mathematics

Article

Relocation Scheduling in a Two-Machine Flow Shop with Resource Recycling Operations

Ting-Chun Lo and Bertrand M. T. Lin *

Institute of Information Management, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan; chunchun.mg08@nycu.edu.tw
* Correspondence: bmtlin@mail.nctu.edu.tw

Abstract: This paper considers a variant of the relocation problem, which is formulated from an urban renewal project. There is a set of jobs to be processed in a two-machine flow shop subject to a given initial resource level. Each job consumes some units of the resource to start its processing on machine 1 and will return some amount of the resource when it is completed on machine 2. The amount of resource released by a job is not necessarily equal to the amount of resource acquired by the job for starting the process. Subject to the resource constraint, the problem is to find a feasible schedule whose makespan is minimum. In this paper, we first prove the NP-hardness of two special cases. Two heuristic algorithms with different processing characteristics, permutation and non-permutation, are designed to construct feasible schedules. Ant colony optimization (ACO) algorithms are also proposed to produce approximate solutions. We design and conduct computational experiments to appraise the performances of the proposed algorithms.

Keywords: resource-constrained scheduling; relocation problem; flow shop; resource recycling; heuristic algorithms; ant colony optimization

Citation: Lo, T.-C.; Lin, B.M.T. Relocation Scheduling in a Two-Machine Flow Shop with Resource Recycling Operations. *Mathematics* **2021**, *9*, 1527. https://doi.org/10.3390/math9131527

Academic Editors: Chin-Chia Wu and Win-Chin Lin

Received: 23 May 2021
Accepted: 23 June 2021
Published: 29 June 2021

1. Introduction

Scheduling is a decision-making process that allocates limited resources to tasks in a given time period to optimize certain objectives in manufacturing as well as service industries [1]. Usually, resources are considered as machines that process the assigned tasks in manufacturing industries. In some scheduling contexts, there could be different extra resources, like capital, crews and technicians, storage space, energy, computer memory, and so on, that are required to support the execution of the tasks. Such scheduling problems are known as resource-constrained scheduling. Resource-constrained project scheduling problems (RCPSP) have received considerable attention for decades. Please refer to Brucker, Drexl, Möhring, Neumann, and Pesch [2], Habibi, Barzinpour, and Sadjadi [3] Herroelen, De Reyck, and Demeulemeester [4], Issa and Tu [5] for comprehensive reviews on RCPSP. The resource constraint featured in the relocation problem is different traditional ones in the sense that the amount of resource released by an activity is not necessarily the same as that acquired for commencing the activity. This study investigates the relocation problem in a two-machine flow shop with the specific feature of a resource recycling mechanism.

The construction industry have various optimization decisions to address in the project course [6,7]. The relocation problem originated from the public house redevelopment project in Boston [8,9]. The project had a set of buildings to be torn down and erected for redevelopment. During the redevelopment process, current tenants of the buildings under reconstruction needed to be relocated to temporary housing units. They could be assigned to new housing units. It was not mandatory for tenants to reside at the same place they lived before. Therefore, the authority had to determine a minimum budget of temporary housing units such that all tenants could be successfully relocated. Kaplan [8] first formulated the relocation problem of determining a feasible redevelopment sequence of the buildings with the initial budget. In the view of optimization, this problem can also be described as finding a feasible sequence of the redevelopment buildings that reflects the minimum initial budget. Kaplan and Amir [10] showed that the relocation problem is mathematically equivalent to the two-machine flow shop scheduling problem for minimizing makespan, implying that the basic relocation problem can be solved by the classical Johnson's algorithm [11].

Lin and Huang [12] first introduced the recycling operations for yielding the resource into the study on the relocation problem. In previous studies on the relocation problem, the resource is consumed when a job starts to process and returns immediately when it is completed. However, the concept of the resource recycling is assumed that we need to have a mechanism or procedure to recycle the resource before the resource can be used for later jobs. Therefore, a job is divided into two separate parts on two dedicated machines: one processed on machine 1 and the other for recycling the resource on machine 2. The operations on machine 1 should have sufficient resource so that they can commence the processing, and the operations on machine 2 should wait for the completion of its corresponding counterpart operations jobs on machine 1. However, the job sequences on the two machines are not necessarily the same. Cheng, Lin and Huang [13] presented an integer linear program formulation for the permutation case, in which the job sequences on all machines are the same. They also investigated the non-permutation case. We continue to study this relocation problem and discuss more theoretical proofs. Then, we present heuristic algorithms and ant colony optimization (ACO) algorithms for both permutation and non-permutation sequences to find feasible schedules with resource constraints for minimum makespan.

The rest of this paper is organised as follows. In Section 2, we present problem statements and give a numerical example followed by a literature review. The complexity results of two special cases are discussed in Section 3. In Sections 4 and 5, we present heuristic algorithms and ant colony optimization algorithms for constructing approximate schedules. Computational experiments and performance statics of the algorithms are given in Section 6. We conclude this search and suggest research directions for future study in Section 7.

2. Problem Definition

In this section, we first introduce the notation that will be used throughout our research. Then, a formal problem formulation follows. An integer programming model is also proposed. The notations are listed below:

Notation:

$\mathcal{N} = \{1, 2, \ldots, n\}$	set of jobs to be processed;
$p_{1,j}$	processing time of job j on machine 1;
$p_{2,j}$	processing time of job j on machine 2;
α_j	resource requirement of job j;
β_j	amount of the resource returned by job j;
$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$	a particular sequence of the jobs (assumed for the case of permutation schedules);
v_0	initial resource level;
v_t	resource level at time $t \geq 0$;
$C_{m,j}$	completion time of job j on machine m, $m = 1, 2$.

We formally state the problem as follows: From time zero onwards, a set of jobs $\mathcal{N} = \{1, 2, \ldots, n\}$ is available to be processed in a two-stage flow shop consisting of machine 1 and machine 2. Initially, the common resource pool contains v_0 units of a single type of resource. Job $j \in \mathcal{N}$ can start processing only if machine 1 is not occupied and the resource level is larger than or equal to α_j. When job j starts processing, it immediately consumes α_j units of the resource and takes p_{1j} units of time on machine 1. After the operation on machine 1 is completed, p_{2j} units of time are required to complete its resource recycling operation on machine 2. When job j completes on machine 2, it produces and returns β_j units of the resource back to the resource pool. No preemption on either machine is permitted. Note that there is no strict relation between α_j and β_j. That is, β_j could be smaller than, equal to, or larger than α_j. The goal is to minimize the makespan. In other words, we want to find a feasible schedule that completes all jobs in the shortest time.

To illustrate the problem definition, we consider an instance of four jobs with an initial resource level $v_0 = 6$. The parameters are shown below in Table 1. We construct two example schedules.

Table 1. An example of four jobs.

	$p_{1,j}$	$p_{2,j}$	α_j	β_j
job 1	3	3	8	9
job 2	1	5	3	10
job 3	5	6	3	8
job 4	6	1	11	4

Figure 1 shows an optimal permutation solution with $C_{\max}$ = 22, and Figure 2 shows an optimal non-permutation solution with $C_{\max}$ = 19. Both of them are feasible. In this example, it is clear that the non-permutation solution can attain a better makespan than the permutation one.

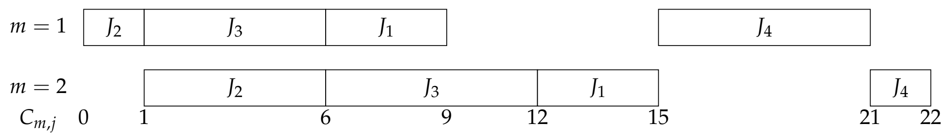

Figure 1. Permutation solution

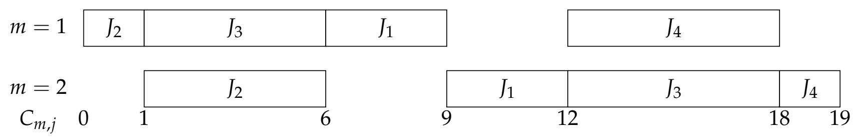

Figure 2. Non-permutation solution

Literature Review

To describe our problem, we denote use the standard three-field notation $F2|rp|C_{\max}$, proposed by Graham et al. [14]. The first field indicates the machine environment of a two-machine flow shop, where the first machine is the operation of the building being torn down and the second machine is about re-constructing buildings corresponding to resource recycling operations. The second field indicates that the specific conditions for the job characteristics, i.e., the relocation problem. The last field specifies the objective function of makespan.

The study on the relocation problem was inaugurated by Kaplan [8] in 1986. The fundamental purpose of the basic relocation problem is to minimize the initial budget required for guaranteeing project feasibility. In Kaplan's study, multiple working crews were considered that if resources were sufficient, i.e., a number of buildings could be simultaneously developed. Kaplan and Amir [10] formulated the application of relocation problem as an integer program. They also noted the relationship between the minimum budget in relocation feasibility and the minimum makespan of two-machine flow shop scheduling, which is solvable in $O(n \log n)$ time [11]. To reflect real situations of the housing redevelopment project in East Boston, Kaplan, and Berman [15] refined the integer programming model and scheduling heuristics. Applications like the financial constraints on single machine scheduling problems [16] which can be reduced to the two-machine flow shop scheduling problem as a special case of the relocation problem. The relocation problem is also related to the memory management issue in database system in practical term [17]. Amir and Kaplan [17] showed that minimizing the makespan on parallel machines is NP-hard. Kononov and Lin [18] proved that parallel-machine setting is strongly NP-hard even if there are only two working crews and all jobs have the same processing time. They also designed approximation algorithms with performance ratio analysis for two special cases.

Cheng and Lin [19] presented more proofs and proposed the concept of composite jobs that can reduce the computational time for handling the relocation problem. Cheng and Lin [20] also demonstrated the concept of relocation scheduling to give an economic interpretation of Johnson's algorithm. The concept can also simplify proofs and reduce time complexity in some two-machine flow shop scheduling problems. There are other extensions from the relocation problem. Lin and Tseng [21] considered the problem with processing times and deadline constraints. Furthermore, they provided a complexity result and two polynomial algorithms to solve the restricted problems. Lin and Tseng [22] proposed a branch-and-bound algorithm to maximize the resource level under a specified due date and considered the precedence constraints [23] that is NP-hard even if the precedence constraints are specified by a bi-partite graph. Lin and Cheng [24] showed two relocation problems of minimizing the maximum tardiness is strongly NP-hard and of minimizing the number of tardy jobs under a due date is NP-hard even when all the jobs have an equal tardy weight and resource requirement. Based on the generalized due dates proposed by Hall [25], Lin and Liu [26] extended the scheduling problem and designed a branch-and-bound algorithm to reduce the computational time required. Sevastyanov, Lin and Huang [27] considered the relocation problem with arbitrary release dates. They developed a multi-parametric dynamic programming algorithm to solve the case with a fixed number if distinct due release dates and analyzed complexity of different problem settings. Kononov and Lin [28] considered minimizing the total weighted completion time and proved four special cases are strong NP-hardness. They established the equivalence between the UET (unit-execution-time) case and the unit-weighted case and presented a 2-approximation algorithm for the restricted special cases.

As per the feature of resource recycling in the relocation problem, there are some existing works. Lin and Huang [12] first introduced the concept of resource recycling. This operation can be processed on a secondary recycling machine that the whole procession can be described as a two-machine flow shop scheduling problem. In this paper, they showed that it is NP-hard and designed three heuristic algorithms to compose approximate schedules. Problem formulation and some complexity results were discussed in Cheng, Lin and Huang [13]. They presented integer linear programming models for finding the feasible permutation sequence and non-permutation sequence with minimum makespan. Lin [29] considered the setting where processing and recycling are carried out on the same single machine. The problem is a generalization of the knapsack problem. He designed a pseudo-polynomial time dynamic programming algorithm and formulated an integer program to solve this recycling problem that operations are processed on the same single machine.

3. Complexity Analysis

This section is dedicated to discussion of the complexity results of the problem of several special cases. First, let instance $\mathcal{I}$ contain n jobs with $p_{1,j}$, $p_{2,j}$, α_j, and β_j given for each job j and an initial resource level v_0. We create another instance $\bar{\mathcal{I}}$ having n jobs with $\overline{p}_{1,j} = p_{2,j}, \overline{p}_{2,j} = p_{1,j}, \overline{\alpha}_j = \beta_j, and\ \overline{\beta}_j = \alpha_j$ that is symmetric to the instance $\mathcal{I}$. Set the initial resource level $\overline{v}_0 = v_0 - \sum_{j=1}^{4q+1}(\beta_j - \alpha_j)$. We claim that the two instances have the optimal makespan. The concept follows the results of Kononov and Lin [28].

Theorem 1. *(Mirror Property) Instance $\mathcal{I}$ and instance $\bar{\mathcal{I}}$ have the optimal makespan.*

Proof. Let $\sigma = (\sigma(1), \sigma(2), \ldots, \sigma(n))$ be a feasible permutation of jobs in the given instance $\mathcal{I}$ case. We show that $\overline{\sigma} = (\sigma(n), \sigma(n-1), \ldots, \sigma(1))$ is feasible for $\bar{\mathcal{I}}$ we created in the above. Assume $\overline{V}_k$ is the resource level after the jobs $\sigma(n), \ldots, \sigma(k+1)$ in $\overline{\sigma}$ complete. We have $\overline{V}_k - \overline{\alpha}_{\sigma(k)} = \overline{v}_0 + \sum_{j=k+1}^{n}(\overline{\beta}_{\sigma(j)} - \overline{\alpha}_{\sigma(j)}) - \overline{\alpha}_{\sigma(k)} = v_0 + \sum_{j=1}^{k}(\beta_{\sigma(j)} - \alpha_{\sigma(j)}) - \beta_{\sigma(k)} = v_0 + \sum_{j=1}^{k-1}(\beta_{\sigma(j)} - \alpha_{\sigma(j)}) - \alpha_{\sigma(k)} \geq 0$. The last inequality is feasible with the schedule σ. Therefore, we can get $\overline{V}_k \geq \overline{\alpha}_{\sigma(k)}$, which shows the schedule $\overline{\sigma}$ is feasible. As we know the schedule $\overline{\sigma}$ and σ are two-stage flow shop, if their sequences are reversed, they have same processing time. Therefore, we can construct another optimal schedule if we get an optimal one. □

For n jobs, there are $(n!)$ possible sequences when the permutation schedules are considered. If we consider the non-permutation variant, the number of schedules will become $O(n! \times n!)$ because the permutations on the two machines could be different. For technical constraints or dispatching fairness, say first comes first served, the processing sequence could be given and fixed [30,31]. In view of implementations, if an optimal schedule can be efficiently obtained from a given job sequence on either machine, we can then reduce the decision tree size from $O(n! \times n!)$ to $O(n!)$. This section will explore the complexity status of the setting with a fixed job sequence.

First, we prove the problems that when the sequence of the jobs on machine 2 is given and fixed, finding the optimal schedule is strongly NP-hard, even if all jobs have the same processing time on machine 2. On the other hand, if the given and fixed sequence of the jobs is on machine 1, we can get the same result that finding optimal schedules is also strongly NP-hard. The proof is given in the following:

3-Partition: Given an integer B and a set A of $3q$ elements $\{1, 2, \ldots, 3q\}$, each $j \in A$ has a size x_j, $B/4 < x_j < B/2$, such that $\sum_{j=1}^{3q} x_j = qB$, is there a partition $A_1, A_2, \ldots, A_q$ of the set A such that $\sum_{x_j \in A_l} x_j = B$, $1 \leq l \leq q$?

Theorem 2. *If a sequence of the jobs on machine 2 is given and fixed, then finding an optimal schedule is strongly NP-hard, even if all jobs have the same processing time on machine 2.*

Proof. Given an instance of 3-Partition, we create a corresponding set of $4q + 1$ jobs as follows:

Enforcer jobs: $p_{1,j} = 0, p_{2,j} = B, \alpha_j = 2B, and\ \beta_j = 3B, \quad 1 \leq j \leq q;$
Ordinary jobs: $p_{1,j} = x_i, p_{2,j} = B, \alpha_j = x_i, and\ \beta_j = 0, \quad q+1 \leq j \leq 4q;$
Final job: $p_{1,j} = 3qB, p_{2,j} = B, \alpha_j = 0, and\ \beta_j = 0, \quad j = 4q + 1.$

The the initial resource level $v_0 = 3B$. The jobs on machine 2 are sequenced in increasing order of their indices. We claim that there is a 3-Partition if there is a feasible schedule whose makespan is no greater than $(4q + 1)B$.

Assume that there exists a desired partition $A_1, A_2, \ldots, A_q$ of 3-Partition. Because the total actual processing length on machine 2 is $(4q + 1)B$, we know that no idle time on machine 2 is permitted. Then, we schedule the enforcer job 1 on machine one first followed by the the ordinary jobs corresponding to the three elements of A_1 and the resource level is brought back to $3B$. Repeat the dispatching pattern and then schedule the last job. It is to see that the schedule is feasible and the makespan is exactly $(4q + 1)B$.

Assume that there is a feasible schedule whose makespan is no larger than $(4q + 1)B$. The total actual processing length on machine 2 of all jobs is $(4q + 1)B$. There is no idle time on machine 2. On the other hand, the total actual processing length on machine 1 is $4qB$. Considering the subsequent operations on machine 2, no idle time is allowed on machine 1. In other words, machine 1 and machine 2 cannot have any idle time in order to attain the makespan $(4q + 1)B$. As given, machine 2 processes the enforcer jobs $1, 2, \ldots, q$ as in their indices. Since all enforcer jobs have the same parameter values, without loss of generality we assume that the enforcer jobs also follow the same processing order on machine 1.

We first note that job 1 should start first on machine 1 for otherwise non-zero idle time will be incurred on machine 2. After completing job 1 on machine 1, the resource level drops from $3B$ to B. To start the next enforcer job 2, machine 1 wait for the previous enforcer job 1 to be completed to accumulate sufficient resources. To avoid the idle time between the first and second enforcer jobs on machine 1, we assign ordinary jobs to fill up the idle period. Let A_1 be the set of elements defining these ordinary jobs. If $\sum_{x_j \in A_1} x_j < B$, then there is an idle time before job 2 on machine 1. On the other hand, if $\sum_{x_j \in A_1} x_j > B$, then the resource is insufficient and the completion time of some ordinary jobs are later than the second enforcer job, leading to the idle time on machine two. As the result, $\sum_{x_j \in A_1} x_j = B$ must hold. Continuing this process, we can find subsets $A_2, \ldots, A_n$ with $\sum_{x_j \in A_l} x_j = B$, $2 \le l \le q$, satisfied for the 3-Partition problem. Figure 3 shows the sequence of the optimal schedule: the purple blocks are enforcer jobs, the red ones are ordinary jobs, and the blue ones are final job. $\square$

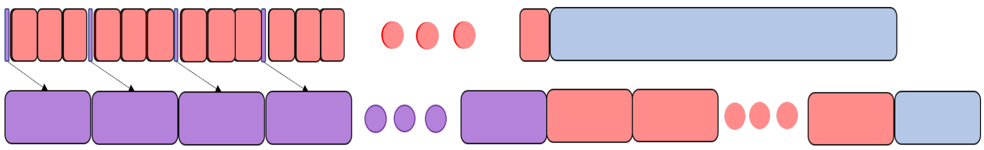

Figure 3. Given and fixed sequence on machine 2.

Theorems 1 and 2 together imply the following result.

Theorem 3. *If a sequence of the jobs on machine 1 is given and fixed, then finding an optimal schedule is strongly NP-hard, even if all jobs have the same processing time on machine 2.*

Proof. Owing to Theorems 1 and 2, we can get the feasibility of a given and fixed sequence of the jobs on machine 1 whose optimal schedule is strongly NP-hard. On the other hand, the idle time before job 1 on machine 2 is inevitable and the total actual processing length is $(4q + 1)B$. Furthermore, the sequence of Theorem 2 whose jobs on machine 2 is given and fixed, if we reverse this two-stage flow shop sequence, we can get the given and fixed sequence of the jobs on machine 1 which is equivalent to Theorem 3. It can seen that Figure 4 is derived form Figure 3 by reversing the Gantt chart from the right. As a result, the sequences of Theorem 2 and Theorem 3 have same total processing time. Then, we get the optimal schedule. $\square$

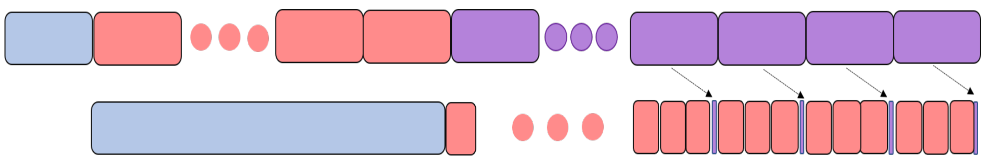

Figure 4. Given and fixed sequence on machine 1.

Theorem 4. *If sequences of the jobs on machine 1 and machine 2 are given and fixed, then the optimal schedule can be found in polynomial time.*

Proof. Assume that there is a feasible schedule whose sequences of the jobs on machine 1 and machine 2 are given and fixed. We schedule the first job on machine 1 followed by the first job on machine 2. If the resource of second job on machine 1 is insufficient, it should wait for previous job which on machine 2 to return the resource. Otherwise, it can be processed immediately when the previous job finished. Then, the job on machine 2 starts when the job on machine 1 completed. Continuing this process, we can schedule all the jobs and the makespan is minimum. On the other hand, if the job on machine 2 return resource is not enough for the next job to be processed, we can know that this sequence is not feasible. As a result, we can get an optimal schedule if sequences of the jobs on machine 1 and machine 2 are given and fixed. □

To simplify the problem, we consider the special case where processing sequences on both machines are given. By problem definition, the two sequences are not necessarily the same. For simplicity in presentation, we re-index the jobs to follow the natural sequence $\pi_1 = (\pi_{1,1}, \pi_{1,2}, \ldots, \pi_{1,n})$ on machine 1. Let $\pi_2 = (\pi_{2,1}, \pi_{2,2}, \ldots, \pi_{2,n})$ denote the sequence on machine 2 and $v(t)$ the resource level at a specific time point t. Notations t_1 and t_2 represent the current time points on machines 1 and 2. Note that if an operation whichever finished on machine 1 or machine 2 and another operation starts on next machine simultaneously at time t, we define $v(t)$ as the resource level after the finished operation on one machine and before the starting operation on the other machine. We will use this method, outlined in Algorithm 1, to calculate makespan for the problem.

Algorithm 1: Two Sequences

1 Let $\pi_1 = (\pi_{1,1}, \pi_{1,2}, \ldots, \pi_{1,n})$ and $\pi_2 = (\pi_{2,1}, \pi_{2,2}, \ldots, \pi_{2,n})$ be the given processing sequences;
2 $t_{1,\pi_{1,1}} = p_{1,\pi_{1,1}}; t_1 = t_{1,\pi_{1,1}};$
3 $v(t_1) = v_0 - \alpha_{\pi_{1,1}}; t_2 = 0;$
4 $i = 2; j = 1;$
5 **while** $i \leq n$ *and* $j \leq n$ **do**
6   **if** $v(t_1) \geq \alpha_{\pi_{1,i}}$ **then**
7         $t_{1,\pi_{1,i}} = \max\{t_1, t_2\} + p_{1,\pi_{1,i}};$
8         $t_1 = t_{1,\pi_{1,i}};$
9         $v(t_1) = v(t_1) - \alpha_{\pi_{1,i}};$
10         $i = i + 1;$
11   **else**
12   **if** *job* $\pi_{1,j}$ *is not yet scheduled on machine 1* **then**
13   Report "No feasible solution!";
14   **else**
15             $t_2 = \max\{t_{1,\pi_{1,j}}, t_2\} + p_{2,\pi_{2,j}};$
16   **if** $t_2 \geq t_1$ **then**
17                 $t_1 = t_2;$
18             $v(t_1) = v(t_1) + \beta_{\pi_{2,j}};$
19             $j = j + 1;$
20 **while** $j \leq n$ **do**
21     $t_2 = \max\{t_{1,\pi_{2,j}}, t_2\} + p_{2,j};$
22     $j = j + 1;$
23 **return** t_2.

In Algorithm 1, the first job of the sequence on machine 1 is processed first and there should be sufficient resource for it to start. Therefore, the first time point $t_{1,1}$ is the processing time of job 1 on machine 1, which is also t_1, and $v(t_1)$ is the resource level when job 1 finishes. In Line 6 we check the resource if we can process the job i on machine 1 or not. If resource is insufficient for job i on machine 1, we execute Line 11 to Line 19 for processing some job j on machine 2 to collect more resource. In Line 12, we need to check if job $\pi_{1,j}$ is scheduled first on machine 1 or not. Because we start to process jobs on machine 2 when the resource is not enough for machine 1, there may be several candidate jobs that can be processed on machine 2. Therefore, sometimes, t_2 is less than t_1 when the resource level is sufficient for the next job. In Lines 16 to 17, if t_2 is larger than t_1, we need to set t_1 equal to t_2, i.e., the next job on machine 1 should wait for the job on machine 2 to recycle its resource. When all the jobs on machine 1 finish, there are still some jobs on machine 2 not yet processed. As a result, we process the remaining jobs on machine 2 in the While loop of Line 20 to Line 22.

4. Heuristic Algorithms

Since the $F2|rp|C_{\max}$ problem is computationally hard, it is hard to find optimal solutions when the problem size is large. We therefore design heuristic algorithms to produce approximate solutions in an acceptable time.

4.1. Permutation

We design two heuristic algorithms, using different sequences to construct feasible schedules for $F2|rp|C_{\max}$ problem. If the job of the sequence does not violate the resource constraint, it must satisfy two conditions that the current resource level is sufficient for it, and that after its processing the resource level is sufficient for all remaining jobs. We denote the job sequence as σ and the remaining jobs sequenced by Johnson's rule using resource parameters α and β as σ_{JR}. Recall that v_0 is the initial resource requirement using Johnson's rule, and V_{needed} denotes the minimum resource requirement for the remaining jobs that the sequence is the same as σ_{JR} excluding job j. If the job violates the constraints, we will remove it. Then, we can get a set of feasible jobs which are the candidates to be processed next. Algorithm 2 examines each of the remaining jobs to determine if they are feasible candidates for the next position.

Algorithm 2: Check Resource

Function `CheckResource(`$v_{now}, \sigma, \sigma_{JR}$`):`
  if σ*.length = 1* **then**
    return σ;
  else
    forall *job* $j \in \sigma_{JR}$ **do**
      if $v_{now} - \alpha_j + \beta_j < V_{needed}$ *or* $v_{now} \prec \alpha_j$ **then**
        σ.remove($job\ j$);
    return σ;

The first heuristic algorithm, JR-time Permutation Heuristic, is outlined in Algorithm 3. We define σ_{time} as the remaining jobs sequenced by Johnson's rule using $p_{1,i}$ and $p_{2,i}$ and σ_{JR} using α_i and β_i. Before a job is processed, we need to run `CheckResource` function for checking whether the job can be processed or not. Then, we append the job to the partial schedule σ. Repeat the same step until all the jobs are processed.

Algorithm 3: JR-time Permutation Heuristic

$\sigma = [\,], v(t_i) = v_0$;
$i = 0$;
while $i \leq n$ **do**
  $\bar{\sigma}$ = `CheckResource(`$v(t_{i+1}), \sigma_{time}, \sigma_{JR}$`);`
  $v(t_{i+1}) = v(t_i) - \bar{\sigma}(\alpha_1) + \bar{\sigma}(\beta_1)$;
  σ.append($\bar{\sigma}(1)$);
  σ_{JR}.remove($\bar{\sigma}(1)$);
  $\bar{\sigma}$.remove($\bar{\sigma}(1)$);
Stop.

The second heuristic, JR-resource Permutation Heuristic, is the same as the previous one except that the job sequence is ordered by Johnson's rule using α_i and β_i.

4.2. Non-Permutation

We design two heuristic algorithms that construct non-permutation schedules for $F2|rp|C_{\max}$. Let σ_1 be the sequence of the jobs processed on machine 1, and σ_2 the job sequence on machine 2. Let $\hat{\sigma_2}$ contain the jobs eligible for processing on machine 2. Algorithm 4 processes the jobs on machine 1 first. If a job satisfies the two constraints, it will be appended to the schedule. Since we want to construct a non-permutation schedule, we create $\hat{\sigma_2}$ to collect the jobs which are finished on machine 1 but not yet on machine 2. When the resource level is insufficient for the candidate job, we need to process the jobs on machine 2 for acquiring more resource. Therefore, we choose the job that has the largest β in $\hat{\sigma_2}$ to be processed first. However, this strategy may lead to an idle time when we only process the selected job. To avoid this situation, we first find the arrival time of the selected job that can be processed and we call it $LargeBetaArrivalTime$ here. Then, check if the completion time of any other job is earlier than $LargeBetaArrivalTime$. Furthermore, we change the $fraction$ such that $1 - fraction$ is the acceptable time range that exceeds $LargeBetaArrivalTime$.

In Line 14, if the acceptable completion time is earlier than $LargeBetaArrivalTime$, then it is appended to σ_2 and removed form $\hat{\sigma_2}$. This process iterates until all the jobs of $\hat{\sigma_2}$ are checked. After that, the job having the largest β is appended to σ_2 and removed form $\hat{\sigma_2}$. We repeat the above steps until all the jobs be processed and then we get a feasible non-permutation schedule.

The second heuristic, JR-time Non-Permutation Heuristic, is similar to the first one except for using $p_{1,i}$ and $p_{2,i}$ to arrange the job sequence.

Algorithm 4: JR-resource Non-Permutation Heuristic

1 Order the jobs by Johnson's rule using α_i and β_i.;
2 $\sigma_1 = [\,], \sigma_2 = [\,], \hat{\sigma_2} = \emptyset$;
3 $i = 0, j = 0, k = 0$;
4 $v(t_{1,i}) = v_0$;
5 **while** $i \leq n$ **do**
6     $\bar{\sigma}$ = `CheckResource`$(v(t_{1,i}), \sigma_{ori}, \sigma_{JR})$;
7   **if** $v(t_{1,i}) \geq \bar{\sigma}(\alpha_1)$ **then**
8         $v(t_{1,i}) = v(t_{1,i}) - \bar{\sigma}(\alpha_1)$;
9   σ_1.append$(\bar{\sigma}(1))$;
10         $\hat{\sigma_2}$.append$(\bar{\sigma}(1))$;
11         $i = i + 1$;
12   **else**
13   **forall** *job* $k \in \hat{\sigma_2}$ **do**
14   **if** $LargeBetaArrivalTime \geq fraction * t_{2,k}$ **then**
15   σ_2.append$(job\ k)$;
16                 $v(t_{2,j}) = v(t_{2,j}) + \beta_k$;
17                 $\hat{\sigma_2}$.remove$(job\ k)$;
18                 $j = j + 1$;
19             $v(t_{2,j}) = v(t_{2,j}) + \beta_{LargeBeta}$;
20   σ_2.append$(J_{2,LargeBeta})$;
21             $\hat{\sigma_2}$.remove$(J_{2,LargeBeta})$;
22             $j = j + 1$;
23 **while** $len(\hat{\sigma_2}) \leq n$ **do**
24     $v(t_{2,j}) = v(t_{2,j}) + \beta_j$;
25   σ_2.append$(job_{2,j})$;
26     $\hat{\sigma_2}$.remove$(job_{2,j})$;
27     $j = j + 1$;
28 Stop.

5. Ant Colony Optimization

In this section, we design an ACO algorithm to solve our problem. We will explain the framework and strategies of the algorithm for producing the approximate sequence.

State transition rule: In the ACO search process, each ant selects the next node to visit by calculating the preference for each path according to the pheromone intensity and heuristic visibility. In the proposed ACO algorithm, the preference P_{ij} of an ant, positioned at node i, for selecting node j is defined as:

$$P_{ij} = \begin{cases} \frac{\tau_{ij}^{w_\tau} \eta_{ij}^{w_\eta}}{\sum_{j \in I} \tau_{ij}^{w_\tau} \eta_{ij}^{w_\eta}}, & \text{if } j \in I; \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where τ_{ij} is the pheromone intensity on the link from node i to node j, and η_{ij} the visibility value from node i to node j, and I the set of remaining admissible jobs to be processed. Parameters w_τ and w_η control the relative importance of τ_{ij} and η_{ij}. The greater a parameter is, the more influence of it to the preference value. In our design, the visibility value η_{ij} is based on a greedy strategy. We prefer less processing times on both machines, less resource requirement, and larger amount of the resource returned by job for priority selection. Visibility value is defined as:

$$\eta_{ij} = \frac{\beta_j}{\alpha_j + p_{1j} + p_{2j}}. \tag{2}$$

We use preference values P_{ij} for our exploration strategy. This method is just like the roulette wheel that every node, i.e., every job has their transition probability, based on which we select the next job randomly. Every node has a chance to be selected, even the probability is low.

Pheromone updating rule: After all the jobs are processed, we update the pheromone tails so that the ants can select their future paths according to previous experience. The trail intensity on link (i, j) is updated as below:

$$\tau_{ij} = (1 - \rho) \times \tau_{ij} + \Delta\tau_{ij}, \tag{3}$$

where ρ represents the pheromone evaporation rate, and $\Delta\tau_{ij}$ the incremental pheromone between nodes i and j given as:

$$\Delta\tau_{ij} = \begin{cases} \frac{Q}{C_k}, & \text{if } j \in \text{I}; \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

In the above definition, Q is an adjustable parameter and C_k the completion time of the last job on machine 2. This strategy is based on policy that the less C_k is, the more pheromone on the path enhanced.

Stopping criterion: The proposed ACO algorithm assigns a colony of ants to probe their own sequences and set a maximum number of iterations. When all the ants complete their routes in one iteration, we select the minimum makespan, i.e., the elite, to be our current best solution. Then, we iterate the process until reaching the maximum number of iterations. If there is a better solution in iterations, this new solution will replace the current best one.

Permutation: To take into account the resource constraints on two machines, we only choose the job that would return the sufficient resource for the remaining jobs to be successfully processed. Therefore, we use the function `CheckResource` before ants select the next job and then enter the ACO algorithm to get the permutation sequence. This method is similar to `JR Permutation Heuristic` except that we use ACO to choose the job sequence.

Non-Permutation: For non-permutation sequences, we divide the algorithm into two parts. In the first part, we use the ACO algorithm to obtain the sequence on machine 1, similar as in Permutation. In second part, the difference form Permutation is when the resource is insufficient for the next job, we use the same method *LargeBetaArrivalTime* in `JR-resource Non-Permutation Heuristic` to select jobs to process on machine 2. Then, we can get a complete sequence and we use it to update the ACO algorithm. As a result, we can get the non-permutation sequence by ACO combined with the heuristic method on machine 2.

6. Computational Experiments

In this section, we present computational experiments on the proposed methods through test data to compare and analyze the performance of these algorithms. The programs were coded in Python and executed on a personal computer with an Intel(R) Core(TM) i7-8700K CPU running at 3.70 GHz with 32.0 GB RAM. The operating system is Windows 10. We will describe how the test data sets were generated. Then, we present the related parameter settings and discuss the experimental results.

Data generation schemes: In our experiments, all parameters are integer. Processing times $p_{1,j}$ and $p_{2,j}$ of jobs on different machines were generated from the uniform distribution, $[1, 10]$. Resource parameters α_j and β_j were generated from the uniform distribution $[1, 20]$. The initial resource level was considered based on 1.1 and 1.4 times the minimum resource requirement that is at least how much the resource is needed for all the jobs of each data set. Test data sets are categorized into 8 different job numbers $n \in \{10, 20, 30, \ldots, 80\}$. For each job number, 5 independent sets were generated. Each set also has different uniform distributions for processing times $p_{1,j}$ and $p_{2,j}$, the resource parameters α_j and β_j, and the initial resource requirement. That means that we have 40 different data sets in all. On each data set, say 10 jobs, heuristic algorithms were run only once since they are deterministic. For a specific setting, the values were averaged over 5 independent sets of the same setting. The ACO algorithm, due to its randomness nature, was exercised 5 runs on each data set to get its average performance.

6.1. Results of Heuristic Algorithms

In this experiment, we apply the four heuristic algorithms on different data sets. We compare permutation solutions with non-permutation ones in two different methods, namely JR-resource and JR-time. The initial resource level in all experiment results set by multiplying the minimum resource requirement by 1.1. For each problem size, the average objective value of derived solution (minimum makespan) are reported. Since the elapsed execution times of four heuristic algorithms are almost negligible, we do not show the execution time in the following tables. All detail experiment results of different data sets are shown in Appendix A.

In Table 2, the makespan of permutation sequence of perm and non-permutation is maxβ. It can be seen that the JR-resource algorithm can get better makespan than the JR-time algorithm. Since the constraint is considered by resource, it is obviously that when the jobs sequenced by processing time, the resource would insufficient and the jobs should wait for the resource returned which lead to idle time. In most of the data sets, the makespans of permutation heuristics are less than non-permutation ones (maxβ in Table 2). However, sometimes, non-permutation can get a better solution that reported in JR-resource with 10 jobs. We speculate that some jobs on machine 2 can fill up the idle time and thus decrease the waiting time on machine 1.

In the experiment, there are different fractions of the bearable exceeding time, which is the acceptable time length that exceeds *LargeBetaArrivalTime*, used in JR-resource and JR-time Non-Permutation Heuristics. The fraction ranges from 5/10 to 10/10. Table 3 is for JR-time Non-Permutation Heuristics focused on processing times, and Table 4 for the heuristics focused on α_i and β_i. If we do not consider the bearable exceeding time, the makespan would be more than others that bear the exceed time because there is longer idle time in the sequences. It is clear that with the 10/10 fraction we get a longer makespan. In most cases, the makespan is the same regardless of the fraction. However, sometimes it is shown that if we bear too much exceeded time, we may get worse makespan with a 5/10 fraction of 20 jobs in JR-time and 5/10 to 6/10 fractions of 10 jobs in JR-resource. The results of fractions are not better than permutation ones, so we do not have further test for different fractions with 1.4 times the initial resource level.

Table 2. JR-resource and JR-time heuristics.

#	JR-Resource		JR-Time	
	perm	**maxβ**	**perm**	**maxβ**
	$C_{\max}$			
10	80	78	87	92
20	137	142	166	177
30	199	202	251	256
40	258	262	332	345
50	307	308	425	439
60	351	357	474	494
70	426	431	596	605
80	467	472	636	642

Table 3. Different fractions in JR-time Non-Permutation Heuristics.

#	Fraction of Exceeding Time					
	10/10	**9/10**	**8/10**	**7/10**	**6/10**	**5/10**
	$C_{\max}$					
10	92	88	87	87	86	86
20	177	169	166	166	166	167
30	256	252	252	252	252	252
40	345	332	332	333	333	332
50	439	424	424	425	425	425
60	494	475	475	475	475	475
70	605	595	597	597	597	597
80	642	638	638	637	637	637

6.2. Results of ACO Algorithms

We discuss the results of ACO algorithms with permutation and non-permutation options. We tuned several parameter values in preliminary tests to determine the setting for further experiments. We observed differences in the results, although not significant. The parameter values leading to better results were adopted as the base setting for the final computational tests. We set the two parameters $w_\tau = 2$ and $w_\eta = 3$ that could get better makespan in the experiment we tested before. The pheromone evaporation ρ is 0.95 to avoid early convergence. Parameter Q is defined as the number of jobs divided by 10 and multiplied by 50. Setting 100 epochs for a solution with the colony size the same as the number of jobs. Then, we have each data set run this process 5 times to get an average makespan and an average elapsed execution time. The time unit here is second and "-" means that no feasible solution was found. All complete experiment results are shown in Appendix A.

Table 4. Different fraction in JR-resource Non-Permutation Heuristics.

#	Fraction of Exceeding Time					
	10/10	**9/10**	**8/10**	**7/10**	**6/10**	**5/10**
	$C_{\max}$					
10	78	78	78	79	80	80
20	142	138	137	137	137	137
30	202	199	199	199	199	199
40	262	258	258	258	258	258
50	308	307	307	307	307	307
60	357	351	351	351	351	351
70	431	428	426	426	426	426
80	472	467	467	467	467	467

In Table 5, perm indicates the permutation method, and maxβ the non-permutation method presented in Section 5. M2-enum is also non-permutation sequence that is different from the method we used in maxβ. The difference between them is that when the resource is insufficient on machine 1, M2-enum will enumerate all the possible sequences of candidate jobs on machine 2 to find the minimum makespan and return resource. This procedure is time-consuming because the number of possible sequences we need to compare is a factorial of the number of candidate jobs on machine 2. Therefore, when the job number is larger than 30, we cannot get a solution in 3600 s. We also experiment on the integer programming method (IP) proposed by Cheng et al. [13] to solve the problem. The IP can get the optimal solutions for data sets. However, when the number of jobs is over 10, it cannot find any feasible solution in 3600 s for most data sets. Therefore, it is regarded as no solution found. With the initial resource is multiplied by 1.1, it is clear that perm can obtain a better makespan and the required run time of perm is also less than maxβ.

Table 5. Results of ACO algorithms and IP with 1.1 $\times$ initial resource levels.

#	**perm**		**maxβ**		**M2-enum**		**IP**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**
10	69	0.41	70	0.80	69	0.84	68	446.81
20	125	2.33	128	4.54	123	27.47	-	-
30	184	8.24	187	13.46	184	3396.50	-	-
40	239	22.09	242	36.99	-	-	-	-
50	299	50.15	300	73.62	-	-	-	-
60	339	97.24	344	139.03	-	-	-	-
70	416	172.08	418	236.40	-	-	-	-
80	456	279.78	459	352.56	-	-	-	-

Table 6 indicates that with more initial resource a better makespan can be achieved, as comapred with those in Table 5. Similarly, M2-enum and IP cannot get any solution when job numbers are over 30 and 10, respectively, in 3600 s. It is also shown that when the initial resource level is larger, for like 30 jobs, M2-enum will waste more time so that it cannot find better solutions in 3600 s (Displayed in Table A18). Then, the permutation method can find an approximate solution with less time. However, results of 1.1 and 1.4 times are within spitting distance when the quantity of jobs increases. It is reckoned that the processing times of the subsequent jobs on machine 2 are larger and the resource is sufficient for them, so their starting times are later than their completion times on machine 1 and they keep processing continuous without idle time. Therefore, the makespan of larger data sets are not quite different when the initial resource increases.

Table 6. Results of ACO algorithms and IP with 1.4 × initial resource levels.

#	perm		maxβ		M2-enum		IP	
	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time
10	66	0.42	67	0.79	69	0.84	65	187.72
20	125	2.44	127	4.44	124	470.49	-	-
30	183	8.31	185	13.95	184	2967.66	-	-
40	239	21.59	241	34.04	-	-	-	-
50	299	49.88	300	69.53	-	-	-	-
60	339	95.94	343	129.10	-	-	-	-
70	416	160.34	418	226.25	-	-	-	-
80	456	263.11	458	364.20	-	-	-	-

Table 7 shows that different fractions of the bearable exceeding time used in ACO Non-Permutation algorithms with different times of the initial resource. In most cases, it would get less makespan with considering the bearable exceeding time. The makespan resulted from fractions between 5/10 and 9/10 are not quite different. It is clearly shown that the makespan of 10/10 fraction is longer than others. As a result, setting the bearable exceeding time can achieve a better performance.

Table 7. Results of different fraction in maxβ of ACO with different initial resource levels.

#	Fraction of Exceeding Time with 1.1 × Initial Resource Levels											
	5/10		6/10		7/10		8/10		9/10		10/10	
	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time
10	69	0.84	69	0.82	69	0.79	69	0.78	70	0.87	70	0.80
20	125	4.65	125	4.61	125	4.81	125	4.35	125	4.71	128	4.54
30	184	14.50	184	15.61	184	15.32	184	13.70	184	14.42	187	13.46
40	239	35.49	239	41.70	239	40.46	239	33.73	239	36.92	242	36.99
50	299	71.12	299	76.58	299	78.63	299	72.06	299	77.92	300	73.62
60	339	128.90	339	141.33	339	144.98	339	133.39	340	139.73	344	139.03
70	416	262.69	417	255.90	417	243.21	416	235.66	416	218.79	418	236.40
80	456	347.84	456	368.73	456	354.50	456	395.34	456	358.64	459	352.56
#	Fraction of Exceeding Time with 1.4 × Initial Resource Levels											
	5/10		6/10		7/10		8/10		9/10		10/10	
	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time
10	66	0.77	66	0.78	66	0.79	66	0.82	66	0.79	67	0.79
20	125	4.52	124	4.51	124	4.56	125	4.49	124	4.54	127	4.44
30	183	13.99	183	13.85	183	13.80	183	13.98	183	13.94	185	13.95
40	239	33.55	239	33.85	239	33.83	239	34.28	239	34.18	241	34.04
50	299	70.50	298	75.06	299	71.54	299	69.69	299	70.03	300	69.53
60	339	132.19	339	130.84	339	130.32	339	130.71	340	129.45	343	129.10
70	417	214.20	416	209.34	417	242.51	416	230.94	417	224.80	418	226.25
80	456	359.57	456	359.48	456	369.24	456	379.99	456	359.53	458	364.20

6.3. Comparison between Heuristics and ACO Algorithms

We discuss the experiment results of the heuristic and the ACO algorithms with special focus on permutation results of JR-time and JR-resource because the makespan of permutation cases are less than other non-permutation heuristic algorithms. ACO-10/10 is the 10/10 fraction of exceeding time bearable in the non-permutation ACO algorithm. We also choose the 5/10 fraction (ACO-5/10) and the permutaion of ACO (ACO-perm) as the control groups.

In Figures 5 and 6, both heuristic algorithms produced larger makespan no matter if the initial resource is multiplied by 1.1 or 1.4, especially for the JR-time. However, there is still some deviations in the ACO algorithms owing to the randomness nature. Except for the above situation, 1.4 times are still better than 1.1 times in most cases with permutation and non-permutation methods, especially JR-time. We reckon that with a higher initial resource level, more jobs on machine 1 can keep continuous processing, thus reducing the idle time waiting for resource return. Concerning the ACO algorithm, it is clear that that the 5/10 fraction is better than the 10/10 case.

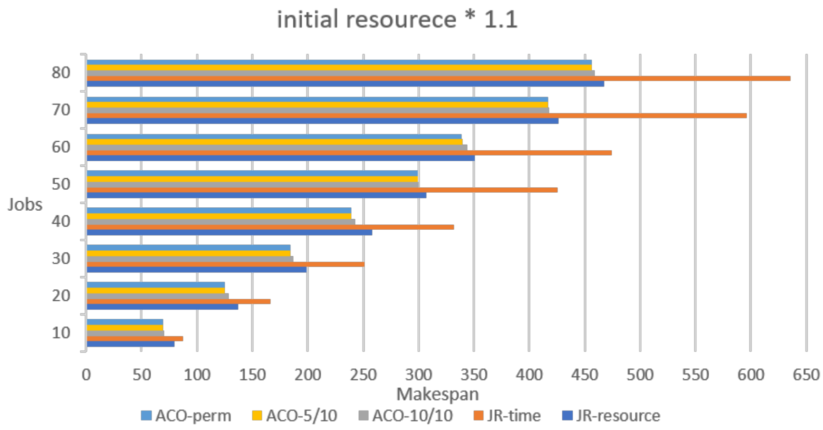

Figure 5. Bar chart of comparison with 1.1 × initial resource levels.

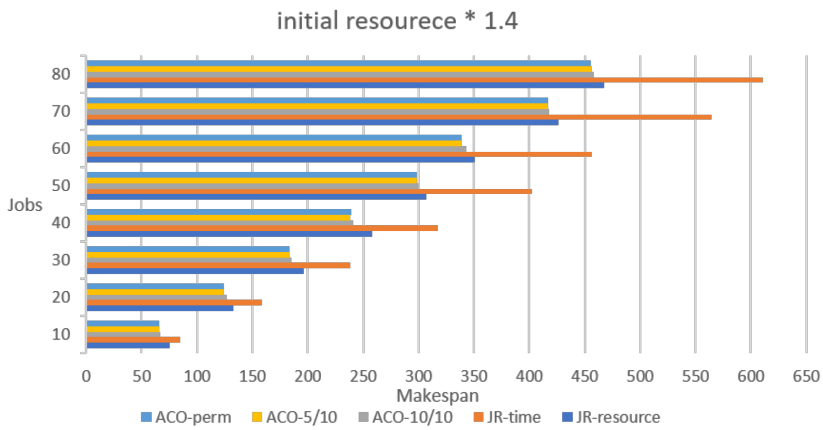

Figure 6. Bar chart of comparison with 1.4 × initial resource levels.

7. Conclusions and Future Works

In this paper, we considered the relocation problem in a two-machine flow shop scheduling problem with the second machine introduced for recycling the resource returned by jobs completed on the first machine. For this problem, we proved that given a sequence of jobs on whichever machine, the problem is still strongly NP-hard. The case with two fixed sequences of jobs on both machines can be solved in polynomial time. For the computationally hard problem, we proposed two heuristic algorithms to construct feasible schedules with permutation and non-permutation sequences. ACO algorithms were designed to find processing sequences on the two machines. Computational experiments indicates that JR-resource produced better makespan than JR-time, and both of their permutation algorithms are better than the non-permutation ones. This result is similar as in ACO algorithm that the makespan of permutation solutions are better. However, when we considered the bearable exceeding time, non-permutation sequences can get better solutions. Between heuristics and ACO algorithms, it is shown that ACO algorithms yielded schedules with less makespan.

For further studies, there is room for further improvements of our proposed solution methods by developing effective methods to arrange the job sequence on machine 2 to mitigate the incurred idle time. Furthermore, we can also combine processing time and resources as a control factor in the heuristic algorithms. It would be also interesting to deploy machine learning and reinforcement learning approaches for finding better parameter settings and updating strategies. We also note that to optimally solve the problem is still limited to small-scale instances. For larger instances, we need tighter lower bounds to facilitate the development of exact methods and provide a tight comparison base for approximation methods. Another direction is identifying application contexts in which the unique type of resource constraints, the amount of consumed resource and the amount of returned resource could be different, is applicable.

Author Contributions: Conceptualization, T.-C.L. and B.M.T.L.; methodology, T.-C.L. and B.M.T.L.; software, T.-C.L.; formal analysis, T.-C.L. and B.M.T.L.; writing—Original draft preparation, T.-C.L. and B.M.T.L.; writing, T.-C.L. and B.M.T.L.; supervision, T.-C.L. and B.M.T.L.; project administration, B.M.T.L.; funding acquisition, B.M.T.L. All authors have read and agreed to the published version of the manuscript.

Funding: Lo and Lin were partially supported by the Ministry of Science and Technology of Taiwan under the grant MOST-109-2410-H-009-029.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets analyzed in this study can be found at http://cpanel-199-19.nctu.edu.tw/~bmtlin/F2RP.zip (accessed on 29 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Detail Experiment Results

All of the experiment results mentioned in Section 6 are summarized from the detail information shown below. We show the makespan ($C_{\max}$) and execution time for each data set followed by the average over five data sets in different job numbers. In the tables, "perm" means permutation method, and "maxβ" non-permutation with a 10/10 fraction. We also set the execution time to stop the program if it exceeds 3600 s. Therefore, if we cannot obtain any solution in 3600 s, we indicate the situation by an "-" entry. In ACO, owing to its randomness property, we run each data sets for five times which is "Run." in the tables. In Tables A18 and A19, when the numbers of jobs are over 30 and 10, we cannot obtain solutions for most cases so that we do not experiment on the cases with more jobs. It is clear that most results are associated with an "-", indicating that no solutions were found within 3600 s.

Table A1. JR-resource with 1.1 × initial resource levels.

Sets	perm		maxβ		9/10		8/10		7/10		6/10		5/10	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
10														
1	78	0.00	78	0.00	78	0.00	78	0.00	78	0.00	78	0.00	78	0.00
2	95	0.00	90	0.00	90	0.00	90	0.00	90	0.00	95	0.00	95	0.00
3	82	0.00	81	0.00	81	0.00	81	0.00	82	0.00	82	0.00	82	0.00
4	72	0.00	72	0.00	72	0.00	72	0.00	72	0.00	72	0.00	72	0.00
5	71	0.00	71	0.00	71	0.00	71	0.00	71	0.00	71	0.00	71	0.00
Avg.	80	0.00	78	0.00	78	0.00	78	0.00	79	0.00	80	0.00	80	0.00
20														
1	144	0.00	154	0.00	144	0.00	144	0.00	144	0.00	144	0.00	144	0.00
2	148	0.00	148	0.00	148	0.00	148	0.00	148	0.00	148	0.00	148	0.00
3	117	0.00	126	0.00	118	0.00	117	0.00	117	0.00	117	0.00	117	0.00
4	160	0.00	165	0.00	165	0.00	160	0.00	160	0.00	160	0.00	160	0.00
5	115	0.00	115	0.00	115	0.00	115	0.00	115	0.00	115	0.00	115	0.00
Avg.	137	0.00	142	0.00	138	0.00	137	0.00	137	0.00	137	0.00	137	0.00
30														
1	196	0.00	196	0.00	196	0.00	196	0.00	196	0.00	196	0.00	196	0.00
2	209	0.00	209	0.00	209	0.00	209	0.00	209	0.00	209	0.00	209	0.00
3	194	0.00	194	0.00	194	0.00	194	0.00	194	0.00	194	0.00	194	0.00
4	201	0.00	211	0.00	201	0.00	201	0.00	201	0.00	201	0.00	201	0.00
5	193	0.00	200	0.00	193	0.02	193	0.00	193	0.00	193	0.00	193	0.00
Avg.	199	0.00	202	0.00	199	0.00	199	0.00	199	0.00	199	0.00	199	0.00
40														
1	267	0.00	270	0.00	267	0.00	267	0.00	267	0.00	267	0.00	267	0.00
2	229	0.00	231	0.00	229	0.00	229	0.01	229	0.00	229	0.00	229	0.00
3	260	0.00	268	0.00	260	0.00	260	0.00	260	0.02	260	0.00	260	0.00
4	278	0.00	286	0.00	278	0.00	278	0.00	278	0.00	278	0.00	278	0.00
5	256	0.00	256	0.02	256	0.00	256	0.00	256	0.02	256	0.00	256	0.00
Avg.	258	0.00	262	0.00	258	0.00	258	0.00	258	0.01	258	0.00	258	0.00
50														
1	310	0.00	311	0.00	310	0.00	310	0.00	310	0.00	310	0.00	310	0.01
2	324	0.00	324	0.02	324	0.00	324	0.00	324	0.02	324	0.00	324	0.00
3	285	0.00	288	0.00	285	0.00	285	0.00	285	0.00	285	0.00	285	0.00
4	307	0.00	307	0.02	307	0.00	307	0.00	307	0.02	307	0.00	307	0.00
5	309	0.00	309	0.00	309	0.02	309	0.00	309	0.00	309	0.01	309	0.00
Avg.	307	0.00	308	0.01	307	0.00	307	0.00	307	0.01	307	0.00	307	0.00
60														
1	340	0.00	350	0.02	340	0.00	340	0.00	340	0.00	340	0.01	340	0.01
2	359	0.00	362	0.00	359	0.02	359	0.00	359	0.02	359	0.01	359	0.01
3	347	0.00	353	0.00	347	0.02	347	0.02	347	0.02	347	0.01	347	0.01
4	348	0.00	348	0.02	348	0.00	348	0.00	348	0.00	348	0.01	348	0.01
5	359	0.00	370	0.00	360	0.00	359	0.00	359	0.02	359	0.01	359	0.01
Avg.	351	0.00	357	0.01	351	0.01	351	0.00	351	0.01	351	0.01	351	0.01

Table A1. *Cont.*

Sets	perm		maxβ		9/10		8/10		7/10		6/10		5/10	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
							70							
1	441	0.00	441	0.00	441	0.02	441	0.02	441	0.02	441	0.01	441	0.01
2	392	0.00	392	0.00	392	0.02	392	0.02	392	0.02	392	0.01	392	0.01
3	448	0.00	459	0.02	448	0.02	448	0.02	448	0.02	448	0.01	448	0.01
4	455	0.00	455	0.02	455	0.02	455	0.02	455	0.02	455	0.01	455	0.01
5	395	0.00	407	0.02	402	0.02	395	0.00	395	0.00	395	0.01	395	0.01
Avg.	426	0.00	431	0.01	428	0.02	426	0.01	426	0.01	426	0.01	426	0.01
							80							
1	505	0.00	505	0.02	505	0.02	505	0.02	505	0.02	505	0.01	505	0.01
2	466	0.00	473	0.02	466	0.02	466	0.02	466	0.02	466	0.01	466	0.01
3	479	0.00	483	0.02	479	0.02	479	0.02	479	0.02	479	0.01	479	0.01
4	438	0.00	441	0.02	438	0.02	438	0.02	438	0.02	438	0.01	438	0.01
5	449	0.00	457	0.02	449	0.02	449	0.02	449	0.02	449	0.01	449	0.01
Avg.	467	0.00	472	0.02	467	0.02	467	0.02	467	0.02	467	0.01	467	0.01

Table A2. JR-time with 1.1 $\times$ initial resource levels.

Sets	perm		maxβ		9/10		8/10		7/10		6/10		5/10	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
							10							
1	89	0.00	89	0.00	89	0.00	89	0.00	89	0.00	89	0.00	89	0.00
2	101	0.00	109	0.00	101	0.00	101	0.00	101	0.00	101	0.00	101	0.00
3	77	0.00	86	0.00	84	0.00	77	0.00	77	0.00	77	0.00	77	0.00
4	78	0.00	84	0.00	78	0.00	78	0.00	79	0.00	74	0.00	74	0.00
5	90	0.00	90	0.00	90	0.00	90	0.00	90	0.00	90	0.00	90	0.00
Avg.	87	0.00	92	0.00	88	0.00	87	0.00	87	0.00	86	0.00	86	0.00
							20							
1	151	0.00	154	0.00	157	0.00	152	0.00	152	0.00	152	0.00	152	0.00
2	211	0.00	236	0.00	216	0.00	212	0.00	212	0.00	212	0.00	212	0.00
3	158	0.00	163	0.02	158	0.00	158	0.02	158	0.02	158	0.00	158	0.00
4	197	0.00	219	0.00	203	0.00	197	0.00	197	0.00	197	0.00	197	0.00
5	115	0.00	113	0.00	110	0.00	110	0.00	111	0.00	111	0.00	115	0.00
Avg.	166	0.00	177	0.00	169	0.00	166	0.00	166	0.00	166	0.00	167	0.00
							30							
1	317	0.02	316	0.00	317	0.00	317	0.00	317	0.00	317	0.01	317	0.00
2	239	0.00	231	0.02	234	0.00	240	0.00	240	0.00	240	0.00	240	0.00
3	220	0.00	229	0.00	222	0.00	222	0.00	222	0.00	222	0.00	222	0.00
4	272	0.00	289	0.02	278	0.00	272	0.02	272	0.02	272	0.00	272	0.00
5	208	0.00	213	0.00	209	0.00	208	0.00	208	0.00	208	0.00	208	0.00
Avg.	251	0.00	256	0.01	252	0.00	252	0.00	252	0.00	252	0.00	252	0.00

Table A2. *Cont.*

Sets	perm		maxβ		9/10		8/10		7/10		6/10		5/10	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
							40							
1	318	0.02	334	0.00	317	0.00	318	0.00	318	0.02	318	0.01	318	0.01
2	272	0.02	276	0.00	271	0.00	272	0.00	272	0.02	272	0.01	272	0.01
3	333	0.02	356	0.00	333	0.00	333	0.00	336	0.02	336	0.01	333	0.01
4	401	0.02	415	0.00	403	0.00	401	0.01	401	0.02	401	0.01	401	0.01
5	337	0.02	343	0.00	334	0.00	337	0.01	337	0.02	337	0.01	337	0.01
Avg.	332	0.02	345	0.00	332	0.00	332	0.00	333	0.02	333	0.01	332	0.01
							50							
1	443	0.00	446	0.02	442	0.02	443	0.02	443	0.02	443	0.02	443	0.02
2	400	0.02	404	0.02	398	0.00	400	0.02	400	0.02	400	0.01	400	0.01
3	416	0.00	441	0.02	417	0.02	414	0.02	418	0.02	418	0.02	418	0.01
4	464	0.00	475	0.02	462	0.02	465	0.02	464	0.02	464	0.02	464	0.02
5	402	0.00	431	0.02	402	0.02	400	0.02	400	0.02	402	0.02	402	0.02
Avg.	425	0.00	439	0.02	424	0.01	424	0.02	425	0.02	425	0.02	425	0.02
							60							
1	583	0.02	596	0.03	588	0.03	585	0.03	585	0.02	583	0.03	583	0.03
2	510	0.02	564	0.03	514	0.03	512	0.03	512	0.03	512	0.03	512	0.02
3	448	0.02	455	0.02	450	0.04	448	0.02	448	0.03	448	0.03	448	0.03
4	391	0.00	385	0.03	382	0.02	392	0.02	392	0.03	392	0.02	391	0.02
5	440	0.02	471	0.03	440	0.02	440	0.03	440	0.02	440	0.02	440	0.03
Avg.	474	0.01	494	0.03	475	0.03	475	0.02	475	0.02	475	0.03	475	0.03
							70							
1	490	0.02	501	0.05	490	0.03	490	0.03	490	0.03	490	0.03	490	0.03
2	614	0.02	625	0.03	615	0.04	615	0.03	615	0.05	615	0.03	615	0.04
3	741	0.03	748	0.05	741	0.05	741	0.05	741	0.03	741	0.04	741	0.04
4	506	0.03	498	0.05	495	0.03	505	0.03	506	0.02	506	0.03	506	0.03
5	631	0.02	654	0.06	634	0.03	634	0.05	633	0.05	632	0.04	631	0.04
Avg.	596	0.02	605	0.05	595	0.04	597	0.04	597	0.03	597	0.04	597	0.04
							80							
1	578	0.02	583	0.06	578	0.05	578	0.06	579	0.05	578	0.05	578	0.05
2	615	0.03	639	0.06	615	0.06	615	0.06	615	0.06	615	0.06	615	0.06
3	824	0.03	831	0.06	827	0.08	827	0.06	824	0.05	824	0.06	824	0.06
4	514	0.03	496	0.05	517	0.03	517	0.06	517	0.03	517	0.04	517	0.04
5	649	0.02	659	0.06	652	0.06	651	0.05	651	0.05	651	0.05	649	0.05
Avg.	636	0.02	642	0.06	638	0.06	638	0.06	637	0.05	637	0.05	637	0.05

Table A3. JR-resource and JR-time with 1.4 × initial resource levels.

Sets	**JR-Resource**				**JR-Time**			
	perm		**maxβ**		**perm**		**maxβ**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**
				10				
1	69	0.00	69	0.00	91	0.00	91	0.00
2	95	0.00	90	0.00	101	0.00	109	0.00
3	73	0.00	75	0.00	77	0.00	85	0.00
4	70	0.00	70	0.00	63	0.00	64	0.00
5	71	0.00	71	0.00	90	0.00	90	0.00
Avg.	76	0.00	75	0.00	84	0.00	88	0.00
				20				
1	136	0.00	145	0.00	141	0.00	149	0.00
2	141	0.00	141	0.00	189	0.00	189	0.00
3	117	0.00	126	0.00	158	0.00	163	0.00
4	160	0.00	165	0.00	197	0.00	219	0.00
5	111	0.00	111	0.00	107	0.00	107	0.00
Avg.	133	0.00	137.6	0.00	158	0.00	165	0.00
				30				
1	191	0.00	191	0.00	316	0.00	321	0.01
2	209	0.00	209	0.00	216	0.00	213	0.00
3	194	0.00	194	0.00	197	0.00	207	0.01
4	201	0.00	211	0.00	272	0.00	289	0.00
5	187	0.00	194	0.00	188	0.00	186	0.00
Avg.	196	0.00	199.8	0.00	238	0.00	243	0.00
				40				
1	267	0.00	270	0.00	318	0.00	334	0.01
2	229	0.00	231	0.00	238	0.00	240	0.01
3	260	0.00	268	0.00	299	0.00	307	0.01
4	278	0.00	286	0.00	401	0.00	415	0.01
5	256	0.00	256	0.00	327	0.00	332	0.01
average	258	0.00	262.2	0.00	317	0.00	326	0.01
				50				
1	310	0.00	311	0.00	443	0.01	446	0.02
2	324	0.00	324	0.00	344	0.01	337	0.01
3	285	0.00	288	0.00	358	0.01	371	0.01
4	307	0.00	307	0.00	462	0.01	464	0.02
5	309	0.00	309	0.00	402	0.01	431	0.02
Avg.	307	0.00	307.8	0.00	402	0.01	410	0.02
				60				
1	340	0.00	350	0.01	589	0.01	604	0.03
2	359	0.00	362	0.01	502	0.01	524	0.03
3	347	0.00	353	0.01	448	0.01	455	0.03
4	348	0.00	348	0.01	347	0.01	352	0.02
5	359	0.00	370	0.01	440	0.01	471	0.03
Avg.	351	0.00	356.6	0.01	465	0.01	481	0.03

Table A3. *Cont.*

Sets	JR-Resource				JR-Time			
	perm		maxβ		perm		maxβ	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
				70				
1	441	0.00	441	0.01	490	0.02	501	0.03
2	392	0.00	392	0.01	569	0.02	587	0.04
3	448	0.00	459	0.01	680	0.02	699	0.04
4	455	0.00	455	0.01	479	0.02	459	0.03
5	395	0.00	407	0.01	603	0.02	621	0.04
Avg.	426	0.00	430.8	0.01	564	0.02	573	0.04
				80				
1	505	0.00	505	0.01	550	0.03	521	0.05
2	466	0.00	473	0.01	615	0.02	639	0.06
3	478	0.00	482	0.01	817	0.02	824	0.06
4	438	0.00	441	0.01	481	0.03	446	0.04
5	449	0.00	457	0.01	591	0.03	610	0.05
Avg.	467	0.00	471.6	0.01	611	0.03	608	0.05

Table A4. ACO Permutation with 1.1 × initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
					10										**50**					
1	66	0.41	82	0.41	73	0.41	61	0.39	64	0.40	303	48.86	321	50.82	277	49.20	294	51.35	300	50.37
2	66	0.41	82	0.41	73	0.41	61	0.41	64	0.41	303	49.31	321	50.27	277	49.09	294	50.77	300	50.55
3	68	0.41	83	0.41	73	0.41	61	0.41	64	0.41	303	49.18	321	50.91	279	50.17	294	50.99	300	50.57
4	66	0.41	82	0.41	73	0.41	60	0.41	64	0.41	303	48.76	321	49.94	277	50.67	294	50.63	298	51.03
5	66	0.41	82	0.42	73	0.39	61	0.41	64	0.39	303	48.91	320	49.75	278	50.36	294	50.90	301	50.48
Avg.	66	0.41	82	0.41	73	0.40	61	0.40	64	0.40	303	49.00	321	50.34	278	49.90	294	50.93	300	50.60
					20										**60**					
1	122	2.37	134	2.32	109	2.31	155	2.34	108	2.31	325	91.75	337	92.80	339	103.47	346	97.93	349	97.29
2	123	2.34	134	2.32	109	2.32	153	2.39	108	2.33	325	92.25	337	93.09	339	100.06	346	101.74	349	98.19
3	122	2.31	134	2.30	111	2.31	154	2.36	108	2.31	325	92.80	337	101.40	339	95.92	343	98.36	349	99.48
4	122	2.30	134	2.32	109	2.31	153	2.38	108	2.31	324	93.05	337	102.11	339	96.97	346	95.87	349	98.85
5	122	2.30	134	2.33	108	2.33	155	2.38	108	2.31	325	93.06	337	101.48	339	98.08	346	95.88	349	99.10
Avg.	122	2.33	134	2.32	109	2.32	154	2.37	108	2.32	325	92.58	337	98.18	339	98.90	345	97.96	349	98.58
					30										**70**					
1	181	8.22	199	8.24	187	8.08	183	8.26	172	8.28	427	237.11	382	232.36	437	243.98	446	232.11	396	244.93
2	178	8.32	199	8.23	185	8.30	187	8.19	172	8.33	426	231.01	383	231.93	436	240.15	446	233.42	396	236.99
3	179	8.34	199	8.02	187	8.15	186	8.34	172	8.19	426	231.26	383	232.20	437	242.17	446	235.37	397	240.01
4	178	8.27	199	8.31	187	8.16	186	8.19	172	8.19	428	233.16	384	235.12	438	243.92	446	228.57	396	235.52
5	180	8.35	199	8.07	187	8.26	185	8.35	172	8.24	426	231.87	382	234.44	437	242.01	446	235.86	395	244.66
Avg.	179	8.30	199	8.18	187	8.19	185	8.27	172	8.25	427	232.88	383	233.21	437	242.45	446	233.06	396	240.42

Table A4. *Cont.*

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
					40										80					
1	234	22.94	226	22.26	237	22.21	244	21.88	255	21.37	493	262.88	459	272.06	474	280.10	429	286.30	426	290.69
2	234	22.34	225	22.15	237	22.17	244	22.18	255	21.70	494	264.76	457	268.81	476	301.40	429	282.64	426	279.41
3	234	22.39	225	22.12	237	22.22	244	21.97	255	21.79	491	269.19	459	274.39	476	289.93	429	286.31	427	279.23
4	234	22.23	226	22.42	237	23.49	244	21.46	255	21.92	491	267.44	459	274.37	476	290.52	429	286.06	426	283.96
5	234	22.20	226	22.37	237	21.27	244	21.66	255	21.58	491	261.35	460	274.82	476	287.02	429	292.30	428	288.61
Avg.	234	22.42	226	22.26	237	22.27	244	21.83	255	21.67	492	265.12	459	272.89	476	289.80	429	286.72	427	284.38

Table A5. 10/10 fraction in maxβ of ACO with 1.1 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
					10										50					
1	66	0.80	83	0.82	76	0.80	61	0.77	64	0.78	304	73.59	322	72.11	281	74.46	295	76.75	298	75.98
2	66	0.81	83	0.84	76	0.81	61	0.77	64	0.78	304	73.20	323	71.61	284	74.40	295	76.59	300	74.93
3	66	0.80	83	0.82	76	0.81	61	0.77	64	0.77	305	72.03	320	72.18	281	74.08	294	76.54	301	73.19
4	66	0.80	83	0.81	76	0.81	61	0.77	64	0.77	305	70.09	321	72.05	281	73.93	294	75.74	300	71.40
5	66	0.81	82	0.83	76	0.81	62	0.78	64	0.77	303	72.64	321	72.12	282	73.60	294	76.02	300	71.38
Avg.	66	0.80	83	0.82	76	0.81	61	0.77	64	0.77	304	72.31	321	72.02	282	74.09	294	76.33	300	73.38
					20										60					
1	125	4.45	134	4.35	109	4.52	165	4.84	108	4.34	335	133.58	342	139.84	345	137.31	346	137.39	353	138.52
2	125	4.47	137	4.33	110	4.50	159	4.92	108	4.36	335	141.68	343	140.46	343	137.01	346	140.27	353	140.43
3	124	4.56	137	4.48	111	4.58	162	4.95	108	4.38	334	139.79	341	138.67	343	140.84	346	136.81	353	142.69
4	125	4.58	135	4.49	110	4.51	162	4.84	108	4.35	333	141.93	341	138.21	345	140.37	346	136.16	352	138.97
5	124	4.44	137	4.59	110	4.56	163	4.86	108	4.34	332	143.57	341	134.99	344	139.70	346	136.96	354	139.60
Avg.	125	4.50	136	4.45	110	4.53	162	4.88	108	4.36	334	140.11	342	138.43	344	139.05	346	137.52	353	140.04
					30										70					
1	183	13.86	199	12.84	187	13.31	189	13.47	173	13.65	427	237.11	382	232.36	437	243.98	446	232.11	396	244.93
2	184	13.72	199	12.80	188	13.28	190	13.20	172	13.71	426	231.01	383	231.93	436	240.15	446	233.42	396	236.99
3	185	13.77	199	12.80	187	13.64	189	13.39	173	13.54	426	231.26	383	232.20	437	242.17	446	235.37	397	240.01
4	184	13.71	199	12.85	187	13.79	190	13.70	174	13.81	428	233.16	384	235.12	438	243.92	446	228.57	396	235.52
5	184	13.82	199	12.84	187	13.63	189	13.80	174	13.64	426	231.87	382	234.44	437	242.01	446	235.86	395	244.66
Avg.	184	13.77	199	12.83	187	13.53	189	13.51	173	13.67	427	232.88	383	233.21	437	242.45	446	233.06	396	240.42
					40										80					
1	240	38.52	226	38.32	242	36.42	249	35.61	255	36.54	492	375.78	462	349.85	477	360.96	431	352.80	432	347.36
2	238	38.52	226	38.49	241	36.89	251	35.49	255	36.70	491	370.97	461	351.44	476	354.39	432	341.28	432	343.65
3	239	37.96	226	39.25	243	37.06	250	35.57	255	36.49	491	346.62	461	353.05	476	371.67	432	346.74	432	346.67
4	237	38.03	227	38.29	241	35.97	253	35.50	255	36.04	491	346.77	462	370.16	476	355.03	432	347.73	433	343.81
5	238	39.12	226	36.22	242	35.79	249	35.57	255	36.28	492	338.27	462	349.98	477	352.28	432	350.91	432	345.74
Avg.	238	38.43	226	38.11	242	36.43	250	35.55	255	36.41	491	355.68	462	354.90	476	358.87	432	347.89	432	345.45

Table A6. 9/10 fraction in maxβ of ACO with 1.1 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time
	10										**50**									
1	66	0.86	82	0.90	75	0.88	61	0.83	64	0.85	303	77.82	321	74.82	278	80.54	294	81.34	300	75.74
2	66	0.91	82	0.91	75	0.88	61	0.83	64	0.85	303	77.56	320	71.95	279	82.37	294	80.57	300	74.94
3	64	0.89	82	0.90	76	0.88	61	0.85	64	0.85	303	76.57	320	74.97	277	80.30	294	80.34	296	75.27
4	67	0.91	82	0.91	75	0.89	62	0.83	64	0.86	303	76.99	321	77.19	279	79.90	294	80.85	300	75.89
5	64	0.89	82	0.91	74	0.89	61	0.84	64	0.85	303	77.53	321	78.59	278	76.85	294	83.66	300	75.41
Avg.	65	0.89	82	0.91	75	0.88	61	0.84	64	0.85	303	77.30	321	75.50	278	79.99	294	81.35	299	75.45
	20										**60**									
1	123	4.71	134	4.64	109	4.68	153	4.95	108	4.51	327	141.19	337	136.65	339	138.21	346	138.00	349	141.32
2	122	4.72	134	4.63	108	4.66	154	5.01	108	4.50	328	139.83	337	137.01	339	138.86	346	141.57	349	140.31
3	122	4.70	134	4.65	107	4.67	152	4.96	108	4.54	325	135.63	337	136.45	339	138.58	346	141.70	349	143.52
4	123	4.72	134	4.65	110	4.70	155	4.98	108	4.53	326	133.17	337	137.07	339	142.03	346	140.06	349	145.28
5	122	4.72	134	4.64	110	4.70	153	4.95	108	4.52	328	138.14	338	138.98	339	143.39	346	140.57	349	145.65
Avg.	122	4.71	134	4.64	109	4.68	153	4.97	108	4.52	327	137.59	337	137.23	339	140.21	346	140.38	349	143.21
	30										**70**									
1	180	14.65	199	13.82	187	14.55	183	14.10	172	14.36	426	226.28	382	221.02	435	219.65	445	221.61	394	223.12
2	179	14.66	199	14.17	187	14.45	185	14.62	172	14.47	426	222.66	382	212.25	435	220.95	446	215.60	394	217.91
3	179	14.91	199	14.14	187	14.30	185	14.25	172	14.38	426	217.61	382	213.16	435	220.95	446	217.37	394	218.23
4	178	14.69	199	13.99	187	14.70	185	14.13	172	14.42	426	218.11	382	215.98	435	219.74	445	218.73	394	218.14
5	178	15.52	199	14.08	187	14.17	184	14.47	172	14.43	426	221.69	382	215.92	435	219.88	445	215.06	394	218.20
Avg.	179	14.89	199	14.04	187	14.43	184	14.31	172	14.41	426	221.27	382	215.66	435	220.23	445	217.67	394	219.12
	40										**80**									
1	234	35.61	226	34.97	237	37.93	245	38.02	255	36.98	492	375.78	462	349.85	477	360.96	431	352.80	432	347.36
2	234	35.83	226	35.40	237	37.82	245	38.44	253	37.02	491	370.97	461	351.44	476	354.39	432	341.28	432	343.65
3	234	35.44	225	34.85	237	37.90	246	38.10	253	36.78	491	346.62	461	353.05	476	371.67	432	346.74	432	346.67
4	234	36.39	226	35.38	237	38.30	245	38.07	255	36.94	491	346.77	462	370.16	476	355.03	432	347.73	433	343.81
5	234	36.14	226	36.89	237	38.16	244	38.38	254	37.28	492	338.27	462	349.98	477	352.28	432	350.91	432	345.74
Avg.	234	35.88	226	35.50	237	38.02	245	38.20	254	37.00	491	355.68	462	354.90	476	358.87	432	347.89	432	345.45

Table A7. 8/10 fraction in maxβ of ACO with 1.1 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time
	10										**50**									
1	66	0.83	82	0.80	74	0.78	60	0.74	64	0.77	303	75.91	320	68.71	278	71.71	294	73.90	300	72.48
2	64	0.81	82	0.80	75	0.78	61	0.73	64	0.75	303	71.79	320	70.20	278	71.97	294	73.30	300	71.27
3	66	0.80	82	0.80	73	0.77	61	0.73	64	0.77	303	71.89	320	71.23	277	71.16	294	75.10	300	71.72
4	66	0.81	82	0.80	74	0.78	61	0.75	64	0.75	303	73.03	320	69.73	277	71.86	294	73.61	300	71.68
5	64	0.80	82	0.80	75	0.78	61	0.73	64	0.75	303	71.47	320	69.86	277	72.09	294	73.92	300	71.85
Avg.	65	0.81	82	0.80	74.2	0.78	61	0.74	64	0.76	303	72.82	320	69.95	277	71.76	294	73.97	300	71.80

Table A7. *Cont.*

Run	**Sets of Job**																			
	1		**2**		**3**		**4**		**5**		**1**		**2**		**3**		**4**		**5**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**
					20										**60**					
1	121	4.43	134	4.24	110	4.31	151	4.61	108	4.14	326	139.02	337	135.00	339	135.43	346	133.79	349	130.65
2	121	4.34	134	4.30	109	4.44	153	4.59	108	4.14	325	138.47	337	136.73	339	129.84	346	130.84	349	128.93
3	121	4.39	134	4.33	109	4.30	154	4.58	108	4.14	327	138.06	337	137.37	339	129.56	343	130.46	349	127.88
4	122	4.36	134	4.27	109	4.38	154	4.56	108	4.09	325	139.67	337	134.98	339	131.12	346	133.02	349	128.04
5	123	4.39	134	4.28	109	4.37	155	4.53	108	4.17	327	135.08	337	134.00	339	138.65	343	130.87	349	127.35
Avg.	122	4.38	134	4.28	109	4.36	153	4.57	108	4.14	326	138.06	337	135.62	339	132.92	345	131.80	349	128.57
					30										**70**					
1	180	14.04	199	13.41	187	13.70	184	13.45	172	13.86	426	241.06	382	221.76	435	234.29	446	231.74	394	231.07
2	178	13.89	199	13.31	187	13.64	184	13.43	172	14.02	426	250.99	382	228.99	435	238.28	446	231.08	394	246.31
3	179	14.05	198	13.30	187	13.73	184	13.71	172	14.00	426	252.71	382	230.86	435	239.93	445	221.51	394	238.70
4	180	14.11	199	13.36	187	13.56	186	13.86	172	13.94	426	232.75	381	228.95	435	239.59	445	226.95	394	239.24
5	179	14.10	199	12.98	187	13.58	184	13.51	172	13.93	426	225.05	382	248.09	435	240.17	445	235.00	394	236.33
Avg.	179	14.04	199	13.27	187	13.64	184	13.59	172	13.95	426	240.51	382	231.73	435	238.45	445	229.25	394	238.33
					40										**80**					
1	234	33.57	226	33.11	237	33.34	244	33.83	255	34.57	492	390.40	456	392.66	476	397.85	429	389.19	428	388.73
2	234	33.64	226	33.13	237	33.66	244	34.47	255	34.15	491	401.52	459	392.01	476	410.50	429	395.19	427	395.45
3	234	33.27	226	33.00	237	33.73	244	34.18	253	34.07	492	386.09	459	403.42	476	408.59	429	386.62	426	391.20
4	234	33.92	226	32.90	237	33.51	244	34.04	255	34.22	491	392.32	461	393.11	476	406.42	429	388.55	426	390.01
5	234	33.55	226	32.66	237	33.83	244	34.17	253	34.72	491	394.83	459	398.15	476	406.28	429	383.86	425	400.62
Avg.	234	33.59	226	32.96	237	33.61	244	34.14	254.2	34.35	491	393.03	459	395.87	476	405.93	429	388.68	426	393.20

Table A8. 7/10 fraction in maxβ of ACO with 1.1 $\times$ initial resource levels.

Run	**Sets of Job**																			
	1		**2**		**3**		**4**		**5**		**1**		**2**		**3**		**4**		**5**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**
					10										**50**					
1	66	0.80	82	0.81	76	0.78	61	0.75	64	0.78	303	81.13	320	76.29	279	78.45	294	80.38	300	78.42
2	64	0.82	82	0.81	73	0.80	61	0.75	64	0.77	303	81.72	319	77.53	278	76.26	294	81.02	300	78.29
3	64	0.80	82	0.83	74	0.78	61	0.78	64	0.77	303	82.78	320	75.95	278	76.19	294	81.27	300	77.83
4	66	0.81	82	0.83	74	0.80	61	0.77	64	0.77	303	80.47	319	75.51	278	76.16	294	81.51	300	77.96
5	66	0.80	82	0.80	74	0.78	60	0.77	64	0.78	303	79.49	320	74.80	277	76.11	294	81.57	300	78.58
Avg.	65	0.81	82	0.82	74.2	0.79	61	0.76	64	0.77	303	81.12	320	76.02	278	76.64	294	81.15	300	78.22
					20										**60**					
1	122	4.75	134	4.78	109	4.94	155	5.00	108	4.66	325	145.04	337	144.45	339	145.83	346	144.10	349	146.28
2	122	4.76	134	4.73	110	4.94	153	4.98	108	4.58	326	147.64	337	142.39	339	144.15	346	140.60	349	144.70
3	123	4.75	134	4.70	110	4.94	152	5.11	108	4.58	325	151.54	337	144.42	339	146.35	346	140.53	349	145.21
4	123	4.67	134	4.84	110	4.89	153	5.20	108	4.64	325	149.09	337	145.36	339	145.84	344	140.23	349	144.35
5	121	4.72	134	4.88	110	4.75	152	5.00	108	4.59	324	148.09	337	145.79	339	146.22	346	141.77	349	144.56
Avg.	122	4.73	134	4.78	110	4.89	153	5.06	108	4.61	325	148.28	337	144.48	339	145.68	346	141.44	349	145.02

Table A8. *Cont.*

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time
	30										70									
1	178	15.69	199	14.88	187	15.30	184	15.28	172	15.81	427	239.67	382	226.74	435	240.19	446	243.32	394	253.61
2	179	15.70	199	14.66	187	15.30	185	15.49	172	15.47	426	243.98	382	231.35	435	247.64	445	261.67	394	252.07
3	178	15.63	199	14.66	187	15.20	183	15.28	172	15.72	426	241.49	382	232.59	435	250.78	446	256.78	394	249.48
4	178	15.54	199	14.82	187	15.15	185	15.36	172	15.56	426	234.57	381	234.01	435	241.50	446	250.07	394	254.81
5	177	15.86	199	14.87	187	15.11	185	15.11	172	15.59	427	226.02	382	238.63	435	229.16	445	246.84	394	253.42
Avg.	178	15.68	199	14.78	187	15.21	184	15.30	172	15.63	426	237.15	382	232.66	435	241.85	446	251.74	394	252.68
	40										80									
1	234	36.13	226	40.48	237	41.07	244	40.77	255	39.08	491	352.66	457	372.15	476	364.67	429	347.80	425	350.14
2	234	41.42	226	40.43	237	41.15	244	40.58	253	39.08	491	362.31	460	359.62	476	356.04	429	353.10	426	350.78
3	234	41.59	226	40.43	237	40.66	244	40.61	255	40.77	491	357.62	457	356.03	476	362.06	429	340.50	427	352.36
4	234	41.07	226	41.04	237	40.04	244	40.48	254	41.65	492	349.59	460	355.83	476	362.85	429	341.16	426	351.01
5	234	41.35	226	40.95	237	40.07	244	38.95	255	41.56	491	350.00	459	363.36	476	358.65	429	340.25	428	352.00
Avg.	234	40.32	226	40.67	237	40.60	244	40.28	254.4	40.43	491	354.44	459	361.40	476	360.85	429	344.56	426	351.26

Table A9. 6/10 fraction in maxβ of ACO with 1.1 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time	C_{max}	Time
	10										50									
1	66	0.82	82	0.87	75	0.84	61	0.77	64	0.80	303	84.10	320	71.40	277	76.13	294	79.02	300	75.71
2	64	0.82	82	0.85	74	0.80	60	0.77	64	0.79	303	78.00	321	73.81	277	75.54	294	80.19	299	76.62
3	66	0.80	82	0.87	74	0.82	61	0.78	64	0.80	303	74.56	321	76.09	277	77.72	294	78.72	300	76.84
4	66	0.81	82	0.86	74	0.83	61	0.80	64	0.78	303	73.70	319	74.11	278	76.30	294	79.37	300	77.50
5	66	0.82	82	0.86	75	0.85	61	0.78	64	0.79	303	73.27	321	74.28	278	74.55	294	79.70	300	77.19
Avg.	66	0.82	82	0.86	74.4	0.83	61	0.78	64	0.79	303	76.73	320	73.94	277	76.05	294	79.40	300	76.77
	20										60									
1	121	4.60	134	4.50	109	4.50	153	4.85	108	4.56	327	141.96	337	142.32	339	142.07	346	138.16	349	141.01
2	121	4.66	134	4.48	107	4.60	153	4.88	108	4.56	326	141.70	337	140.15	339	145.36	346	138.41	349	138.29
3	123	4.65	134	4.55	110	4.62	153	4.83	108	4.46	326	144.69	337	137.77	339	144.49	342	138.68	349	140.56
4	122	4.59	134	4.56	107	4.60	154	4.82	108	4.47	326	148.28	337	138.50	339	144.32	346	139.33	349	141.35
5	122	4.63	134	4.50	109	4.57	154	4.87	108	4.38	326	144.58	337	138.47	339	144.73	346	137.57	349	140.42
Avg.	122	4.63	134	4.52	108	4.58	153	4.85	108	4.49	326	144.24	337	139.44	339	144.20	345	138.43	349	140.33
	30										70									
1	179	14.79	199	15.13	187	15.93	185	15.89	172	16.42	426	250.03	382	255.22	435	253.63	446	249.17	394	263.45
2	178	14.66	199	14.95	187	15.75	185	15.90	172	16.94	427	248.29	382	252.99	435	264.60	445	248.31	394	258.80
3	179	14.74	199	14.91	187	15.78	185	16.66	172	14.83	426	260.56	382	253.93	435	266.52	445	244.39	394	255.52
4	178	15.47	199	15.02	187	15.88	185	16.33	172	15.14	426	257.73	382	259.06	435	258.85	446	250.25	394	262.16
5	178	16.22	199	15.01	187	15.75	183	16.55	172	15.51	426	261.93	382	251.41	435	260.53	446	248.11	394	262.06
Avg.	178	15.18	199	15.01	187	15.82	185	16.26	172	15.77	426	255.71	382	254.52	435	260.82	446	248.05	394	260.40

Table A9. *Cont.*

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
					40										80					
1	234	40.62	226	39.70	237	42.47	244	42.91	254	43.84	491	345.15	459	363.19	476	405.04	429	371.25	427	395.82
2	234	40.83	226	40.25	237	41.81	244	43.00	255	42.47	491	349.86	459	362.35	476	401.10	429	408.69	427	400.23
3	235	40.70	224	40.67	237	42.36	244	42.46	253	43.08	491	363.31	459	380.68	476	406.06	429	396.13	427	398.01
4	234	40.81	226	40.62	237	42.19	244	42.81	254	41.81	491	355.77	459	410.13	476	407.89	429	399.04	426	400.03
5	234	40.64	226	41.40	237	42.97	245	43.13	254	39.00	491	353.87	460	400.75	476	409.60	429	396.60	426	396.07
Avg.	234	40.72	226	40.53	237	42.36	244	42.86	254	42.04	491	353.59	459	383.42	476	405.94	429	394.34	427	398.03

Table A10. 5/10 fraction in maxβ of ACO with 1.1 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
					10										50					
1	66	0.81	82	0.87	73	0.80	61	0.80	64	0.89	303	72.07	320	69.82	277	70.59	294	73.93	300	70.89
2	66	0.80	82	0.84	74	0.84	60	0.80	64	0.97	303	71.26	321	68.68	279	69.98	294	73.50	300	70.51
3	66	0.81	82	0.84	73	0.84	60	0.86	64	0.85	303	71.84	321	68.00	277	70.00	294	73.57	300	71.20
4	66	0.82	82	0.83	73	0.83	60	0.80	64	0.89	303	71.96	321	69.43	277	70.30	294	73.22	300	70.50
5	66	0.81	82	0.83	74	0.82	60	0.84	64	0.84	303	71.82	320	69.12	279	70.78	294	73.73	300	71.41
Avg.	66	0.81	82	0.85	73.4	0.83	60	0.82	64	0.89	303	71.79	321	69.01	278	70.33	294	73.59	300	70.90
					20										60					
1	122	4.50	134	4.48	109	4.71	154	4.92	108	4.42	326	129.69	337	134.31	339	128.27	346	125.95	349	127.23
2	122	4.55	134	4.51	111	4.76	153	4.94	108	4.43	326	128.17	337	128.05	339	128.40	346	125.94	349	130.87
3	122	4.51	134	4.55	108	4.86	155	4.94	108	4.52	326	131.61	337	127.31	339	128.30	346	125.48	349	130.67
4	121	4.47	134	4.57	110	4.70	153	5.12	108	4.45	325	133.03	337	126.95	339	128.40	346	125.64	349	131.79
5	122	4.49	134	4.66	108	4.56	153	4.90	108	4.66	325	132.79	337	127.01	339	128.45	346	125.79	349	132.32
Avg.	122	4.50	134	4.55	109	4.72	154	4.97	108	4.49	326	131.06	337	128.73	339	128.36	346	125.76	349	130.58
					30										70					
1	179	14.93	199	14.20	187	14.44	185	14.93	172	14.59	426	238.87	382	257.53	435	271.36	445	269.61	394	276.76
2	179	14.57	199	13.86	187	14.38	185	14.70	172	14.62	426	223.68	382	270.23	435	279.23	445	267.17	394	254.83
3	178	14.58	198	13.98	187	14.72	185	14.65	172	14.50	426	244.73	382	277.17	435	269.71	446	268.80	394	267.50
4	179	14.68	199	13.87	187	14.32	185	14.69	172	14.59	426	242.79	382	270.89	435	270.17	445	268.87	394	276.46
5	179	14.61	199	14.01	187	14.87	185	14.52	172	14.56	426	234.45	382	269.20	435	274.05	446	267.15	394	256.08
Avg.	179	14.68	199	13.98	187	14.55	185	14.70	172	14.57	426	236.90	382	269.00	435	272.90	445	268.32	394	266.33
					40										80					
1	234	35.81	224	35.41	237	35.25	244	35.76	253	36.26	491	349.43	458	347.66	476	355.28	429	336.65	427	345.07
2	234	35.10	226	35.36	237	35.29	244	35.63	253	37.32	492	348.48	460	346.87	476	356.84	429	340.46	426	347.93
3	234	35.50	226	34.47	237	35.36	244	35.61	255	35.52	493	360.13	460	345.98	476	358.21	429	341.88	426	344.11
4	234	35.01	226	34.38	237	35.33	244	35.16	255	35.23	491	348.64	459	350.96	474	352.55	429	340.05	425	344.19
5	234	36.18	226	35.07	237	35.31	244	35.82	254	36.18	491	346.19	460	346.46	476	353.20	429	342.86	427	345.99
Avg.	234	35.52	226	34.94	237	35.31	244	35.60	254	36.10	492	350.57	459	347.59	476	355.22	429	340.38	426	345.46

Table A11. ACO Permutation with 1.4 × initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
	10										**50**									
1	60	0.40	82	0.42	65	0.43	59	0.42	64	0.47	303	44.12	319	52.14	277	51.69	293	48.35	299	49.90
2	60	0.41	82	0.41	65	0.42	59	0.40	64	0.43	303	44.19	321	51.98	278	51.82	293	49.51	300	50.13
3	60	0.41	82	0.42	65	0.41	59	0.41	64	0.43	303	46.80	321	51.32	277	51.86	293	48.41	300	47.95
4	59	0.41	82	0.42	65	0.42	59	0.45	64	0.44	303	48.62	321	54.77	277	51.95	293	48.42	300	48.78
5	59	0.42	83	0.41	65	0.41	59	0.46	64	0.44	303	50.69	321	53.74	279	50.94	292	48.92	300	49.98
Avg.	60	0.41	82	0.42	65	0.42	59	0.43	64	0.44	303	46.88	321	52.79	278	51.65	293	48.72	300	49.35
	20										**60**									
1	119	2.36	132	2.37	111	2.75	154	2.43	108	2.38	325	94.56	337	94.70	339	96.27	346	98.59	349	96.67
2	119	2.44	132	2.35	111	2.38	156	2.43	108	2.39	326	93.79	337	94.69	339	95.27	346	101.80	349	96.86
3	119	2.40	132	2.33	108	2.39	152	2.44	108	2.37	325	94.10	337	95.00	339	90.30	344	99.10	349	102.86
4	119	2.70	132	2.41	110	2.42	154	2.45	108	2.37	325	94.11	337	95.32	339	97.20	346	96.52	349	96.35
5	119	2.64	132	2.62	110	2.38	154	2.44	108	2.41	324	95.23	337	95.11	339	98.55	341	98.13	349	87.34
Avg.	119	2.51	132	2.41	110	2.46	154	2.44	108	2.38	325	94.36	337	94.96	339	95.52	345	98.83	349	96.02
	30										**70**									
1	175	8.19	199	8.52	187	8.52	186	8.18	171	8.23	426	151.96	382	155.61	435	161.65	445	164.42	394	166.58
2	174	8.66	199	8.36	187	8.57	187	8.11	171	8.22	427	153.84	382	158.69	435	161.05	446	160.26	394	159.41
3	176	8.29	199	8.24	187	8.20	185	8.37	171	8.09	427	161.72	382	160.61	435	162.51	446	158.98	394	165.49
4	174	8.06	199	8.62	187	8.38	184	8.42	171	8.04	426	156.67	381	156.82	435	161.68	445	166.21	394	165.23
5	173	8.33	199	8.25	187	8.48	187	8.20	171	8.10	426	155.25	380	154.27	435	159.76	446	167.91	394	161.85
Avg.	174	8.31	199	8.40	187	8.43	186	8.26	171	8.14	426	155.89	381	157.20	435	161.33	446	163.56	394	163.72
	40										**80**									
1	234	22.11	226	21.04	237	21.19	244	21.40	252	21.31	491	267.29	459	262.07	475	257.31	429	264.95	425	257.91
2	234	22.67	226	22.56	237	21.92	244	21.51	252	21.08	491	271.28	458	261.66	474	260.63	429	271.69	425	251.82
3	234	21.59	226	22.20	237	21.26	244	21.36	252	21.49	491	266.73	459	259.88	474	262.55	429	267.61	425	269.71
4	234	21.46	226	21.55	237	21.93	244	22.19	252	20.70	491	261.10	459	258.10	474	264.23	429	269.63	425	255.35
5	234	21.87	226	21.41	237	21.49	244	21.70	252	20.79	491	267.99	458	255.27	474	270.05	429	263.06	425	259.92
Avg.	234	21.94	226	21.75	237	21.56	244	21.63	252	21.07	491	266.88	459	259.39	474	262.96	429	267.39	425	258.94

Table A12. 10/10 fraction in maxβ of ACO with 1.4 × initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
	10										**50**									
1	60	0.82	83	0.83	69	0.79	59	0.75	64	0.80	304	70.10	321	69.38	281	68.83	294	71.00	300	71.04
2	61	0.81	83	0.81	65	0.79	59	0.74	64	0.80	305	70.57	321	66.71	281	67.84	294	70.75	300	71.30
3	59	0.80	83	0.83	68	0.78	59	0.72	64	0.81	304	70.14	321	66.19	280	67.47	293	70.70	300	71.48
4	60	0.78	83	0.85	68	0.78	59	0.74	64	0.78	303	70.10	321	67.39	281	68.16	292	71.29	300	70.54
5	59	0.80	83	0.82	68	0.79	59	0.73	64	0.81	306	70.25	321	66.72	281	68.49	293	70.75	300	70.94
Avg.	60	0.80	83	0.83	68	0.79	59	0.74	64	0.80	304	70.23	321	67.28	281	68.16	293	70.90	300	71.06

Table A12. *Cont.*

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
	20										60									
1	119	4.38	134	4.40	111	4.49	160	4.71	108	4.12	331	132.65	341	133.61	343	130.06	346	125.04	351	131.37
2	119	4.53	133	4.51	111	4.48	157	4.76	108	4.01	333	132.66	339	126.78	340	127.96	346	130.12	353	130.25
3	119	4.51	134	4.34	111	4.46	162	4.72	108	4.13	333	129.97	339	127.06	345	127.95	346	128.06	354	130.49
4	119	4.62	133	4.29	111	4.45	163	4.78	108	4.05	334	128.33	338	126.99	344	127.21	346	127.61	353	131.08
5	119	4.39	134	4.38	112	4.47	162	4.78	108	4.22	335	128.70	341	127.05	342	127.76	346	126.90	351	131.90
Avg.	119	4.49	134	4.38	111	4.47	161	4.75	108	4.11	333	130.46	340	128.30	343	128.19	346	127.54	352.4	131.02
	30										70									
1	180	14.53	199	13.45	187	13.63	190	14.18	172	13.97	426	223.65	383	224.15	438	224.10	446	225.36	396	228.87
2	181	14.39	199	13.33	187	13.95	189	14.19	172	13.97	427	221.73	383	225.46	437	224.98	446	218.05	395	234.39
3	180	14.48	199	13.20	187	13.84	189	14.35	171	14.08	427	221.17	382	220.01	437	235.77	445	219.83	394	240.44
4	180	14.65	199	13.10	187	13.77	190	14.22	172	14.04	427	222.83	383	219.55	438	235.66	446	219.19	396	234.23
5	179	14.58	199	12.99	187	13.62	188	14.24	172	14.06	428	230.13	383	218.47	436	242.94	446	212.65	395	232.63
Avg.	180	14.52	199	13.21	187	13.76	189	14.23	171.8	14.02	427	223.90	383	221.53	437	232.69	446	219.01	395.2	234.11
	40										80									
1	238	34.79	226	32.29	241	33.85	248	34.13	252	34.58	491	361.10	461	365.65	476	367.03	432	356.03	432	374.91
2	237	33.80	226	32.30	240	34.02	250	34.71	252	35.05	491	351.14	460	379.02	476	384.25	432	358.12	432	361.36
3	238	34.36	226	33.10	241	33.96	250	34.79	252	35.29	492	351.12	461	365.51	476	389.75	431	361.65	431	364.59
4	239	33.40	226	32.46	241	33.98	248	34.44	252	35.94	493	352.39	461	365.27	476	375.20	430	354.07	432	362.76
5	241	33.58	226	32.41	242	33.93	249	34.47	252	35.30	492	349.35	461	358.78	476	370.01	430	359.80	428	366.10
Avg.	239	33.99	226	32.51	241	33.95	249	34.51	252	35.23	492	353.02	461	366.84	476	377.25	431	357.93	431	365.94

Table A13. 9/10 fraction in maxβ of ACO with 1.4 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
	10										50									
1	59	0.79	82	0.83	65	0.78	59	0.74	64	0.80	303	70.98	320	67.09	279	70.00	293	72.83	299	70.42
2	60	0.79	82	0.88	68	0.77	59	0.73	64	0.81	303	71.25	320	67.34	277	69.35	292	72.52	300	70.00
3	60	0.78	82	0.84	68	0.81	59	0.76	64	0.80	303	70.85	321	67.20	277	70.08	293	72.78	300	68.74
4	60	0.76	82	0.85	68	0.80	59	0.79	64	0.80	303	70.94	320	67.33	277	69.82	292	72.58	300	68.60
5	59	0.78	82	0.82	68	0.81	59	0.77	64	0.79	303	70.49	321	67.69	278	69.74	292	72.73	300	69.49
Avg.	60	0.78	82	0.85	67	0.79	59	0.76	64	0.80	303	70.90	320	67.33	278	69.80	292	72.69	300	69.45
	20										60									
1	119	4.43	132	4.53	109	4.58	152	4.81	108	4.27	327	129.64	337	130.01	339	132.17	346	125.70	349	129.38
2	119	4.60	132	4.57	110	4.57	152	4.88	108	4.42	325	129.85	337	127.65	339	131.66	346	125.47	349	130.15
3	119	4.58	132	4.58	107	4.58	155	4.85	108	4.23	328	131.25	337	130.42	339	131.22	346	127.41	349	131.58
4	119	4.47	132	4.51	110	4.62	155	4.80	108	4.16	326	130.20	337	128.24	339	130.18	346	126.64	349	130.51
5	119	4.63	132	4.48	108	4.54	155	4.84	108	4.06	327	130.41	337	129.65	339	131.22	346	125.51	349	130.05
Avg.	119	4.54	132	4.53	109	4.58	154	4.84	108	4.23	327	130.27	337	129.19	339	131.29	346	126.15	349	130.33

Table A13. *Cont.*

Run	**Sets of Job**																			
	1		**2**		**3**		**4**		**5**		**1**		**2**		**3**		**4**		**5**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**
					30										**70**					
1	175	14.43	199	13.19	187	13.89	185	14.45	171	13.45	427	244.28	382	224.02	435	226.72	445	227.96	394	226.79
2	175	14.16	199	12.87	187	13.80	184	14.65	171	13.96	427	228.50	382	223.32	435	225.61	445	227.60	394	222.79
3	177	14.13	199	12.90	187	13.97	184	14.19	171	14.14	428	223.29	382	222.66	435	227.02	445	216.39	394	221.29
4	175	14.98	199	12.96	187	14.08	185	13.91	171	13.99	426	222.01	382	219.02	435	228.56	446	215.56	394	221.98
5	173	14.69	199	13.36	187	13.87	185	14.24	171	14.31	427	222.75	382	223.12	435	236.94	445	218.18	394	223.71
Avg.	175	14.48	199	13.05	187	13.92	185	14.29	171	13.97	427	228.17	382	222.43	435	228.97	445	221.14	394	223.31
					40										**80**					
1	234	33.55	225	32.81	237	33.76	246	35.03	252	37.16	491	349.76	459	351.18	474	359.25	429	339.20	425	361.69
2	234	33.77	226	32.87	237	33.90	246	35.67	252	34.21	491	351.63	460	349.46	476	368.82	429	348.57	425	361.43
3	234	34.07	226	32.98	237	33.99	245	34.85	252	34.40	491	352.73	458	358.00	476	368.09	429	337.87	426	372.25
4	234	33.95	226	32.42	237	33.81	244	35.01	250	34.63	491	351.55	458	364.62	476	362.86	429	345.34	426	363.50
5	235	33.87	226	32.38	237	34.32	244	36.21	252	34.89	491	348.60	460	362.70	476	367.12	429	353.93	427	438.14
Avg.	234	33.84	226	32.69	237	33.95	245	35.36	252	35.06	491	350.85	459	357.19	476	365.23	429	344.98	426	379.40

Table A14. 8/10 fraction in maxβ of ACO with 1.4 $\times$ initial resource levels.

Run	**Sets of Job**																			
	1		**2**		**3**		**4**		**5**		**1**		**2**		**3**		**4**		**5**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**
					10										**50**					
1	60	0.80	82	0.92	65	0.81	59	0.75	64	0.82	303	69.44	321	65.64	277	67.94	293	73.12	300	70.41
2	58	0.79	82	0.92	65	0.83	59	0.78	64	0.80	303	69.44	321	66.30	277	69.99	293	73.27	300	70.45
3	59	0.80	82	0.91	67	0.84	59	0.76	64	0.79	303	68.72	321	65.93	279	69.78	293	73.30	300	71.13
4	60	0.87	82	0.93	66	0.82	59	0.75	64	0.79	303	69.31	321	65.67	279	69.48	292	73.22	298	72.03
5	60	0.85	82	0.88	65	0.80	59	0.74	64	0.79	303	68.95	321	65.64	278	69.60	293	73.21	300	70.32
Avg.	59	0.82	82	0.91	66	0.82	59	0.76	64	0.80	303	69.17	321	65.84	278	69.36	293	73.22	300	70.87
					20										**60**					
1	119	4.37	132	4.36	109	4.37	154	4.97	108	4.06	326	129.48	337	128.02	339	132.21	346	127.52	349	131.45
2	119	4.39	132	4.56	111	4.39	155	5.05	108	4.08	326	130.63	337	132.50	339	135.24	346	132.26	349	131.68
3	119	4.40	132	4.60	110	4.60	156	4.96	108	4.23	328	128.73	337	131.23	339	133.96	346	128.72	349	131.43
4	119	4.27	132	4.54	110	4.64	153	4.82	108	4.24	326	130.76	337	130.80	339	132.43	346	127.90	349	131.03
5	119	4.33	132	4.45	110	4.70	153	4.72	108	4.21	326	130.31	337	127.97	339	131.18	343	127.68	349	132.76
Avg.	119	4.35	132	4.50	110	4.54	154	4.90	108	4.16	326	129.98	337	130.10	339	133.01	345	128.82	349	131.67
					30										**70**					
1	174	14.53	198	13.46	187	13.70	184	14.11	171	13.83	426	231.32	382	229.68	435	237.95	445	225.90	394	235.27
2	174	14.44	199	13.21	187	14.41	185	14.15	171	13.76	426	232.78	382	226.54	435	237.43	445	231.07	394	235.68
3	174	14.69	199	13.29	185	13.84	185	14.29	171	14.22	427	226.33	382	231.25	435	228.80	446	224.58	394	233.44
4	175	14.55	198	12.96	187	14.01	184	14.24	171	14.79	426	228.31	381	228.21	435	235.13	445	224.31	394	233.68
5	175	14.10	199	13.03	187	13.75	185	14.10	171	14.16	427	229.95	382	230.07	435	232.42	445	222.44	394	240.95
Avg.	174	14.46	199	13.19	187	13.94	185	14.18	171	14.15	426	229.74	382	229.15	435	234.35	445	225.66	394	235.81

Table A14. *Cont.*

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
	40										80									
1	234	33.74	226	32.75	237	34.00	244	34.79	252	35.40	491	372.89	459	371.61	476	378.11	429	431.14	427	367.77
2	234	35.39	226	33.79	237	34.33	244	34.56	252	35.25	491	385.00	457	376.62	476	386.75	429	437.42	425	355.19
3	234	34.76	226	32.94	237	33.78	244	35.11	252	35.64	491	367.65	459	379.97	476	380.16	429	433.80	425	366.88
4	234	34.00	226	32.51	237	33.50	244	35.07	252	35.27	491	369.67	459	371.16	474	378.50	429	375.88	424	375.39
5	234	34.47	226	32.77	237	33.62	244	35.29	252	34.29	491	366.16	458	369.34	476	373.08	429	365.21	425	364.34
Avg.	234	34.47	226	32.95	237	33.85	244	34.96	252	35.17	491	372.27	458	373.74	476	379.32	429	408.69	425	365.92

Table A15. 7/10 fraction in maxβ of ACO with 1.4 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
	10										50									
1	60	0.76	82	0.84	65	0.76	59	0.72	64	0.79	303	80.16	321	69.24	277	70.24	292	74.02	300	71.51
2	60	0.78	82	0.81	65	0.77	59	0.74	64	0.82	303	73.05	321	65.99	277	70.62	293	73.80	300	71.51
3	60	0.80	82	0.81	65	0.84	59	0.75	64	0.83	303	72.41	321	66.72	279	70.26	292	74.34	300	72.53
4	61	0.79	82	0.81	65	0.83	59	0.75	64	0.87	302	73.04	321	67.55	277	70.30	293	74.22	300	71.41
5	60	0.77	82	0.86	65	0.78	59	0.73	64	0.83	303	71.77	321	67.53	277	71.02	292	73.75	300	71.42
Avg.	60	0.78	82	0.83	65	0.80	59	0.74	64	0.83	303	74.09	321	67.41	277	70.49	292	74.03	300	71.68
	20										60									
1	119	4.39	132	4.46	108	4.78	153	4.89	108	4.26	326	128.65	337	129.66	339	134.27	346	130.59	349	128.63
2	119	4.26	132	4.51	110	4.65	152	4.76	108	4.38	326	128.75	337	127.03	339	133.06	346	127.68	349	129.51
3	119	4.32	132	4.59	109	4.71	153	4.80	108	4.29	325	128.42	337	128.60	339	135.36	346	127.38	349	130.51
4	119	4.27	132	4.48	109	5.00	154	4.79	108	4.48	324	129.78	337	128.45	339	136.13	346	128.11	349	132.49
5	119	4.67	132	4.49	109	4.83	156	4.76	108	4.26	326	129.25	337	132.78	339	133.66	346	126.74	349	132.60
Avg.	119	4.38	132	4.51	109	4.79	154	4.80	108	4.33	325	128.97	337	129.30	339	134.50	346	128.10	349	130.75
	30										70									
1	174	14.30	199	13.06	187	13.64	186	13.73	171	13.66	427	239.39	382	235.44	435	246.05	445	236.55	394	255.11
2	177	14.71	199	13.04	187	13.63	184	13.88	171	14.31	426	242.99	382	234.77	435	238.37	445	243.76	394	245.57
3	174	14.45	199	12.97	187	13.72	185	13.93	171	13.86	426	240.07	382	232.38	435	253.19	446	241.68	394	244.85
4	174	14.69	199	13.12	187	13.55	183	14.08	171	13.99	428	236.13	382	238.29	435	251.76	445	237.30	394	248.74
5	174	14.42	199	13.07	185	13.60	185	13.97	171	13.73	427	237.71	382	239.42	435	256.39	445	243.75	394	243.09
Avg.	175	14.52	199	13.05	187	13.63	185	13.92	171	13.91	427	239.26	382	236.06	435	249.15	445	240.61	394	247.47
	40										80									
1	234	34.18	226	32.34	237	33.47	244	34.39	252	34.75	491	355.44	459	387.20	476	367.73	429	348.69	425	361.64
2	234	34.21	226	32.27	237	33.84	244	34.22	252	34.78	491	358.90	458	452.33	476	357.15	429	352.06	425	364.45
3	234	33.97	225	32.37	237	33.51	244	34.47	252	34.77	491	352.64	460	446.64	476	365.63	429	341.14	425	367.49
4	234	34.10	226	32.39	237	33.53	244	34.46	252	34.69	491	354.75	460	411.01	473	375.85	429	343.29	425	358.24
5	234	34.00	225	32.48	237	33.61	244	34.21	252	34.71	491	350.60	459	371.80	475	370.45	429	349.52	425	366.38
Avg.	234	34.09	226	32.37	237	33.59	244	34.35	252	34.74	491	354.47	459	413.79	475	367.36	429	346.94	425	363.64

Table A16. 6/10 fraction in maxβ of ACO with 1.4 $\times$ initial resource levels.

Run	**Sets of Job**																			
	1		**2**		**3**		**4**		**5**		**1**		**2**		**3**		**4**		**5**	
	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**	C_{max}	**Time**
						10										**50**				
1	59	0.77	82	0.84	65	0.77	59	0.71	64	0.76	303	70.78	320	67.86	278	69.21	293	88.11	300	77.73
2	60	0.75	82	0.85	65	0.76	59	0.73	64	0.80	303	70.28	321	69.51	277	69.30	292	87.96	298	77.74
3	60	0.75	82	0.85	65	0.77	59	0.74	64	0.79	303	70.48	320	68.20	277	69.85	292	92.54	300	75.20
4	59	0.78	82	0.85	65	0.76	59	0.80	64	0.78	303	71.06	319	67.40	277	69.75	293	90.99	300	74.56
5	60	0.77	82	0.85	65	0.78	59	0.78	64	0.80	303	71.99	320	69.42	277	75.94	293	83.33	300	77.39
Avg.	60	0.77	82	0.85	65	0.77	59	0.75	64	0.79	303	70.92	320	68.48	277	70.81	293	88.59	300	76.52
						20										**60**				
1	119	4.49	132	4.56	109	4.53	153	4.75	108	4.29	327	129.50	337	129.90	339	131.28	346	128.13	349	132.61
2	119	4.46	132	4.45	108	4.54	154	5.02	108	4.33	326	131.09	337	129.46	339	130.31	346	127.91	349	133.17
3	119	4.33	132	4.51	109	4.44	152	4.72	108	4.27	326	131.82	337	131.52	339	132.01	346	128.22	349	132.52
4	119	4.33	132	4.50	109	4.40	155	4.67	108	4.28	324	131.20	337	128.93	339	132.86	346	129.25	349	132.61
5	119	4.52	132	4.61	110	4.44	152	5.05	108	4.31	325	129.73	337	129.75	339	133.50	346	130.48	349	133.23
Avg.	119	4.43	132	4.53	109	4.47	153	4.84	108	4.29	326	130.67	337	129.91	339	131.99	346	128.80	349	132.83
						30										**70**				
1	173	14.08	199	12.86	187	13.64	185	13.96	171	14.39	426	215.26	381	218.51	435	216.39	446	198.86	394	206.93
2	176	14.11	199	12.90	187	13.64	185	14.32	171	14.08	426	215.64	382	207.97	435	208.71	445	197.88	394	207.29
3	175	14.23	199	12.98	187	13.72	185	14.33	171	13.91	427	208.16	382	212.68	435	212.47	446	197.33	394	206.72
4	173	14.35	199	13.02	187	13.68	185	14.70	171	14.08	426	226.05	382	202.80	435	209.57	445	197.59	394	207.20
5	174	14.38	199	12.88	187	13.64	185	14.43	171	13.96	426	230.31	382	216.97	435	205.56	445	198.29	394	208.30
Avg.	174	14.23	199	12.93	187	13.66	185	14.35	171	14.08	426	219.08	382	211.79	435	210.54	445	197.99	394	207.29
						40										**80**				
1	234	34.21	226	32.54	237	33.76	244	34.32	252	34.62	491	348.91	460	365.14	474	376.18	429	345.95	425	366.90
2	234	34.15	226	32.60	237	33.57	244	34.47	252	34.72	491	356.95	460	366.77	475	368.62	429	343.70	425	364.87
3	234	34.26	226	32.66	237	33.57	244	34.58	252	34.80	493	355.42	459	357.71	476	362.79	429	345.14	426	360.93
4	234	33.85	226	32.45	237	33.87	244	34.23	252	34.14	491	361.16	458	365.88	476	375.33	429	344.93	425	358.11
5	234	34.10	226	32.57	237	33.72	244	34.49	252	33.89	491	355.63	456	366.90	475	369.19	429	346.99	425	357.00
Avg.	234	34.11	226	32.57	237	33.70	244	34.42	252	34.43	491	355.61	459	364.48	475	370.42	429	345.34	425	361.56

Table A17. 5/10 fraction in maxβ of ACO with 1.4 $\times$ initial resource levels.

Run	Sets of Job																			
	1		2		3		4		5		1		2		3		4		5	
	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time	$C_{\max}$	Time
					10										**50**					
1	58	0.80	82	0.80	65	0.75	59	0.72	64	0.77	303	70.50	319	68.01	279	69.68	292	74.77	300	71.71
2	60	0.79	82	0.82	65	0.74	59	0.71	64	0.78	303	70.92	321	66.38	277	69.63	292	74.27	300	72.12
3	60	0.79	82	0.83	65	0.76	59	0.70	64	0.77	303	70.80	320	65.30	278	69.15	293	74.18	300	71.46
4	60	0.79	82	0.82	65	0.76	59	0.70	64	0.78	303	70.82	320	65.58	277	70.80	292	73.60	298	70.41
5	60	0.76	82	0.82	65	0.76	59	0.70	64	0.77	302	71.10	320	65.66	279	70.27	292	74.84	300	70.52
Avg.	60	0.79	82	0.82	65	0.75	59	0.70	64	0.77	303	70.83	320	66.19	278	69.91	292	74.33	300	71.24
					20										**60**					
1	119	4.27	132	4.40	110	4.68	155	4.87	108	4.26	324	134.13	337	131.51	339	134.55	346	129.39	349	135.28
2	119	4.53	132	4.63	110	4.62	154	4.90	108	4.19	325	132.49	337	130.62	339	134.62	346	128.56	349	133.57
3	119	4.48	132	4.51	110	4.62	152	4.84	108	4.23	326	133.00	337	131.47	339	130.36	346	127.61	349	134.02
4	119	4.44	132	4.49	109	4.63	154	4.86	108	4.12	325	136.22	337	131.36	339	129.65	344	128.76	349	134.32
5	119	4.34	132	4.47	111	4.67	155	4.86	108	4.15	325	138.10	337	130.66	339	131.27	346	132.26	349	131.01
Avg.	119	4.41	132	4.50	110	4.64	154	4.87	108	4.19	325	134.79	337	131.12	339	132.09	346	129.31	349	133.64
					30										**70**					
1	176	14.57	199	13.36	187	13.96	185	14.06	171	13.99	426	213.89	382	210.60	435	212.37	446	203.49	394	219.23
2	175	14.69	199	13.09	187	13.81	185	14.10	171	14.01	426	212.78	382	211.25	435	213.80	446	204.17	394	228.61
3	175	14.95	199	13.29	187	13.72	186	14.26	171	14.13	427	212.19	382	211.88	435	214.04	446	208.09	394	239.13
4	173	14.50	199	13.33	187	13.84	185	14.12	171	13.87	426	211.30	382	208.96	435	212.23	446	208.38	394	230.80
5	174	14.70	199	13.21	187	14.10	185	14.25	171	13.75	427	215.19	382	209.14	435	212.17	445	208.73	394	222.70
Avg.	175	14.68	199	13.26	187	13.89	185	14.16	171	13.95	426	213.07	382	210.37	435	212.92	446	206.57	394	228.09
					40										**80**					
1	234	34.42	225	32.72	237	33.05	244	34.08	252	34.27	492	353.73	458	366.59	476	367.47	429	341.96	425	354.92
2	234	34.92	226	31.71	237	32.88	244	33.96	252	34.55	491	353.13	460	361.57	474	371.96	429	345.05	425	356.15
3	234	34.76	226	31.76	237	32.93	244	34.09	251	34.17	491	352.95	459	362.61	476	366.73	429	349.67	425	359.46
4	234	34.30	226	32.02	237	33.52	244	33.71	252	33.99	491	373.47	459	353.15	476	376.03	429	347.99	425	370.89
5	234	34.11	226	31.76	237	33.01	244	34.12	252	34.05	491	370.20	459	355.60	476	368.67	429	343.99	425	365.23
Avg.	234	34.50	226	31.99	237	33.08	244	33.99	252	34.21	491	360.70	459	359.90	476	370.17	429	345.73	425	361.33

Table A18. Experiment results of ACO M2-enum.

Run	Sets of Job																			
	1		**2**		**3**		**4**		**5**		**1**		**2**		**3**		**4**		**5**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**	$C_{\max}$	**Time**
	10 jobs with 1.1 $\times$ initial resource										**10 jobs with 1.4 $\times$ initial resource**									
1	64	0.86	82	0.84	73	0.83	61	0.84	64	0.83	59	0.87	82	0.87	65	0.86	59	1.12	64	0.86
2	66	0.84	82	0.83	73	0.83	60	0.86	64	0.84	59	0.85	82	0.87	65	0.87	59	1.22	64	0.87
3	66	0.84	82	0.84	73	0.83	60	0.86	64	0.83	60	0.88	82	0.87	65	0.86	59	1.16	64	0.86
4	66	0.81	82	0.86	73	0.84	61	0.86	64	0.83	60	0.92	82	0.87	65	0.88	59	1.17	64	0.86
5	64	0.83	82	0.86	74	0.84	60	0.86	64	0.83	59	0.86	82	0.86	65	0.85	59	1.20	64	0.86
Avg.	65	0.84	82	0.85	73	0.83	60	0.86	64	0.83	59	0.87	82	0.87	65	0.86	59	1.17	64	0.86
	20 jobs with 1.1 $\times$ initial resource										**20 jobs with 1.4 $\times$ initial resource**									
1	122	8.27	134	17.43	109	7.01	154	5.14	108	106.89	119	14.59	132	35.62	109	6.65	153	5.09	108	1864.11
2	121	8.00	134	17.74	110	6.98	153	5.19	108	105.95	119	13.13	132	37.49	107	6.65	153	5.03	108	2062.52
3	121	8.33	134	18.44	108	6.85	155	5.34	108	96.28	119	15.01	132	39.95	108	6.69	154	5.08	108	2313.29
4	122	8.39	134	18.19	110	6.91	153	5.22	108	92.69	119	14.46	132	41.77	107	6.70	153	5.03	108	2376.40
5	121	8.42	134	18.19	109	6.96	155	5.26	108	92.67	119	14.22	132	40.31	110	6.66	153	5.05	108	2820.85
Avg.	121	8.28	134	18.00	109	6.94	154	5.23	108	98.90	119	14.28	132	39.03	108	6.67	153	5.05	108	2287.43
	30 jobs with 1.1 $\times$ initial resource										**30 jobs with 1.4 $\times$ initial resource**									
1	179	1363.26	200	-	187	-	183	2379.65	172	-	174	1450.22	200	-	187	-	185	2586.26	172	-
2	177	1350.69	200	-	187	-	184	3020.92	172	-	174	1056.23	200	-	187	-	185	2700.09	172	-
3	178	1649.96	200	-	187	-	184	2696.27	172	-	173	1699.81	200	-	187	-	184	2049.86	172	-
4	177	1282.78	200	-	187	-	184	-	172	-	175	1441.64	200	-	187	-	185	2385.19	172	-
5	180	1084.22	200	-	187	-	184	-	172	-	175	1218.19	200	-	187	-	185	3604.12	172	-
Avg.	178	1346.18	200	3600.00	187	3600.00	184	3059.37	172	3600.00	174	1373.22	200	3600.00	187	3600.00	185	2665.10	172	3600.00

Table A19. Experiment results of IP.

IP	Times of Initial Resource			
Sets	**1.1**		**1.4**	
	$C_{\max}$	**Time**	$C_{\max}$	**Time**
		10		
1	64	1518.71	56	344.81
2	81	59.93	81	62.71
3	73	110.96	65	23.72
4	59	74.42	59	15.25
5	64	470.02	64	492.10
Avg.	68	446.81	65	187.72
		20		
1	123	3600.11	119	2864.56
2	-	3600.00	132	1233.87
3	-	3600.27	-	3600.20
4	-	3600.32	x	3600.61
5	-	3600.47	107	366.97
Avg.	-	-	-	-

References

1. Pinedo, M. *Scheduling*; Springer: New York, NY, USA, 2016; Volume 29.
2. Brucker, P.; Drexl, A.; Möhring, R.; Neumann, K.; Pesch, E. Resource-constrained project scheduling: Notation, classification, models, and methods. *Eur. J. Oper. Res.* **1999**, *112*, 3–41. [CrossRef]
3. Habibi, F.; Barzinpour, F.; Sadjadi, S. Resource-constrained project scheduling problem: Review of past and recent developments. *J. Proj. Manag.* **2018**, *3*, 55–88. [CrossRef]
4. Herroelen, W.; Reyck, B.D.; Demeulemeester, E. Resource-constrained project scheduling: A survey of recent developments. *Comput. Oper. Res.* **1998**, *25*, 279–302. [CrossRef]
5. Issa, S.; Tu, Y. A survey in the resource-constrained project and multi-project scheduling problems. *J. Proj. Manag.* **2020**, *5*, 117–138. [CrossRef]
6. Ibadov, N.; Kulejewski, J.; Krzemiński, M. Fuzzy ordering of the factors affecting the implementation of construction projects in Poland. *AIP Conf. Proc.* **2013**, *1558*, 1298–1301.
7. Krzemiński, M. KASS v. 2.2. scheduling software for construction with optimization criteria description. *Acta Phys. Pol. A* **2016**, *130*, 1439–1442. [CrossRef]
8. Kaplan, E.H. Relocation models for public housing redevelopment programs. *Environ. Plan. B Plan. Des.* **1986**, *13*, 5–19. [CrossRef]
9. PHRG. *New Lives for Old Buildings: Revitalizing Public Housing Project*; Public Housing Group, Department of Urban Studies and Planning, MIT: Cambridge, MA, USA, 1986.
10. Kaplan, E.H.; Amir, A. A fast feasibility test for relocation problems. *Eur. J. Oper. Res.* **1988**, *35*, 201–206. [CrossRef]
11. Johnson, S.M. Optimal two- and three-stage production schedules with setup times included. *Nav. Res. Logist. Q.* **1954**, *1*, 61–68. [CrossRef]
12. Lin, B.M.T.; Huang, H.L. On the relocation problem with a second working crew for resource recycling. *Int. J. Syst. Sci.* **2006**, *37*, 27–34. [CrossRef]
13. Cheng, T.C.E.; Lin, B.M.T.; Huang, H.L. Resource-constrained flowshop scheduling with separate resource recycling operations. *Comput. Oper. Res.* **2012**, *39*, 1206–1212. [CrossRef]
14. Graham, R.L.; Lawler, E.L.; Lenstra, J.K.; Kan, A.R. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discret. Math.* **1979**, *5*, 287–326.
15. Kaplan, E.H.; Berman, O. OR hits the heights: Relocation planning at the orient heights housing project. *Interfaces* **1988**, *18*, 14–22. [CrossRef]
16. Xie, J. Polynomial algorithms for single machine scheduling problems with financial constraints. *Oper. Res. Lett.* **1997**, *21*, 39–42. [CrossRef]
17. Amir, A.; Kaplan, E.H. Relocation problems are hard. *Int. J. Comput. Math.* **1988**, *25*, 101–110. [CrossRef]
18. Kononov, A.V.; Lin, B.M.T. On relocation problems with multiple identical working crews. *Discret. Optim.* **2006**, *3*, 366–381. [CrossRef]

19. Cheng, T.C.E.; Lin, B.M.T. Johnson's rule, composite jobs and the relocation problem. *Eur. J. Oper. Res.* **2009**, *192*, 1008–1013. [CrossRef]
20. Cheng, T.C.E.; Lin, B.M.T. Demonstrating Johnson's algorithm via resource-constrained scheduling. *Int. J. Prod. Res.* **2017**, *55*, 3326–3330. [CrossRef]
21. Lin, B.M.T.; Tseng, S.S. Some results of the relocation problems with processing times and deadlines. *Int. J. Comput. Math.* **1991**, *40*, 1–15. [CrossRef]
22. Lin, B.M.T.; Tseng, S.S. Relocation problems of maximizing new capacities under a common due date. *Int. J. Syst. Sci.* **1992**, *23*, 1433–1448. [CrossRef]
23. Lin, B.M.T.; Tseng, S.S. Resource-requirement minimization in relocation problems with precedence constraints. In Proceedings of the ICCI '92: Fourth International Conference on Computing and Information, Toronto, ON, Canada, 28–30 May 1992; pp. 26–29.
24. Lin, B.M.T.; Cheng, T.C.E. Minimizing the weighted number of tardy jobs and maximum tardiness in relocation problem. *Eur. J. Oper. Res.* **1999**, *116*, 183–193. [CrossRef]
25. Hall, N.G. Scheduling problems with generalized due dates. *IIE Trans.* **1986**, *18*, 220–222. [CrossRef]
26. Lin, B.M.T.; Liu, S.T. Maximizing the reward in relocation problems with generalized due dates. *Int. J. Prod. Econ.* **2008**, *115*, 55–63. [CrossRef]
27. Sevastyanov, S.V.; Lin, B.M.T.; Huang, H.-L. Tight complexity analysis of the relocation problem with arbitrary release dates. *Theor. Comput. Sci.* **2011**, *412*, 4536–4544. [CrossRef]
28. Kononov, A.V.; Lin, B.M.T. Minimizing the total weighted completion time in the relocation problem. *J. Sched.* **2010**, *13*, 123–129. [CrossRef]
29. Lin, B.M.T. Resource-constrained scheduling with optional recycling operations. *Comput. Ind. Eng.* **2015**, *90*, 39–45. [CrossRef]
30. Shafransky, Y.M.; Strusevich, V.A. The open shop scheduling problem with a given sequence of jobs on one machine. *Nav. Res. Logist.* **1998** *45*, 705–773 . [CrossRef]
31. Lin, B.M.T.; Hwang, F.J.; Kononov, A.V. Relocation scheduling subject to fixed processing sequences. *J. Sched.* **2016**, *19*, 153–163. [CrossRef]