation problem. From a practical point of view, it has implications related to the memory management issue in database systems (Amir and Kaplan 1988).

We formally introduce the relocation problem in the following. A project is initiated to redevelop B buildings with V_0 temporary housing units. At the beginning of this project, each building i has n_i rooms ($i = 1, \ldots, B$). After reconstruction, each building i can have a capacity of a_i rooms. The new capacity of a building is not necessarily the same as the original one. For example, if building i does not exist before the reconstruction project, then its original capacity $n_i = 0$. On the other hand, if building i is to be demolished, then $a_i = 0$. In general, there is no

*Corresponding author. Email: bmtlin@mail.nctu.edu.tw

International Journal of Systems Science
ISSN 0020–7721 print/ISSN 1464–5319 online © 2006 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/10611860500411100

restricted relation between a_i and n_i, i.e. $a_i < n_i$, $a_i > n_i$ or $a_i = n_i$ is allowed. Although a_i could be greater, equal or less than n_i, the sum of a_b should be greater than the sum of n_b; we denote the restriction as $\sum_1^B a_i \geq \sum_1^B n_i$. If the constraint is not specified, some households would become homeless after the redevelopment project. From the aspect of scheduling theory, we may say that the relocation problem is a generalized version of conventional resource-constrained scheduling problems (Blazewicz *et al.* 1986, 1989) in the following sense. There are V_0 units of resources available in a common resource pool for processing a set of B jobs, each i of which requires a_i units of resources from the pool and returns a_i units to the pool at its completion. The goal seeks to compose a job sequence such that each job can be successfully processed, i.e. when a job is scheduled to start, resources in the pool are sufficient to support its requirement.

The basic relocation problem centers on the feasibility issue instead of temporal considerations as most scheduling problems. However, it is mathematically equivalent to the classical two-machine flowshop scheduling problem of makespan minimization. In Kaplan's study (1986), multiple working crews are assumed, i.e. more than one buildings could be simultaneously developed if resources were available, and the objective is to design a redevelopment schedule of minimum makespan, i.e. the maximum completion time. He proposed a myopic to deliver approximate solutions. Amir and Kaplan (1988) showed that this problem is NP-hard by a reduction from Partition, which is NP-hard in the ordinary sense. Since then, the complexity status of the two-machine case had remained open until Kononov and Lin (2004) confirmed its strong NP-hardness. For other previous works on the relocation problem, the reader is referred to Kaplan (1986), Amir and Kaplan (1988), Kaplan and Berman (1988), Lin and Tseng (1991, 1992, 1993), Lin and Cheng (1999).

In this paper, we study a new generalization that arises from the practical situations where the resources of each completed task are available only after a recycling process that in the mean time needs another type of working crew. In the sense of house redevelopment, we may say that there is a crew for demolishing the buildings and a second team for erecting new buildings from the sites where buildings have been destroyed. The interest of investigation into this generalization lies in another interpretation concerning resource-constrained project management. Each job of the set demands resources for processing and releases some amount of resources. The constraint is that the resources must be recycled before becoming available to successor jobs. Therefore, machine one and machine two in this setting can be referred to as processing machine and recycle machine, respectively. When the process of a job is split into two parts on two different machines, the complexity of the overall decision increases to a certain degree. A very interesting characteristic behind the studied problem is that while the relocation problem, without any temporal constraints, is equivalent to two-machine flowshop scheduling problem, we now have another dimension involving a flowshop of two machines. In other words, two dimensions related two-machine flowshop are interrelated and considered simultaneously.

The rest of this paper is organized as follows. In Section 2, we shall give a formal definition of the two-machine flowshop relocation problem and an example to illustrate the definition. In Section 3, we present some complexity results, including proofs for NP-hardness. As the problem turns out to be computationally challenging, we develop in Section 4 heuristic algorithms and computational experiments. Finally, we give some concluding remarks in Section 5.

2. Problem statements

In this section, a formal formulation will be defined. The notation that will be used throughout this paper will also be introduced.

With a number of initial resources V_0 in a common resource pool, a set of jobs $N = \{1, 2, \ldots, n\}$ is to be processed on a two-machine flowshop. Each job i has two subtasks where task one and task two must be processed by machine one and machine two, respectively. The processing times needed are p_i and q_i. The processing of the first task of job i will consume n_i units of resources immediately when the processing starts. That is, the processing can be started only if machine one is available and there are at least n_i units of resources available in the common resource pool. The processing of the second subtask cannot commence unless its first subtask is finished and machine two is available. At the completion of the second subtask on machine two, a_i units of resources will be produced and returned to the common pool. The value of a_i is not necessarily equal to that of n_i. For an illustration of the problem definition, we consider the set of three jobs under $V_0 = 5$ shown in table 1. Gantt charts and resource levels of two example schedules S_1: 1-2-3 and S_2: 3-2-1 are shown in figure 1. Schedules S_1 and S_2 are both feasible, but S_1 achieves a better makespan.

Adopting the three-field notation used in Graham *et al.* (1979), we denote the problem of concern by $F2RP//C_{\max}$. The first field indicates the problem's environment, i.e. a two-machine flowshop for the relocation problem. The second field denotes specific conditions on job characteristics, such as all jobs have the same resource requirement or the same

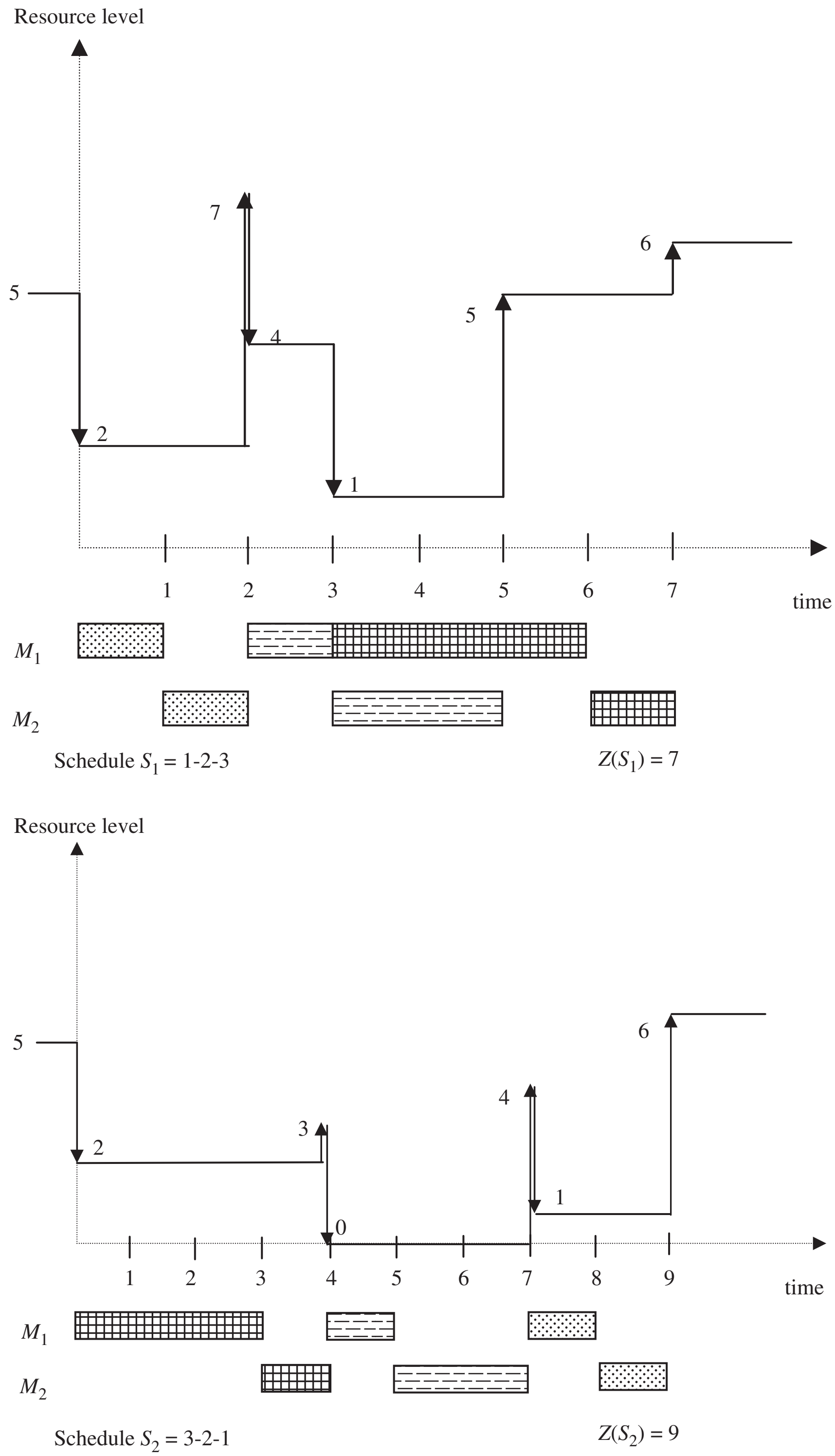

Figure 1. Example schedules with evolution of resource levels.

processing time. The last field specifies the objective function that we are seeking to optimize.

For the classical two-machine flowshop problem, Johnson (1954) proposed an elegant solution algorithm, which arranges the jobs such that for jobs i and j, i precedes j if $\min\{p_i, q_j\} \leq \max\{p_j, q_i\}$. In another interpretation, the algorithm consists of three steps: 1. Partition the jobs set into $N^+ = \{i \mid i \in N \text{ and } p_i \leq q_i\}$ and $N^- = N - N^+$; 2. Arrange the jobs of N^+ in non-decreasing order of p_i, and arrange the jobs

Table 1. Example set of three jobs.

Jobs	p_i	q_i	n_i	a_i
1	1	1	3	5
2	1	2	3	4
3	3	2	3	1

of N^- in non-increasing order of q_i; and 3. Append the subsequence of N^- into that of N^+. Regardless of processing times, the basic relocation problem is to determine the minimum number of initial resources that guarantees the existence of a feasible schedule. We call this basic guarantee for job set N as the minimum resource requirement of N and denote it by $V(N)$. Kaplan and Amir (1988) have shown that to find a schedule achieving $V(N)$, Johnson's algorithm can be applied by letting $p_i = n_i$ and $q_i = a_i$. The above algorithm has a practical implication because the jobs that are beneficial in terms of resources ($p_i \leq q_i$ or $n_i \leq a_i$) are of course scheduled first. The two problems are mathematically equivalent by the fact that the cumulated idle time of a schedule in a two-machine flowshop problem is equal to the minimum resource requirement of its corresponding instance in the relocation problem. Although polynomial time algorithms exist for the two problems, when we have a two-dimensional flowshop (one for temporal consideration and the other for resource constraint), the problem's structures become sophisticated and hinder the development of efficient solution algorithms.

3. Complexity results

As we mentioned in the previous section, like Johnson's problem and the financial constraint problem, the feasibility issue of the relocation problem can be resolved in polynomial time by a simple ordering procedure. However, the complexity of the relocation problem arises when the demolishing time and reconstruction time are taken into consideration. In this section, we present the complexity results of the problem and some special cases.

First, we present a strong NP-hardness proof of the problem under study through a polynomial time reduction from the 3-Partition problem, which is known as strongly NP-hard (Garey and Johnson 1979).

3-Partition: Given an integer B and a set A of $3n$ positive integers $x_1, x_2, \ldots, x_{3n}$, $B/4 < x_i < B/2$, $1 \leq i \leq 3n$, such that $\sum_{i=1}^{3n} x_i = nB$, is there a partition $A_1, A_2, \ldots, A_n$ of the set A such that $|\sum_{x_i \in A_l} x_i = B$, $1 \leq l \leq n$?

Theorem 1: *The special case $F2RP//C_{\max}$ is strongly* NP-*hard, even if all jobs have the same processing time on machine one.*

Proof: We use $F2RP/p_i = p/C_{\max}$ to denote the conditions specified in the theorem. It is not hard to see that the decision version of the special case $F2RP/p_i = p/C_{\max}$ is in NP. Next, we perform a polynomial-time reduction from 3-Partition. For a given instance of 3-Partition, we construct a corresponding set of $4n$ jobs for the $F2RP/p_i = p/C_{\max}$ problem as follows:

$$\begin{aligned} \text{Ordinary jobs:} \quad & p_i = B, q_i = B - x_i, n_i = 0, \\ & \text{and } a_i = x_i, \quad 1 \leq i \leq 3n; \\ \text{Enforcer jobs:} \quad & p_i = B, q_i = 2B, n_i = B, \\ & \text{and } a_i = 0, \quad 3n + 1 \leq i \leq 4n; \end{aligned}$$

Let the number of initial resources $V_0 = B$. We claim that there is a partition as specified in 3-Partition if and only if there is a feasible schedule whose makespan is no greater than $(4n+1)B$. Because the sum of processing times on machine two of all jobs is $4nB$, only those schedules whose total idle time is equal to or less than B will be considered in the following proof.

If there is a desired partition $A_1, A_2, \ldots, A_n$ of the set A in 3-Partition, then we schedule the enforcer job $3n+1$ first. The jobs corresponding to the three elements of A_1 follow. The second enforcer job $3n+2$ is then scheduled. Successively, the jobs corresponding to the three elements of A_2 are scheduled. Continuing the dispatching, we can work out a schedule that has only an idle time of B before the first job on machine two, i.e. a makespan of $(4n+1)B$.

Assume that there is a schedule for the $F2RP/p_i = p/C_{\max}$ problem whose makespan is no greater than $(4n+1)B$. As all jobs have the same processing time B on machine one, an idle time of B before the first job on machine two is inevitable. This fact implies that no further idle time is allowed in the schedule. If an ordinary job is scheduled first, then there will be non-zero idle time before the second job on machine two. Therefore, some enforcer job, say $3n+1$, must be assigned to the first position. At the completion, the resource level reduces to zero, which constrains that only ordinary jobs can be scheduled next. Let us focus on the jobs, say $j_1, j_2, \ldots, j_k$, scheduled between the first and the second enforcer jobs. To accumulate sufficient resources for the processing of the second enforcer job, the inequality $x_{j1} + x_{j2} + \cdots + x_{jk} \geq B$ must hold. On the other hand, if $x_{j1} + x_{j2} + \cdots + x_{jk} > B$, then the difference between the completion times of job j_k on machine one and machine two will be less than B. It readily implies a non-zero idle time before the second enforcer job on machine two, a contradiction. Therefore, we have $x_{j1} + x_{j2} + \cdots + x_{jk} = B$. The fact that $B/4 < x_i < B/2$

for any x_i forces the number of ordinary jobs between the two enforcer jobs to be exactly three. We let the three ordinary jobs form subset A_1. Continuing this process, we can similarly find subsets $A_2, \ldots, A_n$ and then conclude the theorem. □

By symmetry, we can also learn from this theorem that the special cases (1) all jobs have the same q_i; (2) all jobs have the same n_i; and (3) all jobs have the same a_i are all strongly NP-hard. The complexity results imply that the $F2RP//C_{\max}$ problem exhibits sophisticated structures and is computationally intractable. In other words, it is very unlikely to develop polynomial or pseudo-polynomial time algorithms for deriving optimal solutions.

4. Heuristic algorithms and computational experiments

4.1. *Heuristic algorithms*

As the generic $F2RP//C_{\max}$ is strongly NP-hard, it is very unlikely to develop polynomial time algorithms for producing optimal solutions. Seeking near-optimal approximate solutions in a reasonable time therefore is a viable alternative for decision makers. In this section, we will design three heuristic algorithms and study their effectiveness.

The first algorithm constructs a feasible schedule in a greedy fashion. That is, the algorithm picks up the most beneficial job, in terms of makespan, from the unscheduled ones. If the decision does not violate the feasibility constraints, then the selected job is appended to the partial schedule. The feasibility testing consists of two issues: the current resource level is sufficient for the job under consideration, and the resource level after processing the job under consideration is sufficient for all of the other remaining jobs. The selection process iterates until the set of unscheduled jobs is emptied. Note that before the algorithm starts, a feasibility test is conducted. Therefore, we assume the initial resources are sufficient for the overall processing. The outline of the algorithm, denoted as Algorithm H_1, is given as below.

Algorithm H_1:

Step 1: Order the jobs by Johnson's algorithm using p_i and q_i;
Step 2: Schedule $S = \phi$;
Step 3: For $k = 1$ to n do
  3.1: Select from N the first job, say i, satisfying (1) $n_i \le V_{k-1}$ and (2) $(V_{k-1} + a_i - n_i) \ge V(N - \{i\})$.
  3.2: Set $S_{[k]} =$ job i; $V_k = V_{k\text{-}1} + a_i - n_i$;
  3.3: Set $N = N - \{i\}$;
Step 4: Report schedule S and $Z(S)$, and stop.

The second heuristic H_2 is similar to the first one except that the jobs are initially arranged in non-decreasing order of $p_i + q_i$. Heuristics H_1 and H_2 both produce approximate schedules in a constructive way so as to ensure the whole process can guarantee feasibility. The third heuristic is simply focused on feasibility and provides an initial schedule in which jobs are arranged by using Amir and Kaplan's algorithm with parameters a_i and n_i. This approach is called heuristic H_3.

To enhance the performance of the proposed algorithms, a 2-OPT procedure is applied when an approximate is produced. Given a feasible solution, the 2-OPT procedure considers all pairs of jobs that will not cause infeasibility if their positions in the solutions are exchanged. The job-pair that can reduce the makespan most is selected and a new solution is obtained by switching the positions of the two jobs. The process iterates until no further improvement is attainable. For convenience, stage one (constructing a feasible schedule) and stage two (improving the solution) together is called algorithm H_i, $i = 1$, 2 or 3.

4.2. *Lower bounds*

Before we proceed to the development of computational experiments, we introduce some lower bounds on the optimal solutions. For decision makers, lower bounds on the solutions of a combinatorial optimization problem not only provide a way to underestimating the final outcome, but also serve as an alternative gauge to measure the quality of approximate solutions. For a given job instance, define two subsets

$$N_1 = \{i | n_i \le V_0, i \in N\},$$

and

$$N_2 = \left\{ i | n_i \le V_0 + \sum_{j \in N \setminus \{i\}} (a_j - n_j), \quad i \in N \right\}.$$

Subset N_1 contains the jobs that can be scheduled at the first position. Subset N_2 is similarly defined based on the last position. Then, two lower bounds are accordingly defined as

$$LB_1 = \min_{i \in N_1}\{p_i\} + \sum_{j \in N} q_j,$$

and

$$LB_2 = \min_{i \in N_2}\{q_i\} + \sum_{j \in N} p_j.$$

The first bound comes from the fact that an idle time before the first job to be processed on the second machine is inevitable. The second bound is defined as the total processing length on machine one plus the last operation on machine two. Combining the two lower bounds, we can define an aggregate lower bound as:

$$LB = \max\{LB_1, LB_2\}.$$

The lower bound will be used to evaluate approximate solutions derived from heuristic or meta-heuristic methods.

4.3. *Computational evaluation*

To examine the efficiency and effectiveness of the proposed heuristic algorithms, we designed and conducted a series of computational experiments with randomly generated test instances. The platform of the experiments is a personal computer with an i586 CPU of 1.01 GHz running Microsoft Windows XP Version 2002. The programs were coded in C++. Values of resource-related parameters n_i and a_i were randomly drawn from the uniform interval [1, 100]. Values of the processing lengths of the jobs on the two machines were randomly generated from the setting $p_i \in [1, 100]$ and $q_i \in [1, 100]$. The intervals used in the instance generation scheme were commonly adopted in the scheduling literature concerning flowshop problems. In our experimental setting, two specific dimensions are considered to reflect the problem scale and level of resource availability.

1. The problem size n varies from 50 up to 500 with an increment of 50.
2. As the degree of availability of resources plays a key role in the feasibility issue, the problem-solving sessions is differentiated by letting $V_0 = 1.2 \times V(N)$ and $1.4 \times V(N)$. That is, the number of initial resources is either 120% or 140% of the minimum resource requirement that ensures the existence of a feasible schedule.

For every combination of the above-mentioned dimensions, we generated thirty instance sets and take the average of their output. All of the three proposed heuristic algorithms H_1, H_2, and H_3 were evaluated. We kept track of the average error ratios of the derived solutions. Because it is very unlikely to attain optimal solution values in a reasonable time, the relative error ratios are defined by $(Z_{Hi} - LB)/LB * 100\%$, where Z_{Hi}, $i = 1$, 2, or 3, is a solution value achieved by Algorithm H_i, and LB is the lower bound presented in the previous subsection. As the heuristic algorithms run quite fast, we also consider an aggregate heuristic, called $H_{\min}$, that reports the value of $\min\{Z_{H1}, Z_{H2}, Z_{H3}\}$.

Tables 2 and 3 summarize the numerical results of our experiments for the two settings classified in the two dimensions, respectively. In the tables, parameter α is the factor defining the availability level of initial resources. In table 2, the entries in each column denote the average values taken over each thirty instances. The columns include average values achieved by the three heuristics, the aggregate heuristic and the lower bound. The columns entitled *Max_Err*, *Avg_Err* and *Std_Err* contain the maximum error, the average error and the standard deviation of errors reported by $H_{\min}$ from each thirty instances for each problem size. The statistics clearly evince that the average errors of the solutions reported by the aggregate $H_{\min}$ is almost negligible except for some problem scale which indicates an average error of about 0.32%. Furthermore, the worst or maximum error ever attained is only about 3.5%, while most of the maximum errors are below 1%. The deviations are rather minor, suggesting the stability of the achieved solution quality. As a general observation, all heuristic algorithms can provide solutions of better qualities for large-scale problems.

The relative performances of three heuristics are compared using the number of times the heuristics performed best out of each thirty instances. The results are summarized in table 4. Heuristic H_1 clearly outperforms the other two in the sense that it produced the best solutions in at least twenty-one of the

Table 2. Solution values and relative errors of different heuristics with $\alpha = 1.2$.

n	H_1	H_2	H_3	$H_{\min}$	Lower bound	*Max_Err* (%)	*Avg_Err* (%)	*Std_Err* (%)
50	2701.93	2704.70	2704.57	2698.97	2691.23	3.50	0.28	0.67
100	2520.40	5252.87	5249.63	5247.37	5230.50	2.54	0.32	0.59
150	7737.97	7741.23	7737.03	7734.53	7723.80	0.78	0.14	0.22
200	10282.27	10284.90	10285.07	10281.63	10272.93	0.66	0.08	0.17
250	12864.87	12866.70	12863.23	12861.53	12852.67	0.69	0.07	0.14
300	15547.40	15549.53	15546.97	15543.79	15533.03	0.37	0.07	0.12
350	18014.47	18017.37	18013.53	18010.37	18004.90	0.25	0.03	0.07
400	20431.63	20431.57	20432.3	20428.63	20425.03	0.21	0.02	0.05
450	22960.43	22961.33	22963.60	22958.47	22953.63	0.23	0.02	0.04
500	25595.03	25598.97	25598.70	25594.10	25586.57	0.21	0.03	0.06

Table 3. Solution values and relative errors of different heuristics with $\alpha = 1.4$.

n	H_1	H_2	H_3	$H_{\min}$	Lower bound	*Max_Err* (%)	*Avg_Err* (%)	*Std_Err* (%)
50	2740.80	2744.40	2743.50	2738.13	2715.23	3.82	0.84	1.12
100	5186.87	5189.83	5192.73	5185.87	5179.20	1.56	0.13	0.33
150	7848.57	7854.33	7850.37	7846.83	7832.57	1.08	0.18	0.33
200	10297.03	10302.53	10297.07	10294.83	10280.87	0.94	0.14	0.22
250	12816.30	12817.57	12814.90	12808.00	12797.20	0.60	0.08	0.16
300	15483.93	15484.37	15483.27	15480.80	15470.20	0.62	0.07	0.16
350	17885.57	17889.67	17885.47	17881.67	17870.83	0.41	0.06	0.12
400	20453.87	20454.20	20453.60	20451.77	20444.50	0.24	0.03	0.07
450	23026.10	23026.23	23029.10	23022.67	23017.33	0.19	0.02	0.05
500	25537.87	25532.07	25534.10	25529.10	25522.77	0.17	0.02	0.05

Table 4. Relative performances among the three heuristics.

	$\alpha = 1.2$			$\alpha = 1.4$		
n	H_1	H_2	H_3	H_1	H_2	H_3
50	25	13	19	24	12	18
100	21	12	21	27	16	14
150	25	12	21	27	12	16
200	28	12	19	26	11	17
250	24	14	21	22	16	20
300	25	11	18	26	13	21
350	24	14	22	26	12	16
400	26	13	20	26	13	16
450	26	15	23	25	15	19
500	27	12	22	23	16	19

thirty replicates. The observation suggests that Johnson's sequence is still a promising starting point for searching for quality approximate solutions.

5. Concluding remarks

This paper has addressed a hybrid generalization, $F2RP//C_{\max}$, of the relocation problem and two-machine flowshop problem. While the generalized problem reflects a practical resource-constrained operations management environment, it also theoretically shapes a two-dimensional flowshop framework. In this paper, we have established the computational intractability of $F2RP//C_{\max}$, even if some specific restrictions are given. We have also developed three heuristic algorithms to produce approximate solutions in an acceptable time. Three lower bounds on the makespan have been developed. Through computational experiments, we have learned that the proposed algorithm is effective in composing quality solutions with small error ratios from lower bounds.

As aforementioned, the relocation problem exhibits theoretical as well as practical interest. However, the relocation problem did not receive considerable attention in the past decades. Recently, some researchers in the area of resource-constrained project management and scheduling recognized the challenging research issues brought forth by the relocation problems. As a sequel, incorporating different measures and different machine environments could move the study in a worthy direction.

Acknowledgements

The authors are grateful to the anonymous referees for their constructive comments that have indicated some errors in the early version of this paper.

This research was partially supported by the National Science Council of the ROC under contract number 92-2416-H-260-006.

References

A. Amir and E.H. Kaplan, "Relocation problems are hard", *International Journal of Computer Mathematics*, 25, pp. 101–110, 1988.

J. Blazewicz, W. Cellary, R. Slowinski and J. Weglarz. *Annals of Operations Research*: *Scheduling Under Resource Constraints – Deterministic Models*, P.L. Hammer and J.C. Baltzer, Eds., Basel, Switzerland: Scientific Publishing Company, 1986.

J. Blazewicz, J.K. Lenstra and A.H.G. Rinnoy Kan, "Scheduling subject to resource constraints: classification and complexity", *Discrete Applied Mathematics*, 5, pp. 11–24, 1989.

M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco, CA: Freedman, 1979.

R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey", *Annals of Discrete Mathematics*, 5, pp. 287–326, 1979.

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

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

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

E.H. Kaplan and O. Berman, "Orient heights housing projects", *Interfaces*, 18, pp. 14–22, 1988.

A.V. Kononov and B.M.T. Lin, "Relocation problems with parallel machines", *Proceedings of The 9th EURO Workgroup International Conference on Project Management and Scheduling*, Nancy, France, April 2004, pp. 49–52.

B.M.T. Lin and T.C.E. Cheng, "Relocation problems to minimize the maximum tardiness and the number of tardy jobs", *European Journal of Operational Research*, 116, pp. 183–193, 1999.

B.M.T. Lin and S.S. Tseng, "Some results of relocation problems with processing times and deadlines", *International Journal of Computer Mathematics*, 41, pp. 1–15, 1991.

B.M.T. Lin and S.S. Tseng, "On the relocation problems of maximizing new capacities under a common due-date", *International Journal of Systems Science*, 23, pp. 1433–1448, 1992.

B.M.T. Lin and S.S. Tseng, "Generating the best K sequences in the relocation problems", *European Journal of Operational Research*, 69, pp. 131–137, 1993.

PHRG. *New Lives for Old Buildings: Revitalizing Public Housing Project*, Public Housing Group, Department of Urban Studies and Planning, MIT, Cambridge, MA, 1986.

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

B.M.T. Lin is a Professor of the Department of Information and Finance Management and the Institute of Information Management, National Chiao Tung University, Taiwan, R.O.C. His research interest is mainly in production scheduling, combinatorial optimization and parallel computing. Dr. Lin has published papers in such professional journals as *IIE Transactions*, *Management Science*, *Naval Research Logistics*, *SIAM Journal on Optimization* and this journal.

H.L. Huang received her MBA degree from the Department of Information Management at National Chi Nan University, Taiwan, R.O.C. She is currently a Ph.D. student at the Institute of Information Management, National Chiao Tung University. Her research interest includes production scheduling and operation research.