Scheduling with Subcontracting Options

Zhi-Long Chen[1]

Department of Decision and Information Technologies
Robert H. Smith School of Business
University of Maryland
College Park, Maryland 20742-1815
Email: *zchen@rhsmith.umd.edu*

Chung-Lun Li[2]

Department of Logistics, Faculty of Business
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
Email: *lgtclli@polyu.edu.hk*

October 2006
Revised: May 2007
Revised: December 2007
Final Version: January 2008

[1] Supported in part by the National Science Foundation under grants DMI-0196536 and DMI-0421637
[2] Supported in part by the Research Grants Council of Hong Kong under grant PolyU6132/02E

ABSTRACT

Motivated by a problem commonly faced by time-sensitive product manufacturers, we propose an analytical model to study the joint decisions of subcontracting and detailed job scheduling. In our model, a manufacturer operates in a make-to-order fashion and receives a set of orders from its customers at the beginning of the planning horizon. The orders can be either processed by the manufacturer in-house or subcontracted to one of several available subcontractors, possibly at a higher cost. The manufacturer needs to determine which orders should be produced in-house and which orders should be subcontracted. Furthermore, it needs to determine a production schedule for the orders to be produced in-house. The objective is to minimize the total production and subcontracting cost, subject to a constraint on the maximum completion time of the orders. We analyze the computational complexity of the model, develop a heuristic for solving it, and analyze worst-case and asymptotic performances of the heuristic. We also study the value of subcontracting by comparing our model and a model where no subcontracting option is available to the manufacturer. Computational results demonstrate that the subcontracting option gives the manufacturer a significant performance improvement. Related managerial insights are also provided.

Key words*: Scheduling; subcontracting; computational complexity; worst-case analysis; asymptotic analysis*

1. INTRODUCTION

Subcontracting is the procurement of an item or service that a firm is normally capable of producing using its own facilities and that requires the firm to make specifications available to the supplier (Day 1956). A more popular term, “outsourcing,” refers to the special case of subcontracting where the manufacturer has no in-house capability and is dependent on the subcontractor for the entire product volume. The practice of subcontracting and outsourcing is widespread in many industries because of the many advantages this can bring to a firm. When a firm subcontracts out some of its tasks, this allows it to concentrate on its core competencies. Subcontracting lowers investment requirements, and thus, the financial risk of the firm. It also helps the firm improve its response to customer demand. Furthermore, if a firm subcontracts an entire operation to a subcontractor, the demand uncertainty of the supply chain is reduced through the risk-pooling effect.

In making subcontracting decisions, a firm needs to take into account many factors including, among others, its capacity and the cost for in-house production, customer demand, available subcontractors and their production costs and delivery lead times, and so forth. Clearly, analytical models and problem-solving tools are needed if a firm is to optimize the tradeoffs from those factors.

In this paper, we present an analytical model to study the joint decisions of subcontracting and detailed job scheduling. Our model is motivated by the following problem commonly faced by manufacturers who make time-sensitive products, such as toys, fashion apparel, and high-tech consumer electronic products that typically have a large variety, a short life-cycle, and are sold in a very short selling season. On the one hand, because of the high demand uncertainty of the products, retailers typically do not place orders until reliable market information is available shortly before a selling season. On the other hand, since there are significant markdowns for unsold products at the end of the selling season, the manufacturer runs a high risk if it starts production early, before it has some information about retailers’ orders. As a result, the manufacturer will operate in a make-to-order fashion; i.e., it will not start production until orders from the retailers have been placed shortly before the selling season. Due to the fact that the available production time is limited, the manufacturer’s own capacity is usually not enough to satisfy a peak in demand in the upcoming selling season. Consequently, a good strategy often used by manufacturers is to subcontract some orders to subcontractors, but possibly at a higher cost than if the products had been produced in-house. Given a set of distinct orders from the retailers, the manufacturer needs to determine which orders should be produced in-house and which should be subcontracted. Furthermore, the manufacturer needs to determine a production schedule for the orders to be produced in-house. In order to deliver the orders to the retailers as soon as possible before the selling season ends, while controlling the production and subcontracting costs, the manufacturer needs to consider subcontracting and scheduling simultaneously.

In our model, there are multiple identical production lines available at the manufacturer's own plant where the in-house orders are processed, and there are multiple subcontractors available, each capable of processing all the orders. The objective of the model is to minimize the total cost of production and subcontracting, subject to a constraint on the delivery lead time of the orders. We analyze the computational complexity of the model, develop a heuristic for solving it, and analyze worst-case and asymptotic performances of the heuristic. We also study the value of subcontracting by conducting a computational comparison between our model and a model where no subcontracting option is available to the manufacturer. As our literature review below indicates, this is one of few papers that examine the joint decisions of subcontracting and detailed order scheduling, and is the first paper that studies the value of subcontracting for this type of problem.

In the following we give a brief review of related studies. A large body of literature discusses the benefits and issues of subcontracting qualitatively. Some recent articles include, among others, Ioannou (1995), Webster *et al.* (1997), Bazinet *et al.* (1998), Craumer (2002), and Kolawa (2004). However, analytical models that study issues and decisions related to subcontracting are limited. In the literature on operations management, most existing analytical models study strategic or aggregate production planning decisions in a make-to-stock environment where inventory plays an important role in meeting customer demand. Kamien and Li (1990) introduced a multi-period game-theoretic model that incorporates subcontracting and aggregate production planning decisions. They showed that the option of subcontracting results in production smoothing. Van Mieghem (1999) studied a single-period game-theoretic model with subcontracting, production, and capacity-investment decisions. Atamturk and Hochbaum (2001) analyzed dynamic lot-sizing problems with tradeoffs among capacity acquisition, subcontracting, production, and inventory holding. Kogan (2000), Bradley (2004), and Tan and Gershwin (2004) presented optimal control models for aggregate production, inventory, and subcontracting decisions. Yang *et al*. (2005) studied a similar production-inventory-outsourcing model with Markovian in-house production capacity. Alp and Tan (2008) considered an integrated capacity management and inventory planning model where in addition to permanent capacity, contingent capacity can be acquired by hiring temporary workers (which can be viewed as some type of subcontracting). Lee *et al.* (1997), Logendran and Puvanunt (1997), and Logendran and Ramakrishna (1997) considered subcontracting jointly with production planning decisions in the context of cellular and flexible manufacturing systems. Our model differs from all these existing models because they consider a make-to-stock environment and study aggregate planning decisions, whereas our model considers detailed scheduling decisions in a make-to-order environment with no finished product inventory involved.

There are a handful of existing articles that incorporate subcontracting into detailed scheduling decisions. Bertrand and Sridharan (2001) considered a make-to-order manufacturing environment where orders arrive over time randomly, and can either be processed in-house on a single machine or subcontracted. The objective is to maximize the utilization of in-house capacity while minimizing tardiness in fulfilling orders. Simple heuristic rules are proposed and computational results are reported. Lee *et al.* (2002) proposed a multi-stage scheduling model where each order requires multiple operations and each operation can be processed on a number of alternative machines in-house or subcontracted. The objective is to minimize the makespan for completing a given set of orders. They proposed a genetic algorithm based heuristic solution approach. Chung *et al.* (2005) considered a job shop scheduling problem where each order has a due date which must be satisfied, but operations of orders can be subcontracted at a certain cost. The objective is to minimize the total subcontracting cost. They proposed a heuristic algorithm. Qi (2007a) studied a problem where there is a single in-house machine and a single subcontractor with a single machine. Subcontracted orders need to be shipped back in batches. The objective is to minimize the weighted sum of a delivery lead time performance measure of orders and total subcontracting and transportation cost. He proposed dynamic programming algorithms for four problems where the time-based performance measure is total completion time, makespan, maximum lateness, and number of tardy orders, respectively. Qi (2007b) studied a two-stage flow shop scheduling problem with options of subcontracting some operations to subcontractors. He considered a minimum makespan objective and analyzed various models for different situations of subcontracting. Bukchin and Hanany

(2007) considered a decentralized scheduling problem involving competition among multiple decision makers, each having a set of jobs to be processed either by an in-house common machine or by a subcontractor. The objective is to minimize the total completion time of the jobs. They analyzed the ratio between the objective value of the decentralized problem and the objective value of the centralized system.

Our model differs from the above-reviewed models that integrate subcontracting with scheduling decisions in the following aspects. The machine configuration at the in-house plant in our model follows a parallel-machine configuration, whereas all the existing models consider a different configuration. Similar to the model of Qi (2007a), our model takes into account both customer service level and total cost. The other existing models do not consider both performance measures. However, it is assumed in Qi (2007a) that the subcontracting cost is always more expensive than the in-house production cost for processing a job, whereas we do not make such an assumption. Also, in Qi's model, there is only one subcontractor available and the subcontractor has only a single machine, whereas in our model there are multiple subcontractors available. In addition, we study the value of subcontracting and derive related managerial insights, while none of the existing papers that consider detailed scheduling does this. Furthermore, our solution approaches and analyses are different from the ones used in the existing papers.

The rest of this paper is organized as follows: In Section 2, we provide a formal description of our problem. In Section 3, we show that the problem is NP-hard even with a special structure, and design an efficient heuristic for it. We analyze the worst-case and asymptotic performances of the heuristic. In Section 4, we conduct computational experiments to evaluate the performance of the proposed heuristic and examine the value of subcontracting. Computational results show that the heuristic often generates optimal solutions. Also, we find that subcontracting option leads to a significant improvement in performance for the manufacturer. In Section 5, we discuss other variants of our model. We conclude our paper in Section 6. Proofs of all theorems and lemmas are provided in the Appendix.

2. MODEL DESCRIPTION

At the beginning of a planning horizon, a manufacturer receives a set of n jobs (i.e., customer orders with distinct product configurations) $N = \{1,2,\ldots,n\}$ from its customers downstream in the supply chain. The manufacturer can either process each job at its own plant or subcontract it to a subcontractor for processing. There are m identical parallel machines (i.e., production lines) $M = \{1, 2, \ldots, m\}$ available at the manufacturer's plant, such that if a job is processed at the manufacturer's plant, it only needs to be processed by one of these machines. There are k subcontractors $K = \{1, 2, \ldots, k\}$ available. We assume that each job is indivisible; that is, it has to be processed either by an in-house machine as a whole or subcontracted to one of the subcontractors entirely. This assumption makes sense in many practical situations where there are a large number of product configurations such that each job represents a distinct configuration and needs to be processed and delivered together in order to minimize setup and coordination costs. It is also reasonable for situations where each job comes from a different customer and it must be handled by the same entity (either the in-house facility or a subcontractor) for convenience of accounting, packaging, and delivery. Such an assumption appears in all of the papers reviewed in Section 1 that consider joint scheduling and subcontracting decisions.

If job $j \in N$ is processed at the manufacturer's own plant, a processing time of p_{0j} units is required and the cost to the manufacturer is q_{0j} dollars. If job $j \in N$ is subcontracted to subcontractor $h \in K$, then the delivery lead time is p_{hj} (i.e., it takes p_{hj} units of time to complete the job using subcontractor h) and the cost incurred is q_{hj} dollars. We note that we do not explicitly define the workload or size of each job. However, the workload of a job $j \in N$ can be easily incorporated into the time and cost parameters p_{0j}, q_{0j}, p_{hj}, and q_{hj}. We can set these parameters such that they depend on (e.g., increase with) the workload of job j. For example, if the workload of job j is w_j, a possible instance of these parameters can be: $p_{ij} = \alpha_i w_j$, and $q_{hj} = \beta_i w_j$ for $i \in \{0,1,..., K\}$ and $j \in N$, where $\alpha_i, \beta_i > 0$ are facility-dependent parameters.

We assume that for each subcontractor $h \in K$, the delivery lead time p_{hj} and subcontracting cost q_{hj} for a job $j \in N$ are independent of the total workload of the subcontractor. This assumption is made and justified in some of the existing papers in the literature. For example, Bertrand and Sridharan (2001) and Buckchin and Hanany (2007) make the same assumption on the subcontracting lead time of a job. Buckchin and Hanany assume that the subcontractor has an unlimited capacity and that the subcontracting lead time of a job is a constant factor multiplied by the in-house processing time of the job (which is independent of the total workload of the subcontractor). They justify the unlimited capacity assumption by stating that "The assumption that the subcontractor has unlimited capacity is adequate when the subcontractor's available capacity, related to the type of jobs in question, is significantly higher than the firm's required capacity. In this case, a contract can be signed that clearly defines a due-date commitment on the subcontractor's side." In the models of Chung *et al.* (2005), Buckchin and Hanany (2007), and Qi (2007a), it is also assumed that the subcontracting cost of a job is independent of the other jobs processed by the same subcontractor. This assumption is appropriate if each subcontractor makes independent offers to the individual jobs received from the manufacturer with a promise in delivering a finished job within a certain deadline at a certain cost, where the deadline and cost for a job are determined by the workload of the job independent of the other jobs.

In our model, the manufacturer needs to determine (i) a subset of jobs to be subcontracted to each of the k subcontractors, and (ii) a production schedule for the jobs processed on the m machines at its own plant, such that a certain delivery lead performance is guaranteed and the total production cost is minimized. Each subcontractor takes care of its own production scheduling for the jobs subcontracted from the manufacturer and delivers completed jobs to the manufacturer by the pre-specified lead times.

The problem can be viewed as a generalized version of the classical parallel-machine scheduling problems. In classical parallel-machine scheduling problems, no production costs are involved, while some performance measure of job completion times is optimized. In our problem, the objective is to minimize the total operating costs, and each subcontractor $h \in K$ can be viewed as having a sufficient number (e.g., n) of identical parallel machines, such that each of these machines will handle at most one job.

Given a solution to this problem, let C_j denote the completion time of job $j \in N$. If job $j \in N$ is subcontracted to some subcontractor $h \in K$, then $C_j = p_{hj}$. If job $j \in N$ is processed in-house, then C_j is determined by the schedule of the jobs processed by the manufacturer itself. Let $C_{\max} = \max\{C_j \mid j \in N\}$ denote the maximum completion time of the jobs (also known as makespan, which is the time when all the jobs are completed) in a given solution. We note that makespan $C_{\max}$ is one of the most widely used delivery lead time performance measures in the scheduling literature (see, e.g., Pinedo 2002). In our case, $C_{\max}$ represents the time where all the customer orders are completed. By imposing a constraint on $C_{\max}$, the production for a particular season (for fashion apparel) or the production of a particular product line (for short life-cycle toys and high-tech consumer products) is completed within a given time limit. Let G denote the total cost of a given solution including the production costs at the manufacturer's plant and the subcontracting costs. Our problem is to minimize the total cost G subject to the constraint of $C_{\max} \le C$, where C is a pre-specified threshold value. We denote this problem as $Min\{G \mid C_{\max} \le C\}$. This problem can be used to model situations where there is a desired level of customer service, represented by the constraint that all the jobs must be completed within a certain time frame C. It can also be used to determine all Pareto-optimal solutions with respect to G and $C_{\max}$ if we solve this problem repeatedly for different values of C.

3. ANALYSIS OF PROBLEM $Min\{G \mid C_{\max} \le C\}$

We first show that this problem is NP-hard even with a special structure. Then, we propose a heuristic and analyze its performance.

Theorem 1: Problem $Min\{G \mid C_{\max} \le C\}$ is NP-hard even if there is only a single machine at the manufacturer, there is a single subcontractor, and the in-house production costs are proportional to the production times.

We now propose a heuristic for this problem with a general number of parallel machines at the manufacturer and multiple subcontractors. The heuristic is based on an integer programming formulation of the problem, and exploits a special property of the LP-relaxation of the IP formulation.

We note that linear programming based approaches have been used in the literature in developing approximation algorithms for classical parallel machine problems without subcontracting options but with makespan as the objective function (Potts 1985 and Lenstra *et al.* 1990) or as part of the objective function (Shmoys and Tardos 1993). However, the technical details we develop here are different from the ones in these papers. Our problem involves production costs, whereas the problems considered by Potts and Lenstra *et al.* do not. Hence, their techniques do not work for our problem. The techniques developed by Shmoys and Tardos may be applicable to our problem. However, their approach requires solving up to $\log(nP_{\max})$ LP-relaxation problems, where $P_{\max}$ is the maximum processing time of jobs. So, their approach is not strongly polynomial. As we will see below, our approach only requires solving one LP-relaxation problem. As a result, our heuristic is strongly polynomial, and is therefore much more efficient than their solution method. Furthermore, we develop a simple procedure to solve the LP-relaxation problem involved in our approach, whereas all the existing papers rely on an LP solver (ellipsoid method).

Finally, we use a different approach from the ones used in these papers to scheduling the jobs for which the variables in the LP-relaxation problem have fractional values.

For each job $j \in N$, define a subset of subcontractors $S_j = \{h \in K \mid p_{hj} \le C\}$, which are the subcontractors that can process job j with a delivery time no later than the required threshold of the makespan C. Define $V = \{j \in N \mid S_j \ne \emptyset\}$, which is the set of jobs that can be subcontracted. For each $j \in V$, define a subcontractor $s_j = \arg\min_{v \in S_j} \{q_{vj}\}$, which is the subcontractor that can process job j at a minimum cost with a delivery time no later than C. The problem can then be formulated as the following mixed integer linear program:

$$
\begin{aligned}
IP(C)\textbf{:} \quad & \text{Minimize} \quad && \sum_{i=1}^{m} \sum_{j=1}^{n} q_{0j} x_{ij} + \sum_{j \in V} q_{s_j j} y_j && (1) \\
& \text{subject to} && \sum_{i=1}^{m} x_{ij} + y_j = 1, \text{ for } j \in V && (2) \\
& && \sum_{i=1}^{m} x_{ij} = 1, \text{ for } j \in N \setminus V && (3) \\
& && C_{\max} \ge \sum_{j=1}^{n} p_{0j} x_{ij}, \text{ for } i \in M && (4) \\
& && C_{\max} \le C && (5) \\
& && x_{ij} \in \{0,1\}, \text{ for } i \in M \text{ and } j \in N && (6) \\
& && y_j \in \{0,1\}, \text{ for } j \in V. && (7)
\end{aligned}
$$

In this formulation, each binary variable x_{ij} is defined to be 1 if job j is assigned to machine $i \in M$ at the manufacturer's own plant, and 0 otherwise. Variable y_j is defined to be 1 if job j is assigned to subcontractor s_j, and 0 otherwise. Variable $C_{\max}$ is the makespan. Objective function (1) is the total production and subcontracting cost. Constraints (2) and (3) ensure that each job $j \in V$ is either assigned to an in-house machine or subcontracted to s_j, and that each job $j \in N \setminus V$ is assigned to an in-house machine. Constraints (4) and (5) define the makespan and ensure that the threshold on the makespan is not violated. We will focus on the LP-relaxation problem of $IP(C)$, denoted as $LP(C)$, which is the same formulation as $IP(C)$ except that variables x_{ij} and y_j are allowed to take any nonnegative values. Constraints (2) and (3) enforce these variables to take values between 0 and 1. We have the following result.

Lemma 1: In an optimal basic solution of $LP(C)$, if $n \ge m$ then among all x_{ij} and y_j variables, at least $n - m$ of them take the value of 1.

Lemma 1 implies that in an optimal basic solution of $LP(C)$, there are at most m jobs with some fractional values assigned to the corresponding x_{ij} or y_j variables. We show in the following that we do not have to use the simplex method to obtain an optimal basic solution of $LP(C)$. Instead, there is a fairly simple polynomial-time algorithm that can find an optimal basic solution of $LP(C)$ with at most m jobs having fractional values of the corresponding x_{ij} or y_j variables.

We first consider the following auxiliary problem, denoted as AUX : We are given a set of t jobs $T = \{1,2,\ldots,t\}$ to be scheduled on the m parallel identical machines at the manufacturer, where each job $j \in T$ has a processing time e_j. Each job j can be split into subjobs, where each subjob is allowed to have a fractional amount of processing time as long as the total processing time of all subjobs of job j is equal to e_j. The subjobs of a job are mutually independent and can be processed on different machines at the same time. The problem is to split at most m jobs into subjobs and find a schedule for all the resulting subjobs and unsplit jobs on the m machines, such that the makespan of the schedule is equal to $E_T = \frac{1}{m}\sum_{j\in T} e_j$.

We now present a procedure for solving problem AUX. The idea is quite simple. We assign jobs sequentially to machines, and divide a job into subjobs when necessary in order to make the total processing time of each machine equal to E_T. A detailed description of the procedure is given as follows.

Procedure $P1$ for Problem AUX :

Step 1: For $i = 1,2,\ldots,m$, determine $z(i) = \min\{u \mid \sum_{j=1}^{u} e_j \ge iE_T\}$, $b_1(i) = iE_T - \sum_{j=1}^{z(i)-1} e_j$, and $b_2(i) = \sum_{j=1}^{z(i)} e_j - iE_T$. (Note that $b_1(i) > 0$, $b_2(i) \ge 0$, and $b_1(i) + b_2(i) = e_{z(i)}$.)

Step 2: Assign jobs $1,2,\ldots,z(1)$ to machine 1. If $b_2(1) > 0$, then split job $z(1)$ into two subjobs with $b_1(1)$ units of processing time in the first subjob and $b_2(1)$ units of processing time in the second subjob, and then remove the second subjob from machine 1 while retaining the first subjob on the machine.

Step 3: For $i = 2,3,\ldots,m$, assign the leftover subjob of job $z(i-1)$, along with jobs $z(i-1)+1, z(i-1)+2,\ldots,z(i)$, to machine i. If $b_2(i) > 0$, then consider the following cases:

Case (i): If $z(i) > z(i-1)$, then split job $z(i)$ into two subjobs with $b_1(i)$ units of processing time in the first subjob and $b_2(i)$ units of processing time in the second subjob. Remove the second subjob from machine i and retain the first subjob on the machine.

Case (ii): If $z(i) = z(i-1)$, then further split the subjob of $z(i)$ that has been assigned to machine i

into two subjobs, with E_T units of processing time in the first subjob and $b_2(i)$ units of processing time in the second subjob. Remove the second subjob from machine i and retain the first subjob on the machine.

Step 4: For $i = 1,2,\ldots,m$, schedule the jobs and subjobs assigned to machine i in an arbitrary sequence without inserting any idle time between jobs.

In Step 2, at most one job is split into two subjobs. In each iteration of Step 3, at most one new job is split into subjobs. In the last iteration of Step 3 (i.e., when $i = m$), no job is split into subjobs. Hence, at most $m-1$ jobs are split into subjobs by this procedure. In Step 1, $z(i)$, $b_1(i)$, and $b_2(i)$ ($i = 1,2,\ldots,m$) can be determined recursively in $O(t+m)$ time. It is easy to see that Steps 2 and 3 can be implemented in $O(t+m)$ time. Hence, procedure $P1$ requires $O(t+m)$ time.

The following algorithm either identifies the infeasibility of problem $LP(C)$ or finds an optimal solution of the problem if it is feasible. In this algorithm, U denotes the set of in-house jobs and subjobs, while N_1 and N_2 denote the sets of unsplit and split jobs, respectively, as defined in the proof of Lemma 1.

Procedure $P2$ for $LP(C)$:

Step 1: Set $U \leftarrow N$, $N_1 \leftarrow \varnothing$, and $N_2 \leftarrow \varnothing$.

Step 2: For each job $j \in V$, if $q_{s_j j} \le q_{0j}$, then assign job j to subcontractor s_j, i.e., set $y_j = 1$ and $x_{ij} = 0$ for $i \in M$, and set $U \leftarrow U \setminus \{j\}$ and $N_1 \leftarrow N_1 \cup \{j\}$. Let $W = U \cap V$, which is the set of jobs that have not been subcontracted but can be subcontracted if needed.

Step 3: If $\frac{1}{m}\sum_{j\in U} p_{0j} > C$ and W is empty, then $LP(C)$ is infeasible and stop. Otherwise, if $\frac{1}{m}\sum_{j\in U} p_{0j} \le C$, then all the jobs in U are assigned to the manufacturer's own plant. Call procedure $P1$ to schedule these jobs onto the m machines. This results in at most $m-1$ jobs being split into subjobs. These jobs are added to set N_2. The other jobs are added to set N_1. For each job $j \in U \cap N_1$, let $x_{ij} = 1$, where i is the machine to which job j is assigned. For each job $j \in U \cap N_2$ and each machine $i \in M$, let $x_{ij} = \tau_{ij} / p_{0j}$, where τ_{ij} is the total processing time of the subjobs of job j assigned to machine i. Stop.

Step 4: If $\frac{1}{m}\sum_{j\in U} p_{0j} > C$ and W is not empty, then we may need to subcontract some jobs in W.

Compute $\Delta = \sum_{j \in U} p_{0j} - mC$, which is the total processing time of jobs in W that needs to be subcontracted in an optimal solution of $LP(C)$. Suppose that there are w jobs in W. Reindex the jobs in W as $[1],[2],\ldots,[w]$ such that

$$\frac{q_{s_{[1]},[1]} - q_{0,[1]}}{p_{0,[1]}} \le \frac{q_{s_{[2]},[2]} - q_{0,[2]}}{p_{0,[2]}} \le \cdots \le \frac{q_{s_{[w]},[w]} - q_{0,[w]}}{p_{0,[w]}}.$$

Set $v \leftarrow 1$, and consider the following two cases:

(a) If $\Delta = 0$, then call procedure $P1$ to schedule all of the jobs in U on the m machines at the manufacturer's plant. This results in at most $m-1$ split jobs. These jobs are added to set N_2. The other jobs are added to set N_1. For each job $j \in U \cap N_1$, set $x_{ij} = 1$, where i is the machine to which job j is assigned. For each job $j \in U \cap N_2$ and each machine $i \in M$, let $x_{ij} = \tau_{ij} / p_{0j}$, where τ_{ij} is the total processing time of the subjobs of job j assigned to machine i. Stop.

(b) If $\Delta > 0$, then consider the following two subcases:

(i) If $\Delta < p_{0,[v]}$, then split job $[v]$ into two subjobs $g_1([v])$ and $g_2([v])$ with the processing times being $p_{0,[v]} - \Delta$ and Δ, respectively. Assign subjob $g_2([v])$ to subcontractor $s_{[v]}$ (and let $y_{[v]} = \Delta / p_{0,[v]}$). Set $U \leftarrow (U \setminus \{[v]\}) \cup \{g_1([v])\}$. Call procedure $P1$ to schedule all of the jobs in U (including subjob $g_1([v])$) on the m machines at the manufacturer's plant. This results in at most m split jobs (including job $[v]$). These jobs are added to set N_2. All the other jobs are added to set N_1. For each job $j \in U \cap N_1$, set $x_{ij} = 1$, where i is the machine to which job j is assigned. For each job $j \in U \cap N_2$ and each machine $i \in M$, let $x_{ij} = \tau_{ij} / p_{0j}$, where τ_{ij} is the total processing time of the subjobs of job j assigned to machine i. Stop.

(ii) If $\Delta \ge p_{0,[v]}$, then assign job $[v]$ to subcontractor $s_{[v]}$. Let $y_{[v]} = 1$, and add job $[v]$ to N_1. Update $\Delta \leftarrow \Delta - p_{0,[v]}$ and $U \leftarrow U \setminus \{[v]\}$. If $v < w$, then set $v \leftarrow v + 1$ and repeat Steps 4(a) and 4(b). If $v = w$ and $\Delta = 0$, then repeat Step 4(a) and stop. If $v = w$ and $\Delta > 0$, then $LP(C)$ is infeasible and stop.

Lemma 2: If $LP(C)$ is feasible, then procedure $P2$ finds an optimal solution to $LP(C)$ with no more than m split jobs in $O(n \log n + m)$ time.

In the following, we describe our heuristic for solving problem $Min\{G \mid C_{\max} \le C\}$. The heuristic assigns each job either to an in-house machine or to a subcontractor for processing based on the solution of $LP(C)$ obtained by procedure $P2$ (which generates a set of unsplit jobs N_1 and a set of split jobs N_2). More specifically, the heuristic assigns each job $j \in N_1$ with $x_{ij} = 1$ for some $i \in M$ to machine i of the manufacturer, and each job $j \in N_1 \cap V$ with $y_j = 1$ to subcontractor s_j. This forms a partial schedule σ_1. Next, each job in set N_2 is either assigned to a subcontractor or an in-house machine based on its cost parameters and the workload of each in-house machine in schedule σ_1.

Heuristic $H1$ for Problem $Min\{G \mid C_{\max} \le C\}$:

Step 1: Solve $LP(C)$ by procedure $P2$. If $LP(C)$ is infeasible, then problem $Min\{G \mid C_{\max} \le C\}$ is infeasible and stop. Otherwise, based on the solution of $LP(C)$, define $J_i = \{ j \in N_1 \mid x_{ij} = 1\}$ for $i \in M$.

Step 2: For each $i \in M$, assign each job $j \in J_i$ to machine i at the manufacturer. Schedule the jobs on each machine in an arbitrary sequence. For each job $j \in N_1 \cap V$ with $y_j = 1$, assign job j to subcontractor s_j. Denote the resulting partial schedule (containing the jobs from N_1 only) by σ_1. Define the workload of each in-house machine under solution σ_1 as the total processing time of the jobs assigned to this machine by σ_1.

Step 3: For each $j \in N_2$: If $S_j \ne \emptyset$ and $q_{s_j j} \le q_{0j}$, then assign job j to subcontractor s_j. If $S_j \ne \emptyset$ and $q_{s_j j} > q_{0j}$, or if $S_j = \emptyset$, then assign job j to an in-house machine with the minimum workload under solution σ_1 and update the workload of this machine accordingly. Denote the final schedule as σ.

Determining $S_1, S_2, \ldots, S_n$ and $s_1, s_2, \ldots, s_n$ takes $O(nk)$ time. It is easy to see that once these values are known, Steps 2 and 3 of the heuristic take $O(n)$ time. Since the LP-relaxation problem in Step 1 of heuristic $H1$ is solved by procedure $P2$ in $O(n \log n + m)$ time, the overall time complexity of heuristic $H1$ is $O(nk + n \log n)$.

If problem $LP(C)$ is infeasible (which can be detected by Step 1 of heuristic $H1$), then problem $Min\{G \mid C_{\max} \le C\}$ is infeasible as well. If problem $Min\{G \mid C_{\max} \le C\}$ is feasible, then problem $LP(C)$ is feasible and heuristic $H1$ generates schedule σ. However, schedule σ generated by $H1$ may not always be feasible for problem $Min\{G \mid C_{\max} \le C\}$ even if this problem is feasible. In fact, it is highly unlikely that there exists a polynomial-time heuristic which can always generate a feasible schedule for every feasible instance of $Min\{G \mid C_{\max} \le C\}$. The reason is as follows: If there is such a heuristic H, then this heuristic can always find a feasible schedule for the special case in which all in-house production costs are zero (i.e., $q_{ij} = 0$ for all $i \in M$ and $j \in N$), all subcontracting costs are strictly positive (i.e., $q_{0j} > 0$ for all $j \in N$), and the threshold $C = C^*$, where C^* is the minimum makespan of the jobs if all the jobs are processed in-house. It can be seen that finding a feasible schedule for this problem is equivalent to finding an optimal schedule for the classical *m*-parallel-machine makespan minimization problem, which is known to be NP-hard (Garey and Johnson 1978). Thus, heuristic H can be used to solve this NP-hard problem optimally, which would mean that P must be equal to NP.

We show below that when the given threshold of the makespan C satisfies a certain condition, then schedule σ generated by $H1$ is guaranteed to be feasible for problem $Min\{G \mid C_{\max} \le C\}$.

Theorem 2: If $C \ge \frac{1}{m}\sum_{j \in N} p_{0j} + p_{0,\max}$, where $p_{0,\max} = \max\{p_{0j} \mid j \in N\}$, then the solution generated by $H1$ is guaranteed to be a feasible solution to problem $Min\{G \mid C_{\max} \le C\}$.

We realize that it is not meaningful to discuss worst-case or asymptotic performance of a heuristic if the solution generated by the heuristic may not even be feasible. However, by Theorem 2, under the condition "$C \ge \frac{1}{m}\sum_{j \in N} p_{0j} + p_{0,\max}$," our heuristic is guaranteed to generate feasible solutions. Hence, it is interesting to study worst-case or asymptotic performance of our heuristic under this condition.

Let $Z(\sigma)$ denote the total cost of schedule σ obtained by $H1$, and Z^* denote the optimal total cost of problem $Min\{G \mid C_{\max} \le C\}$. We have the following results.

Theorem 3: $Z(\sigma) \le 2Z^*$ for problem $Min\{G \mid C_{\max} \le C\}$ with $C \ge \frac{1}{m}\sum_{j \in N} p_{0j} + p_{0,\max}$.

Theorem 4: If the processing times and production costs of the jobs are distributed over the intervals $[X_1, X_2]$ and $[Y_1, Y_2]$, respectively, where $0 < X_1 < X_2 < \infty$ and $0 < Y_1 < Y_2 < \infty$, and if m and k are fixed, then, as n approaches infinity, the solution σ generated by heuristic $H1$ is asymptotically optimal to problem $Min\{G \mid C_{\max} \le C\}$ with $C \ge \frac{1}{m}\sum_{j \in N} p_{0j} + p_{0,\max}$.

4. COMPUTATIONAL RESULTS AND RELATED INSIGHTS

In this section, we conduct computational experiments to test the performance of heuristic $H1$. We also evaluate the value of subcontracting based on the model $Min\{G \mid C_{\max} \le C\}$. Heuristic $H1$ is coded in C with running time always less than 1 CPU second on a 2-GHz processor for every problem tested. Therefore, we do not report CPU times of these heuristics.

4.1 Performance of Heuristic $H1$

To test the performance of heuristic $H1$, we generate test problems randomly as follows:

a. Number of jobs $n \in \{50, 100, 200, 400\}$, number of parallel machines at the manufacturer $m \in \{2, 4, 8\}$, and number of subcontractors available $k \in \{1, 3, 9\}$. We note that a small (large) m relative to k means the relative small (large) in-house capacity compared to available subcontracting options. The nine possible combinations of m and k here represent a wide range of this relative relationship.

b. Job processing times at the manufacturer p_{0j} are drawn from the discrete uniform distribution over the interval $[1,100]$, and job processing times at the subcontractors $p_{1j}, p_{2j}, \ldots, p_{kj}$ are drawn from the discrete uniform distribution over the interval $[1,100\mu]$, where $\mu \in \{1, 2, 4\}$. These values of μ represent a situation in which the delivery lead time of a job, when subcontracted, is about the same as, two times, or four times the processing time of the job if it is processed in-house.

c. Production costs of jobs at the manufacturer q_{0j} are drawn from the discrete uniform distribution over the interval $[1,100]$, and production costs charged by the subcontractors $q_{1j}, q_{2j}, \ldots, q_{kj}$ follow the discrete uniform distribution over the interval $[1,100\nu]$, where $\nu \in \{0.5, 1, 4\}$. These values of ν represent a situation in which the production cost of a job charged by a subcontractor is about half, the same as, or four times the production cost of the job if it is processed in-house.

d. The makespan threshold value $C = \left\lceil \alpha (\sum_{j \in N} p_{0j}) / m \right\rceil + p_{0,\max}$, where $\alpha \in \{0.3, 0.6, 1.0\}$. It can be estimated that with C generated this way, roughly speaking, at least $(1-\alpha) \times 100\%$ of the jobs have to be subcontracted in order to meet the given constraint on the makespan.

For each of the 972 ($= 4 \times 3 \times 3 \times 3 \times 3 \times 3$) possible combinations of the values of parameters n, m, k, μ, ν, and α involved in a problem, we generate 10 test problems randomly. The LP-relaxation problem $LP(C)$ is infeasible for some of the test problems. We note that even if $LP(C)$ is feasible for a test problem, the test problem itself may not be feasible; and even if the test problem is feasible, the solution generated by the heuristic may not be feasible. We report in Table 1 the percentage of test instances for which $H1$ generates a feasible solution among all the problems where $LP(C)$ is feasible. Since a problem may still be infeasible even if the corresponding $LP(C)$ is feasible, the percentage reported in Table 1 is a lower bound of the percentage of feasible problems for which the heuristic generates a feasible solution. Each entry in the table corresponds to a given combination of n, m, k, and α based on 90 test problems (9 combinations of μ and ν, and 10 test problems for each combination). By Theorem 2, when α = 1.0, the solution generated by the heuristic is always feasible. Hence, only the problems with α = 0.3 or 0.6 are reported in Table 1. From this table, we can conclude that the heuristic often (with an overall chance of more than 90%) generates a feasible solution when the given problem instance is feasible. We also observe that heuristic $H1$ tends to be more successful in generating feasible solutions when $\alpha = 0.6$ than when $\alpha = 0.3$. This is because when α is larger, the test instances generated are more likely to be feasible.

Among those test instances where heuristic $H1$ generated feasible solutions, the heuristic solution values always equal the optimal objective values of the corresponding $LP(C)$ solutions. In other words, for all of our test problems, if the solution generated by the heuristic is feasible, then it is optimal. This indicates that heuristic $H1$ is highly effective.

4.2 Value of Subcontracting

The value of subcontracting is well understood in a qualitative sense. In contrast, very few works have quantitatively evaluated the value of subcontracting. Alp and Tan (2008) have evaluated the value of contingent capacity (which can be viewed as subcontracting) in the context of an integrated capacity and inventory planning model. However, no quantitative study in subcontracting has appeared in the context of detailed operations scheduling. Here, we build on the results developed in Section 3 to evaluate the possible benefits that can be obtained by using the subcontracting option. The results we obtain in this section provide a tool for deriving various managerial insights and performing sensitivity analysis in practice (e.g., how the solution with or without the subcontracting option will change with various parameters of the problem).

Let ω and σ denote the optimal solution to a problem without the subcontracting option and the optimal solution to the same problem with the subcontracting option, respectively. We investigate the relative reduction in the objective value that can be achieved when subcontracting is an available option. The relative reduction in the objective value is given as

$$\frac{Z(\omega)-Z(\sigma)}{Z(\omega)}=1-\frac{G(\sigma)}{G(\omega)}, \tag{8}$$

where $Z(\rho)$ denotes the objective value of a solution ρ and $G(\rho)$ denotes the total production and subcontracting cost of schedule ρ. Without the subcontracting option, all of the jobs are processed on the m machines at the manufacturer. Hence, the total production cost is fixed and independent of how the jobs are scheduled. Therefore, $G(\omega)=\sum_{j\in N} q_{0j}$. With the subcontracting option, we can apply heuristic $H1$ to obtain a solution σ^H whose objective value $G(\sigma^H)$ is an upper bound on the optimal objective value $G(\sigma)$. Then (8) implies that

$$\frac{Z(\omega)-Z(\sigma)}{Z(\omega)}\geq 1-\frac{G(\sigma^H)}{G(\omega)}. \tag{9}$$

Hence, given a problem instance, the right-hand side of (9) provides a lower bound for the relative cost reduction that can be computed efficiently.

We use the same set of test instances with $\alpha=1.0$ generated in the experiments reported in Section 4.1. The instances with $\alpha=0.3$ or 0.6 are not used here because the problem without the subcontracting option is likely to be infeasible with those values of α. With $\alpha=1.0$, the problem without subcontracting option is always feasible, and heuristic $H1$ is also guaranteed to generate a feasible solution. We report in Tables 2 and 3 both average relative cost reduction (shown in columns with "Avg Red") and minimum relative cost reduction (shown in columns with "Min Red") due to the subcontracting option for problem $Min\{G\,|\,C_{\max}\leq C\}$. We found that the relative cost reduction due to subcontracting remains almost the same as the number of machines m or the average subcontractors' processing time (represented by μ) changes (for any fixed combination of the other parameters). Thus, we do not report the results in terms of m or μ. The results in Table 2 are based on 90 test problems with $m=4$ (10 for each of the 9 combinations of the values of μ and ν). The results in Table 3 are based on 360 test problems with $\mu=2$ (10 for each of the 36 combinations of the values of n, m, and k).

The results in these tables demonstrate that there is a significant cost reduction by using the subcontracting option for almost every problem tested. The relative cost reduction increases quickly with the number of subcontractors k and decreases quickly with the average subcontracting costs (represented by ν). However, the average relative cost reduction does not change much with the number of jobs n. This can be explained as follows. Since the threshold value is set as $C = \left\lceil \sum_{j \in N} p_{0j} / m \right\rceil + p_{0,\max}$, the in-house capacity of the manufacturer relative to the set of jobs to be processed remains the same as the number of jobs increases. Hence, on average the value of subcontracting does not change much with n.

5. OTHER VARIANTS OF THE MODEL

Note that in our model there is a pre-specified threshold value C on the makespan of the schedule. As mentioned earlier, this model works for applications where there is an explicit service constraint and for applications where one needs to know the Pareto-optimal solutions with respect to G and $C_{\max}$. However, a slightly different model may be more appropriate for some other applications. In this section we discuss two such problem variations: (i) We consider the problem of minimizing a weighted sum of makespan and the total cost, i.e., $\lambda C_{\max} + (1-\lambda)G$, where $0 \le \lambda \le 1$, and denote this problem as $Min\{\lambda C_{\max} + (1-\lambda)G\}$. This problem represents the situation where the manufacturer has a known relative preference over the lead time performance $C_{\max}$ and the total cost G. He can choose a weighting parameter $\lambda \in [0,1]$, assign λ as the preference weight to $C_{\max}$, assign $(1-\lambda)$ as the preference weight to G, and consider the weighted sum of these two measures. (ii) We consider the problem of minimizing the total cost G, subject to the constraint that each job j has to be completed no later than a given deadline d_j. We denote this problem as $Min\{G \mid C_j \le d_j\}$. Note that problem $Min\{G \mid C_{\max} \le C\}$ is a special case of this problem, where every job has a common deadline C.

5.1 Problem $Min\{\lambda C_{\max} + (1-\lambda)G\}$

Using reduction from the Subset Sum problem with a similar construction as in the proof to Theorem 1, it is not difficult to show that when $0 < \lambda < 1$, problem $Min\{\lambda C_{\max} + (1-\lambda)G\}$ is NP-hard even if there is only a single machine at the manufacturer, there is a single subcontractor, and the in-house production costs are proportional to the production times. To solve this problem with a general number of parallel machines at the manufacturer and multiple subcontractors, we propose the following heuristic method. The main idea of the heuristic is to divide the problem into a number of subproblems and then use a similar idea as the heuristic developed in Section 3 to solve each subproblem.

Suppose that there are β ($\beta \le nk$) distinct values among all delivery lead times $\{p_{hj} \mid j \in N, h \in K\}$ specified by the subcontractors. Let $\pi_1, \pi_2, \ldots, \pi_\beta$ denote these β values, where $\pi_1 > \pi_2 > \cdots > \pi_\beta$. Define $\pi_{\beta+1} \equiv 0$. We consider $\beta+1$ subproblems. For $\ell = 1,2,\ldots,\beta+1$, subproblem ℓ is defined to be the original problem with two additional requirements:

Requirement (I): The maximum delivery time among all of the subcontracted jobs is no more than π_ℓ.

Requirement (II): The makespan $C_{\max}$ is calculated as the maximum of π_ℓ and the completion time of the last job completed at the manufacturer's own plant.

For each job $j \in N$ and each $\ell = 1,2,\ldots,\beta+1$, define a subset of subcontractors $S_{\ell j} = \{h \in K \mid p_{hj} \le \pi_\ell\}$, which are the subcontractors that can process job j with a delivery time no later than π_ℓ. Define $V_\ell = \{j \in N \mid S_{\ell j} \ne \varnothing\}$, which is the set of jobs that can be subcontracted in subproblem ℓ. For each $j \in V_\ell$, define a subcontractor $s_{\ell j} = \arg\min_{v \in S_{\ell j}} \{q_{vj}\}$, which is the subcontractor that can process job j at a minimum cost with a delivery time no later than π_ℓ. Requirements (I) and (II) of subproblem ℓ imply that if a job $j \in V_\ell$ is to be subcontracted, it is optimal to assign it to subcontractor $s_{\ell j}$. In subproblem $\beta+1$, since $\pi_{\beta+1} = 0$, no job can be subcontracted and, hence, $V_{\beta+1} = \varnothing$.

It is easy to see that any feasible solution to the original problem is also a feasible solution to at least one of the subproblems with the same objective value. Thus, the minimum of the optimal objective values of the subproblems is the optimal objective value of the original problem. This means that we can solve the original problem by solving these $\beta+1$ subproblems.

For $\ell = 1,\ldots,\beta+1$, subproblem ℓ can be formulated as the following mixed integer program which is similar to the formulation $IP(C)$ given in Section 3:

$$
\begin{aligned}
IP_\ell: \quad & \text{Minimize} \quad && \lambda C_{\max} + (1-\lambda)\sum_{i=1}^{m}\sum_{j=1}^{n} q_{0j} x_{ij} + (1-\lambda)\sum_{j \in V_\ell} q_{s_{\ell j} j} y_j \\
& \text{subject to} && \sum_{i=1}^{m} x_{ij} + y_j = 1, \text{ for } j \in V_\ell \\
& && \sum_{i=1}^{m} x_{ij} = 1, \text{ for } j \in N \setminus V_\ell \\
& && C_{\max} \ge \sum_{j=1}^{n} p_{0j} x_{ij}, \text{ for } i \in M \\
& && C_{\max} \ge \pi_\ell \\
& && x_{ij} \in \{0,1\}, \text{ for } i \in M \text{ and } j \in N \\
& && y_j \in \{0,1\}, \text{ for } j \in V_\ell.
\end{aligned}
$$

We focus on the LP-relaxation problem of IP_ℓ, denoted as LP_ℓ. By a similar proof to Lemma 1 for the LP-relaxation of $IP(C)$, we can show that in an optimal basic solution of LP_ℓ, if $n \geq m$ then among all x_{ij} and y_j variables, at least $n-m$ of them take the value of 1. Therefore, in an optimal basic solution of LP_ℓ, there are at most m jobs with some fractional values assigned to the corresponding x_{ij} or y_j variables. It is not difficult design an efficient procedure similar to $P2$ to determine an optimal solution to problem LP_ℓ with no more than m split jobs. Based on this optimal solution, we can design a heuristic similar to $H1$ to solve subproblem ℓ. The heuristic for the overall problem $Min\{\lambda C_{\max} + (1-\lambda)G\}$ can then be summarized as follows.

Heuristic $H2$ for Problem $Min\{\lambda C_{\max} + (1-\lambda)G\}$:

Step 1: For $\ell = 1,2,\ldots,\beta+1$, solve subproblem ℓ by a heuristic similar to $H1$ and get the solution σ_ℓ.

Step 2: Pick the solution with the minimum objective value among the $\beta+1$ solutions $\sigma_1, \sigma_2, \ldots, \sigma_{\beta+1}$ generated in Step 1.

Let σ denote the solution generated by heuristic $H2$. Let $Z(\sigma)$ denote the objective value of solution σ. Let Z^* denote the optimal objective value of this problem. We have the following worst-case results.

Theorem 5: $Z(\sigma)/Z^* \leq 2$, and this bound is tight.

Theorem 6: If the processing times and production costs of the jobs are distributed over the intervals $[X_1, X_2]$ and $[Y_1, Y_2]$, respectively, with $0 < X_1 < X_2 < \infty$ and $0 < Y_1 < Y_2 < \infty$, and if m and k are fixed, then the solution σ generated by heuristic $H2$ is asymptotically optimal as n approaches infinity.

The proofs of Theorems 5 and 6 follow the same arguments as in the proofs of Theorems 3 and 4, respectively, and are therefore omitted. Note that heuristic $H2$ always generates a feasible solution to problem $Min\{\lambda C_{\max} + (1-\lambda)G\}$. Thus, although both $Min\{\lambda C_{\max} + (1-\lambda)G\}$ and $Min\{G \mid C_{\max} \leq C\}$ are NP-hard, it is easier to obtain feasible solutions with worst-case performance guarantee for problem $Min\{\lambda C_{\max} + (1-\lambda)G\}$ than obtaining feasible solutions for problem $Min\{G \mid C_{\max} \leq C\}$.

We tested the performance of heuristic $H2$ using various test problems similar to the ones used in Section 4.1. The average relative error among all 9720 test instances equals 0.62%, indicating that heuristic $H2$ is overall highly effective. The computational results also show that the relative error of the heuristic decreases as n increases, and it approaches 0 as n tends to infinity. This is consistent with the fact that the heuristic is asymptotically optimal.

5.2 Problem $Min\{G \mid C_j \le d_j\}$

As mentioned before, problem $Min\{G \mid C_j \le d_j\}$ is a generalization of problem $Min\{G \mid C_{\max} \le C\}$. Thus, it is an NP-hard problem. In fact, solving this generalized problem is substantially harder. The following theorem states a property of the optimal solution to this problem, which is essentially the "earliest due-date first" (EDD) rule commonly used in the scheduling literature (Pinedo 2002).

Theorem 7: There exists an optimal solution to $Min\{G \mid C_j \le d_j\}$ in which the jobs on each in-house machine are arranged in nondecreasing order of d_j.

The proof of Theorem 7 follows a straightforward pairwise job interchange argument and is thus omitted. We re-index the jobs such that $d_1 \le d_2 \le \cdots \le d_n$. Thus, by Theorem 7, there exists an optimal schedule where lower-index jobs always precede higher-index jobs on every machine.

We define $S_j = \{h \in K \mid p_{hj} \le d_j\}$ as the subset of subcontractors that can process job j with a delivery time no later than its deadline. Define $V = \{j \in N \mid S_j \ne \emptyset\}$, and let $s_j = \arg\min_{v \in S_j}\{q_{vj}\}$ for $j \in V$. Similar to the IP formulation given in Section 3 for problem $Min\{G \mid C_{\max} \le C\}$, the following mixed integer program formulates problem $Min\{G \mid C_j \le d_j\}$, where x_{ij} and y_j are defined the same way as in Section 3:

$$
\begin{aligned}
IP: \quad & \text{Minimize} && \textstyle\sum_{i=1}^{m}\sum_{j=1}^{n} q_{0j}x_{ij} + \sum_{i \in V} q_{s_j j}\, y_j && (10)\\
& \text{subject to} && \textstyle\sum_{i=1}^{m} x_{ij} + y_j = 1, \text{ for } j \in V && (11)\\
& && \textstyle\sum_{i=1}^{m} x_{ij} = 1, \text{ for } j \in N \setminus V && (12)\\
& && \textstyle\sum_{k=1}^{j} p_{0k}x_{ik} \le d_j, \text{ for } i \in M \text{ and } j \in N && (13)\\
& && x_{ij} \in \{0,1\}, \text{ for } i \in M \text{ and } j \in N && (14)\\
& && y_j \in \{0,1\}, \text{ for } j \in V. && (15)
\end{aligned}
$$

Objective function (10) and constraints (11)-(12) are identical to (2)-(3) in formulation $IP(C)$. Constraint set (13) ensures that all jobs complete no later than their deadlines.

Note that formulation IP has a different structure from formulation $IP(C)$, because constraint set (13) contains a total of nm constraints. Due to this fact, the LP-relaxation of IP does not have a similar property to the one stated in Lemma 1 for the LP-relaxation of $IP(C)$. In fact, the LP-relaxation solution of IP is much more fractional than that of $IP(C)$. One can follow the idea of heuristic $H1$ to design a similar heuristic for problem $Min\{G \mid C_j \le d_j\}$ by first solving the LP-relaxation of IP to obtain a partial schedule of unsplit jobs and then heuristically assigning the split jobs to either a subcontractor or an in-house machine. However, there are many more split jobs to consider than in the case of heuristic $H1$. As a result, to guarantee a constant worst-case performance bound of the resulting heuristic for problem $Min\{G \mid C_j \le d_j\}$, one may have to consider an exponential number of possibilities for assigning the split jobs to proper machines. It remains a challenging future research topic for developing a polynomial-time heuristic with some performance guarantee.

Note also that an alternative way of modeling delivery performance is to consider a due date d_j for each job j and impose a threshold constraint on the job tardiness. Such a model has an objective of minimizing the total cost G and a constraint of $C_j - d_j \le F$ for every job j. Letting $d'_j = d_j + F$, the threshold constraint becomes $C_j \le d'_j$. Thus, this model is mathematically equivalent to problem $Min\{G \mid C_j \le d'_j\}$.

6. CONCLUSIONS

In this paper we have proposed an analytical model that integrates job scheduling and subcontracting decisions where the objective is to minimize the total production and subcontracting cost with a constraint on the delivery performance. The problem is NP-hard even when there is a single machine at the manufacturer and there is a single subcontractor available. A heuristic solution approach has been developed for the problem. Both worst-case and asymptotic performances of the heuristic have been analyzed. Our computational experiments have demonstrated that the heuristic can often generate a feasible solution, and when it generates a feasible solution, the solution is always optimal for all the problem instances tested. We have also studied the value of the subcontracting option in the context of this model. Our computational results have shown that for most test instances, a significant reduction in the objective value occurs when the subcontracting option is used.

We have also discussed two variants of our model. The first variant, where the objective is to minimize a weighted sum of the total cost and the makespan, can be solved by a similar heuristic. However, it is much more difficult to solve the second variant where the objective is to minimize the total cost subject to the constraint that each job is completed no later than a job specific deadline.

We note that in practice, it is likely that customer orders may come in dynamically over time, and hence not all the orders are known at the beginning of the planning horizon. Thus, our model may need to be implemented in a rolling-horizon basis, such that an updated schedule is determined periodically with all known customer orders.

Several related topics deserve future research. Problems under the same model but with job delivery performance measured by other criteria such as maximum or total tardiness of jobs can be considered. Those problems are generally NP-hard, and hence heuristics will need to be developed. We have assumed in this paper that the subcontractors have unlimited capacity. A different model with a limited capacity at each subcontractor can also be developed. In such a case, the delivery lead time of a job by a subcontractor will depend on how many jobs it receives from the manufacturer. The solution generated by such a model will tend to have more jobs processed by the in-house machines, and the subcontracted jobs are likely to be distributed more evenly among different subcontractors. A detailed analysis of such issues is an interesting future research topic.

ACKNOWLEDGEMENT

The authors would like to thank the associate editor and two referees for their helpful comments on the earlier versions of the paper.

REFERENCES

Alp, O. and T. Tan (2008). Tactical capacity management under capacity flexibility in make-to-stock systems. *IIE Transactions* **40**, 221-237.

Atamturk, A. and D.S. Hochbaum (2001). Capacity acquisition, subcontracting, and lot sizing. *Management Science* **47,** 1081-1100.

Bazinet, C.G., S.A. Kahn and S.J. Smith (1998). Measuring the value of outsourcing. *Best's Review* **99,** 85-88.

Bertrand, J.W.M. and V. Sridharan (2001). A study of simple rules for subcontracting in make-to-order manufacturing. *European Journal of Operational Research* **128,** 509-531.

Bradley, J.R. (2004). A Brownian approximation of a production-inventory system with a manufacturer that subcontracts. *Operations Research* **52,** 765-784.

Bukchin, Y. and E. Hanany (2007). Decentralization cost in scheduling: A game-theoretic approach. *Manufacturing and Service Operations Management* **9**, 263-275.

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* **98,** 238-250.

Craumer, M. (2002). How to think strategically about outsourcing. *Harvard Management Update*, May 2002.

Day, J.S. (1956). *Subcontracting Policy in the Airframe Industry*. Graduate School of Business Administration, Harvard University, Boston, MA.

Garey, M.R. and D.S. Johnson (1978). "Strong" NP-completeness results: Motivation, examples, and implications. *Journal of the Association for Computing Machinery* **25,** 499-508.

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

Ioannou, L. (1995), Subcontracting's subtle subversions. *International Business*, June 1995, pages 20-22.

Kamien, M.I. and L. Li (1990). Subcontracting, coordination, flexibility, and production smoothing in aggregate planning. *Management Science* **36,** 1352-1363.

Kogan, K. (2000). Continuous-time models for production scheduling in constrained subcontracting conditions. *INFOR* **38,** 113-125.

Kolawa, A. (2004). Outsourcing is not the enemy. *Wall Street Journal*, February 24, 2004, page B2.

Lee, D.-H., S.-K. Lim, G.-C. Lee, H.-B. Jun and Y.-D. Kim (1997). Multi-period part selection and loading problems in flexible manufacturing systems. *Computers and Industrial Engineering* **33,** 541-544.

Lee, Y.H., C.S. Jeong and C. Moon (2002). Advanced planning and scheduling with outsourcing in manufacturing supply chain. *Computers and Industrial Engineering* **43,** 351-374.

Lenstra, J.K., D.B. Shmoys and E. Tardos (1990). Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming* **46,** 259-271.

Logendran, R. and V. Puvanunt (1997). Duplication of machines and subcontracting of parts in the presence of alternative cell locations. *Computers and Industrial Engineering* **33,** 235-238.

Logendran, R. and P. Ramakrishna (1997). A methodology for simultaneously dealing with machine duplication and part subcontracting in cellular manufacturing systems. *Computers and Operations Research* **24,** 97-116.

Pinedo, M. (2002). *Scheduling: Theory, Algorithms, and Systems*, 2nd Edition, Prentice-Hall, Upper Saddle River, NJ.

Potts, C.N. (1985). Analysis of a linear programming heuristic for scheduling unrelated parallel machines. *Discrete Applied Mathematics* **10,** 155-164.

Qi, X. (2007a). Coordinated logistics scheduling for in-house production and outsourcing. *IEEE Transactions on Automation Science and Engineering*, forthcoming.

Qi, X. (2007b). Outsourcing and production scheduling for a two-stage flow shop. Working paper, Hong Kong University of Science and Technology.

Shmoys, D.B. and E. Tardos (1993). An approximation algorithm for the generalized assignment problem. *Mathematical Programming* **62,** 461-474.

Tan, B. and S.B. Gershwin (2004). Production and subcontracting strategies for manufacturers with limited capacity and volatile demand. *Annals of Operations Research* **125,** 205-232.

Van Mieghem, J.A. (1999). Coordinating investment, production, and subcontracting. *Management Science* **45,** 954-971.

Webster, M., C. Alder and A.P. Muhlemann (1997). Subcontracting within the supply chain for electronics assembly manufacture. *International Journal of Operations and Production Management* **17,** 827-841.

Yang, J., X. Qi and Y. Xia (2005). A production-inventory system with Markovian capacity and outsourcing option. *Operations Research* **53,** 328-349.

APPENDIX

Proof of Theorem 1. We show this by a reduction from the Subset Sum problem defined as follows: Given a set of items $R = \{1,2,\ldots,r\}$, a size $a_j \in \mathbf{Z}^+$ for each item $j \in R$, and a positive integer A, the problem asks if there is a subset $Q \subseteq R$ such that $\sum_{j\in Q} a_j = A$. Subset Sum is known to be NP-complete (Garey and Johnson 1979). We define $B = \sum_{j\in R} a_j$ and assume that $B \neq A$ (otherwise, the problem is trivial).

Given an instance of Subset Sum, we construct the following instance for our problem:

number of machines at the manufacturer, $m = 1$;

number of subcontractors, $k = 1$;

job set: $N = R \cup \{r+1\}$ (number of jobs, $n = r+1$);

processing time at the manufacturer: $p_{0j} = La_j$ for $j \in N$, and $p_{0,r+1} = LB$;

delivery lead time if subcontracted: $p_{1j} = La_j$ for $j \in N$, and $p_{1,r+1} = LA$;

production cost if processed in-house: $q_{0j} = a_j$ for $j \in N$, and $q_{0,r+1} = B$;

production cost if subcontracted: $q_{1j} = 2a_j$ for $j \in N$, and $q_{1,r+1} = 2A$;

threshold for the makespan: $C = LA$;

threshold for the objective value: $Z_0 = A + 2B$;

where L is any positive integer. Since in the above instance $q_{0j} = p_{0j}/L$ for all $j \in N$, in-house production cost of a job is proportional to its production time. We prove that there is a solution to the constructed instance of our problem with the objective value being no more than Z_0 if and only if there is a solution to Subset Sum.

("If" part) Given a solution to Subset Sum, we construct a solution to our problem as follows: Subcontract all the jobs in $N \setminus Q$ to the only subcontractor available, and process the jobs in Q in-house in an arbitrary sequence. Evidently, in this solution $C_{\max} = LA$ and $G = A + 2B = Z_0$.

("Only if" part) Given a solution to our problem with an objective value no more than Z_0, let the subset of the jobs processed in-house be H. Clearly, $r+1 \notin H$, since otherwise $C_{\max} \geq LB \geq L(A+1)$, implying that the objective value exceeds Z_0. Thus, $C_{\max} = p_{1,r+1} = LA = C$. Also, $\sum_{j\in H} a_j \leq A$, since otherwise $C_{\max} \geq L(A+1) > C$. Suppose that $\sum_{j\in H} a_j < A$. Then, the total cost is

$$G=\sum_{j\in H} q_{0j}+\sum_{j\in N\setminus H} q_{1j}=\sum_{j\in H} a_j+2A+\sum_{j\in R\setminus H} 2a_j=2A+2B-\sum_{j\in H} a_j>A+2B=Z_0,$$

which is a contradiction. This concludes that $\sum_{j\in H} a_j=A$. ■

Proof of Lemma 1: Note that besides the nonnegativity constraints, there are $n+m+1$ constraints in $LP(C)$. Hence, in an optimal basic solution (which can be obtained by the simplex method), no more than $n+m+1$ variables may take positive values. Since variable $C_{\max}$ takes a positive value, among all x_{ij} and y_j variables, at most $n+m$ of them have a positive value in the optimal basic solution. Given an optimal basic solution of $LP(C)$, let Γ_j denote the subset of variables among $x_{1j}, x_{2j}, ..., x_{mj}, y_j$ that take a nonzero value when $j\in V$, and let Γ_j denote the subset of variables among $x_{1j}, x_{2j}, ..., x_{mj}$ that take a nonzero value when $j\in N\setminus V$. Define $N_1=\{j\in N \mid |\Gamma_j|=1\}$ and $N_2=\{j\in N \mid |\Gamma_j|\geq 2\}$. If $j\in N_1$, then there is only one nonzero variable among the variables $x_{1j}, x_{2j}, ..., x_{mj}, y_j$. Constraints (2) and (3) imply that this nonzero variable must have a value of 1. Clearly, each Γ_j contains a distinct set of variables (i.e., there is no overlap between Γ_j and $\Gamma_{j'}$ if $j\neq j'$). Thus,

$$n+m\geq |N_1|+2|N_2|=|N_1|+2(n-|N_1|)=2n-|N_1|,$$

which implies that $|N_1|\geq n-m$. This completes the proof of the lemma. ■

Proof of Lemma 2: Clearly, in an optimal solution of $LP(C)$, for each job $j\in V$, if $q_{s_j j}\leq q_{0j}$ then job j should be assigned to subcontractor s_j. This shows the validity of Step 2 of the algorithm. After Step 2, if $\frac{1}{m}\sum_{j\in U} p_{0j}>C$ and W is empty, then the jobs in U must be processed in-house, but the makespan of the in-house schedule for these jobs would be larger than C, and hence the problem is infeasible. If $\frac{1}{m}\sum_{j\in U} p_{0j}\leq C$, then all of the jobs in U should be scheduled at the manufacturer's own plant. This is because the objective would increase if a job in W is subcontracted. This shows the correctness of Step 3. In Step 4, if $\frac{1}{m}\sum_{j\in U} p_{0j}>C$ and W is not empty, then to be optimal we need to subcontract exactly Δ units of the total processing time of the jobs in W. The jobs in W are sorted so that job $[1]$ can bring the largest reduction in cost if one unit of it is subcontracted. Hence, the jobs in W are considered in the order $[1],[2],\ldots,[w]$ in Step 4. This minimizes the total cost of the subcontracted jobs. This shows the validity of Step 4.

In procedure $P2$, Steps 1 and 2 take no more than $O(n)$ time. If Step 3 is executed, it calls procedure $P1$, which takes $O(n+m)$ time. If Step 4 is executed, since there are at most n jobs in W, sorting these jobs takes $O(n\log n)$ time, while procedure $P1$ is called exactly once, which takes $O(n+m)$ time. Therefore, the overall time complexity of procedure $P2$ is bounded from above by $O(n\log n+m)$. ■

Proof of Theorem 2: We first show that if $C \geq \frac{1}{m}\sum_{j\in N} p_{0j} + p_{0,\max}$, then problem $Min\{G \mid C_{\max} \leq C\}$ is feasible. To this end, we construct a feasible solution to the problem as follows: We process all the jobs by the m in-house machines using the following scheduling rule: For $j=1,2,\ldots,n$, schedule job j to a machine with the minimum workload and update the workload of that machine accordingly. In the resulting schedule π, let $C_{\max}(\pi)$ denote the makespan and C^i denote the total processing time of the jobs assigned to machine i, for $i \in M$. According to this scheduling rule, it is evident that the start time of the last job on each machine must be no more than $\frac{1}{m}\sum_{j\in N} p_{0j}$. Thus, $C^i \leq \frac{1}{m}\sum_{j\in N} p_{0j} + p_{0,\max}$ for every $i \in M$, which implies that $C_{\max} = \max\{C^i \mid i\in M\} \leq \frac{1}{m}\sum_{j\in N} p_{0j} + p_{0,\max} \leq C$. Therefore, schedule π is feasible. This means that problem $Min\{G \mid C_{\max} \leq C\}$ is feasible and, hence, $LP(C)$ is feasible. It is evident that schedule σ_1 generated for the jobs set N_1 in Step 2 is feasible. Let $C^i(\sigma_1)$ denote the total processing time of the jobs assigned to machine i (i.e., the total workload of machine *i*) under schedule σ_1. Define $\Delta^i(\sigma_1) = C - C^i(\sigma_1)$ to be the slack of machine i, for $i \in M$. By definition of slack, we can see that a schedule is feasible if the slack of every machine is nonnegative. The fact that σ_1 is feasible means that $\Delta^i(\sigma_1) \geq 0$, for all $i \in M$. Since some jobs of N_1 may be subcontracted under schedule σ_1, we have

$$\sum_{i=1}^{m} C^i(\sigma_1) \leq \sum_{j\in N_1} p_{0j} = \sum_{j\in N} p_{0j} - \sum_{j\in N_2} p_{0j}\,. \tag{16}$$

Since $C \geq \frac{1}{m}\sum_{j\in N} p_{0j} + p_{0,\max}$, by (16), we have

$$\sum_{i=1}^{m} \Delta^i(\sigma_1) = mC - \sum_{i=1}^{m} C^i(\sigma_1) \geq mp_{0,\max} + \sum_{j\in N_2} p_{0j}\,. \tag{17}$$

Let N_3 be the subset of the jobs in N_2 that are assigned to an in-house machine by Step 3 of the heuristic. Define $w = |N_3|$ as the number of jobs in N_3. Clearly, $w \leq m$. Let $[h]$ denote the index of the *h*-th job of N_3 added to an in-house machine in Step 3 of $H3$, for $h=1,2,\ldots,w$. By the scheduling rule used in Step 3, whenever a job in N_3 is added to an in-house machine, it is added to a machine with the minimum workload (i.e., a machine with the maximum slack). Let i_h denote the in-house machine where job $[h]$ of

N_3 is added ($h = 1, 2, \ldots, w$). Thus, immediately before job $[1]$ is added, machine i_1 has the minimum total workload under schedule σ_1. By (17), this means that $\Delta^{i_1}(\sigma_1) \geq p_{0,\max}$. This implies that after job $[1]$ is added, the updated total workload of machine i_1 is no more than C, and hence the resulting schedule remains feasible. By (17), it is easy to see that after job $[1]$ is added, the total updated slack of the m machines is

$$\sum_{i=1}^{m} \Delta^i(\sigma_1) \geq m p_{0,\max} + \sum_{j \in N_2 \setminus \{[1]\}} p_{0j} . \tag{18}$$

Next, we consider job $[2]$. By a similar argument and using (18), we can prove that immediately after job $[2]$ is added to an in-house machine, the resulting schedule remains feasible, and the total updated slack of the m machines is

$$\sum_{i=1}^{m} \Delta^i(\sigma_1) \geq m p_{0,\max} + \sum_{j \in N_2 \setminus \{[1],[2]\}} p_{0j} .$$

By repeatedly applying this argument, we conclude that the schedule remains feasible after all the jobs of N_3 are added to the schedule. ■

Proof of Theorem 3: Let $Z(\sigma_1)$ denote the total cost of schedule σ_1 generated by heuristic $H1$. Let $Z(N_2)$ denote the total cost of the jobs in N_2 under schedule σ. Let $Z_{LP(C)}$ denote optimal total cost of problem $LP(C)$. Clearly, $Z_{LP(C)}$ is a lower bound of Z^*. Consider the partial schedule σ_1 generated by Step 2 of heuristic $H1$. Since σ_1 only includes the jobs in N_1, we have $Z(\sigma_1) \leq Z_{LP(C)}$. In Step 3 of heuristic $H1$, each job in N_2 is assigned to a subcontractor or an in-house machine in a way that it incurs the minimum possible cost. Thus, $Z(N_2) \leq Z^*$. Therefore, $Z(\sigma) = Z(\sigma_1) + Z(N_2) \leq Z_{LP(C)} + Z^* \leq 2Z^*$. ■

Proof of Theorem 4: Since m is fixed, the contribution of the jobs in N_2 to the total cost of the problem under any schedule is always bounded from above. On the other hand, the optimal total cost of the problem under any schedule approaches infinity as n tends to infinity. Therefore,

$$\lim_{n \to \infty} \frac{Z(N_2)}{Z^*} = 0 . \tag{19}$$

As we have shown in the proof of Theorem 3, $Z(\sigma_1) \leq Z^*$. This implies that

$$\lim_{n \to \infty} \frac{Z(\sigma) - Z^*}{Z^*} \quad \leq \quad \lim_{n \to \infty} \frac{Z(\sigma_1) + Z(N_2) - Z^*}{Z^*}$$

$$= \lim_{n\to\infty} \frac{Z(\sigma_1) - Z^*}{Z^*} \qquad \text{(by (19))}$$

$$\leq 0.$$

This completes the proof. ■

Table 1. Percentage of test instances of $Min\{G \mid C_{\max} \le C\}$ for which heuristic $H1$ generates feasible/optimal solutions

		k = 1		k = 3		k = 9	
n	m	α = 0.3	α = 0.6	α = 0.3	α = 0.6	α = 0.3	α = 0.6
50	2	90.0%	100%	100%	96.0%	95.7%	100%
	4	88.9%	95.0%	96.7%	100%	96.1%	100%
	8	67.4%	73.8%	83.6%	92.8%	87.3%	100%
100	2	100%	100%	100%	91.7%	100%	100%
	4	90.0%	100%	90.0%	92.3%	91.7%	100%
	8	73.3%	100%	95.0%	93.1%	82.9%	100%
200	2	83.3%	95.2%	85.7%	100%	100%	100%
	4	100%	100%	100%	87.5%	95.2%	100%
	8	100%	90.9%	100%	91.7%	95.0%	100%
400	2	90.9%	100%	100%	100%	100%	100%
	4	90.9%	100%	100%	100%	100%	100%
	8	90.9%	100%	95.2%	90.9%	100%	100%

Table 2. Relative cost reduction due to the subcontracting option for $Min\{G \mid C_{\max} \le C\}$: Results in terms of n and k

n	k	Avg Red	Min Red
50	1	34.7%	4.6%
	3	53.8%	12.4%
	9	71.8%	37.8%
100	1	34.1%	4.4%
	3	53.0%	16.0%
	9	72.2%	43.2%
200	1	33.1%	7.1%
	3	53.0%	17.1%
	9	72.0%	43.6%
400	1	32.8%	7.3%
	3	52.6%	18.1%
	9	72.6%	44.9%

Table 3. Relative cost reduction due to the subcontracting option for $Min\{G \mid C_{\max} \le C\}$: Results in terms of ν

ν	Avg Red	Min Red
0.5	74.5%	52.1%
1	57.9%	28.0%
4	25.8%	4.4%