International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

Job shop scheduling with the option of jobs outsourcing

Hamid Safarzadeh & Farhad Kianfar

To cite this article: Hamid Safarzadeh & Farhad Kianfar (2019): Job shop scheduling with the option of jobs outsourcing, International Journal of Production Research, DOI: 10.1080/00207543.2019.1579934

To link to this article: https://doi.org/10.1080/00207543.2019.1579934

Published online: 01 Apr 2019.

International Journal of Production Research, 2019
https://doi.org/10.1080/00207543.2019.1579934

Taylor & Francis
Taylor & Francis Group

Check for updates

Job shop scheduling with the option of jobs outsourcing

Hamid Safarzadeh and Farhad Kianfar*

Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran

(*Received 13 May 2018; accepted 25 January 2019*)

Incorporating outsourcing in scheduling is addressed by several researchers recently. However, this scope is not investigated thoroughly, particularly in the job shop environment. In this paper, a new job shop scheduling problem is studied with the option of jobs outsourcing. The problem objective is to minimise a weighted sum of makespan and total outsourcing cost. With the aim of solving this problem optimally, two solution approaches of combinatorial optimisation problems, *i.e.* mathematical programming and constraint programming are examined. Furthermore, two problem relaxation approaches are developed to obtain strong lower bounds for some large scale problems for which the optimality is not proven by the applied solution techniques. Using extensive numerical experiments, the performance of the solution approaches is evaluated. Moreover, the effect the objectives's weights in the objective function on the performance of the solution approaches is also investigated. It is concluded that constraint programming outperforms mathematical programming significantly in proving solution optimality, as it can solve small and medium size problems optimally. Moreover, by solving the relaxed problems, one can obtain good lower bounds for optimal solutions even in some large scale problems.

Keywords: job shop scheduling; outsourcing; subcontracting; mathematical programming; constraint programming; relaxation; lower bounds

1. Introduction

Outsourcing is a common practice in manufacturing plants due to the existing constraints in the shop floors and the advantages of the external firms. By benefiting subcontractors' capabilities, this strategy can improve the performance of the manufacturing systems in various ways. It can be useful in the reduction of production bottlenecks and compensation of the disruption effects. Moreover, outsourcing can decrease the manufacturing costs as the subcontractors may process some operations with lower costs than the manufacturer. This strategy can also lead to focus on the core value added processes via subcontracting the subsidiary tasks (Chen and Li 2008).

Outsourcing can be associated to different aspects of factory management. One of the key planning issues which can be influenced by operational outsourcing is scheduling. This topic generally addresses the assignment of resources to activities over the time (Pinedo 2012). In the scheduling problems, there are normally various constraints to be satisfied such as activity precedence relationships, limitedness of resources, or due dates for the jobs. On the other hand, one or more objectives should be optimised like the resources utilisation, system throughput, and/or total production cost. Accordingly, it is necessary to develop an effective schedule in order to meet this variety of criteria. In the presence of outsourcing, the subcontractors can be regarded as extra resources to fulfil the jobs over the time. This option may provide a worthwhile possibility to enhance the effectiveness of the schedule in satisfying the problem constraints and improving the considered objectives.

There exist several studies in the literature that consider outsourcing and scheduling simultaneously. Especially, this subject has attracted more attention in recent years, so that almost all of the related studies are conducted in the past decade. A detailed review of the related literature is presented in the next section. It will be discussed that scheduling with outsourcing is not studied thoroughly in the literature. Particularly, job shop environment is rarely addressed by the researchers, as to the best of our knowledge, only three papers have investigated outsourcing for this machine environment. Job shop scheduling problem (JSSP) is one of the most complicated scheduling problems with various industrial applications (Çaliş and Bulkan 2015). In this problem, there are several workstations from which the available *jobs* are passed to accomplish their required works, *i.e. operations*. The jobs don't need to have similar process plans. Hence, this model is useful for shops with a high level of variety in the product types, which is a prevalent case in today's industry. To incorporate outsourcing in this problem, like the other multi-operation scheduling problems, two general types of outsourcing can be considered: 1)

---

*Corresponding author. Email: fkianfar@sharif.edu

jobs outsourcing, 2) operations outsourcing (Choi and Park 2014). In the former, a job is subcontracted as a whole, whereas in the latter, the operations are subject of outsourcing so that each operation can be either processed in-house or assigned to one of the subcontractors. As will be discussed in the next section, in the literature of scheduling with the outsourcing option, several researchers like Choi and Chung (2011), Mokhtari, Abadi, and Amin-Naseri (2012), Chung and Choi (2013), Tavares Neto and Godinho Filho (2011), Choi and Park (2014), Guo and Lei (2014) and Lei and Guo (2016) have considered jobs outsourcing. However, operations outsourcing is also studied by some researchers such as Qi (2009), Qi (2011), Mokhtari, Abadi, and Amin-Naseri (2012) and Chung et al. (2005).

In this paper, a job shop scheduling problem with the option of jobs outsourcing is studied. A solution of this problem is specified by determining the subcontracted jobs and scheduling the other ones in the manufacturer's shop. Makespan and total outsourcing cost are the considered objectives for this problem, which are minimised together as a weighted sum objective function. To the best of our knowledge, it is the first time that this problem is being studied in the literature. To solve the problem, we use two well-known exact solution approaches of combinatorial optimisation problems, *i.e.* mathematical programming and constraint programming (CP). For the former, a mixed integer linear programming (MILP) model of the problem is developed based on disjunctive formulation, and for the latter, a model is proposed using ILOG CP elements of scheduling. These developed models are examined numerically in this research for solving the problem optimally. Moreover, in order to obtain some strong lower bounds for the problem when it is not solved to optimality, two efficient relaxation approaches are developed. To solve the relaxed problems, the related MILP and CP models are also presented. The relaxed problems comparing to the original problem can be solved for larger problem scales, and when they are solved, some strong lower bounds are obtained for the original problem. These lower bounds are useful to approximate the optimal solution of the original problem as well as to estimate the absolute or relative deviations of the obtained feasible solutions from optimality. Comprehensive numerical experiments are conducted to evaluate the efficiency of the solution methods. The problem scales for which the solution approaches are applicable are examined. Meanwhile, the effect of how to weight the objectives in the objective function is also analysed on the performance of the solution methods.

In short, the major contributions of this research are summarised as follows:

- The research gap for considering outsourcing in JSSP is reduced by analysing a new problem.
- In order to solve the problem optimally, new MILP and CP models are developed and examined through extensive numerical experiments..
- CP approach is used for the first time in the context of scheduling with the outsourcing option.
- Two innovative relaxation approaches are presented to estimate optimal solutions for larger problem dimensions.
- The effect of the weights of the objectives in the objective function is examined thoroughly on the problem solving approaches.

The remainder of the paper proceeds as follows: In Section 2, a literature review of scheduling with outsourcing is presented. In Section 3, the considered problem is defined formally. Section 4 is devoted to formulating the proposed MILP model, and in Section 5, the developed CP model is presented. In Section 6, the proposed relaxation approaches are explained, and the related MILP and CP models are presented. In Section 7, comprehensive computational experiments are conducted to examine the efficiency of the developed solution methods. Finally, the paper is concluded in Section 8, and some directions for future studies are proposed.

2. Literature review

The research of outsourcing in scheduling is in its infancy; however, there is an increasing trend to consider this subject in the literature. The papers reviewed in the following include most of the researches we have seen in the literature for this scope. These papers can be classified in respect of the machine environment type. Accordingly, the related literature is reviewed in the following with regard to the major environment types in scheduling which are single machine, parallel machine, flow shop, and job shop. Meanwhile, the flow shop environment is considered in turn as two-stage and multi-stage cases. After reviewing the literature, some aspects of our research are discussed regarding the reviewed literature.

In the single machine environment, Lee and Sung (2008a, 2008b) investigated two problems with the possibility of jobs outsourcing. The problem considered in the former paper aims to minimise a weighted sum of total completion time and total outsourcing cost regarding a budget limit. However, total completion time is substituted by total tardiness or maximum lateness of jobs in the problem of the latter paper. In both papers, the authors use heuristic and branch and bound algorithms to solve the problems. Qi (2008) considered batch outsourcing in a single machine scheduling problem to mitigate the transportation effects of subcontracting. Dynamic programming techniques are used to analyse the problem with several objective functions, and some general results and optimal algorithms are derived for the problems. Zhong and Huo (2013) studied a problem of single machine scheduling with the option of outsourcing. The subcontractors process each job with a different cost from the in-house processing case, and there exists a delivery time as a stepwise function of outsourcing time for an outsourced job. The problem objective is a weighted sum of total processing cost and either makespan or the number of tardy jobs. The authors have proven NP-hardness of the problem. Moreover, for the case of considering makespan objective, they have proposed a pseudo-polynomial algorithm, while for the other case, proposed such an algorithm for a special case of the problem. Hong and Lee (2016) addressed a single machine scheduling problem with subcontractor selection. In this problem, total outsourcing cost is minimised considering due dates for the jobs and capacity limits for the subcontractors. Some characteristics of the optimal solution are presented, and a pseudo-polynomial algorithm is proposed to optimise the problem. The superiority of the algorithm is demonstrated to the dynamic programming approach via numerical experiments.

Chen and Li (2008) considered an identical parallel machine scheduling problem with the option of outsourcing. The time or cost of processing a job depends on the job performer *i.e.* manufacturer or each of the subcontractors. The problem objective is to minimise total cost while the maximum completion time of the jobs is limited by an upper bound. The authors proved that the problem is NP-hard, and proposed a heuristic algorithm to solve it. Mokhtari and Abadi (2013) studied a problem of parallel machine scheduling with outsourcing. In this problem, the manufacturer has some unrelated parallel machines whereas each of the subcontractors owns a single machine. The jobs are scheduled on both the manufacturer and the subcontractors' resources in order to minimise the weighted sum of total completion time and total outsourcing cost. An integer programming model is developed for this problem and enhanced using some characteristics of the problem. Moreover, a heuristic method is presented based on the decomposition of the problem via Lagrangian Relaxation approach. Neto, Filho, and da Silva (2015) considered an identical parallel machine scheduling problem in which each job may be subcontracted with a specific cost, while total outsourcing budget is limited. The problem is solved to minimise the sum of total outsourcing cost and total weighted tardiness. To this end, an Ant Colony Algorithm (ACO) is developed whose performance is evaluated by comparing it to the mathematical programming approach. Liu, Lee, and Wang (2016) addressed a green scheduling problem with outsourcing in the context of uniform parallel machine scheduling. The problem objective is to minimise the aggregate of total resource consumption and total outsourcing cost regarding an upper bound for the maximum tardiness of the jobs. In order to solve the problem exactly or approximately, a branch and bound and a metaheuristic algorithm, which hybridises genetic algorithm (GA) and simulated annealing (SA), are developed respectively.

Qi (2009) investigated a two stage flow shop scheduling problem in which the first-stage operations are possible to be outsourced. Moreover, it is assumed that the subcontracted jobs are transported back to the shop in a single batch with a delay time. The problem objective is to minimise both makespan and total outsourcing cost simultaneously in order to obtain the Pareto optimal solutions. An optimal algorithm and a heuristic method are developed to solve the problem according to the characteristics of the optimal solution. Furthermore, the author has studied an extension of this problem in another research work in which outsourcing is allowed for both stages (Qi 2011). In this study, several scenarios are considered for outsourcing and some optimisation algorithms are developed to solve the defined problems. Choi and Chung (2011) analysed a two stage flow shop scheduling problem with the option of jobs outsourcing. The problem aims to minimise the sum of makespan and total outsourcing cost. The complexity of the problem is analysed by the authors, and several optimal conditions and polynomial algorithms are presented for some special cases of the problem. Three other special cases of this problem are investigated by Chung and Choi (2013), in two of which the problem is demonstrated to be NP-hard and for the other one to be polynomially solvable. Moreover, an approximation algorithm is proposed to solve the general problem.

Tavares Neto and Godinho Filho (2011) analysed a multi stage permutation flow shop scheduling problem with the possibility of jobs outsourcing. The optimisation criterion is to minimise a weighted sum of makespan and total outsourcing cost. To solve the problem, the authors proposed an algorithm consisting of two sequential ACO algorithms, each of which solves a certain part of the problem. Mokhtari, Abadi, and Amin-Naseri (2012) addressed a multi stage flow shop scheduling problem with the goal of minimising the sum of total outsourcing cost and weighted mean flow time. Outsourcing can be assumed for the jobs or operations, and besides, the existent subcontractor may have single machine or parallel machines in its shop. Combining these assumptions results in four scenarios which are solved in this paper using mathematical programming and a metaheuristic algorithm called Team Process Algorithm (TPA). Choi and Park (2014) considered a problem of multi stage permutation flow shop scheduling in which each job may be subcontracted as a whole. The objective function is the sum of total outsourcing cost and either makespan or total completion time. There exists a special property for the in-house processing times in the problem such that they are the sum of two parameters; one depends on the machine type and the other on the job type. The authors have proven that the problems are polynomially solvable considering this assumption.

For the machine environment addressed in this paper, *i.e.* job shop, to the best of our knowledge, Chung et al. (2005) conducted the earliest study. They assumed that each of the operations can be subcontracted independently, and it is required to fulfil the jobs before the assigned due dates. The problem objective is to minimise total outsourcing cost. A two stage algorithm is proposed to solve the problem in which the first stage seeks to minimise the maximum lateness of the jobs through scheduling the in-house operations. In the second stage, the tardinesses of the jobs, if exists any, are diminished by outsourcing and re-scheduling strategies. Guo and Lei (2014) studied a bi objective job shop scheduling problem in which a job may be outsourced as a whole to a certain single subcontractor. The problem addresses scheduling of the jobs in the shops of both the manufacturer and the subcontractor. The problem objectives are to minimise total tardiness and total outsourcing cost together. A two phased neighbourhood search algorithm is proposed in order to obtain the Pareto optimal solutions. In the first phase, several random solutions are generated and improved using four neighbourhood structures, and in the second phase, the best solution of the previous phase is adjusted via outsourcing and scheduling decisions. The authors have studied this problem once more in another paper while altering the structure of the problem objective (Lei and Guo 2016). In this respect, total outsourcing cost is transferred to the constraint set by considering an upper bound for it, and total tardiness is minimised as the objective function of the problem. A metaheuristic algorithm which is called shuffled frog leaping algorithm is utilised to solve the problem.

To the best of our knowledge, among the vast literature of job shop scheduling problem, only the three reviewed papers above have considered the possibility of outsourcing. Therefore, it demands further attention from the academic community to investigate this subject. In this regard, in our study, we have addressed outsourcing and scheduling in the job shop environment to help fill this gap. Makespan is one of the most, and probably the most common criterion in the literature of scheduling and should be investigated for JSSP with outsourcing as well. However, it is not taken into account by any of the abovementioned JSSP papers, and it is the first time that this objective is being studied for this context. On the other hand, mathematical programming, as one of the solution approaches of this paper, is not studied well for JSSP with outsourcing. Indeed, among the reviewed JSSP papers, Chung et al. (2005) have pointed to a descriptive model which is not rigorous enough to be implemented for solving the problem. Besides, Guo and Lei (2014) have proposed a mathematical model with a four index binary variable which seems to be non-efficient regarding the ordinary MILP formulation approaches of JSSP (Ku and Beck 2016). For the other utilised solution approach, CP, to the best of the authors' knowledge, it is the first time that this approach is being utilised in the whole literature of integrating scheduling and outsourcing.

3. Problem statement

Classical job shop scheduling problem investigates allocation of n jobs to m machines over a period of time. Each job consists of m consecutive operations, each of which processed by one of the machines. The order of the machines to process the operations of a job is predetermined and may be different than those of the other jobs. The assumptions for typical scheduling problems are also considered for this problem such as no preemption condition and readiness of jobs and machines at the beginning of the scheduling horizon.

To extend the problem for the case of outsourcing, we assume that each job may be subcontracted entirely to the available subcontractors. If so, the subcontracted job has a specified delivery time, and incurs a cost for outsourcing, which are named respectively *outsourcing time* and *outsourcing cost*. No capacity constraint is assumed for the subcontractors, so outsourcing a job doesn't restrict the other ones for outsourcing. Consequently, the selected jobs are outsourced at the time zero in order to reduce the completion times of the jobs. The considered objectives for the problem are *makespan* and *total outsourcing cost*. These objectives are summed together to form a single objective function, having a particular weight for the latter one. Note that the makespan objective is defined as the maximum completion time of all the in-house and the outsourced jobs.

The utilised notations for the problem are presented below:

Notations:

m:	Number of machines
n:	Number of jobs
p_{ij}:	Processing time of job j on machine i
ot_j:	Outsourcing time of job j
oc_j:	Outsourcing cost of job j
δ_h^j:	The machine number related to h th operation of job j
w:	Weight of total outsourcing cost in the objective function
$C_{\max}$:	Maximum completion time of the entire jobs; makespan

4. Mathematical model

Mathematical programming is used extensively to model and solve the scheduling problems. It is often the primal strategy to analyse the new scheduling problems (Ku and Beck 2016). Moreover, mathematical models are useful to comprehend the structure of a problem and can be helpful for developing heuristic algorithms (Özgüven, Özbakır, and Yavuz 2010). Although the complexity of scheduling problems may limit the utilisation of this approach, still many practical size problems can be solved using this approach. A special advantage of mathematical programming is the ability to prove solution optimality. Moreover, if the problem is not solved to optimality, at least a lower bound is reported by the solver for the optimal solution in minimisation problems.

In recent years, several researchers have used mathematical programming as a principal solution approach for the scheduling problems such as Keha, Khowala, and Fowler (2009), Unlu and Mason (2010), and Demir and Kürşat İşleyen (2013). Particularly for the job shop scheduling problem with the makespan objective, Ku and Beck (2016) analysed three types of MILP formulation methods, *i.e. disjunctive*, *time-indexed* and *rank-based*. They compared these methods through detailed numerical experiments in terms of proving solution optimality and demonstrated that the disjunctive approach is the most efficient one. In this type of formulation, the sequence of the jobs on the machines are expressed by a three-index binary variable like z_{ijl}, which is equal to 1 if job j is processed before, but not necessarily just before, job l on machine i. Here, we extend this well-known approach to model the problem under investigation. The developed model is presented in the following:

Decision Variables:

x_{ij}:	Starting time of job j on machine i, if it is processed by the manufacturer.
z_{ijl}:	Binary variable, equals to 1 if job j precedes job l on machine i; 0 otherwise.
f_j:	Binary variable, equals to 1 if job j is outsourced; 0 otherwise.
$C_{\max}$:	Maximum completion time of the in-house processed and the outsourced jobs.

Model:

$$\min \left(C_{\max} + w \sum_j oc_j\, f_j \right) \tag{1}$$

Subject to

$$x_{\delta_h^j, j} \geq x_{\delta_{h-1}^j, j} + p_{\delta_{h-1}^j, j} \quad \forall j\,, 2 \leq h \leq m \tag{2}$$

$$x_{ij} \geq x_{il} + p_{il} - M z_{ijl} - M(f_j + f_l) \quad \forall i, j, l\,;\, j < l \tag{3}$$

$$x_{il} \geq x_{ij} + p_{ij} - M(1 - z_{ijl}) - M(f_j + f_l) \quad \forall i, j, l\,; j < l \tag{4}$$

$$C_{\max} \geq x_{\delta_m^j, j} + p_{\delta_m^j, j} - M f_j \quad \forall j \tag{5}$$

$$C_{\max} \geq ot_j - M(1 - f_j) \quad \forall j \tag{6}$$

$$x_{ij} \geq 0 \quad \forall i, j \tag{7}$$

$$z_{ijl} \in \{0, 1\} \quad \forall i, j, l\,; j < l \tag{8}$$

$$f_j \in \{0, 1\} \quad \forall i, j \tag{9}$$

In this model, M is a big enough value which makes the constraints inactive when needed. The objective function of the problem is stated in Expression 1. Constraint 2 indicates the precedence relationships between the operations in the jobs. Here, this constraint is not restricted to the in-house jobs since in the other constraints, x_{ij} has no effect if its related job is outsourced. Constraints 3 and 4 guarantee that the operations processed on a machine are sequenced with no overlap. The constraints get inactive if job j or l is subcontracted. Constraint 5 ensures that makespan is not lower than the completion time of each of the in-house processed jobs, and Constraint 6 indicates this for the outsourced jobs. Constraints 7–9 specify the domains of the decision variables.

5. Constraint programming model

Constraint Programming (CP) is an alternative approach to solve the combinatorial optimisation problems, which originates in computer science and artificial intelligence community. It was initially developed to solve *constraint satisfaction problems* (CSP), *i.e.* the constrained problems aiming to find a feasible solution. However, later on CP was utilised to solve the optimisation problems too (Gass and Fu 2013). This approach has been applied successfully in different areas such as production planning, scheduling, timetabling, supply chain management, production configuration, molecular biology, robotics, etc. (Benhamou, Jussien, and O'Sullivan 2007). The major solution strategy in CP, called *constraint propagation*, is based on analysing the existing constraints to infer new ones, reducing the domains of the decision variables based on the new constraints, and communicating the reduced domains to the constraints. This procedure continues iteratively with the aim of restricting the solution space and solving the problem (Lustig and Puget 2001). CP likewise MILP, and in contrast to the metaheuristic algorithms and most of the heuristic algorithms, is capable of proving solution optimality. However, CP may be more efficient than MILP when the problem is constraint intensified, since it gains the constraints effectively in solving the problem.

CP is used extensively to solve the scheduling problems (Baptiste, Pape, and Nuijten 2012). Particularly, for classical JSSP with the makespan objective, Ku and Beck (2016) have examined CP along with the MILP approaches and demonstrated that CP outperforms MILP for this problem. One of the advantages of CP is a simple declaration of the variable relationships and problem constraints (Apt 2003). In fact, many logical relations may be expressed directly through the CP statements, while by mathematical programming it may require defining several additional variables and constraints to formulate them. Particularly, IBM ILOG has provided a dedicated set of CP modelling statements for the scheduling problems which are embedded in the ILOG CPLEX CP Optimizer software. These statements which are very scheduling oriented can express various types of scheduling concepts explicitly such as time durations, precedence constraints, jobs sequencing, parallel servers, resource consumptions and so on. Several researchers have used this approach to model and solve the scheduling problems in recent years such as Zeballos, Novas, and Henning (2011), Unsal and Oguz (2013), Gedik et al. (2016), and Ham, Fowler, and Cakici (2017). The basic element in this modelling approach is *interval variable* which is a type of decision variable. Interval variables are utilised to represent the time durations in the model like the length of an activity or a time window. When defining an interval variable, its length may be specified as a fixed parameter or remained as a decision variable in the model. Moreover, an interval variable can be defined as *optional*; in this case, the presence of the interval variable is also a choice to be determined by solving the problem. After declaring the interval variables and other ordinary decision variables, based on which several scheduling oriented relations can be expressed in the model. For example, it is possible to restrict the starting or finishing time of an interval variable, define precedence relationships between the interval variables or state no-overlapping condition for a set of interval variables. For detailed knowledge of this modelling approach we refer the readers to CPLEX 12.6.0 Manual; Laborie and Rogerie 2008; and Laborie and Rogerie 2009, and in the following, we present a model for the current problem based on this approach:

Decision Variables

t_{ij}:	Interval variable for processing job j on machine i (optional)
$C_{\max}$:	Integer variable, makespan

Model:

$$\min\left(C_{\max} + w\sum_{j} oc_j(1 - \text{Presence Of}(t_{0j}))\right) \tag{10}$$

Subject to

$$\text{Interval}(t_{ij}\ ;\text{size}: p_{ij}; \text{optional}) \quad \forall i, j \tag{11}$$

$$\text{End Before Start}(t_{\delta^j_{h-1}j}, t_{\delta^j_h j}) \quad \forall j\ , 2 \le h \le m \tag{12}$$

$$\text{No Overlap}(\{t_{ij} | 1 \le j \le n\}) \quad \forall i \tag{13}$$

$$\text{Presence Of}(t_{ij}) = \text{Presence Of}(t_{lj}) \quad \forall i, j, l \tag{14}$$

$$C_{\max} = \max\ (\{\text{End Of}(t_{\delta^j_m j}) | 1 \le j \le n\} \cup \{ot_j(1 - \text{Presence Of}(t_{0j})) | 1 \le j \le n\}) \tag{15}$$

In the model above, the interval variables represent the in-house operations. In this regard, if an interval variable is absent in the problem solution, it means that the corresponding operation is outsourced. Note that due to the rules of ILOG CP

Optimizer, the time points must be integer, so the $C_{\max}$ variable is also defined as an integer variable. The lengths of the intervals are also assumed to be integer; however, if their actual values are real, it is possible to multiply them by a large value and then round them to the integers. In the proposed model, Expression 10 states the objective function of the problem. The element $\text{Presence Of}(t)$ is an ILOG Boolean variable which is equal to 1 if the interval t is present in the solution and 0 otherwise (CPLEX 12.6.0 Manual). Accordingly, we use $1 - \text{Presence Of}(t_{0j})$ to represent the outsourcing state of job j in the model. Expression 11 defines the properties of the interval variables. As is evident, the lengths of these variables are specified, and the variables are set to be optional. Constraint 12 expresses the order of operations in the jobs. Constraint 13 ensures sequencing the in-house jobs on the machines without overlapping. Constraint 14 indicates that the whole operations of a job are either processed in the shop or subcontracted. Finally, makespan is defined in Constraint 15 as the maximum completion time of the in-house and the outsourced jobs.

6. Problem relaxation

In the experimental results, it is observed that some large scale problems cannot be solved optimally by the MILP or CP approaches. In these cases, it is worthwhile to derive some strong lower bounds for the problem in order to approximate the optimal objective values. Moreover, these bounds can be utilised to estimate the gap of the resulted solutions from optimality. Accordingly, two types of relaxations are developed for the studied problem in this section. When a relaxed problem is solved optimally, the objective value of its optimal solution can be used as a lower bound for the objective value of the optimal solution in the original problem. In general, the relaxation approaches are developed to have a sense of optimality for larger scales of the problem than the scales for which the MILP or CP approaches solve the problem optimally.

In both of the proposed relaxation approaches, the precedence relationships between the operations in the jobs are relaxed. At the same time to offset this simplification, some explicit constraints are added to the relaxed problems for the starting times of the operations and completion times of the jobs. Similar relaxation approach is also used for classical JSSP in the literature (Lageweg, Lenstra, and Kan 1976; Carlier and Pinson 1989; Brucker and Jurisch 1993). This approach is extended here for the studied problem through the proposed relaxation methods as follows:

Relaxation 1**:** The first relaxation is based on the decomposition of the original problem into m distinct single machine scheduling problems. Each of these problems corresponds to one of the machines. The problem related to machine i is called here as P_i. Solving each problem results in a lower bound for the original problem, and the largest one is selected as the proposed lower bound by this relaxation named $lb1$.

The relaxed problem P_i addresses which jobs to be outsourced and how to sequence the in-house jobs on machine i. If a job is outsourced, it has the same outsourcing time and cost as in the original problem. Otherwise, its processing time is equal to the original processing time of this job on machine i. Accordingly, by decomposing the problem in such a way the precedence relationships of the operations in the jobs are ignored as well. In order to improve the induced lower bound, some valid constraints of the original problem are added to the relaxed problem P_i as is explained in the following. It is known from the original problem that if job j is processed internally, it has a minimum possible starting time on machine i due to the order of its operations. This value, called r_{ij}, is calculated by aggregating the processing times of this job's operations prior to machine i. Moreover, job j requires a minimum time to be completed after releasing from machine i, which is called q_{ij} here and is equal to the total processing time of its operations after machine i. In order to include these constraints in problem P_i, the starting time of job j, if it is not outsourced, is constrained to be greater than or equal to r_{ij}. Furthermore, the completion time of job j is defined as an event that takes place q_{ij} units of time after finishing its work on the machine. The objective function of the relaxed problem is also defined in a similar way as in the original problem, using the newly defined completions times for the in-house jobs.

For each feasible solution of the original problem, the outsourcing plan of the jobs together with the schedule of the in-house jobs on machine i clearly induces a feasible solution for problem P_i. According to the definition of P_i, it is evident that the objective value in this feasible solution is not greater than the objective value in the feasible solution of the original problem. So it is concluded that solving each of the single machine relaxed problems results in a lower bound for the original problem. As previously stated, the maximum one is selected as the output of the first relaxation approach $lb1$.

Relaxation 2**:** In the previous relaxation, the single machine problems are solved independently, so they may have different outsourcing plans. However, in the original problem, because a job is subcontracted entirely, outsourcings of the jobs on the machines are conducted in an identical way. Accordingly, in order to fill this gap and improve the resulted lower bound, in the second relaxation approach, the original problem is relaxed as a whole without machine decomposition. This strategy leads to a centralised outsourcing plan for the entire machines. Similarly, this relaxation is performed again by omitting the operations' precedence relationships in the jobs. Besides, the derived starting and completion time constraints for the previous relaxation are also utilised for the relaxed problem. Therefore, job j is required to start on machine i after r_{ij} units of time. Furthermore, the completion time of an in-house job j is defined as an event that occurs at least q_{ij} units of time after releasing from each machine i. The objective function of the problem is also defined likewise the original problem.

This relaxation is evidently more powerful than the previous one. Indeed, the new relaxed problem includes all the previous single machine relaxed problems together which are now dependent to each other through the jobs outsourcing decisions. Hence, the generated lower bound by relaxation 2, called $lb2$, is greater than or equal to $lb1$. However, the problem gets more complicated, and subsequently harder to solve.

It is worth mentioning that in order to obtain the intended lower bounds by the relaxation approaches, it is required to solve the corresponding relaxed problems optimally. To this end, we examine both mathematical programming and constraint programming approaches to solve the problems. Accordingly, the MILP and CP models of the relaxed problems are presented in the following:

***MILP model of the relaxed problems*:** The model presented below is primarily formulated for relaxation 2; nonetheless, it is also applicable to the relaxed problem P_k in relaxation 1 if i indexes are fixed to k. The decision variables are similar to those of the original problem MILP model.

$$\min \left(C_{\max} + w \sum_j oc_j \, f_j \right) \tag{16}$$

Subject to

$$x_{ij} \ \geq \ r_{ij} \quad \forall i, j \tag{17}$$

$$x_{ij} \geq x_{il} + p_{il} - M z_{ijl} - M (f_j + f_l) \quad \forall i, j, l \,; j < l \tag{18}$$

$$x_{il} \ \geq \ x_{ij} + p_{ij} - M (1 - z_{ijl}) - M (f_j + f_l) \quad \forall i, j, l \,; j < l \tag{19}$$

$$C_{\max} \ \geq x_{ij} + p_{ij} + q_{ij} - M f_j \quad \forall i, j \tag{20}$$

$$C_{\max} \ \geq \ ot_j - M (1 - \ f_j) \quad \forall j \tag{21}$$

$$z_{ijl} \in \{0, 1\} \quad \forall i, j, l \,; j < l \tag{22}$$

$$f_j \in \{0, 1\} \quad \forall j \tag{23}$$

Expression 16 declares the objective function of the model. Constraint 17 indicates the minimum possible starting times for the jobs on the machines. Constraints 18 and 19 are for sequencing the jobs on the machines with no overlap. Constraint 20 ensures that makespan is greater than or equal to the defined completion times for the in-house jobs. Constraint 21 indicates that the completion times of the outsourced jobs are not greater than makespan. Finally, Constraints 22 and 23 declare the decision variables' domains. As is evident, comparing to the MILP model of the original problem, the constraint that declares the precedence relationships of the operations in the jobs are omitted in this model. Instead, Constraint 17 is added to the model, and Constraint 20 is modified.

***CP models of the relaxed problems*:** Similar to the MILP model, the proposed CP model below is also applicable to both relaxations 1 and 2, except that for the former, i is fixed to the number of the desired machine. The same decision variables of the original problem's CP model are used here as well.

$$\min \left(C_{\max} + w \sum_j oc_j (1 - \text{Presence Of}(t_{0j})) \right) \tag{24}$$

Subject to

$$\text{Interval}(t_{ij} \,; \text{size} : p_{ij}; \text{optional}) \quad \forall i, j \tag{25}$$

$$\text{Start Of}(t_{ij} \,, \ M) \geq r_{ij} \quad \forall i, j \tag{26}$$

$$\text{No Overlap}(\{t_{ij} | 1 \leq j \leq n\}) \quad \forall i \tag{27}$$

$$\text{Presence Of}(t_{ij}) = \text{Presence Of}(t_{lj}) \quad \forall i, j, l \tag{28}$$

$$C_{\max} = \max\ (\{\text{End Of}(t_{ij}, -M) + q_{ij} | \forall i, j\} \cup \{ot_j.(1 - \text{Presence Of}(t_{0j})) | \forall j\}) \tag{29}$$

In this model, M is considered as a big enough value. The objective function of the model is declared in Expression 24. Expression 25 specifies the properties of the utilised interval variables to represent the operations of the in-house jobs. Constraint 26 indicates the minimum possible starting times of the in-house jobs on the machines. Note that the element $\text{Start Of}(t, a)$, where t and a are respectively an interval variable and an integer value, returns the starting time of t if it is present, and a if it is absent. Besides, in the case of omitting the second argument, zero is returned when t is absent (CPLEX 12.6.0 Manual). Accordingly, Constraint 26 becomes trivial when the related operation is outsourced. Constraint 27 is for sequencing the jobs on the machines. Constraint 28 ensures that all the operations of a job have similar outsourcing states. Constraint 29 indicates that makespan is the maximum value of the defined completion times for the in-house jobs and the outsourcing times of the subcontracted jobs. In this constraint, the element End Of() with two arguments acts like Start Of() with two arguments as was explained.

7. Experiments

In this section, comprehensive numerical experiments are conducted to evaluate the efficiency of the proposed solution methods. To solve the MILP and CP models, ILOG CPLEX Optimizer and ILOG CP Optimizer software in versions 12.6.0 are utilised respectively. Using Concert technology, the models are implemented in Visual Studio 2010 via C#.NET programming language. The experiments are run on the systems with CPU 3.1 GHz, 4 GB RAM and operating system of Windows 7.

7.1. Data generation

Before conducting the numerical experiments, it is required to generate some proper test problems. To this end, we extend two commonly used data sets of JSSP in the literature for the problem under investigation. The original characteristics of these data sets are described in the following:

(1) *Taillard instances* (Taillard 1993): the problem dimensions are $m \times n = 15 \times 15$, 15×20, 20×20, 15×30, 20×30, 15×50, 20×50, 20×100. The processing times of the operations are generated randomly from a discrete uniform distribution with the range of 1 to 99, shown by *Uni* [1 99]. For each problem size, 10 instances are generated randomly, which totally results in 80 problem instances named consecutively as TA01 to TA80.

(2) *Lawrence instances* (Lawrence 1982): the problem dimensions are $m \times n = 10 \times 5$, 15×5, 20×5, 10×10, 15×10, 20×10, 30×10 and 15×15. The generating distribution of the operations' processing times is discrete *Uni* [5 99]. Meanwhile, 5 instances are generated for each problem size, which totally results in 40 instances named consecutively as LA01 to LA40.

The Taillard instances related to the first six dimensions (TA01-TA60) and the entire Lawrence instances are chosen to be extended for the studied problem. To this end, it is required to add outsourcing time and cost quantities as well as the weight of total outsourcing cost in the objective function to the basic instances. In order to generate the outsourcing times and costs, first, we assume hypothetically that there exist distinct outsourcing times and costs for the operations of the jobs. Then, after generating these values, they are aggregated to attain the jobs' outsourcing times and costs. In this respect, for the Taillard and Lawrence instances, the outsourcing times of the operations are generated randomly from the same distributions used in their original data sets. Moreover, the outsourcing costs of the operations are generated randomly from *Uni* [1 49] and *Uni* [10 100] for the Taillard and Lawrence instances respectively. The set of the extended instances related to a problem size in the Taillard or Lawrence data sets are called a *problem class*. In the experiments related to the relaxation approaches, some other critical problem dimensions are observed besides the abovementioned problem dimensions. So, three other *extra problem classes* with the dimensions of $m \times n = 20 \times 25$, 25×25, 15×35 are also added to the set of the problem classes. For each extra problem class, 5 instances are generated randomly by the same generating distributions used for the Taillard instances. Meanwhile, all the generated time or cost parameters mentioned in this section are produced using the similar random generator, seeds, and algorithms utilised by Taillard (1993).

For the weight of total outsourcing cost in the objective function, three different values are chosen for each problem class. The weights are selected through extensive preliminary experiments in the aim of having varied levels of jobs outsourcing rates in the test problems. The selected values for each problem class are numbered from 1 to 3, which generally lead to low, medium and high outsourcing rates respectively. These values are reported in Table 1 along with the characteristics of the problem classes.

Table 1. Summary of the problem classes' characteristics.

	Name	Basic Instances	Number of Instances	Dimensions		Selected Weights		
				m	n	0	1	2
Lawrence Problem Classes	LPC1	LA01 – LA05	5	5	10	0.28	0.23	0.07
	LPC2	LA06 – LA10	5	5	15	0.31	0.25	0.12
	LPC3	LA11 – LA15	5	5	20	0.35	0.275	0.175
	LPC4	LA16 – LA20	5	10	10	0.145	0.06	0.025
	LPC5	LA21 – LA25	5	10	15	0.13	0.09	0.005
	LPC6	LA26 – LA30	5	10	20	0.15	0.11	0.03
	LPC7	LA31 – LA35	5	10	30	0.16	0.12	0.04
	LPC8	LA36 – LA40	5	15	15	0.1	0.06	0.02
Taillard Problem Classes	TPC1	TA01 – TA10	10	15	15	0.14	0.085	0.01
	TPC2	TA11 – TA20	10	15	20	0.17	0.1	0.01
	TPC3	TA21 – TA30	10	20	20	0.12	0.06	0.01
	TPC4	TA31 – TA40	10	15	30	0.18	0.135	0.06
	TPC5	TA41 – TA50	10	20	30	0.115	0.09	0.03
	TPC6	TA51 – TA60	10	15	50	0.21	0.16	0.11
Extra Problem Classes	EPC1	–	5	20	25	0.15	0.1	0.03
	EPC2	–	5	25	25	0.1	0.07	0.01
	EPC3	–	5	15	35	0.18	0.15	0.1

7.2. Experiments and results

After specifying the test problems, they are solved by the solution methods proposed in the previous sections. In this regard, for the original problem, the MILP and CP models of the problem instances are solved by the solvers considering a time limit of 1 h. To obtain the optimal solutions of the relaxed problems, the related MILP or CP models can be applied. However, in the preliminary experiments, it is observed that the CP approach outperforms the MILP approach significantly in solving the relaxed problems to optimality. So, we use only the CP model for the relaxed problems. In this respect, a time limit of 400 s is considered for solving each of the single machine relaxed problems, while 1 h solution is allowed for the relaxed problem of relaxation 2.

The experiment results are reported in Tables 2 and 3, respectively for the Lawrence problem classes, and Taillard and extra problem classes. Each row corresponds to the instances of a problem class with a particular weight. Accordingly, the indicators, which are defined in the following, are reported in average among the instances, except the ones which are explicitly mentioned. In the MILP segments of Tables 2 and 3, T denotes either the assigned time limit when the problem is not solved to optimality, or the solution time otherwise. The indicator *obj* represents the objective value of the best obtained solution. Moreover, *lb*0 indicates the best lower bound attained by the MILP solver for the problem over the solution time. The indicator *gap* denotes the optimal gap percentage reported by the solver at the end of solution, which is equal to $\frac{\text{objective value} - \text{lower bound}}{\text{objective value}} \times 100$. Finally, the number of instances that are solved to optimality is presented by *opt#*. For the CP solution approach, the indicators T, *obj* and *opt#* are defined likewise those of the MILP approach, while the optimal gap is not declared by the CP solver and is ignored. In the segments of the tables related to relaxation 1, T denotes the total solution time of the entire single machine problems in a problem instance. The obtained lower bound by relaxation 1 is reported by *lb*1. Moreover, the number of optimally solved single machine problems is represented by the indicator *sol#*. Note that the obtained lower bound for an instance is the maximum of the objective values among its optimally solved single machine problems. Hence, if none of the single machine problems are solved to optimality, then clearly, the lower bound is not reported for that instance. In this respect, the number of instances for which at least a single machine problem is solved to optimally, *i.e.* the lower bound is obtained for, is reported by *opt#*. The percentage of the gap between the CP solution and the obtained lower bound that is equal to $\frac{\text{CP objective} - \text{lower bound 1}}{\text{lower bound 1}} \times 100$ is denoted by *gap*1. For relaxation 2, T and *lb*2 indicate respectively the solution time and the resulted lower bound. The indicator *opt#* is the number of instances for which the relaxed problem is solved to optimality and subsequently the lower bound is reported. Similarly, *gap*2 denotes the percentage of the gap between the CP solution and the obtained lower bound calculated by the relation $\frac{\text{CP objective} - \text{lower bound 2}}{\text{lower bound 2}} \times 100$. Note that for both relaxations, in the case of not relating the lower bound for an instance, its solution time is excluded in calculating the average solution time of the instances (T). Clearly, the indicators *gap*1 and *gap*2 are also calculated using the instances that the corresponding lower bounds are reported for. The indicator *dev* represents the percentage of the deviation of the MILP solution from the CP solution which is equal to $\frac{\text{MILP objective} - \text{CP objective}}{\text{CP objective}} \times 100$. As the last indicator, the percentage of the outsourced jobs in the CP solution of the original problem is represented by *out*. This

Table 2. Experiment Results of Lawrence problem classes.

				MILP					CP			Relaxation 1					Relaxation 2					
P.C.	*m*	*n*	*wNo*	*T*	*obj*	*lb*0	*gap*0%	*opt#*	*T*	*obj*	*opt#*	*T*	*lb*1	*sol#*	*opt#*	*gap*1%	*T*	*lb*2	*opt#*	*gap*2%	*dev*%	*out*2%
LPC1	5	10	1	369	613	613	0	5	0.5	613	5	0.4	579	5	5	6	0.1	603	5	1.7	0	12
			2	67	594	594	0	5	0.6	594	5	0.4	552	5	5	7.8	0.2	585	5	1.5	0	20
			3	40	424	424	0	5	0.2	424	5	0.2	407	5	5	4.2	0.1	415	5	2.3	0	50
LPC2	5	15	1	3600	877	593	33	0	5.0	877	5	8.4	872	5	5	0.6	5.7	877	5	0	0	17
			2	3600	833	589	29	0	6.4	832	5	8.2	804	5	5	3.5	7.4	832	5	0	0.1	25
			3	3322	630	571	9	1	4.9	630	5	3.9	573	5	5	10.1	4.4	613	5	2.7	0	57
LPC3	5	20	1	3600	1165	493	58	0	45	1158	5	49	1153	5	5	0.5	45	1158	5	0	0.6	11
			2	3600	1110	465	58	0	348	1102	5	77	1074	5	5	2.6	197	1100	5	0.2	0.7	23
			3	3600	925	453	50	0	239	924	5	43	860	5	5	7.5	284	919	5	0.6	0.1	49
LPC4	10	10	1	11.7	851	851	0	5	3.7	851	5	0.3	770	10	5	10.7	0.9	774	5	10	0	10
			2	3.4	750	750	0	5	1.5	750	5	0.2	690	10	5	8.7	0.6	702	5	6.7	0	42
			3	1.3	652	652	0	5	0.3	652	5	0.2	631	10	5	3.3	0.4	640	5	2.0	0	50
LPC5	10	15	1	3600	983	823	16	0	44	972	5	2.6	904	10	5	7.7	5.7	922	5	5.5	1.2	8
			2	3600	933	818	12	0	28	930	5	2.5	835	10	5	11.4	8.9	874	5	6.4	0.3	21
			3	146	657	657	0	5	1	657	5	0.6	648	10	5	1.4	1.8	651	5	0.8	0	53
LPC6	10	20	1	3600	1289	805	37	0	1379	1221	5	37	1179	10	5	3.7	53	1200	5	1.8	5.5	6
			2	3600	1218	776	36	0	1731	1187	4	33	1058	10	5	12.3	107	1134	5	4.7	2.7	14
			3	2888	821	752	8	1	29	820	5	9	759	10	5	8.0	19	789	5	4	0.1	58
LPC7	10	30	1	3600	1810	721	60	0	3600	1759	0	2965	1387	3	4	27.1	–	–	0	–	2.8	8
			2	3600	1775	725	59	0	3600	1681	0	3575	1234	1.4	4	35.8	–	–	0	–	5.6	17
			3	3600	1095	697	36	0	3600	1093	0	2230	988	6	5	12.2	–	–	0	–	0.3	67
LPC8	15	15	1	3600	1271	1070	16	0	162	1260	5	1.6	1155	15	5	9.3	7.5	1167	5	8.1	0.9	8
			2	3600	1243	1063	14	0	171	1227	5	1.3	1104	15	5	11.3	9.2	1125	5	9.1	1.2	20
			3	1103	1020	1012	0.8	4	10	1020	5	0.7	960	15	5	6.3	4.1	977	5	4.4	0	56

Table 3. Experiment results of Taillard and extra problem classes.

				MILP					CP			Relaxation 1					Relaxation 2					
P.C.	*m*	*n*	*wNo*	*T*	*obj*	*lb*0	*gap*0%	*opt#*	*T*	*obj*	*opt#*	*T*	*lb*1	*sol#*	*opt#*	*gap*1%	*T*	*lb*2	*opt#*	*gap*2%	*dev%*	*out*2%
TPC1	15	15	0	3600	1202	1067	11	0	178	1187	10	0.7	1031	15	10	15.2	5.7	1061	10	11.9	1.3	13
			1	2689	1102	1054	4.4	3	28	1102	10	0.6	983	15	10	12.1	6.5	1015	10	8.6	0	43
			2	396	893	893	0.1	9	2.1	893	10	0.3	869	15	10	2.8	1.8	876	10	2.1	0	59
TPC2	15	20	0	3600	1437	987	31	0	3562	1383	1	9.1	1203	15	10	15	39	1242	10	11.3	4	9.5
			1	3600	1281	979	24	0	1979	1267	6	6.6	1085	15	10	16.9	42	1138	10	11.3	1.2	41
			2	2199	913	900	1.3	4	14	912	10	0.9	881	15	10	3.6	10	890	10	2.5	0	63
TPC3	20	20	0	3600	1674	1264	24	0	3600	1634	0	4	1371	20	10	19.2	24	1404	10	16.4	2.5	7.5
			1	3600	1464	1273	13	0	896	1461	9	3	1286	20	10	13.6	36	1332	10	9.7	0.2	45
			2	2738	1194	1183	0.8	3	13	1194	10	0.8	1159	20	10	3	7.7	1167	10	2.3	0	61
EPC1	20	25	0	3600	1955	1272	35	0	3600	1752	0	19	1499	20	5	16.9	129	1543	5	13.6	11	0
			1	3600	1801	1259	30	0	3600	1764	0	16	1408	20	5	25.3	255	1487	5	18.7	2.1	13
			2	3600	1352	1222	9.8	0	1038	1343	4	5	1238	20	5	8.5	73	1271	5	5.6	0.7	58
EPC2	25	25	0	3600	2116	1499	29	0	3600	1986	0	5.5	1643	25	5	20.9	64	1682	5	18.1	6.6	4.8
			1	3600	1989	1501	25	0	3600	1972	0	5.2	1599	25	5	23.4	78	1648	5	19.7	0.9	34
			2	3600	1521	1465	3.8	0	1070	1515	4	1.9	1451	25	5	4.5	41	1464	5	3.5	0.4	52
TPC4	15	30	0	3600	1967	987	50	0	3600	1812	0	2181	1497	11.6	10	21.2	–	–	0	–	8.6	0.6
			1	3600	1842	979	47	0	3600	1793	0	1723	1409	13.4	10	27.2	–	–	0	–	2.9	28
			2	3600	1366	975	29	0	3600	1357	0	413	1151	14.8	10	17.9	2397	1208	2	9.4	0.7	67
TPC5	20	30	0	3600	2118	1242	41	0	3600	1993	0	5548	1653	19.9	10	20.6	–	–	0	–	6.5	6.9
			1	3600	2011	1237	38	0	3600	2006	0	223	1563	20	10	28.5	2927	1653	1	19.3	0.3	43
			2	3600	1448	1210	17	0	3273	1452	1	34	1295	20	10	12.1	815	1346	10	7.9	– 0.2	68
EPC3	15	35	0	3600	2231	985	56	0	3600	2106	0	5453	1452	2.6	5	45.2	–	–	0	–	6	8
			1	3600	2115	976	54	0	3600	2042	0	5352	1385	2.8	5	47.5	–	–	0	–	3.6	17
			2	3600	1835	983	47	0	3600	1814	0	4931	1268	4.2	5	43.2	–	–	0	–	1.2	68
TPC6	15	50	0	3600	3368	1020	70	0	3600	2863	0	–	–	–	0	–	–	–	0	–	18	3.8
			1	3600	3203	1011	68	0	3600	2804	0	–	–	–	0	–	–	–	0	–	14	7.4
			2	3600	2619	992	62	0	3600	2537	0	–	–	–	0	–	–	–	0	–	3.2	61

indicator helps in perceiving the rate of outsourcing in the experimented problems. To calculate it, the solutions obtained by CP are used since almost in all cases, CP results in better or equal solutions than MILP. It is worth noting that in filling the Tables 2 and 3, some quantities are rounded, to an ignorable extent, regarding the scale of the values and the required precision.

In addition to the numerical results, some quantities of Tables 2 and 3 are shown graphically in Figures 1–3. Each figure consists of 3 plots, each of which corresponding to a specified weight number. In the entire plots, the problem classes are put on the horizontal axes in order of their difficulty observed generally in the experiments. In these axes, the names of some problem classes are starred to show optimality solution states. In this respect, a problem class is signed by one star when at least 80% of its instances with the related weight are solved to optimality by the CP approach. If this occurrs for the MILP approach as well, then one other star is also added to the name of the problem class. Figure 1 compares the average objective values obtained by the MILP and CP approaches for the problem classes. The lower bounds attained by the MILP approach, relaxation 1 and relaxation 2 are depicted in Figure 2. Moreover, Figure 3 illustrates the indicators *gap*1 and *gap*2 presented in the result tables.

7.3. Results analysis

Now we analyse the experimental results reported in Tables 2 and 3. According to these results, MILP approach has solved all the problem instances with 5 or 10 machines and 10 jobs to optimality. Moreover, when the ratio of jobs outsourcing is high, it can prove solution optimality for some other instances with larger dimensions, even up to 20 machines and 20 jobs. CP significantly outperforms MILP in solving the problems to optimality. It proves solution optimality in much more instances than MILP and has considerably lower solution times when both approaches solve the problem to optimality. Indeed, the entire instances with 15 machines and 15 jobs, 5 machines and 20 jobs and almost all the instances with 10 machines and 20 jobs are solved optimally by the CP approach. Furthermore, when the outsourcing ratio is not low, CP attains optimality for many instances with 15 or 20 machines and 20 jobs, some instances with 20 or 25 machines and 25 jobs, and an instance with 20 machines and 30 jobs. Although MILP doesn't prove solution optimality well and normally reports large optimal gaps, it gives qualified solutions compared to the CP solutions. In fact, regarding the *dev* indicator in the Tables 2 and 3, and the plots of Figure 1, it is observed that the MILP solutions are close to the CP solutions, especially for the small and medium size problems. Moreover, as the ratio of the outsourced jobs increases, the gap between the MILP and CP solutions decreases, insofar as for the third weight, the MILP and CP solutions become almost equal for all the problem classes.

The weight of total outsourcing cost and the ratio of jobs outsourcing also affect the solution times of the MILP and CP models. In this respect, the results of Tables 2 and 3 indicate that normally the solution time is decreased when the outsourcing ratio is increased. Especially for the third weight which usually ends in more than 50% of jobs outsourcing ratio, the problem gets much easier to solve via both solution approaches.

To analyse the performance of the relaxation approaches, it is evident in Figure 2 that the lower bounds obtained by the relaxation approaches are significantly larger, *i.e.* stronger than the MILP's lower bounds in most of the problem classes. As previously stated, in order to obtain the lower bounds by the relaxation approaches, it is required to solve the relaxed problems to optimality. In this respect, it is observed that the relaxed problem in relaxation 2 is solved optimally in all the instances having up to 25 jobs, and some instances with 30 jobs in the case of high level outsourcing ratio. Moreover, relaxation 1 has been successful to give the intended lower bound for all the instances having up to 35 jobs. However, according to the indicator *sol#*, some single machine problems have not been solved optimally for some large scale problems, so the quality of the lower bound is degraded in these cases. This is occurred especially for the instances with 10 machines and 30 jobs or 15 machines and 35 jobs. The efficiency of the proposed relaxation approaches can be evaluated by investigating the deviations of the resulted lower bounds from optimality. In this regard, we can analyse the values of the indicators *gap*1 and *gap*2 for the problem instances for which the optimal solution is attained by CP. Accordingly, Figure 3 shows that for the starred problem classes in the horizontal axes, almost all of the relative average gaps for relaxation 2 is below 10%. Furthermore, according to the plot related to w_3, this value is even lower than 5% for the problem classes with high outsourcing ratio. In relaxation 1, the results are a bit worse so that the relative average gaps are below 15% for all the starred problem classes. Moreover, they are lower than 10% for the third weight. So, it is concluded that in both cases when the relaxed problems are solved optimally, the resulted lower bounds are close to the optimal solutions. It is worth mentioning that the obtained lower bounds are also useful in estimating the gaps of the obtained solutions from optimality when they are not proven to be optimal. For example, in Figure 3, for many non-starred problem classes, it is demonstrated by relaxation 1 or 2 that the obtained solutions, in average, have at most 15% to 20% error with respect to the optimal solutions. Accordingly, we can guess that the real average errors in these problem classes are around 5% to 10%, according to the observed optimal gaps for the lower bounds in the starred problem classes.

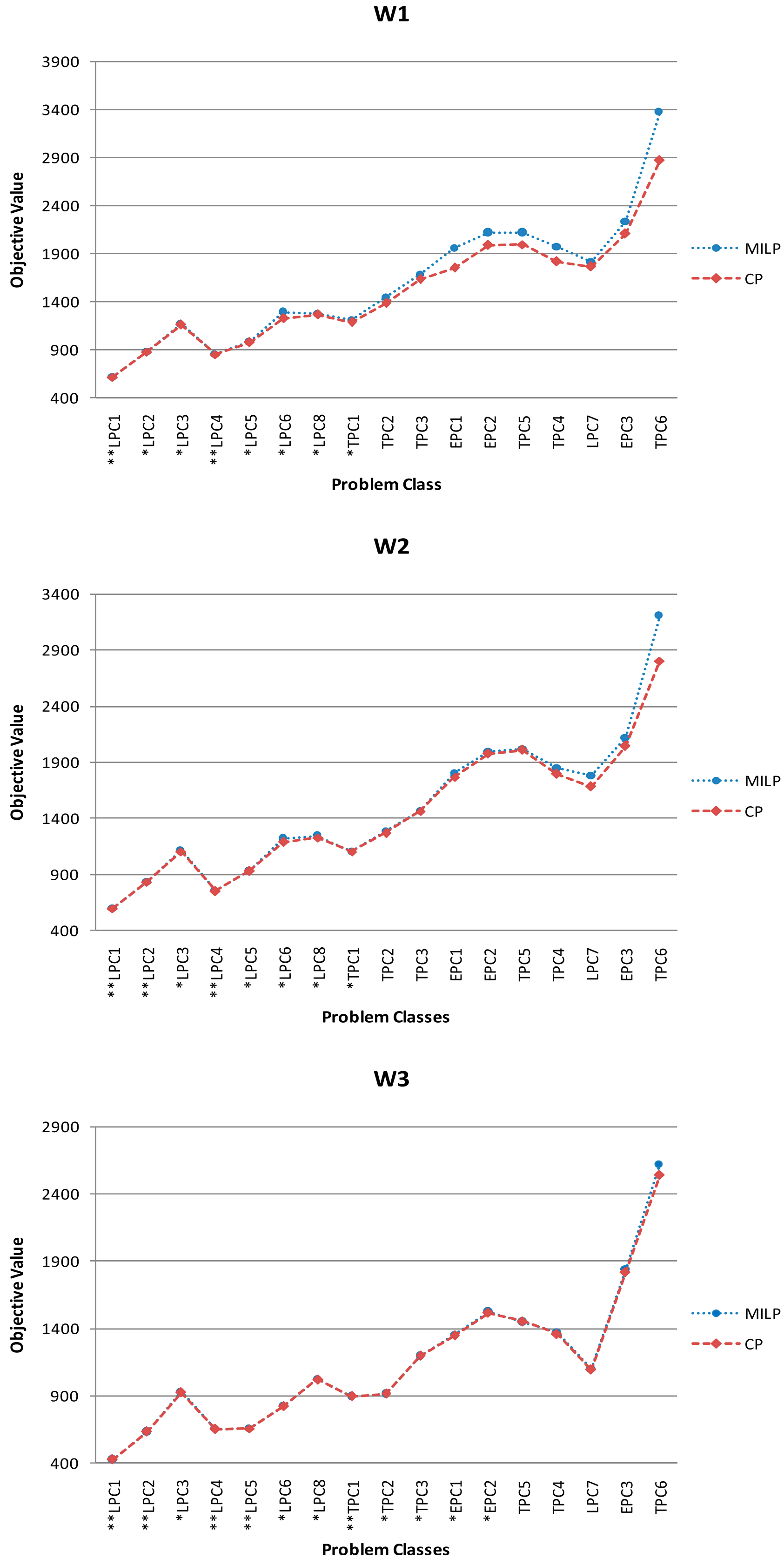

Figure 1. Diagrams of the average objective values obtained by MILP and CP approaches for the problem classes.

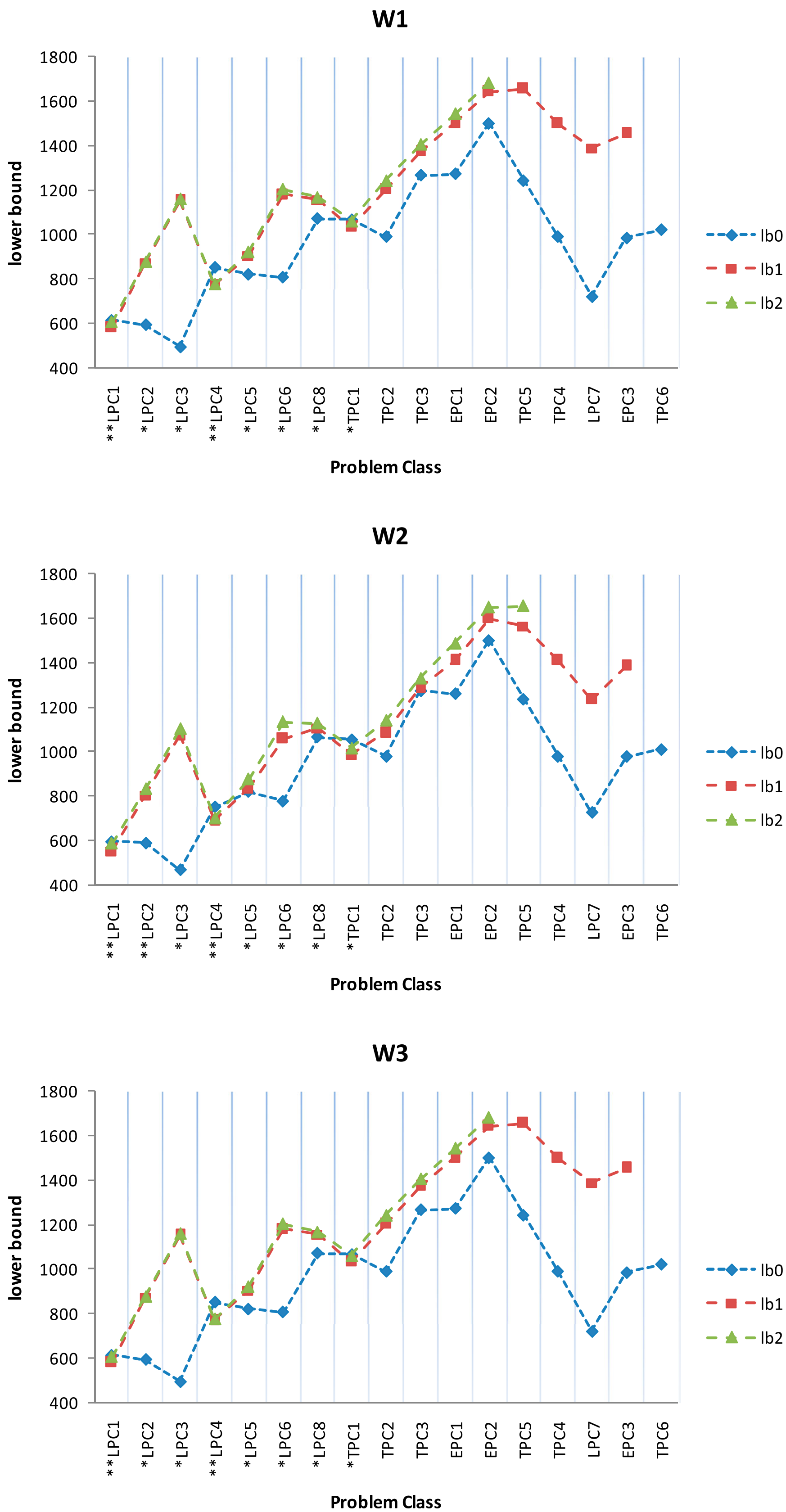

Figure 2. Diagrams of the generated lower bounds by MILP, relaxation 1 and relaxation 2 approaches.

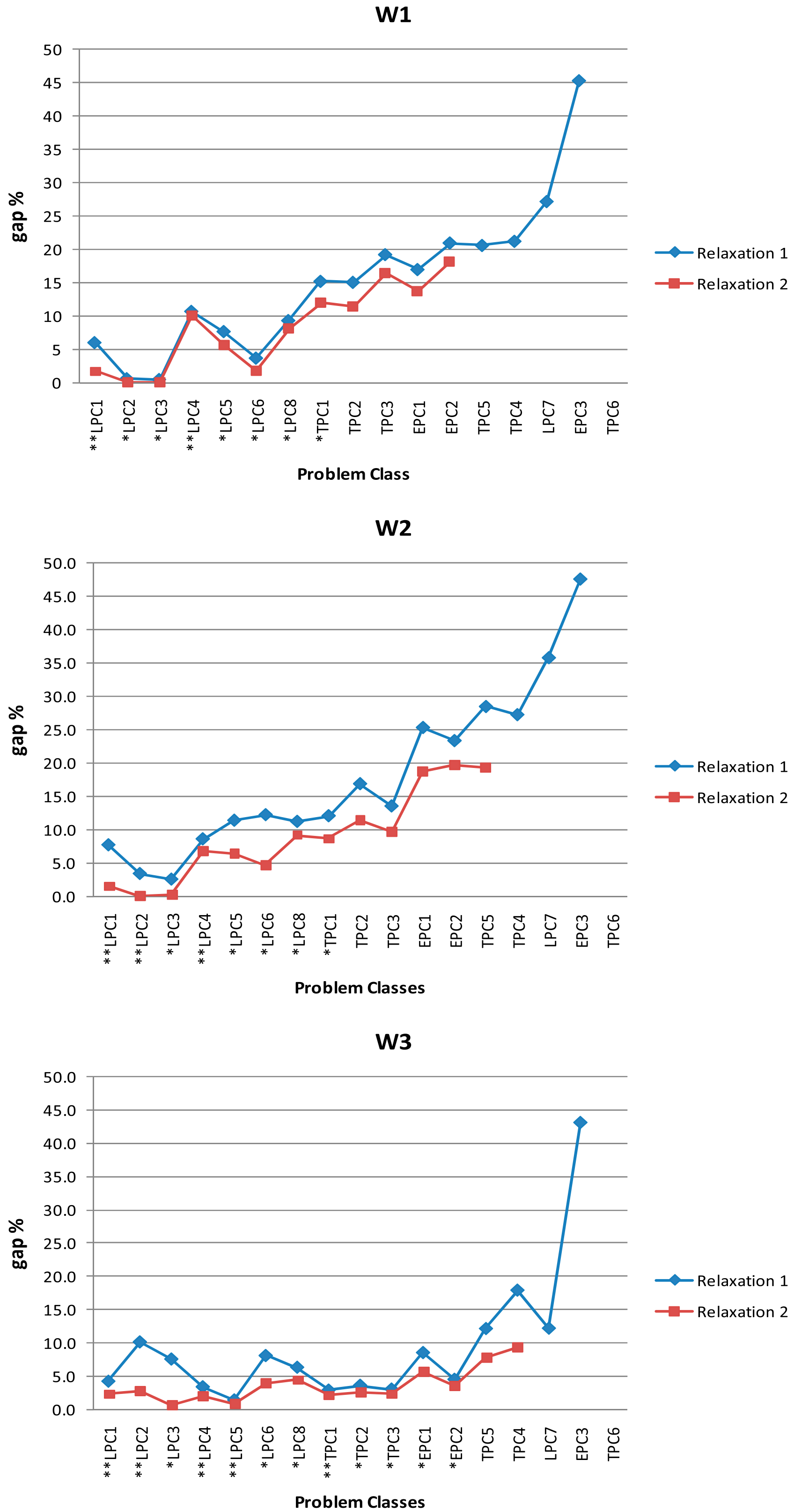

Figure 3. Diagrams of the optimal gaps generated by the relaxation approaches and CP solutions.

8. Conclusion

In this paper, a new job shop scheduling problem is studied with the option of jobs outsourcing. The objective function of the problem is a weighted sum of makespan and total outsourcing cost. A MILP model based on disjunctive formulation and a CP model using ILOG CP elements of scheduling are proposed to solve the problem optimally. Moreover, two relaxation approaches are developed for the problem. These approaches help to obtain strong lower bounds for some problem scales for which the exact methods cannot attain optimal solutions in a reasonable time.

Extensive computational experiments are conducted at the end of the paper to evaluate the performances of the proposed solution techniques. In this regard, two sets of JSSP test problems are extended for the studied problem. Furthermore, for each problem class, three different values are selected for the weight of total outsourcing cost in the objective function. These weights result in varied ratios of jobs outsourcing to analyse the effect of this factor on the performance of the solution techniques. For the original problem, it is observed that CP is significantly superior to MILP in proving solution optimality, and is applicable for much larger problem dimensions, *i.e.* medium size problems. Although MILP approach cannot prove solution optimality properly, the obtained solutions are close to the CP solutions especially in small and medium size problems and in the problems not having low levels of outsourcing. It is also concluded that the problem generally gets easier to solve by both MILP and CP approaches when the rate of jobs outsourcing is increased in the optimal solution, *i.e.* the weight of outsourcing cost is decreased. For the relaxation approaches, it is observed that both can obtain lower bounds for some problem scales for which CP cannot solve the original problem optimally. In these cases, the obtained lower bounds are useful to approximate the optimal objective value as well as the deviations of the resulted solutions from optimality. The efficiency of the relaxation approaches in approximating the lower bounds are evaluated using the instances that are solved to optimality via CP approach. In this respect, it is observed that the lower bounds reported by relaxations 1 and 2 generally have average optimal errors below 15% and 10% respectively. Moreover, when the outsourcing ratio is high, the mentioned percentages drop to 5% and 10% respectively.

For future studies, we propose examining other solution approaches like metaheuristic algorithms to tackle this newly studied problem, when it is not intended to prove solution optimality. Maybe the utilised solution methods in this paper can also be improved to obtain better results. As JSSP with outsourcing is studied only in three papers previously to the best of our knowledge, many other forms of this problem can be investigated to fill the research gap.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Apt, K. 2003. *Principles of Constraint Programming*. Cambridge: Cambridge University Press.

Baptiste, P., C. L. Pape, and W. Nuijten. 2012. *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*. New York: Springer Science & Business Media.

Benhamou, F., N. Jussien, and B. A. O'Sullivan. 2007. *Trends in Constraint Programming*. London: Wiley.

Brucker, P., and B. Jurisch. 1993. "A New Lower Bound for the Job-Shop Scheduling Problem." *European Journal of Operational Research*, Project Management and Scheduling 64: 156–167.

Çaliş, B., and S. Bulkan. 2015. "A Research Survey: Review of AI Solution Strategies of Job Shop Scheduling Problem." *Journal of Intelligent Manufacturing* 26: 961–973.

Carlier, J., and E. Pinson. 1989. "An Algorithm for Solving the Job-Shop Problem." *Management Science* 35: 164–176.

Chen, Z.-L., and C.-L. Li. 2008. "Scheduling with Subcontracting Options." *IIE Transactions* 40: 1171–1184.

Choi, B.-C., and J. Chung. 2011. "Two-Machine Flow Shop Scheduling Problem with an Outsourcing Option." *European Journal of Operational Research* 213: 66–72.

Choi, B.-C., and M.-J. Park. 2014. "Outsourcing Decisions in m-Machine Permutation Flow Shop Scheduling Problems with Machine-Dependent Processing Times." *Asia-Pacific Journal of Operational Research* 31 (4): 1450028.

Chung, D.-Y., and B.-C. Choi. 2013. "Outsourcing and Scheduling for Two-Machine Ordered Flow Shop Scheduling Problems." *European Journal of Operational Research* 226: 46–52.

Chung, D., K. Lee, K. Shin, and J. Park. 2005. "A New Approach to Job Shop Scheduling Problems with Due Date Constraints Considering Operation Subcontracts." *International Journal of Production Economics*, Production Research: Facing the Challenges in the New Millennium 98: 238–250.

CPLEX 12.6.0 Manual. *ILOG Reference of Modeling Scheduling Problems*. https://www.ibm.com/support/knowledgecenter/en/SSSA5P_12.6.0/ilog.odms.ide.help/OPL_Studio/opllangref/topics/opl_langref_scheduling.html.

Demir, Y., and S. Kürşat İşleyen. 2013. "Evaluation of Mathematical Models for Flexible Job-Shop Scheduling Problems." *Applied Mathematical Modelling* 37: 977–988.

Gass, S. I., and M. C. Fu. 2013. *Encyclopedia of Operations Research and Management Science*. New York: Springer US.

Gedik, R., C. Rainwater, H. Nachtmann, and E. A. Pohl. 2016. “Analysis of a Parallel Machine Scheduling Problem with Sequence Dependent Setup Times and Job Availability Intervals.” *European Journal of Operational Research* 251: 640–650.

Guo, X., and D. Lei. 2014. “Bi-Objective Job Shop Scheduling with Outsourcing Options.” *International Journal of Production Research* 52: 3832–3841.

Ham, A., J. W. Fowler, and E. Cakici. 2017. “Constraint Programming Approach for Scheduling Jobs with Release Times, Non-Identical Sizes, and Incompatible Families on Parallel Batching Machines.” *IEEE Transactions on Semiconductor Manufacturing* 30: 500–507.

Hong, J. M., and J. H. Lee. 2016. “Outsourcing Decisions in Single Machine Scheduling Problem with Multiple External Facilities.” *Journal of Marine Science and Technology* 24: 603–609.

Keha, A. B., K. Khowala, and J. W. Fowler. 2009. “Mixed Integer Programming Formulations for Single Machine Scheduling Problems.” *Computers & Industrial Engineering* 56: 357–367.

Ku, W.-Y., and J. C. Beck. 2016. “Mixed Integer Programming Models for Job Shop Scheduling: A Computational Analysis.” *Computers & Operations Research* 73: 165–173.

Laborie, P., and J. Rogerie. 2008. “Reasoning with Conditional Time-Intervals.” Presented at the In FLAIRS Conference, 555–560.

Laborie, P., and J. Rogerie. 2009. “Reasoning with Conditional Time-Intervals. Part II: An Algebraical Model for Resources.” Presented at the In FLAIRS Conference, 201–206.

Lageweg, B. J., J. K. Lenstra, and A. H. G. R. Kan. 1976. “Minimizing Maximum Lateness on One Machine: Computational Experience and Some Applications.” *Statistica Neerlandica* 30: 25–41.

Lawrence, S. 1982. *Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques*. Pittsburgh: Graduate School of Industrial Administration, Carnegie Mellon University.

Lee, I. S., and C. S. Sung. 2008a. “Minimizing Due Date Related Measures for a Single Machine Scheduling Problem with Outsourcing Allowed.” *European Journal of Operational Research* 186: 931–952.

Lee, I. S., and C. S. Sung. 2008b. “Single Machine Scheduling with Outsourcing Allowed.” *International Journal of Production Economics* 111 (2): 623–634.

Lei, D., and X. Guo. 2016. “A Shuffled Frog-Leaping Algorithm for Job Shop Scheduling with Outsourcing Options.” *International Journal of Production Research* 54: 4793–4804.

Liu, Z., W.-C. Lee, and J.-Y. Wang. 2016. “Resource Consumption Minimization with a Constraint of Maximum Tardiness on Parallel Machines.” *Computers & Industrial Engineering* 97: 191–201.

Lustig, I. J., and J.-F. Puget. 2001. “Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming.” *Interfaces* 31: 29–53.

Mokhtari, H., and I. N. K. Abadi. 2013. “Scheduling with an Outsourcing Option on Both Manufacturer and Subcontractors.” *Computers & Operations Research* 40: 1234–1242.

Mokhtari, H., I. N. K. Abadi, and M. R. Amin-Naseri. 2012. “Production Scheduling with Outsourcing Scenarios: A Mixed Integer Programming and Efficient Solution Procedure.” *International Journal of Production Research* 50: 5372–5395.

Neto, R. F. T., M. G. Filho, and F. M. da Silva. 2015. “An Ant Colony Optimization Approach for the Parallel Machine Scheduling Problem with Outsourcing Allowed.” *Journal of Intelligent Manufacturing* 26: 527–538.

Özgüven, C., L. Özbakır, and Y. Yavuz. 2010. “Mathematical Models for Job-Shop Scheduling Problems with Routing and Process Plan Flexibility.” *Applied Mathematical Modelling* 34: 1539–1548.

Pinedo, M. L. 2012. *Scheduling: Theory, Algorithms, and Systems*. New York: Springer Science & Business Media.

Qi, X. 2008. “Coordinated Logistics Scheduling for In-House Production and Outsourcing.” *IEEE Transactions on Automation Science and Engineering* 5: 188–192.

Qi, X. 2009. “Two-Stage Production Scheduling with an Option of Outsourcing from a Remote Supplier.” *Journal of Systems Science and Systems Engineering* 18: 1–15.

Qi, X. 2011. “Outsourcing and Production Scheduling for a Two-Stage Flow Shop.” *International Journal of Production Economics* 129: 43–50.

Taillard, E. D. 1993. “Benchmarks for Basic Scheduling Problems.” *European Journal of Operational Research* 64: 278–285.

Tavares Neto, R. F., and M. Godinho Filho. 2011. “An Ant Colony Optimization Approach to a Permutational Flowshop Scheduling Problem with Outsourcing Allowed.” *Computers & Operations Research* 38: 1286–1293.

Unlu, Y., and S. J. Mason. 2010. “Evaluation of Mixed Integer Programming Formulations for Non-Preemptive Parallel Machine Scheduling Problems.” *Computers & Industrial Engineering* 58: 785–800.

Unsal, O., and C. Oguz. 2013. “Constraint Programming Approach to Quay Crane Scheduling Problem.” *Transportation Research Part E: Logistics and Transportation Review* 59: 108–122.

Zeballos, L. J., J. M. Novas, and G. P. Henning. 2011. “A CP Formulation for Scheduling Multiproduct Multistage Batch Plants.” *Computers & Chemical Engineering* 35: 2973–2989.

Zhong, W., and Z. Huo. 2013. “Single Machine Scheduling Problems with Subcontracting Options.” *Journal of Combinatorial Optimization* 26: 489–498.