Technical Memorandum Number 820

Cooperative Strategies for Manufacturing Planning with Negotiable Third-Party Capacity

by

Xiaoqiang Cai
George L. Vairaktarakis

October 2007

Department of Operations
Weatherhead School of Management
Case Western Reserve University
330 Peter B Lewis Building
Cleveland, Ohio 44106

Cooperative Strategies for Manufacturing Planning with Negotiable Third-Party Capacity *

Xiaoqiang Cai † George L. Vairaktarakis ‡

Abstract

We consider a model where a group of manufacturers outsource finishing operations to a single third-party. Depending on the demand for her capacity, the third-party announces daily booking costs that may take on one of two prices - regular or peak price. Each manufacturer books chunks of production capacity (called manufacturing windows) so as to optimize his booking and tardiness costs subject to window availability at the third party. For days that capacity is booked the manufacturer has the option to purchase overtime at a higher hourly rate. Upon booking her capacity to the manufacturers, the third-party computes the monetary benefit that could be achieved if all manufacturers cooperated to obtain an overall optimal solution. Then, using cooperative game theory she devises a savings sharing scheme so that, in monetary terms, every manufacturer is better off with the coordinated schedule than what each could get for himself by cooperating with a subset of manufacturers. For her work, the third party withholds a portion ρ of the booking revenue paid by the manufacturers for days that were utilized in the original schedule but released after coordination. Subsequently, the third-party has the option to resell her capacity for those days. Through this model we come up with insights that indicate that cooperation amongst manufacturers reduces costs by about 20%.

*This research was initiated while the second author visited The Chinese University of Hong Kong, and was supported in part by the Research Grants Council of HK under Earmarked Grants No. CUHK 4166/01E and CUHK 4170/03E.

†Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, NT, Hong Kong, xqcai@se.cuhk.edu.hk

‡Weatherhead School of Management, Department of Operations, Case Western Reserve University , 10900 Euclid Avenue, Cleveland, OH 44106-7235, gxv5@case.edu

1 Introduction and Literature Review

Let M be a given set of manufacturers and P a single third-party. Each $m \in M$ out sources to P a set N_m of finishing operations according to the following protocol: At time 0 party P announces the available days of production as well as booking costs h_t for each day, $t = 1, 2, \ldots, T$. Let $W_t = [a_t, b_t]$ be the t-th available day of production; also referred to as the t-th *manufacturing window.* If window W_t is booked, overtime is available at the hourly rate α for up to O hours. Following the announced availability of P, each $m \in M$ books a set of windows $\mathcal{W}_m \subseteq \{1, 2, \ldots, T\}$ so as to minimize the total booking cost $\sum_{t \in \mathcal{W}_m} h_t$ paid to P to reserve some of P's available capacity, plus tardiness penalty β for each job j completed after its due-date d_j. Following the determination of $\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_m$, at time 0 party P reschedules jobs in $N = \cup_{m \in M} N_m$ so that the coordinated schedule minimizes the total booking plus tardiness penalty over all $m \in M$ and all $j \in N$. The cost savings produced by P are then shared amongst the manufacturers so that each manufacturer is given at least as much money as he could obtain by himself by forming a coalition with other manufacturers. Here we assume that the objective function value of every manufacturer is expressed in terms of dollars, and that every manufacturer is indifferent between a dollar paid (saved) and a unit gain (loss) in his objective.

The above model significantly differs from research in scheduling in several aspects. First, existing literature carries the common hypothesis that one possesses all capacity needed to process all jobs. While this assumption may be applicable to traditional manufacturing settings, it is not accurate in a variety of situations where manufacturers do not own some machines required to process jobs. Consequently, they need to book additional processing capacity from a third party. Problems of this nature are more common today as a result of factory focused manufacturing, globalization, and the emergence of internet technologies that allow negotiation and trading with third-parties. Second, the main questiona ddressed by our model is the coordination of activities between 2 members of the production chain. The second member (third-party) announces her capacity availability schedule and cost structure, and subsequently the first member (manufacturer) seeks to find the best balance between service (expressed in terms of tardiness penalties) and booking and overtime capacity costs. Finally, our model differs from the literature in manufacturing planning and control in that an incentive payment scheme is identified by P so that every manufacturer is better off to abide to the coordinated schedule than to form coalitions on his own. This payment scheme is said to be an element of the *core* of the sequencing game among manufacturers. Unlike in our model, a typical single machine sequencing game found in the literature assumes that there are $|M|$ players each with a single job, each player occupying a specific position in the initial job schedule, and it is assumed that every player has his own cost criterion.

Curiel *et al.* (2002) present a survey of sequencing games and consider core issues. Sequencing games were introduced by Curiel *et al.* (1989) who considered a single machine game with total weighted flow-time objective for each player. They showed that the corresponding sequencing game is convex which implies that the core of it is guaranteed to be non-empty; see Shapley (1971). For arbitrary regular objective function and for a special class of games referred to as σ_0-Component Additive Games Curiel *et al.* (1994) prove that the core is non-empty and propose one such core element. Hamers, Borm and Tijs (1995) extend the class of single machine sequencing to the case where jobs have release times. It is shown that these games are not convex except for the case of unit processing times or unit weights for all jobs. Instead of ready times, Borm *et al.* (2002) consider the case with due dates. For three different due date related cost criteria they have shown that the core is non-empty and proved convexity for only a special subclass. Hamers *et al.* (2002) imposed precedence constraints on the jobs and proved that the corresponding games are convex, if the precedence relations are parallel chains. The first work where each player may have more than 1 job to be processed is presented by Calleja *et al.* (2004) where each job might be of interest to more than one player. They showed that the core is non-empty if the underlying cost functions are additive with respect to the initial order of jobs. However, convexity does not necessarily hold. Hamers *et al.* (1999) are the first to consider games with more than 1 machines. For m parallel identical machines and the weighted flow time objective for every player they prove that the core of the associated games is non-empty when $m = 2$ and for some special cases when $m > 2$. Another model using multiple machines is presented by Calleja *et al.* (2002) where every player has precisely two operations, each processed on a specific machine and the objective is the maximum between the weighted completion times of the two operations. It's shown that these games are not convex, but the core is non-empty.

A major assumption of this research is that operations in N_m are of similar nature for every $m \in M$ and hence the setup time incurred by P to switch between jobs of different manufacturers is negligible. This is the case for instance in computer testing, testing of mechanical parts and products, quality control operations, certain industrial finishing processes, assembly operations, etc.

Another key assumption of our model is the information sharing protocol amongst the various parties. We assume that there is end-to-end visibility of the booking schedule of P and that all manufacturers have the opportunity to partner so as to maximize the benefit of their coalition. There is ample evidence that many industry sectors move towards such protocol. In June 2000 Cisco launched its *eHub* initiative as the central point for planning and execution of tasks across the company's extended manufacturing supply chain (Grosvenor and Austin, 2001). EHub is an enterprise resource planning system (ERP) that facilitates end-to-end visibility, event alerting, cost and inventory reductions to more than 2000 of the company's partners. The benefits due to the system were \$1.3B by June 2001 and customer satisfaction is at all-time high. To bring eHub to life, Cisco partnered with Manugistics - a developer of ERP systems - to help create its intelligent private trading network. EHub evolved in 3 phases: Phase 1 incorporates information and management capabilities of the system, phase 2 is the heart of the ERP system and incorporates production planning, joint optimization, and some procurement capabilities, and phase 3 extends the system capabilities to product data and changes management. Grosvenor and Austin, 2001 state that "eHub will take Cisco to the next level of value delivery by enabling Cisco and its partners to plan and optimize joint operations collaboratively. By leveraging information in the common data model and using the Manugistics advanced planning tools, Cisco's contract manufacturing base will be able to optimize production plans and allocations to meet customer satisfaction goals at the lowest possible cost". Motorola's Semiconductor sector joined eHub in June 2000 and RosettaNet formed a cross-industry work group in late 2000 to define requirements for an international Hub (iHub). Today OEMs, contract electronic manufacturers and component suppliers sponsor the iHub design. As a result, eHub and iHub are popular information exchange protocols with thousands of subscribers that facilitate the modeling assumptions stated earlier.

We start the next section with a formal definition of the game between the manufacturers and a formal description of our model together with supporting assumptions. Preliminary results are presented in Section 3. An algorithm is presented in Section 4 to solve optimally the coordination problem faced by P. The same algorithm is used by eac manufacturer to solve his booking problem. Comparing the two costs over all manufcturers yields the savings due to coordination. In Section 5 a savings sharing scheme is presented for a fair allocation of savings to the manufacturers. This scheme is then extended in Section 6 to the case where overtime is made available by P during booked days. In Section 7 we report our insights on the value of coordination. Concluding remarks are made in Section 8.

2 Model Assumptions

Let us first define the savings game amongst the manufacturers. For every coalition $S \subseteq M$ of manufacturers let $v(S)$ denote the cost savings obtained by rescheduling jobs in $\cup_{m \in S} N_m$ in windows from $\cup_{m \in M} \mathcal{W}_m$. These cost savings include a *refund* ρh_t that the coalition S receives if window W_t is freed up in the coordinated schedule for S, $0 \leq \rho \leq 1$. Then, the pair (M, v) defines a game. The release of a window allows P to resell it in the future at full price and still withhold a fee $(1-\rho)h_t$ for the original booking. Hence, releasing unnecessary windows presents a win-win proposition for P and the manufacturers in S. Still, the schedule proposed by P may worsen the objective function value of one or more manufacturers. To make coordination possible, party P seeks a balanced allocation $x_1, x_2, \ldots, x_{|M|}$ of the savings $v(M)$ such that

$$x_m \geq 0, \quad \sum_{m \in M} x_m = v(M), \quad and \quad \sum_{m \in S} x_m \geq v(S) \quad for \quad all \quad S \subset M.$$

The last 3 (in)equalities completely describe the coordination game and are referred to as the *core* (in)equalities. Next we discuss the assumptions of our model.

So far we assumed that P announces her availability at time 0, every $m \in M$ determines his manufacturing windows $\mathcal{W}_m$ at time 0, then P replaces the original schedule with the coordinated one at time 0, and the resulting schedule is executed starting at time 0. This is done for modeling simplicity only. In reality, at time 0 party P announces her (future) availability following day t. Between days 0 and t, manufacturers in M reserve capacity amongst the available days in a first-come-first-serve (FCFS) fashion, say $1, 2, \ldots, |M|$. Then,

$$\mathcal{W}_m \subseteq \{W_1, W_2, \ldots, W_T\} - \mathcal{W}_1 - \ldots - \mathcal{W}_{m-1} \quad for \quad m \in M.$$

At time t, party P reschedules all jobs so as to maximize the savings for set M as a whole. Without loss of generality we assume that all activities that take place during the first t production days $[0, t]$ happen instantaneously at time 0. Similarly, we assume that all jobs in N are available at time 0.

Further, we assume that the total processing requirements of jobs in N_m do not exceed the third-party available capacity for which booking prices are announced. Otherwise m has to seek service at another third-party. We make the following important modeling assumptions:

1. Every job may be preempted during a manufacturing window and resume during a subsequent one.

2. All manufacturing windows $W_t = [a_t, b_t]$ have the same amount of available daily capacity, i.e., $b_t = a_t + L$ for $t = 1, 2, \ldots, T$.

3. For every $t = 1, 2, \ldots, T$ the booking cost h_t may only take on the values $\{h, h_P\}$ where h is the booking cost on a regular demand day while h_P is the cost on a peak demand day, and $h_P > h$.

4. For every job $j \in N_m,\ m \in M$, its due-date d_j coincides with the end of a manufacturing window, i.e., $d_j = b_t$ for some $1 \leq t \leq T$.

Assumption 1 reflects common practice where operations stop at the end of the day and resume the next day. If additional setup is required to restart operations, it is performed prior to the beginning of the morning shift thus preserving intact the productive daily capacity of P. Assumption 2 captures a usual work day of $L = 8$ or 16 hours (later on we extend our base model to the case where overtime is allowed up to a given limit). Assumption 3 allows P to segment her capacity so as to charge a premium during peak demand periods. Assumption 4 is in line with the practice where jobs are delivered in batches at the end of each day. When the time between the end of the working day and the start of the next is enough to complete delivery of jobs, then the distribution of non-tardy jobs does not alter their status as early or tardy. On the other hand, if the transportation times are longer, they are subtracted from the due-date committed to the customer. In both cases, we assume that the transportation times are already subtracted from the d_i values. For example, suppose that customer i needs job j on day 51 and the transportation lead time is 2 days. Then, we let $d_j = 49$. According to assumption 4, as long as the job is completed by the end of day 49 it will be delivered to the customer on time. Otherwise it will be tardy and will incur a lateness penalty.

As mentioned earlier, the booking problem faced by each manufacturer $m \in M$ and by P is the same except for the set of windows available. For simplicity, we limit this part of our presentation to the problem faced by P and the case that overtime is not available (i.e., $O = 0$ or $\alpha = \infty$). The case where $O > 0$ is considered explicitly in Section 6. Party P selects windows from $\mathcal{W}_0 = \{1, 2, \ldots, T\}$ and optimizes the total booking plus tardiness penalty for jobs in $N = \cup_{m \in M} N_m$. The following decision variables are needed to formulate this *third-party* model (3P):

$C_j(\sigma) > 0$: completion time of $j \in N$ at the third-party in a job schedule σ,

y_j^t: grouping variable that equals 1 if $j \in N$ is completed during W_t and 0 otherwise,

p_j: the processing time of job $j \in N$

$U(z)$: 1 if $z > 0$; 0 otherwise.

We seek a permutation σ of the jobs in N and a subset of manufacturing windows among those offered by the third party. The subset of manufacturing windows selected can be determined by the set $Y = \{y_j^t\}$ of variables. Our objective is to minimize the total cost $C(Y, \sigma)$ which includes the cost of booking manufacturing windows plus the tardiness penalty for jobs completed after their desired due-date. In other words,

$$C(Y, \sigma) = \sum_{t=1}^{T} [h_t U(\sum_{j \in N} y_j^t)] + \beta \sum_{j \in N} U(C_j(\sigma) - d_j).$$

To formulate our model we pre-process the values $d_j : j \in N$ and $a_t,\ b_t : 1 \leq t \leq T$ so as to produce an equivalent problem where all manufacturing windows available to the third-party are consecutive, i.e., $b_t = a_{t+1},\ t = 1, 2, \ldots, m-1$. Recall that the due dates d_j are already revised and account for the transportation delay from the third-party to the customer. Then, consider the transformation

$$\begin{aligned} a'_t &= (t-1)L \qquad for \ \ t = 1, 2, \ldots, T, \\ b'_t &= tL \qquad for \ \ t = 1, 2, \ldots, T, \\ d'_j &= b'_{t_j} \qquad for \ \ j \in N, \end{aligned} \tag{1}$$

where t_j is the value for which $d_j = b'_{t_j}$, $1 \leq t_j \leq T$. Then, all manufacturing windows available to the third-party are consecutive. In the rest of this paper we will assume that the parameters

$a_t,\ b_t$ for $t = 1, 2, \ldots, T$ and d_j for $j \in N$ already satisfy (1).

$$(3P) \quad Min \quad \sum_{t=1}^{T} h_t \max_{j \in N} y_j^t + \beta \sum_{j \in N} \sum_{\substack{t_j \le t \le T \\ t_j : d_j = b_{t_j}}}^{T} y_j^t \tag{2}$$

$$s.t. \qquad \sum_{r=1}^{t} \sum_{j \in N} p_j y_j^r \le tL \qquad 1 \le t \le T \tag{3}$$

$$\sum_{t=1}^{T} y_j^t = 1 \qquad j \in N \tag{4}$$

$$y_j^t \in \{0, 1\} \qquad j \in N, \quad 1 \le t \le T. \tag{5}$$

Objective (2) includes the booking and tardiness costs respectively in the 2 summation terms. By definition, only one of the variables $y_j^t :\ 1 \le t \le T$ equals 1, job j completes within W_t and cost h_t is paid to book window W_t. The second term of (2) adds a penalty β for every tardy job. Constraints (3) state that the total processing time of jobs completed during windows $W_1, W_2, \ldots, W_t$ does not exceed the available capacity of these windows. Equations (4) ensure that every job is completed in some window. In the above formulation is accurate only if we assume that $\max_j p_j \le L$. This condition ensures that at least one job completes in every manufacturing window among those booked. Otherwise, one large job may consume a window W_{t_0} entirely and complete in a subsequent window. In this case the term $\sum_{t=1}^{T} h_t \max_{j \in N} y_j^t$ would not add the cost h_{t_0} associated with W_{t_0}. Resolving this peculiarity complicates the presentation of (3P) and is unnecessary for our purposes. Our solution algorithms however are not affected by this peculiarity.

3 Preliminary Results

In this section we present several structural properties of optimal solutions for problem (3P).

Property 1 *There exists an optimal set of windows $\mathcal{W}^\star$ for (3P) that utilizes $w = \lceil \frac{\sum_j p_j}{L} \rceil$ windows such that the following properties hold:*
i) There exists a critical window W_c with booking cost $h_c = h_P$ such that windows $W_1, \ldots, W_{c-1}$ are also booked.
ii) All windows in $\mathcal{W}^\star$ following W_c have booking cost h.
iii) All windows in $\mathcal{W}^\star$ except possibly the last are fully utilized (i.e., the total workload assigned to these windows equals L).
iv) Non-tardy jobs are scheduled in EDD order and precede all tardy jobs.

Properties i-iii follow from straightforward interchange arguments and are given without proof. Property iv results from the mechanics of an algorithm by Moore and Hodgson [12] that solves the single machine problem $1|d_i|\sum_i U_i$ of minimizing the number of tardy jobs; see [13] for the 3-field notation for scheduling problems. For a given collection of windows $\mathcal{W}$, problem (3P) is equivalent to $1|d_i|\sum_i U_i$ because transformations (1) render (3P) equivalent to a single processor and hence Property iv holds. Algorithm MH will be used repeatedly and is presented next.

Algorithm MH (Moore, 1968)

1. Order jobs in N in the earliest due-date (EDD) order; let $\mathcal{L}$ be the resulting ordered set.
2. Schedule the first job in $\mathcal{L}$, say j, next to the last non-tardy job in $N - \mathcal{L}$. Let C_j be the resulting completion time and set $\mathcal{L} := \mathcal{L} - \{j\}$.
3. If $C_j > d_j$ then select a non-tardy job in $N - \mathcal{L}$ with longest processing time, say j_0, declare it tardy and schedule it last.
4. If $\mathcal{L} \neq \emptyset$ then goto step 2.

Algorithm MH takes $\mathcal{O}(|N|\log|N|)$ time due to the ordering in step 1. Whenever the currently scheduled job becomes tardy, the longest preceding job is made tardy in step 3 so as to allow as many of the jobs left in $\mathcal{L}$ to complete on-time. When the jobs to be scheduled are specified, it is convenient to use the following notation:

MH($\mathcal{W}$)= the schedule produced when MH is applied to the single machine problem associated with the collection $\mathcal{W}$ of windows.

$Z(\mathcal{W})$= the total booking plus tardiness penalty cost corresponding to schedule MH($\mathcal{W}$).

The following structural property holds for (3P).

Proposition 1 *There exists an optimal solution for (3P) where the optimal tardy set $T^\star$ is $T^\star \supseteq T_{MH}$, where T_{MH} is the tardy set in MH($\{1, 2, \ldots, w\}$).*

Proof: Consider an optimal schedule $\sigma^\star$ for (3P) with associated tardy set T^* such that $T_{MH} - T^\star \neq \emptyset$. Let $j_0 \in T_{MH} - T^\star$ be a job with smallest due date d_0 - break ties with largest processing time p_0. Suppose that j_0 is the l-th job identified as tardy by algorithm MH($\{1, 2, \dots, w\}$) and d_0 the associated due date of the job j_0 in step 3 (i.e. $C_j > d$). Since algorithm MH finds the minimum number of tardy jobs, the fact that j_0 is nontardy in $\sigma^\star$ and j_0 has the smallest due-date among jobs in $T_{MH} - T^\star$, there must exist job $j'_0 \in T^\star - T_{MH}$ with processing time $p'_0 \leq p_0$ and due-date $d'_0 \leq d$. We will interchange the positions of jobs j_0, j'_0 in $\sigma^\star$ and show that the resulting schedule σ has total cost no more than the cost of $\sigma^\star$.

Indeed, after the interchange, in σ job j_0 completes at the same time as j'_0 completes in $\sigma^\star$. Since j'_0 is tardy in $\sigma^\star$ and $d'_0 \leq d_0$ we conclude that job j_0 must be tardy in σ. On the other hand, observe that when j_0 became tardy in MH($\{1, 2, \dots, w\}$), job j'_0 was non-tardy because $j'_0 \notin T_{MH}$. Also, $d'_0 \leq d_0$. Therefore, when jobs in $N - T_{MH}$ (and hence job j'_0) are scheduled in EDD order in $\sigma^\star$, none becomes tardy. Then, in σ all jobs $j \in N - (T_{MH} \cap T^\star) - \{j_0\}$ with due-date $d_j \leq d_0$ and processing time $p_j \leq p'_0$ are nontardy. Therefore, j'_0 is nontardy in σ but j_0 is not which implies that, after the interchange the number of tardy jobs in σ is no greater than the number of tardy jobs in $\sigma^\star$. Therefore, the total tardiness cost of σ is no more than that of $\sigma^\star$. Moreover, interchanging j_0, j'_0 does not affect the booking cost. Hence, the total (booking plus tardiness) cost of σ is no worse than that of $\sigma^\star$. Repeating this argument for all jobs in $T_{MH} - T^\star$ yields the result. This completes the proof of the proposition. $\square$

4 Optimal Capacity Booking

In this section we develop a polynomial time algorithm called *Regular* for (3P). In this problem overtime is prohibited either because it is not available, or because its cost is too high. The solution to (3P) is a building block for the case where overtime is available at the end of a regular production day.

Algorithm *Regular* starts with a collection $\mathcal{W}(0)$ of windows and proceeds with pair wise exchanges that increase the number of non-tardy jobs but at the same time increase the total booking cost. Exchanges are made so as to identify the critical window W_c; i.e., the last window that has booking cost h_P. Then, Properties i-iii specify an optimal set of windows.

Consider the sets

$$\mathcal{W}_R = \{t | h_t = h\}, \quad and \quad \mathcal{W}_P = \{t | h_t = h_P\}.$$

Let $\mathcal{W}^*$ be an optimal solution for (3P). The number of windows in $\mathcal{W}^* \cap \mathcal{W}_P$ cannot exceed $l_P = |\mathcal{W}_P \cap \{1, 2, \ldots, w\}|$ because then earlier and cheaper windows in $\mathcal{W}_R$ could replace later more expensive windows in $\mathcal{W}^*$. Similarly, the number of regular windows in $\mathcal{W}^*$ that start after window W_w cannot exceed $l_R = |\mathcal{W}_R \cap \{w+1, w+2, \ldots, T\}|$. Let

$$\mathcal{W}_P \cap \{1, 2, \ldots, w\} = \{\tilde{t}_1, \ldots, \tilde{t}_{l_P}\}$$

ordered in increasing order of start times of the corresponding manufacturing windows, and

$$\mathcal{W}_R \cap \{w+1, \ldots, T\} = \{\hat{t}_1, \ldots, \hat{t}_{l_R}\}$$

ordered in decreasing order of start times of the corresponding windows. Note that,

$$\tilde{t}_1 < \tilde{t}_2 < \ldots < \tilde{t}_{l_P} < \hat{t}_{l_R} < \hat{t}_{l_R - 1} < \ldots < \hat{t}_1.$$

Let $\mathcal{W}(0)$ be the collection of the w earliest windows in $\mathcal{W}_R$. If $w_R < w$, then $\mathcal{W}(0)$ includes the earliest $w - w_R$ windows in $\mathcal{W}_P$. Using $\mathcal{W}(0)$, define the collections

$$\mathcal{W}(r) = \mathcal{W}(0) + \{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_r\} - \{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_r\} \quad for \;\; r = 1, 2, \ldots, l = \min\{l_R, l_P\}.$$

Properties i-iii suggest that one of the collections $\mathcal{W}(0)$, $\mathcal{W}(1)$, ..., $\mathcal{W}(l)$ solves (3P) optimally. The following algorithm produces an optimal solution for (3P) without necessarily checking all of these collections.

Algorithm Regular

Input : Collections $\mathcal{W}_R$ and $\mathcal{W}_P$, job set N and booking costs $h_t \;:\; 1 \le t \le T$
Output : Optimal collection $\mathcal{W}^*$ and schedule $\sigma^\star$ for (3P)
Begin
[1] Identify sets $\mathcal{W}(0)$, $\{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_l\}$ and $\{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_l\}$ and let $r = 0$, $\mathcal{W}^* = \mathcal{W}(0)$
[2] Set $r = r + 1$
[3] **If** $Z(\mathcal{W}(r)) \le Z(\mathcal{W}^*)$ **then** $\mathcal{W}^* = \mathcal{W}(r)$ **and goto** [2]
else $\sigma^* = MH(\mathcal{W}^*)$ and **Stop**
End

The algorithm starts with $\mathcal{W}(0)$ and exchanges windows $W_{\tilde{t}_r}$ and $W_{\hat{t}_r}$ for $r = 1, 2, \ldots, l$. The first time such exchange is not profitable the algorithm reports the optimal collection $\mathcal{W}^*$ and schedule σ^* and terminates. The equality in the if-condition in line [3] ensures that a minimum-cost schedule is found that has as many non-tardy jobs as possible. The computational effort spent on each iteration of *Regular* is dominated by the effort to compute $Z(\mathcal{W}(r))$ in line [3]. This takes $\mathcal{O}(n \log n)$ time because it involves an application of algorithm MH. The number of iterations does not exceed $l \leq \min\{w, |\mathcal{W}_R|, |\mathcal{W}_P|\}$ because the search for schedules that satisfy properties i-iii terminates when we run out of unused peak or regular windows. Hence, the total effort required is no more than $\mathcal{O}(\min\{n_R, n_P\} n \log n)$.

In our next theorem we prove that *Regular* solves (3P) optimally. This means that the cost associated with the collections $\mathcal{W}(0)$, $\mathcal{W}(1)$, ..., $\mathcal{W}(s) = \mathcal{W}^*$ is non increasing (for some $0 \leq s \leq l$) and that the collections $\mathcal{W}(s+1)$, ..., $\mathcal{W}(l)$ have cost larger than $Z(\mathcal{W}^*)$. We need the following lemma. Let E_r be the set of non-tardy jobs in schedule MH($\mathcal{W}(r)$) and $\tau(\mathcal{W}(r))$ the number of tardy jobs, $r = 0, 1, \ldots, l$. Then,

Lemma 1 *The following properties hold for collections $\mathcal{W}(r)$, $r = 0, 1, \ldots, l$.*
i) $E_1 \subseteq E_2 \subseteq \ldots \subseteq E_l$, and
ii) $2\tau(\mathcal{W}(r)) \leq \tau(\mathcal{W}(r-1)) + \tau(\mathcal{W}(r+1))$ for $r = 1, \ldots, l-1$.

Proof: To prove i) notice that for $r = 1, \ldots, l$ we have $\mathcal{W}(r) = \mathcal{W}(r-1) + \{\tilde{t}_r\} - \{\hat{t}_r\}$. Moreover, every window with index in $\{\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_l\}$ starts later than any window with index in $\{\tilde{t}_1, \tilde{t}_2, \ldots, \tilde{t}_l\}$. Therefore, $W_{\tilde{t}_r}$ starts before $W_{\hat{t}_r}$ and hence all non-tardy jobs in schedule MH($\mathcal{W}(r-1)$) remain nontardy in MH($\mathcal{W}(r)$), i.e., $E_{r-1} \subseteq E_r$ for $r = 1, \ldots, l$. This establishes part i).

To prove part ii) we will first show that

$$|E_{r+1} - E_r| \leq |E_r - E_{r-1}|.$$

Consider the collections $\mathcal{W}(r-1)$ and $\mathcal{W}(r)$ and the associated sets of nontardy jobs E_{r-1} and E_r found by MH($\mathcal{W}(r-1)$) and MH($\mathcal{W}(r)$) respectively. According to the mechanics of algorithm MH, in schedule MH($\mathcal{W}(r)$) the nontardy jobs in $A_r^E = E_r - E_{r-1}$ are chosen from set $A_r = \{j \in N - E_{r-1} \ : \ d_j \geq b_{\tilde{t}_r}\}$ and A_r^E has maximum cardinality among subsets of A_r such that the jobs in $E_{r-1} + A_r^E$ are nontardy when booking the windows in $\mathcal{W}(r)$. Similarly, In schedule MH($\mathcal{W}(r+1)$) the nontardy jobs in $A_{r+1}^E = E_{r+1} - E_r$ are chosen from $A_{r+1} = \{j \in N - E_r \ : \ d_j \geq b_{\tilde{t}_{r+1}}\}$ and A_{r+1}^E has maximum cardinality among subsets of A_{r+1}

such that the jobs in $E_r + A^E_{r+1}$ are nontardy when using the windows in $\mathcal{W}(r+1)$. Note that $\tilde{t}_r < \tilde{t}_{r+1}$ implies $b_{\tilde{t}_r} < b_{\tilde{t}_{r+1}}$ and therefore $A_{r+1} \subseteq A_r$. Then, the maximality of A^E_r implies

$$\begin{aligned}
|A^E_r| &\geq |A^E_{r+1}| \\
|E_r - E_{r-1}| &\geq |E_{r+1} - E_r| \\
[|N| - \tau(\mathcal{W}(r))] - [|N| - \tau(\mathcal{W}(r-1))] &\geq [|N| - \tau(\mathcal{W}(r+1))] - [|N| - \tau(\mathcal{W}(r))] \\
\tau(\mathcal{W}(r)) - \tau(\mathcal{W}(r-1)) &\leq \tau(\mathcal{W}(r+1)) - \tau(\mathcal{W}(r))
\end{aligned}$$

which yields ii). This completes the proof of the lemma. □

Theorem 1 *Algorithm Regular correctly solves (3P)*

Proof: *Regular* terminates in line [4] when $Z(\mathcal{W}(s)) > Z(\mathcal{W}^*)$ for some $1 \leq s \leq l$. We first consider the special case where $\mathcal{W}^\star = \{1, 2, \ldots, w\}$. The remaining cases will be examined subsequently. In this special case $\mathcal{W}^\star$ consists of the w earliest windows and hence the number of tardy jobs is the smallest possible (see also Proposition 1). Collection $\mathcal{W}(0)$ on the other hand consists of the w cheapest windows with respect to booking cost. Moreover, both collections consist of precisely w windows and hence $\mathcal{W}^*$ differs from $\mathcal{W}(0)$ in precisely $l_P = |\mathcal{W}_P \cap \{1, 2, \ldots, w\}|$ windows.

In this case, observe that for $r > l_P$, the booking cost of $\mathcal{W}(r)$ is greater than the booking cost of $\mathcal{W}(r-1)$ by precisely $h_P - h$ units. This means that $\mathcal{W}(l_P)$ has smaller booking cost than $\mathcal{W}(r)$ for $r > l_P$. Also, Proposition 1 implies that the number of tardy jobs in $\mathcal{W}(l_P) = \{1, 2, \ldots, w\}$ is the smallest possible. On the other hand, the fact that $\mathcal{W}^* = \{1, 2, \ldots, w\}$ means that $Z(\mathcal{W}^*) = Z(\mathcal{W}(l_P)) \leq Z(\mathcal{W}(r))$ for $r < l_P$. Therefore, if $\mathcal{W}^* = \{1, 2, \ldots, w\}$ we have

$$Z(\mathcal{W}^*) = Z(\mathcal{W}(l_P)) \leq Z(\mathcal{W}(r)) \quad for \ \ r \neq l_P,$$

i.e., $\mathcal{W}^*$ is optimal. The only other possibility is that *Regular* terminates after $s < l$ successful verifications of the if-statement in line [3]. This case (i.e., $\mathcal{W}^* = \mathcal{W}(s)$) is considered next by showing

$$Z(\mathcal{W}(s)) < Z(\mathcal{W}(r)) \ \ for \ \ r = 1, 2, \ldots, s-1, s+1, \ldots, l.$$

The first s visits of line [3] yield

$$Z(\mathcal{W}(0)) \geq Z(\mathcal{W}(1)) \geq \ldots \geq Z(\mathcal{W}(s-1)) \geq Z(\mathcal{W}(s)).$$

In the $(s+1)$-st visit of line [3] *Regular* terminates because $Z(\mathcal{W}(s+1)) > Z(\mathcal{W}(s))$. It suffices to show that $Z(\mathcal{W}(s+1)) \leq Z(\mathcal{W}(s+2)) \leq \ldots \leq Z(\mathcal{W}(l))$. This is done iteratively starting with $Z(\mathcal{W}(s+1)) \leq Z(\mathcal{W}(s+2))$. Expanding this inequality we get

$$\begin{aligned}
\sum_{t\in\mathcal{W}(s+1)} h_t + \beta\tau(\mathcal{W}(s+1)) &\leq \sum_{t\in\mathcal{W}(s+2)} h_t + \beta\tau(\mathcal{W}(s+2)) \\
\beta[\tau(\mathcal{W}(s+1)) - \tau(\mathcal{W}(s+2))] &\leq \sum_{t\in\mathcal{W}(s+2)} h_t - \sum_{t\in\mathcal{W}(s+1)} h_t = h_P - h. \qquad (6)
\end{aligned}$$

Expanding $Z(\mathcal{W}(s)) < Z(\mathcal{W}(s+1))$ in a similar fashion yields

$$\beta[\tau(\mathcal{W}(s)) - \tau(\mathcal{W}(s+1))] \quad < \quad h_P - h. \qquad (7)$$

In light of (7), to prove inequality (6) suffices to show

$$\beta[\tau(\mathcal{W}(s+1)) - \tau(\mathcal{W}(s+2))] \quad \leq \quad \beta[\tau(\mathcal{W}(s)) - \tau(\mathcal{W}(s+1))].$$

This holds for every $s = 0, 1, \ldots, l-2$ due to part ii) of Lemma 1. Hence, $Z(\mathcal{W}(s+1)) \leq Z(\mathcal{W}(s+2))$. Iteratively we establish that $Z(\mathcal{W}(s)) < Z(\mathcal{W}(s+1)) \leq \ldots \leq Z(\mathcal{W}(l))$. This completes the proof of the theorem. $\square$

5 Sharing the Savings

In this section we study the cooperative game (M, v). We first show that this game is not *convex.* A game is said to be convex when

$$\forall\ S, T \subseteq M, \quad v(S \cup T) + v(S \cap T) \geq v(S) + v(T).$$

Indeed, consider instance with $M = \{1, 2, 3\}$, $N = \{1, 2, 3\}$, each manufacturer owns a single job that has unit processing time, $d_j = 9$ for $j = 1, 2, 3$, $T = L = 3$, $h_1 = h_2 = 0$, $h_3 = 100$, $\rho = 1$ and β any positive number. Recall that the all parameters are rescaled so that the 3 manufacturing windows have no idle time between them (i.e., window $t+1$ starts at time $a_{t+1} = b_t$). Before coordination manufacturers 1 and 2 incur no cost while manufacturer 3 incurs cost of 100. By construction, all jobs finish early. After coordination all 3 jobs are scheduled in window W_1. Consider the coalitions $S = \{1, 3\}$ and $T = \{2, 3\}$. Then, $v(T) = v(S)$=100 because both release W_3 thus earning refund savings of 100. The same is true for coalition $S \cup T = \{1, 2, 3\}$. However, $v(S \cap T) = v(\{3\}) = 0$ because manufacturer 3 cannot create savings all by himself.

Thus, $v(S \cup T) + v(S \cap T) = 100 < 200 = v(S) + v(T)$. This non convexity result leaves open the question of whether an allocation $x_i : i \in M$ exists that satisfies the core (in)equalities. An empty core would render coordination hard to achieve because then some manufacturers would be better off to join more profitable coalitions than the grand coalition M. Fortunately, this is not the case. We prove this by first showing that game (M, v) is superadditive, i.e.,

$$\forall\ S, T \subset M, \quad with \quad S \cap T = \emptyset, \quad v(S \cup T) \geq v(S) + v(T).$$

To prove this we need the following notation:

$\mathcal{W}_i^\star$ = optimal subset of windows in $\mathcal{W}_0 - \cup_{m=1}^{i-1} \mathcal{W}_m$ booked by manufacturer i for jobs in N_i, $i \in M$,

$\mathcal{W}^\star(S)$ = optimal subset of windows in $\cup_{i \in S} \mathcal{W}_i^\star$ booked by members of the coalition S for jobs in $\cup_{i \in S} N_i$,

$\tau(\mathcal{W}_i^\star)$ = the number of tardy jobs when MH is applied to windows in $\mathcal{W}_i^\star$ for the job set N_i.

Note that by changing the set of windows available and the jobs to be scheduled algorithm *Regular* can be used to compute $\mathcal{W}_i^\star$, $\mathcal{W}^*(S)$ and the associated optimal schedules. Then, the total savings obtained by a coalition S are

$$v(S) = \sum_{i \in S} [\rho \sum_{t \in \mathcal{W}_i^\star} h_t + \beta \sum_{i \in S} \tau(\mathcal{W}_i^\star)] - [\rho \sum_{t \in \mathcal{W}^\star(S)} h_t + \beta \tau(\mathcal{W}^\star(S))].$$

Let us partition these savings into those due to reduction in tardiness penalties, say $v_T(S)$, and those due to savings in booking costs, say $v_h(S)$:

$$v_T(S) = \beta \sum_{i \in S} \tau(\mathcal{W}_i^\star) - \beta \tau(\mathcal{W}^\star(S)), \quad v_h(S) = \sum_{i \in S} \sum_{t \in \mathcal{W}_i^\star} h_t - \sum_{t \in \mathcal{W}^\star(S)} h_t.$$

Then, $v(S) = \rho v_h(S) + v_T(S)$ for every $S \subseteq M$. We have the following results.

Proposition 2 *Games (M, v), (M, v_T) and (M, v_h) are superadditive.*

Proof: Note that for every $S, T \subseteq M$ with $S \cap T = \emptyset$ we have

$$\sum_{t \in \mathcal{W}^\star(S)} h_t + \sum_{t \in \mathcal{W}^\star(T)} h_t \leq \sum_{t \in \mathcal{W}^\star(S \cup T)} h_t$$

because with respect to booking costs, cooperation amongst members of S means that the manufacturers in S reschedule their jobs so as to maximize their refunds due to windows in $\mathcal{W}^\star(S)$ (and same holds for T). Therefore, $\mathcal{W}^*(S) \cup \mathcal{W}^*(T) \subseteq \cup_{i\in S\cup T}\mathcal{W}_i^\star$. The latter inequality implies

$$v_h(S \cup T) \quad \geq \quad v_h(S) + v_h(T), \tag{8}$$

i.e., (M, v_h) is superadditive. Similarly, observe that

$$\tau(\mathcal{W}^\star(S \cup T)) \quad \leq \quad \tau(\mathcal{W}^\star(S)) + \tau(\mathcal{W}^\star(T)) \tag{9}$$

because one can produce a feasible schedule $\sigma_{S\cup T}$ for coalition $S \cup T$ by concatenating the schedules $\mathrm{MH}(\mathcal{W}^\star(S))$ and $\mathrm{MH}(\mathcal{W}^\star(T))$ obtained by MH for coalitions S and T respectively. Therefore,

$$\begin{aligned} v_T(S \cup T) &= \beta \sum_{i\in S\cup T} \tau(\mathcal{W}_i^*) - \beta\tau(\mathcal{W}^\star(S \cup T)) \\ &\geq \beta \sum_{i\in S\cup T} \tau(\mathcal{W}_i^*) - \beta\tau(\mathcal{W}^\star(S)) - \beta\tau(\mathcal{W}^\star(T)) \quad \textit{due to} \ \ (9) \\ &= \beta\sum_{i\in S} \tau(\mathcal{W}_i^*) - \beta\tau(\mathcal{W}^\star(S)) + \beta\sum_{i\in T} \tau(\mathcal{W}_i^*) - \beta\tau(\mathcal{W}^\star(T)) \\ &= v_T(S) + v_T(T), \end{aligned} \tag{10}$$

i.e., (M, v_T) is superadditive. Adding (8) and (10) yields that (M, v) is superadditive. □

We now define a distribution for each of the games (M, v_T) and (M, v_h) which will be used to define a distribution for game (M, v). Let $[a, b] = \{a, a + 1, \ldots, b\}$, $1 \leq a \leq b \leq |M|$. Then, the 2 distributions are defined as follows:

$$x_i' = v_h([1, i]) - v_h([1, i - 1]), \quad \textit{and}$$

$$x_i'' = v_T([i, |M|]) - v_T([i + 1, |M|]) \ \textit{for} \ \ i \in M.$$

Proposition 3 *Distributions* $\{x_i'\}_{i\in M}$ *and* $\{x_i''\}_{i\in M}$ *are in the core of the games* (M, v_h) *and* (M, v_T) *respectively.*

Proof: We first prove the proposition for distribution $\{x_i'\}_{i\in M}$. The total savings $v_h([1, i-1])$ obtained by coordinating the schedules of manufacturers in $[1, i - 1]$ are no greater than the savings for $[1,i]$. This is because we can always use an optimal schedule just for $[1,i-1]$ and assign player i his original windows (i.e., $\mathcal{W}_i^\star$) and schedule. However, letting player i use his original windows is not necessarily optimal for coalition $[1,i]$. Therefore $v_h([1,i]) \geq v_h([1,i-1])$ or $x'_i \geq 0$ for every $i \in M$.

Consider arbitrary coalition $S \subseteq M$. Without loss of generality we assume that $S = [a,b]$ otherwise we can apply transformations (1) and restate the following argument as if players in S form a contiguous set $[a,b]$. Then, Proposition 2 is bound to hold. Note that

$$\sum_{i \in S} x'_i = \sum_{i \in [a,b]} [v_T([1,i]) - v_T([1,i-1])] = v_h([1,b]) - v_h([1,a-1]).$$

Then, for $S_1 = [1,a-1]$ and $T_1 = [a,b]$ Proposition 2 yields $v_h([1,b]) - v_h([1,a-1]) \geq v_h([a,b])$. Equivalently, $\sum_{i \in S} x'_i \geq v_h(S)$ for every $S \subseteq M$. In particular, for $a = 1$ and $b = |M|$ the above equalty gives us $\sum_{i \in M} x'_i = v_h([1,|M|]) - v_h(\emptyset) = v_h(M)$. Therefore, all 3 core (in)equalities hold and $\{x'_i\}_{i \in M}$ is in the core of (M, v_h). A similar argument holds for $\{x''_i\}_{i \in M}$. This completes the proof of the proposition. □

The above 2 distributions are very different in regards to the way they distribute savings amongst the players. Tardiness savings are achieved by reducing the number of tardy jobs. This is done by letting each coalition better utilize the windows of its earlier members (i.e. the manufacturers who are early in the FCFS order). Savings are obtained when some late comers schedule more of their jobs early by displacing jobs of some early takers. Then, the number of tardy jobs increases for early takers and decreases for late comers. If this difference is negative the number of tardy jobs for the coalition is smaller after coordination, and tardiness penalty savings are obtained. It is therefore reasonable to reward the early takers by awarding them the savings exclusively. This idea can be implemented using $\{x''_i\}_{i \in M}$ where player i is awarded all the savings beyond $v_T([i+1,|M|])$ as a reward for allowing players in $[i+1,|M|]$ to compete for use of his windows. Contrary to this concept of "fairness", distribution $\{x'_i\}_{i \in M}$ allocates all the savings beyond those obtained by members of $[1,i-1]$ to player i, even though i contributes the least in achieving these benefits. Player i's only contribution is to schedule some of his jobs earlier by displacing early takers, something that is definitely to his advantage with respect to customer service. We conclude that distribution $\{x''_i\}_{i \in M}$ offers a reasonable way to share tardiness savings while $\{x'_i\}_{i \in M}$ appears inappropriate.

On the other hand, late comers may want to obtain a larger portion of the refund savings because it is their own additional unused capacity that allows a coalition to release a window. Without them, the rest of the coalition does not have enough unused capacity to release a window. It is possible in this case, that the late comer will ask to keep the entire savings. This is sensible especially when early takers demand exclusivity for tardiness savings. Distribution $\{x'_i\}_{i \in M}$ allocates all refund gains to the last player among $[1, i]$. To the contrary, distribution $\{x''_i\}_{i \in M}$ allocates all savings to the first player among $[i, |M|]$ even though his own unused capacity is not the one that makes the release possible. We conclude that distribution $\{x''_i\}_{i \in M}$ is not a reasonable way to split booking cost savings while $\{x''_i\}_{i \in M}$ is.

The above discussion suggests a hybrid allocation of savings that uses $\{x'_i\}_{i \in M}$, $\{x''_i\}_{i \in M}$ only for the respective savings for which they are sensible. Namely,

$$x_i = \rho x'_i + x''_i \quad for \quad i \in M.$$

Theorem 2 *Distribution $\{x_i\}_{i \in M}$ is in the core of (M, v).*

Proof: The non negativity of x_i is implied by the nonnegativity of x'_i, x''_i and the fact that $\rho \geq 0$. In Proposition 3 we saw that for any $S = [a, b]$ we have $\sum_{i \in S} x'_i \geq v_h(S)$ and $\sum_{i \in S} x''_i \geq v_T(S)$ because x'_i, x''_i are in the core of the corresponding games. Therefore,

$$\sum_{i \in S} x_i = \rho \sum_{i \in S} x'_i + \sum_{i \in S} x''_i \geq \rho v_h(S) + v_T(S) = v(S).$$

When $S = M$ the above holds as equality because it does so for each of the value functions v_h and v_T. This means that the core equations are satisfied and hence $\{x_i\}_{i \in M}$ is in the core of (M, v). This completes the proof of the theorem. $\square$

6 Optional Overtime Booking

In this section we extend problem (3P) to provide each manufacturer with the option to book overtime during manufacturing windows he already booked for regular production. The availability of overtime provides the opportunity to finish more jobs earlier thus reducing the number of tardy jobs. Also, it allows a manufacturer to utilize the necessary amount of overtime instead of booking a window and utilizing it partially. We assume that each unit of overtime costs α dollars which is assumed to be more expensive than the regular hourly cost in a peak window, i.e., $\alpha > h_P/L$. This assumption captures the reality that overtime is significantly more expensive than regular production. We further assume that a fixed limit of O overtime hours is allowed each day, and that the amount O_t actually booked in window W_t is such that $0 \leq O_t \leq O$. Finally, we assume that the departure time for orders completed during W_t is $b_t + O_t$. All other operations follow the same protocol as in (3P). We refer to this problem as the *third party problem with overtime* or (3PO).

In what follows we develop a polynomial time algorithm for (3PO) and extend the coordination results presented in Section 5. Properties i-iv hold true even when the third-party offers overtime. The following additional property holds.

Property 2
v) There exists an optimal solution for (3PO) such that a critical window W_F utilizes overtime and all preceding ones have $O_t = O$ for every $t < F$.

This is because of properties i-iii and the fact that it is always beneficial to utilize overtime earlier rather than later. The following important lemma narrows down the candidate windows for W_F and highlights a relationship between optimal solutions for (3P) and (3PO).

Lemma 2 *Let $\mathcal{W}^*$ be the collection of windows booked by Regular. Then, there exists an optimal collection $\mathcal{W}_O^*$ for (3PO) such that*

$$\begin{cases} \mathcal{W}_O^* \cap \mathcal{W}_P \subseteq \mathcal{W}^* \cap \mathcal{W}_P & \text{if } \sum_{t \in \mathcal{W}_O^*} O_t \geq \frac{h_P}{\alpha} \\ \mathcal{W}_O^* \cap \mathcal{W}_P \subseteq \mathcal{W}^* \cap \mathcal{W}_P + \{k\} & \text{otherwise,} \end{cases}$$

where W_k is the earliest peak window not in $\mathcal{W}^*$.

Lemma 2 implies that the last peak window used in $\mathcal{W}_O^*$ starts no later than the last peak window in $\mathcal{W}^*$ except possibly when $\sum_{t \in \mathcal{W}_O^*} O_t < \frac{h_P}{\alpha}$. Proofs of the results presented in this section are provided in the appendix. In what follows we assume that an optimal collection $\mathcal{W}_O^*$ has been determined and one needs to construct an optimal schedule σ_O^*. The following 2 schedules facilitate this construction.

- σ_R: Optimal schedule when no overtime is used and MH($\mathcal{W}_O^*$) is found assuming that the last window in $\mathcal{W}_O^*$ has unlimited capacity.

- σ_O: Optimal schedule when the overtime of each window in $\mathcal{W}_O^*$ is fully utilized before allocating work to the following window.

Note that when no overtime is used in the windows of $\mathcal{W}_O^*$, we may need additional capacity to process all workload. This is the reason for allowing unlimited capacity in the last window of $\mathcal{W}_O^*$ in σ_R. Let T_R and E_R be the set of tardy and non-tardy jobs in σ_R respectively. Similarly, let T_O and E_O be the corresponding sets in σ_O and T^*, E^* the corresponding sets in σ_O^*. The next 2 results are important for the identification of T^* which in turn is key in finding $\mathcal{W}_O^*$.

Lemma 3 $T_O \subseteq T^* \subseteq T_R$

In light of Lemma 3 finding T^* is equivalent to identifying jobs in $T_R - T_O$ that must be processed early in σ_O^*. Suppose that the SPT order of $T_R - T_O$ is

$$T_R - T_O = \{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$$

and

$$E^* = E_R + E \quad where \quad E \subseteq T_R - T_O.$$

The following lemma indicates that T^* consists of the first few jobs in the ordered set $\{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$.

Lemma 4 *Suppose that* $T_R - T_O = \{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$. *If* $J_{i_{k+1}} \in E^*$ *then* $J_{i_k} \in E^*$ *for* $1 \leq k < r$.

We can now present an optimal algorithm for (3PO). The algorithm examines all possible schedules $\mathcal{W}_{f,F}$ that satisfy properties i and ii, where f denotes the number of peak windows and F the number of windows where overtime is used. Number F cannot exceed $w = \lceil \frac{\sum_i p_i}{L} \rceil$. These 2 numbers together with properties i and ii completely specify the collection $\mathcal{W}_{F,f}$. Then, algorithm *Overtime* identifies the optimal nontardy set $E^* = E_R + E$ for $E \subseteq \{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$.

Algorithm Overtime

Input : Collections $\mathcal{W}_R$ and $\mathcal{W}_P$, associated booking/overtime costs and values α, β
Output : Optimal collection $\mathcal{W}_O^*$ and schedule σ_O^*
Begin
[1] Aplly *Regular* to find $\mathcal{W}^*$, E_R, T_R
Aplly *Regular* to $\{1, 2, \ldots, w\}$ when each window has length $L + O$; let T_O be the tardy set
Order $T := T_R - T_O = \{\{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}\}$ in SPT order
Let $\mathcal{W}_R := \mathcal{W}^* \cap \mathcal{W}_R$, $\mathcal{W}_P := \mathcal{W}^* \cap \mathcal{W}_P$, $E^* = \emptyset$, $Z_O = \infty$

[2] **For** $F = 0$ **to** $|\mathcal{W}^* \cap \mathcal{W}_P| + 1$ **do**

[3] **For** $f = F$ **to** w **do begin**

[4] $\mathcal{W}_{f,F} := \{F \textit{ earliest windows in } \mathcal{W}_P\} \cup \{f - F \textit{ earliest windows in } \mathcal{W}_R\}$

[5] Set $O_t := 0$ for $1 \le t \le T$, $o := 0$

[6] **For** $k = 1$ **to** r **do begin**

[7] Find smallest integer k_0 such that $PW_k = \sum_{j \in E_R} p_j + \sum_{l=1}^{k} p_{i_l} \le k_0(L + O)$

[8] **If** $k_0 > f$ **then Goto** [11]

[9] **Else if** $PW_k \ge k_0 L + o$ **then begin**

Let t_0 be the k_0-th index in $\mathcal{W}_{f,F}$

$o =: PW_k - k_0 L$

$$O_t = \begin{cases} O & \text{for } t < t_0 \\ PW_k - (k_0 - 1)(L + O) - L & \text{for } t = t_0 \\ 0 & \text{for } t > t_0 \end{cases}$$

end

[10] Find schedule σ with non-tardy set $E_R + \{J_{i_1}, \ldots, J_{i_k}\}$,

using windows $\mathcal{W}_{f,F}$ with overtime schedule $\{O_t \ : \ t \in \mathcal{W}_{f,F}\}$

In σ, delay tardy jobs as late as possible within windows in $\mathcal{W}_{f,F}$

If $Z_O(\sigma) \le Z_O$ **then** $\mathcal{W}_O^* = \mathcal{W}_{f,F}$, $Z_O = Z(\sigma)$, $\sigma_O^* = \sigma$ and $E^* = E_R + \{J_{i_1}, \ldots, J_{i_k}\}$

end

[11] **end**

End

In line [2] *Overtime* exploits Lemma 2 to iterate over candidate peak windows. The total number of windows in $\mathcal{W}_{f,F}$ is F of which $f - F$ are regular. The k_0-th window in $\mathcal{W}_{f,F}$ is the critical window described in property iv and is found in line [7]. Then, the overtime schedule is found in [9]. Knowing T_O, E_O, and $\mathcal{W}_{f,F}$ with its overtime schedule, one schedules nontardy jobs in EDD order and tardy jobs as late as possible (so that tardy jobs do not consume overtime unnecessarily) to obtain σ_O^*. The best schedule is retained in [10]. Correctness of *Overtime* is proved next.

Theorem 3 *Algorithm Overtime solves (3PO) optimally in* $\mathcal{O}(n^2 n_P w)$ *time.*

In what follows we use subscript O to indicate that the corresponding parameter (e.g., $\tau(\cdot)$, $\mathcal{W}^*$, $\mathcal{W}(S)$, etc.) is computed when overtime capacity is available. To simplify notation, instead of $\mathcal{W}_i^*$ we use $\hat{\mathcal{W}_i}^*$ to denote the optimal collection of windows found by algorithm *Overtime* for $i \in M$.

$$v_O(S) = \sum_{i\in S}[\rho \sum_{t\in\hat{\mathcal{W}_i}^\star} h_t + \beta\sum_{i\in S}\tau_O(\hat{\mathcal{W}_i}^\star) + \alpha \sum_{t\in\hat{\mathcal{W}_i}^*} O_t] - [\rho \sum_{t\in\mathcal{W}_O^\star(S)} h_t + \beta\tau_O(\mathcal{W}_O^\star(S)) + \alpha \sum_{t\in\mathcal{W}_O^*(S)} O_t].$$

If we divide these savings into savings due to tardiness plus overtime, say $v_{TO}(S)$ and (as before) those due to savings in booking costs, we have

$$v_{TO}(S) = \sum_{i\in S}[\beta\tau_O(\hat{\mathcal{W}_i}^*) + \alpha \sum_{t\in\hat{\mathcal{W}_i}^*} O_t] - \beta\tau_O(\mathcal{W}_O^\star(S)) - \alpha \sum_{t\in\mathcal{W}_O^*(S)} O_t, \quad v_O(S) = \rho v_h(S) + v_{TO}(S)$$

for every $S \subseteq N$. Then, we have

Proposition 4 *Game (M, v_{TO}) is superadditive.*

Also, by defining

$$\bar{x}_i'' = v_{OT}([i, |M|]) - v_{OT}([i+1, |M|]) \ for \ \ i \in M$$

and

$$\bar{x}_i = \rho x_i' + \bar{x}_i'' \ \ for \ \ i \in M$$

we have

Theorem 4 *Distribution $\{\bar{x}_i\}_{i\in M}$ is in the core of (M, v_O).*

The proof of the latter result is similar to Theorem 2. Evidently, to coordinate game (M, v_O) we bundled overtime with tardiness savings in v_{TO}. It is interesting to note that the overtime savings game is not superadditive. Namely, let

$$v_O'(S) = \alpha\sum_{i\in S}\sum_{t\in\hat{\mathcal{W}_i}^*} O_t - \alpha \sum_{t\in\mathcal{W}_O^*(S)} O_t.$$

Consider the game (M, v_O') where $M = \{1,2\}$, $S = \{1\}$, $T = \{2\}$, $L = 6$, $O = 3$, $T = 2$, $h_1 = h_2 = 9$, $\alpha = 2$, $N_1 = \{1,2\}$, $N_2 = \{3\}$, $p_1 = p_2 = 4$, $p_3 = 1$ and $d_j = \infty$ for $j = 1,2,3$. Manufacturer 1 optimally schedules jobs J_1 and J_2 in W_1 with $O_1 = 2$. Manufacturer 2 books W_2 with $O_2 = 0$. None of the manufacturers can create savings by himself upon his already optimal schedule and hence $v_O'(S) = v_O'(T) = 0$. On the other hand, the optimal schedule for $S \cup T = M$ is to only book W_1 with $O_1' = 3$. Then, prior to coordination the total overtime is $O_1 + O_2 = 2$ while after coordination it becomes $O_1' = 3$. Hence, $v_O'(S\cup T) = 2\alpha - 3\alpha = -\alpha$. Evidently, not only $v_O'(S\cup T) \ngeq v_O'(S) + v_O'(T)$ but also $v_O'(S\cup T)$ is not even nonnegative.

7 The Value of Coordination in Production Planning

In this section we perform an experiment that helps us assess the value of coordination to the various parties involved. In our experiment we generate problem instances by varying the values of parameters L, O, T, p_j and d_j for $j \in N$, M and N_i for $i \in M$, h, h_P, α and β. In what follows we explain the levels given to each parameter in an effort to limit the number of combinations without loosing essential aspects of the problem.

First, we want to let L simulate a usual production day of 8 hours for a single shift or 16 hours for double shift. In the former case we want to allow for up to $O = 4$ hours of overtime. This produces three (L, O) combinations; namely $(8, 0)$, $(16, 0)$ and $(8, 4)$. The first 2 combinations only differ in the number of jobs that can be processed in a day. Instead, we will count time in multiples of 15 minute intervals and let job processing times be multiples. Specifically, we will use $L = 32, 64$ and $O = 16$ and consider only combinations $(32, 0)$ and $(32, 16)$. Processing times will be drawn uniformly from $[1, 5]$ for small jobs and $[4, 10]$ for large jobs. Then, the average small job takes 45 minutes while the average large job takes 2 hours. Due dates for jobs are given so that at least 40% of the workload of each manufacturer is completed late. This is done by letting window $W_t = [(t-1)(L+O), t(L+O)-O]$ for $1 \le t \le T$, ordering all jobs in N_i according to shortest processing time (SPT), and assigning

$$d_j = \left(\sum_{m<i} \left\lceil \frac{\sum_{j \in N_m} p_j}{L} \right\rceil + \left\lfloor \frac{3\sum_{j \in N_i} p_j}{5L} \right\rfloor\right)(L+O) \quad \forall\ j \in N_i.$$

The first term above accounts for windows booked by manufacturers who booked capacity before i and the second term ensures that about 40% of i's workload is processed after the due-date. This will be the case when overtime is not available. When overtime is available algorithm *Overtime* optimally trades tardiness penalties with overtime costs.

Second, after drawing processing times p_j for all $j \in N$, we let T be the smallest integer such that $T \cdot L \ge 1.1 \sum_j p_j$. Then, we randomly draw 15% or 30% of the T windows and designate them as peak windows (i.e., $|\mathcal{W}_P|/T \in \{0.15, 0.3\}$) with cost $h_P = 1.3h$. We solve problems with $|M| = 4$ or 8 to test how the number of manufacturers affects total system savings. We would like to vary $|N_i|$, $i \in M$ so as to assess the effect of the number of windows per manufacturer to system profits. Multiple $|N_i|$ values however, dramatically increase the size of our experiment. Similar effect is achieved if we limit our experiment to $|N_i| = 20$ for every $i \in M$ but experiment with both of the 2 possible ranges of values for p_j. This means that on average every manufacturer needs about 2 regular production windows if p_j is drawn from $[1,5]$ and about 5 windows if drawn from $[4,10]$.

Finally, we select values for α and β. We will set the hourly overtime rate to 50% over the hourly rate h/L on a non-peak day because this is often the case in practice. Namely, $\alpha = 1.5h/L$. In selecting β it is important to consider β values in connection to the threshold value $\delta h = h_P - h$. Whether a peak window is booked instead of a regular one depends on how $h+\beta$ compares to h_P. If $h+\beta < h_P$ then cheap late windows are preferred over earlier peak windows. In this case peak windows would only be booked if the capacity of the third-party due to cheap windows is exhausted. This is not likely, however, since by design we set $T \cdot L = 1.1 \sum_j p_j$. A potential motivation for booking peak windows is to capture a larger share of the cost savings after coordination. It is for this reason that we consider the value $\beta = 0.8\delta h$ in our experiment. On the other hand, if $h+\beta > h_P$ manufacturers would be willing to book earlier peak windows over later cheap ones. This motivates the value $\beta = 1.2\delta h$. Therefore, we select $\beta = 0.8\delta h$ or $1.2\delta h$.

With the above parameter choices there is a total of 32 combinations. For each combination we randomly generate 10 problems and report in Table 1 the following statistics for $O = 0$:

$\Delta C = \frac{\sum_i Z(\mathcal{W}_i^*) - Z(\mathcal{W}^*)}{\sum_i Z(\mathcal{W}_i^*)} 100\%$; percentage savings due to coordination,

$\Delta \tau = \frac{\sum_i \tau(\mathcal{W}_i^*) - \tau(\mathcal{W}^*)}{\sum_i \tau(\mathcal{W}_i^*)} 100\%$; percentage tardiness savings,

$A^+ = 100\% |\{i \; : \; \tau_i(\mathcal{W}_i^*) > \tau_i(\mathcal{W}^*)\}|/|M|$; percentage of manufacturers whose tardiness penalty decreases.

The corresponding statistics for $O > 0$ are denoted by ΔC_O, $\Delta \tau_O$, A_O^+ respectively where $Z(\cdot)$, $\tau(\cdot)$ are replaced by $Z_O(\cdot)$, $\tau_O(\cdot)$ and $\mathcal{W}_i^*$, $\mathcal{W}^*$ are replaced by $\hat{\mathcal{W}_i}^*$, $\mathcal{W}_O^*$ respectively. Moreover, we report

$\Delta O = \frac{\sum_i \sum_{t \in \mathcal{W}_i^*} O_t - \sum_{t \in \mathcal{W}^*} O_t}{\sum_i \sum_{t \in \mathcal{W}_i^*} O_t} 100\%$; the percentage overtime savings,

$\Delta Z = \frac{Z(\mathcal{W}^*) - Z_O(\mathcal{W}_O^*)}{Z(\mathcal{W}^*)} 100\%$; percentage savings due to overtime availability.

Based on the above experiments we make the following observations: i) The value ΔC is of the order of 20%, is greater when $O > 0$, and decreases with the mean processing time $\bar{p}_j$. ii) Note that, if we multiply the numerator and denominator in $\Delta \tau$ by β we obtain the percentage

Table 1: Computational experiment

				$\frac{\lvert\mathcal{W}_P\rvert}{T} = 15\%$					$\frac{\lvert\mathcal{W}_P\rvert}{T} = 30\%$				
(L, O)	$\lvert M\rvert$	β	p_j	ΔC	$\Delta\tau$	A^+			ΔC	$\Delta\tau$	A^+		
$(32, 0)$	4	$0.8\delta h$	$[1, 5]$	18.7	72.3	62.5	-	-	16.8	65.4	65.5	-	-
			$[4, 10]$	16.3	71.1	72.8	-	-	15.0	60.1	69.0	-	-
		$1.2\delta h$	$[1, 5]$	21.2	76.2	67.0	-	-	17.4	67.2	70.8	-	-
			$[4, 10]$	18.1	75.8	63.8	-	-	15.4	70.7	68.5	-	-
	8	$0.8\delta h$	$[1, 5]$	19.1	74.5	65.8	-	-	20.2	77.4	69.5	-	-
			$[4, 10]$	17.5	73.9	62.5	-	-	16.1	74.2	71.8	-	-
		$1.2\delta h$	$[1, 5]$	20.6	77.1	67.5	-	-	21.8	75.6	69.0	-	-
			$[4, 10]$	18.4	76.7	62.0	-	-	18.5	71.0	75.5	-	-
				ΔC_O	$\Delta\tau_O$	A_O^+	ΔZ	ΔO	ΔC_O	$\Delta\tau_O$	A_O^+	ΔZ	ΔO
$(32, 16)$	4	$0.8\delta h$	$[1, 5]$	20.1	48.4	68.0	11.8	32.2	17.4	42.1	70.5	13.4	36.2
			$[4, 10]$	18.6	44.3	64.5	10.7	34.1	16.3	38.2	68.0	12.5	36.7
		$1.2\delta h$	$[1, 5]$	22.1	49.0	62.0	14.3	34.8	19.2	42.4	68.5	15.6	37.3
			$[4, 10]$	19.5	48.5	63.8	13.1	37.4	16.7	41.3	66.8	16.2	39.2
	8	$0.8\delta h$	$[1, 5]$	23.3	52.5	64.5	12.5	27.3	20.5	49.5	72.0	14.1	35.5
			$[4, 10]$	19.0	51.2	63.0	12.1	29.5	17.1	48.8	66.5	14.3	34.6
		$1.2\delta h$	$[1, 5]$	24.9	53.0	64.5	14.3	30.4	22.3	47.0	62.0	15.8	38.8
			$[4, 10]$	20.9	52.8	72.5	16.3	33.9	19.7	46.6	62.0	17.4	40.4

tardiness savings due to coordination. Clearly, $\Delta\tau$ is greater when $O = 0$. This is because ΔC savings are due to tardiness and refund savings only while ΔC_O includes overtime savings. iii) The percentages A^+, A_O^+ of manufacturers with fewer tardy jobs decreases with $|\mathcal{W}_P|/T$. This means that the overall service benefits due to coordination decrease when peak demand periods are longer. iv) The value ΔZ of overtime is significant, increases with β and with the proportion $|\mathcal{W}_P|/T$ of peak windows. v) The ΔO values suggest that the overtime usage after coordination is significantly smaller than before coordination. Observations i-v highlight the importance of coordination among manufacturers and the additional benefits for them and the third-party when overtime is made available. The magnitude of the savings suggests that lack of coordination increases costs significantly and unnecessarily.

8 Conclusion

In this paper we presented a model for outsourcing operations to a single third-party. Our mathematical and empirical analyses offer strong monetary and customer service incentives to effect coordination as well as "fair" saving sharing schemes that facilitate such coordination. To the best of our knowledge this is one of the first works that deals with coordination at the production planning level that accounts for scheduling costs. A wealth of other models is of interest. These models may include different resources (as opposed to a single third party), different objectives (as opposed to booking and tardiness costs), different modes of cooperation (e.g. when manufacturers do not book entire windows but rather reserve capacity every time that a new job enters their system) and different modes of information sharing (e.g., when the manufacturers do not share information about the windows booked by each). Such models will be the focus of our future research.

References

[1] Borm, P., Fiestras-Janeiro, G., Hamers, H., Sánchez, E. and Voorneveld, M. On the Convexity of Games Corresponding to Sequencing Situations with Due Dates, *European Journal of Operational Research*, 136, 616-634, 2002.

[2] Calleja, P., Estévez-Fernández, M. A., Borm, P. and Hamers, H. Job Scheduling, Cooperation and Control, *CentER Discussion Papers 2004-65*, Tilburg University, Tilburg, The Netherlands, 2004.

[3] Calleja, P., Borm, P., Hamers, H., Klijn, F. and Slikker, M. On a New Class of Parallel Sequencing Situations and Related Games, *Annals of Operations Research*, 109, 263-276, 2002.

[4] Curiel, I., Pederzoli, G. and Tijs, S. Sequencing Games, *European Journal of Operational Research*, 40, 344-351, 1989.

[5] Curiel, I., Potters, J., Rajendra Prasad, V., Tijs, S. and Veltman, B. Sequencing and Cooperation, *Operations Research*, 42, 566-568, 1994.

[6] Curiel, I., Hamers, H. and Klijn, F. *Sequencing Games: A Survey* in *Chapters in Game Theory: In Honor of Stef Tijs*, Kluwer Academic Publishers (eds. P. Borm and H. Peters), Boston, pp. 27-50, 2002.

[7] Grosvenor, F., Austin, T.A. Cisco's eHub Initiative, *Supply Chain Management Review*, July/August, 28-35, 2001.

[8] Hamers, H., Borm, P. and Tijs, S. On Games Corresponding to Sequencing Situations with Ready Times, *Mathematical Programming*, 70, 1-13, 1995.

[9] Hamers, H., Klijn, F. and Suijs, J. On the Balancedness of Multimachine Sequencing Games, *European Journal of Operational Research*, 119, 678-691, 1999.

[10] Hamers, H., Klijn, F. and van Velzen, B. On the Convexity Precedence Sequencing Games, *CentER Discussion Papers 2002-112*, Tilburg University, Tilburg, The Netherlands, 2002.

[11] Shapley, L. Cores of Convex Games, *International Journal of Game Theory*, 1, 11-26, 1971.

[12] Moore, J.M., An n job, one machine sequencing for minimizing the number of late jobs, *Management Science* **15**, 102-109, 1968.

[13] Pinedo, M., *Scheduling: Theory, Algorithms and Systems*, Prentice Hall, Englewood Cliffs, NJ. 1995.

Appendix

Proof of Lemma 2: Let σ^*, σ_O^* be optimal schedules for (3P) and (3PO) resp. and $\mathcal{W}^*$, $\mathcal{W}_O^*$ the associated collection of windows. Moreover, suppose that σ_O^* is an optimal solution for (3PO) such that $\mathcal{W}_O^*$ has as many windows in common with $\mathcal{W}^*$ as possible; i.e., σ_O^* is maximal. Define

$Z_O(\sigma)$, $\tau_O(\sigma)$= the optimal cost and the number of tardy jobs resp. associated with schedule σ when overtime is available.

We distinguish 2 cases:

Case i: $\sum_{t \in \mathcal{W}_O^*} O_t \geq x = \frac{h_P}{\alpha}$.

In this case we will start with σ_O^* and construct a schedule with cost no more than $Z_O(\sigma_O^*)$ that has one more window in common with $\mathcal{W}^*$. This would contradict the maximality of σ_O^*. Towards this end let W_k be the earliest peak window in $\mathcal{W}_O^* - \mathcal{W}^*$ and W_y the earliest window in $\mathcal{W}^* - \mathcal{W}_O^*$ (see Figure 1). Window W_y exists because overtime is not available in (3P), i.e., $|\mathcal{W}_O^*| \leq |\mathcal{W}^*|$, and $k \notin \mathcal{W}^*$. We first consider the case where W_k precedes W_y as in Figure 1 where windows are indicated by rectangles, solid lines indicate that a window is booked and dashed lines the possible use of overtime. The case where W_y precedes W_k will be considered later.

INSERT FIGURE 1 HERE

We start with some observations: The booking cost of W_k is h_P by construction. All windows booked in σ^* after W_k have booking cost h otherwise W_k would be included in $\mathcal{W}^*$ in place of a later window with the same booking cost. Windows W_y and all windows after it are not booked in σ_O^* since otherwise W_y would have been booked in place of a later window with same or greater booking cost. We will prove that the inclusion of W_y in σ_O^* produces a schedule whose total cost is no greater than $Z_O(\sigma_O^*)$.

Start with σ_O^* and eliminate the last $x = h_P/\alpha < L$ units of overtime - this may involve more than one windows. Instead, use x units of regular capacity in window W_y. Let $\tilde{\sigma}_O$ be the resulting schedule after applying MH algorithm, and Δ be the number of additional tardy jobs over σ_O^*. We have $\Delta \geq 0$ because W_y starts after any of the x units of overtime eliminated from σ_O^*. Let E_O be the set of non-tardy jobs in $\tilde{\sigma}_O$ and E be the set of non-tardy jobs in the schedule $\tilde{\sigma}$ resulting after MH is applied to collection $\mathcal{W}^* + \{k\} - \{y\}$.

Clearly, $E \subseteq E_O$ because by construction, the k-th unit of production in $\tilde{\sigma}_O$ is available no later than the k-th unit of production in $\tilde{\sigma}$. Therefore, the number Δ of tardy jobs in $\tilde{\sigma}_O$ when x units of processing are delayed until W_y increases by no more than the number of tardy jobs when $L > x$ units are delayed from W_k to W_y (as in σ^*). Hence,

$$\begin{aligned} \Delta &= \tau_O(\sigma_O^*) - \tau_O(\tilde{\sigma}_O) \\ &\leq \tau(\sigma^*) - \tau(\tilde{\sigma}). \end{aligned} \tag{11}$$

The optimality of $\mathcal{W}^*$ implies

$$\beta[\tau(\sigma^*) - \tau(\tilde{\sigma})] \quad < \quad h_P - h \tag{12}$$

otherwise one would book W_k and release W_y thus improving over $Z(\sigma^*)$; contradiction to optimality of $\mathcal{W}^*$. Then,

$$\begin{aligned} Z_O(\tilde{\sigma}_O) &= Z_O(\sigma_O^*) + h + \beta\Delta - \alpha x \\ &\leq Z_O(\sigma_O^*) + h + \beta[\tau(\sigma^*) - \tau(\tilde{\sigma})] - \alpha x \quad \textit{because of } (11) \\ &< Z_O(\sigma_O^*) + h_P - ax = Z_O(\sigma_O^*) \end{aligned}$$

because by choice of x, $\alpha x = h_P$. Hence, $Z_O(\tilde{\sigma}_O) < Z_O(\sigma_O^*)$ contradicting the optimality of σ_O^*. Therefore, in this case $\mathcal{W}_O^* \cap \mathcal{W}_P \subseteq \mathcal{W}^* \cap \mathcal{W}_P$.

Consider now the case where W_y precedes W_k. For the same reasons as before, the fact that $W_y \notin \mathcal{W}_O^*$ implies that all windows following W_y in $\mathcal{W}_O^*$ must have booking cost h and that W_y has booking cost h_P. Since W_y is not booked in σ_O^*, none of the later peak windows are booked either. Then, properties i-iv imply $\mathcal{W}_O^* \cap \mathcal{W}_P \subseteq \mathcal{W}^* \cap \mathcal{W}_P$. This completes case i.

Case ii: $\sum_{t \in \mathcal{W}_O^*} O_t = x < \frac{h_P}{\alpha}$.

In this case $x < L$ because $\alpha > h_P/L$. We make the following observations: First, the subcase where W_y precedes W_k carries through as is. It suffices to consider the case where W_k precedes W_y. Then, in σ^* there exists exactly one booked window, say W_z, following W_y (see Figure 1). This is because the total workload processed in windows in σ^* and σ_O^* is the same and $x < L$. Also, window W_z is utilized for x periods and as before none of the windows following W_y is included in σ_O^* (because $x < L$). Then, W_k is the first and only peak window booked in σ_O^* but not in σ^*. Hence, $\mathcal{W}_O^* \cap \mathcal{W}_P \subseteq \mathcal{W} \cap \mathcal{W}_P + \{k\}$. This completes case ii, and the lemma. □

To avoid confusion, in the following 3 results we use $J_i,\ J_j$ to denote jobs $i, j \in N$.

Proof of Lemma 3: Suppose that jobs are indexed according to the EDD order and $J_i \in T_O$. We will prove that $J_i \in T_O^*$. If not, let J_i be the l-th tardy job when MH obtained σ_O. From the mechanics of MH there must exist $r > i$ such that the completion time of J_r is $C_r > d_r$ when J_r is considered for scheduling. Such J_r must exist otherwise J_i wouldn't become tardy by MH. Let A_l be the set of jobs scheduled in σ_O prior to C_r when J_r is first introduced and A_l^* the set of jobs scheduled in σ_O^* when J_r is first introduced. The total workload prior to C_r in σ_O is no less than the corresponding workload in σ_O^* when J_r is first considered for scheduling. This is because the total overtime used prior to C_r in σ_O is no less than in σ_O^*. Therefore, $A_l^* \subseteq A_l$.

Without loss of generality suppose that in constructing σ_O and σ_O^* ties are broken by MH so that the set A_l^* has as many jobs in common with A_l as possible. This maximality property implies that, whenever possible, the same longest job is made tardy in σ_O and σ_O^*. Since $A_l^* \subseteq A_l$, we have

$$\max\{p_j :\ J_j \in A_l^*\} \leq \max\{p_j :\ J_j \in A_l\} = p_i.$$

If the latter inequality holds strictly, then MH must have already rendered J_i tardy in σ_O^* prior to scheduling J_r, i.e., $J_i \in T_O^*$. On the other hand, if $J_i \in A_l$, the maximality of A_l^* implies that all other jobs in A_l should also be in A_l^* because their processing time is no longer than p_i and the fact that MH always removes the longest among the jobs scheduled so far. Equivalently, $A_l \subseteq A_l^*$ and hence $A_l^* = A_l$. But then, J_r must be tardy when introduced to σ_O^* since it is already tardy when introduced to σ_O which uses at least as much overtime as σ_O^* prior to introducing J_r. Then, MH would render J_i tardy in σ_O^* because it is the longest job in A_l^*. In all cases, $J_i \in T_O^*$. This completes the left hand side of the lemma. The proof for the right hand side is analogous to our argument above except that MH is applied to windows of σ_R. $\square$

Proof of Lemma 4: Suppose that $J_{i_{k+1}}$ is early in σ_O^* but J_{i_k} is tardy. Because of the SPT ordering we have $p_{i_k} \leq p_{i_{k+1}}$. If $d_{i_k} \geq d_{i_{k+1}}$ then swap $J_{i_{k+1}}$ and J_{i_k}. Then, J_{i_k} becomes non-tardy because so is $J_{i_{k+1}}$ even though its processing time is not less than p_{i_k}. Hence, there is an optimal schedule such that if $J_{i_{k+1}} \in E^*$ then $J_{i_k} \in E^*$.

On the other hand, if $d_{i_k} < d_{i_{k+1}}$, then consider the EDD order of jobs in $E_R + E + \{J_{i_k}\} - \{J_{i_{k+1}}\}$. The amount x of overtime needed to process jobs in $E_R + E + \{J_{i_k}\} - \{J_{i_{k+1}}\}$ is no more than the amount x' needed to process jobs in $E_R + E$ because $p_{i_k} \leq p_{i_{k+1}}$. Also, none of the jobs in $E_R + E + \{J_{i_k}\} - \{J_{i_{k+1}}\}$ is tardy because this is the case when x' units of overtime are used in σ_O (because $J_{i_k} \notin T_O$). Therefore, there is an optimal schedule where $J_{i_k} \in E^*$ and $J_{i_{k+1}}$ may or may not be tardy. This completes the proof of the lemma. $\square$

Proof of Theorem 3: Properties i-ii imply that there exists an optimal collection W_O^* that uses the f earliest windows in $\mathcal{W}^* \cap \mathcal{W}_P$, and that the remaining windows are from $\mathcal{W}_R$. Exhaustive enumeration on f, F with $0 \leq f \leq |\mathcal{W} \cap \mathcal{W}_P|$ (as in lines [2], [3]) ensures that an optimal collection that satisfies properties i and ii is evaluated. In the rest of this proof we assume that $\mathcal{W}_O^*$ is such a collection. Then, Lemma 4 implies that

$$E^* \subseteq E_R + \{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$$

and that jobs in $\{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$ are considered for insertion to E^* in SPT order but scheduled in σ_O^* in the EDD order of nontardy jobs as in line [10].

When job J_{i_k} is inserted in EDD order, the total preceding workload PW_k of jobs J_i with $i \geq i_k$ is recomputed and the necessary additional overtime is calculated in line [9]. If the total required overtime exceeds the allowable overtime (i.e., $F \cdot O$), then the iteration is aborted in line [8] and no more jobs from $\{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$ are considered (due to Lemma 4). Otherwise we test in line [10] whether the revised nontardy set produces a better solution and continue by considering the smallest unscheduled job in $\{J_{i_1}, J_{i_2}, \ldots, J_{i_r}\}$. According to Lemma 4, exhaustive enumeration on k in line [6] ensures that E^* is identified.

The possible values of f are $\mathcal{O}(\mathcal{W}'_P)$ while those of F are $\mathcal{O}(w)$. The loop in line [6], involves $\mathcal{O}(n)$ iterations. Lines [7]-[9] take $\mathcal{O}(1)$ time if PW_k and k_0 are computed recursively using

PW_{k-1}. Line [10] takes $\mathcal{O}(n)$ time because E_R and $\{J_{i_1}, \ldots, J_{i_k}\}$ are already ordered. Therefore, lines [4]-[10] take a total of $\mathcal{O}(n^2)$ time. Line [1] takes $\mathcal{O}(n \log n)$ time (asymptotically less than $\mathcal{O}(n^2)$) because it involves 2 applications of MH algorithm and the SPT ordering of jobs in T. Hence, the computational complexity of *Overtime* is $\mathcal{O}(n^2 n_P w)$. This completes the proof of the theorem. □

Proof of Proposition 4: For any 2 given collections of windows $\mathcal{W}_O^*(S)$ and $\mathcal{W}_O^*(T)$, algorithm *Overtime* can improve tardiness and overtime costs by optimizing over all windows available to manufacturers in $S \cup T$. In other words,

$$\begin{aligned}
\beta\tau_O(\mathcal{W}_O^\star(S \cup T)) + \alpha \sum_{t \in \mathcal{W}_O^\star(S\cup T)} O_t \quad \leq \quad & \beta\tau_O(\mathcal{W}_O^\star(S)) + \alpha \sum_{t \in \mathcal{W}_O^\star(S)} O_t \\
& + \beta\tau_O(\mathcal{W}_O^\star(T)) + \alpha \sum_{t \in \mathcal{W}_O^\star(T)} O_t. \qquad (13)
\end{aligned}$$

Therefore,

$$\begin{aligned}
v_{TO}(S \cup T) \quad = \quad & \sum_{i \in S\cup T} [\beta\tau_O(\hat{\mathcal{W}}_i^*) + \alpha \sum_{t \in \hat{\mathcal{W}}_i^\star} O_t] - \beta\tau_O(\mathcal{W}_O^\star(S \cup T)) - \alpha \sum_{t \in \mathcal{W}_O^\star(S\cup T)} O_t \\
\geq \quad & \sum_{i \in S\cup T} [\beta\tau_O(\hat{\mathcal{W}}_i^*) + \alpha \sum_{t \in \hat{\mathcal{W}}_i^\star} O_t] - \beta\tau_O(\mathcal{W}_O^\star(S)) - \alpha \sum_{t \in \mathcal{W}_O^\star(S)} O_t \\
& - \beta\tau_O(\mathcal{W}_O^\star(T)) - \alpha \sum_{t \in \mathcal{W}_O^\star(T)} O_t \qquad \qquad because \;\; of \;\; (13) \\
= \quad & \sum_{i \in S} [\beta\tau_O(\hat{\mathcal{W}}_i^*) + \alpha \sum_{t \in \hat{\mathcal{W}}_i^\star} O_t] - \beta\tau_O(\mathcal{W}_O^\star(S)) - \alpha \sum_{t \in \mathcal{W}_O^\star(S)} O_t \\
& + \sum_{i \in T} [\beta\tau_O(\hat{\mathcal{W}}_i^*) + \alpha \sum_{t \in \hat{\mathcal{W}}_i^\star} O_t] - \beta\tau_O(\mathcal{W}_O^\star(T)) - \alpha \sum_{t \in \mathcal{W}_O^\star(T)} O_t \\
= \quad & v_{TO}(S) + v_{TO}(T)
\end{aligned}$$

i.e., (M, v_{TO}) is superadditive. This completes the proof of the proposition. □

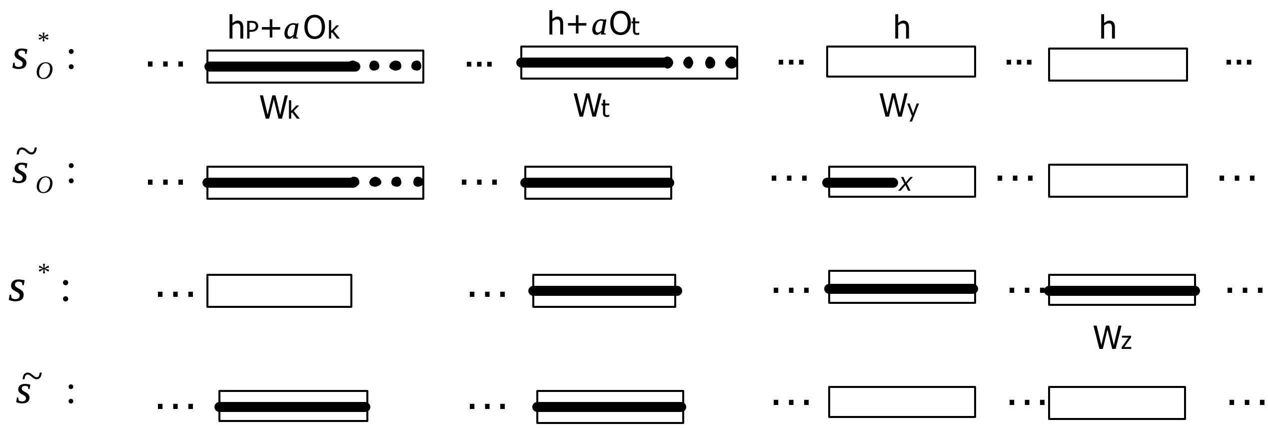

Figure 1