Yunpeng Pan

A combinatorial auctions perspective on min-sum scheduling problems

Abstract In combinatorial auctions, prospective buyers bid on bundles of items for sale, including but not limited to singleton bundles. The bid price given by a buyer on a particular bundle reflects his/her perceived utility of the bundle of items as a whole. After collecting all the bids, the auctioneer determines the revenue-maximizing assignment of winning bidders to bundles subject to nonoverlapping of bundles. To accomplish this, the auctioneer needs to solve a *winner determination problem* (WDP). The exactly same way of thinking can be taken to the context of min-sum scheduling, where jobs can be viewed as bidders who bid on bundles of discrete time periods on machines.

Particular problems often permit only a subset of bundles. By putting appropriate restrictions on the collection of permissible bundles, we can derive from the WDP, various integer programming (IP) formulations for nonpreemptive as well as preemptive min-sum scheduling problems. We thus obtain the well-known time-indexed IP formulation in the nonpreemptive case, and further, a new strong IP formulation in the preemptive case.

1 Introduction

In combinatorial auctions, a set of T items, $\mathcal{T} = \{0, \ldots, T-1\}$, is put up for sale. Prospective buyers numbered $1, \ldots, n$ simultaneously bid on nonempty subsets of the items called *bundles*. More precisely, buyer j casts a collection of bids $\mathcal{B}_j = \{b_{j0}, \ldots, b_{j(K_j-1)}\}$ where each bid $b_{jk} = (\mathcal{S}_{jk}, f_{jk})$ consists of a bundle $\mathcal{S}_{jk} \subseteq \mathcal{T}, \mathcal{S}_{jk} \neq \emptyset$ and a price f_{jk} that buyer j is willing to pay for the bundle. Note that bundles (regardless of their bidders) that contain the same item are mutually exclusive. A revenue-maximizing auctioneer then determines which bids on bundles are won by which buyers, by solving a

Y. Pan: TomoTherapy Inc., 1240 Deming Way, Madison, WI 53717-1954, USA, Tel.: 608-824-3011, E-mail: ypan@tomotherapy.com

combinatorial optimization problem called the *winner determination problem* (WDP) [21,15,11].

By different designs of auction mechanisms, buyers may be allowed to win any number of bundles, ranging from no restriction (e.g., [15]) to at most one bundle (e.g., [11]) to a somewhat hybrid of the two extremes (e.g., [21]). Here we consider a fictitious variant of combinatorial auctions where each buyer will win exactly one bundle (see further discussion in Section 6). Let 0-1 variable y_{jk} $(k = 0, \ldots, K_j - 1; j = 1, \ldots, n)$ indicate whether bid b_{jt} is won. The WDP can be formulated as the following integer programming (IP) problem:

$$\text{(WDP)} \quad \max \sum_{j=1}^{n} \sum_{k=0}^{K_j-1} f_{jk} y_{jk}$$

$$\text{s.t.} \quad \sum_{k=0}^{K_j-1} y_{jk} = 1, \ \forall j = 1, \ldots, n, \tag{1}$$

$$\sum_{j=1}^{n} \sum_{k:\mathcal{S}_{jk} \ni t} y_{jk} \leq 1, \ \forall t = 0, \ldots, T-1, \tag{2}$$

$$y_{jt} \in \{0, 1\}, \ \forall k = 0, \ldots, K_j - 1; j = 1, \ldots, n \tag{3}$$

In the above WDP model, there can be as many as $2^T - 1$ different bundles. The richness of the model lies in the ability to impose different restrictions on permissible bundles.

2 New perspective on min-sum scheduling problems

Let us now turn our focus to min-sum scheduling problems. We will limit our discussion to the single-machine environment with the understanding that it carries over easily to parallel machines. Also, to conform to the convention of the scheduling literature, we switch to an equivalent minimization formulation by redefining f_{jk}.

2.1 Definitions of permissible bundles

In min-sum scheduling problems, we are given n jobs $1, \ldots, n$ with processing times $p_1, \ldots, p_n$ to be processed on a machine that can only process one job at a time. Let $[0, T]$ be the time window during which all jobs must be finished. We assume integrality of the processing times and T. From the perspective of combinatorial auctions, the jobs are the buyers that bid on bundles of the time periods $\mathcal{T} = \{0, \ldots, T-1\}$. Our objective is then to maximize the sum of the prices on the bundles won by the jobs, or equivalently to minimize the sum of bundle costs of the jobs. Without ambiguity, we also use f_{jk} to denote bundle costs.

For problems in which the total processing time of job j must stay constant, each $\mathcal{S}$ in the collection of *permissible* bundles satisfies $|\mathcal{S}| = p_j$. For the nonpreemptive case, it is further required that each bundle comprises p_j consecutive periods; i.e., $\mathcal{S}_{jk} \equiv \{k, \ldots, k + p_j - 1\}, K_j = T - p_j + 1$. For the preemptive case, any bundle of p_j periods is permissible; therefore, the number of permissible bundles for job j is $K_j = \binom{T}{p_j}$.

In the nonpreemptive case, the bundle cost f_{jk} of job j follows from the job completion time cost $F_j(\cdot)$ as part of the problem data; i.e., $f_{jk} \equiv F_j(k + p_j)$ for bundle $\mathcal{S}_{jk} = \{k, \ldots, k + p_j - 1\}$. For the preemptive case, we define the bundle cost as

$$f_{jk} \equiv F_j(1 + \max\{t : t \in \mathcal{S}_{jk}\}), \tag{4}$$

where $F_j(\cdot)$ is from its corresponding nonpreemptive problem. This definition conforms to the conventional interpretation of job completion time cost for preemptive problems. Meanwhile, we realize that the literature on preemptive scheduling is almost exclusively dedicated to *regular* objectives. For preemptive problems with *nonregular* objectives, this definition is not adequate, which perhaps warrants further study in the future. Our subsequent discussion only deals with regular objectives.

If job processing times can be compressed or elongated, possibly at some costs, we can still formulate it as a WDP; we just need to to expand the collection of permissible bundles for a job. Machine downtime can be dealt with in the same spirit, although downtime is likely to reduce the number of permissible bundles. Similarly, release dates and deadlines can be modeled by eliminating bundles that conflict with them, or by setting the bundle costs to a sufficiently large M.

3 New IP formulation for preemptive problems

Following the above definitions of permissible bundles and costs, the WDP gives the same formulation as the time-indexed IP formulation for nonpreemptive problems [7,19,1,5,2], which has been extensively studied. However, a more interesting result is that the WDP leads to an IP formulation for preemptive problems, which to the best of our knowledge, has not been proposed before.

Even though they are relaxations of their nonpreemptive counterparts, preemptive problems can be intractable; e.g., $1|r_j, pmtn|\sum w_j C_j$, the preemptive version of the total weighted completion time problem with release dates ($1|r_j|\sum w_j C_j$) is known to be unary NP-hard. IP formulation and enumerative solution procedures are justified for such problems. To facilitate our subsequent discussion, we write down the linear programming (LP) relaxation of the IP, as well as the associated LP dual problem.

$$\text{(P)} \quad \min \sum_{j=1}^{n} \sum_{k=0}^{K_j-1} f_{jk} y_{jk}$$

$$\text{s.t.} \quad \sum_{k=0}^{K_j-1} y_{jk} = 1, \ \forall j = 1, \ldots, n, \tag{5}$$

$$\sum_{j=1}^{n} \sum_{k:\mathcal{S}_{jk} \ni t} y_{jk} \leq 1, \ \forall t = 0, \ldots, T-1, \tag{6}$$

$$y_{jt} \geq 0, \ \forall k = 0, \ldots, K_j - 1; j = 1, \ldots, n. \tag{7}$$

$$\text{(DP)} \quad \max \sum_{j=1}^{n} u_j + \sum_{t=0}^{T-1} v_t$$

$$\text{s.t.} \quad u_j + \sum_{s \in \mathcal{S}_{jk}} v_s \leq f_{jk}, \ \forall k = 0, \ldots, K_j - 1; j = 1, \ldots, n, \tag{8}$$

$$u_j \text{ free}, \ \forall j = 1, \ldots, n \text{ and } v_t \leq 0, \forall t = 0, \ldots, T-1. \tag{9}$$

Primal-dual relations of the same form have been established recently for nonpreemptive problems in [14]. Furthermore, essentially the same argument as in [14] can be used to show that (P) and (DP) are also equivalent to the following max-min transportation problem:

$$\text{(MMT)} \quad \max z(c)$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}_{jk}} c_{js} \leq f_{jk}, \ \forall k = 0, \ldots, K_j - 1; j = 1, \ldots, n, \tag{10}$$

where

$$\text{(TP)} \quad z(c) = \min \sum_{j=1}^{n} \sum_{t=0}^{T-1} c_{jt} x_{jt} \tag{11}$$

$$\text{s.t.} \quad \sum_{t=0}^{T-1} x_{jt} = p_j, \ \forall j = 1, \ldots, n, \tag{12}$$

$$\sum_{j=1}^{n} x_{jt} \leq 1, \ \forall t = 0, \ldots, T-1, \tag{13}$$

$$x_{jt} \geq 0, \ \forall j = 1, \ldots, n; t = 0, \ldots, T-1. \tag{14}$$

(MMT) is defined in c variables and (TP) is in x variables with c fixed. Consequently, we have the following result:

Theorem 1 *The strongest transportation problem relaxation satisfying (10) provides the same lower bound as (P) for the preemptive problem.*

Proof This is a consequence of the equivalence relations, whose proof is the essentially the same as that of Theorem 1 in [14]. □

Theorem 1 is a general result that holds for any preemptive problem with a regular scheduling objective. Consequently, it holds for $1|r_j|\sum w_j C_j$. Meanwhile, Dyer and Wolsey [7] and Goemans et al. [10] consider a particular transportation problem relaxation of this problem with the transportation costs defined as

$$c_{jt} \equiv \frac{w_j}{p_j}\left(t + \frac{1}{2}\right) + \frac{w_j}{2}, \ \forall t \geq r_j. \tag{15}$$

The "simple split" lower bound by Belouadah et al. [4] for $1|r_j|\sum w_j C_j$ is also equivalent to this transportation problem relaxation. The following corollary compares (P) to this transportation problem relaxation.

Corollary 1 *The transportation problem relaxation in [7,4,10] is dominated by (P) for the* $1|r_j, pmtn|\sum w_jC_j$ *problem.*

Proof Let $\mathcal{S}_{jk}$ be any bundle and let $t_{\max} = \max\{t : t \in \mathcal{S}_{jk}\}$. Suppose that c_{jt} is defined as (15). Note that $\sum_{s\in\mathcal{S}_{jk}} c_{js} \leq \sum_{s=t_{\max}-p_j+1}^{t_{\max}} c_{js} = w_j(1 + t_{\max}) = f_{jk}$. That is, such defined c_{jt} verifies (10). Hence, the result follows from Theorem 1. □

To find further evidence of the strength of (P), now let us turn to the polynomially solvable special case of $1|r_j, pmtn|\sum w_jC_j$ with unit weights; i.e., $1|r_j, pmtn|\sum C_j$. The preemptive list schedule based on the shortest-remaining-processing-time (SRPT) rule [16,18] is optimal to this problem. The theorem below indicates that (P) is tight for this special case.

Theorem 2 *The optimal value of Problem (P) is equal to the optimal value of* $1|r_j, pmtn|\sum C_j$.

Proof The result can be shown using essentially the same argument given by Smith [18] to prove the optimality of SRPT for $1|r_j, pmtn|\sum C_j$. Smith's proof instead deals with an equivalent total flow time problem (the objective functions differ by a constant). The proof goes through without any problems for our situation where multiple jobs may arrive at a time and the remaining processing times of multiple jobs may be decreased during a period. □

Although $1|r_j, pmtn|\sum C_j$ is solvable in polynomial time, the fact that Problem (P) provides a tight relaxation with zero gap is not trivial. In contrast, the transportation problem relaxation of [7,4,10] has a nonzero worst-case relaxation gap.

Theorem 3 *The transportation problem relaxation in [7,4,10] is at best a* $9/8$*-relaxation for* $1|r_j, pmtn|\sum C_j$.

Proof Consider the following example: For $j = 1$, $r_j \equiv 0$, $p_j \equiv p > 1$; for $j = 2, \ldots, n$, $r_j \equiv p-1$, $p_j \equiv q > 1$; the parameters p and q are integral with $p > q$. The transportation problem has the property that optimal solutions stay unchanged after all c_{jt} for a fixed j are increased (or decreased) by the same constant. Using this fact and an interchange argument, the solution obtained first by processing job 1 up to time $p-1$, then by preempting job 1 with jobs $2, \ldots, n$, and finally by finishing off job 1, can be shown to optimally solve the transportation problem relaxation, yielding an objective value of

$$\mathrm{TP} = \frac{n(n-1)q}{2} + n(p-1) + \frac{(n-1)q}{p} + 1.$$

Meanwhile, the optimal value of $1|r_j, pmtn|\sum C_j$ obtained using the SRPT rule is

$$\mathrm{OPT} = \frac{n(n-1)q}{2} + np.$$

By varying over all integral parameters n, p, q satisfying $n > 1, p > q > 1$, the maximal OPT/TP ratio of 9/8 is attained at $n = 3, p = 2, q = 1$. □

4 Column management

Problem (P) can have a great deal more variables than the time-indexed LP of the nonpreemptive problem, even for for relatively small processing time range. The sheer number of columns in (P) makes it highly impractical to explicitly form all the columns in the coefficient matrix, perhaps with the exception of very small T and processing time range. In this section, we discuss two methods for reducing the number of explicit columns.

4.1 Elimination of bundles using problem structure

For a regular preemptive problem, it is always advantageous to keep the machine busy whenever jobs are available for processing. To a static problem with $r_j = 0$ for all j, this implies that the time horizon can be limited to $T = \sum_{j=1}^{n} p_j$. The same idea extends to a dynamic problem in which not all jobs are available at time zero.

Without loss of generality, it is assumed that (A1) the jobs are renumbered in nondecreasing order of release dates; i.e., $i < j \Rightarrow r_i \leq r_j$. Suppose that the jobs are processed in a no-wait, first-come-first-serve manner, and let C_j be the resulting completion time of job j. We define a *block* $\Gamma = \{a, a+1, \ldots, b\}$ as a maximal subset of consecutively ordered jobs such that $r_{j+1} < C_j$ for all $j \in \Gamma, j \neq n$. Clearly, any regular preemptive problem that comprises more than one block can be decomposed into independent subproblems in the same number. Therefore, we only need to consider problems where (A2) all jobs form a single block. For such problems, it suffices to define $T \equiv r_1 + \sum_{j=1}^{n} p_j$; any bundle of job j that does not lie entirely within interval $[r_j, T]$ is superfluous. (A1) and (A2) will be assumed hereafter.

The block-based decomposition and the selection of the smallest T can be useful in solving problems where blocks consist of relatively few jobs that are well spread out by release dates. If the number of bundles remains large, we will have to resort to column generation.

4.2 Column generation for Problem (P)

Column generation is a useful, and sometimes the only viable, technique for solving LP problems with a huge number of variables. We rely on the literature for the exposition of this important technique; e.g., the surveys in [3, 13] and references therein. At times, it may be convenient to take the dual view and think in terms of constraint generation for the dual problem (DP).

Suppose that we have optimally solved the *restricted master problem* (RMP), which has the same structure as (P) except that only a small subset V of columns (which, we recall, correspond to bundles) participate in the problem. Let u_j ($j = 1, \ldots, n$) and v_t ($t \in \mathcal{T}$) denote the optimal dual values associated with the constraints of the RMP. At this point, we need to determine if there exists a column y_{jk} (not in V) with a negative reduced cost that if added to V, can potentially reduce the objective value of the RMP. If no such column exists, we have solved the original problem (P). Otherwise, a column with the most negative reduced cost is added to V, as suggested by the standard approach; this is accomplished by solve the *pricing problem.* In our case, the pricing problem can be formulated as one of finding a bundle $\mathcal{S}_{jk}$ such that the reduce cost

$$f_{jk} - u_j - \sum_{s \in \mathcal{S}_{jk}} v_s \tag{16}$$

is minimized. Note that the quantity u_j is not affected by the choice of bundles and that the pricing problem can be solved for each fixed j independently. Furthermore, due to the special structure of bundle costs (4), the pricing problem can be solved using the following algorithm:

PRICING(j)

let $\alpha_j^* = \infty$; $W_j^* = \emptyset$;
for $t = T - 1$ to $r_j + p_j - 1$ *stepsize* $= -1$,
  do let W be the set of indices of the $(p_j - 1)$ largest $v_s (r_j \leq s < t)$;
    let $\alpha = F_j(t+1) - u_j - v_t - \sum_{s \in W} v_s$;
    if $\alpha < \alpha_j^*$,
      then set $\alpha_j^* = \alpha; W_j^* = W \cup \{t\}$;
return α_j^* and W_j^*.

Let α_{j_0} denote the minimum of all returned $\alpha_j^*, j = 1, \ldots, n$. If $\alpha_{j_0} < 0$, we add to the RMP, the column that corresponds to the bundle as defined by the index set $W_{j_0}^*$. From the dual perspective, the above is a *separation algorithm* for finding a constraint among (8) that has the most violation. With a straightforward implementation, the pricing problem can be solved in $O(nT^2 \log T)$ time. However, by utilizing the heap data structure, the running time can be reduced to $O(nT \log T)$.

The optimal value of the RMP is an upper bound on the optimal value of (P), and the two bounds equal only after column generation terminates. However, the convergence of column generation may become very slow after a certain number of iterations (*tailing-off*, that is). When tailing-off occurs, we may terminate column generation early, and the simple method of [12, 20] can be exploited to get a lower bound. We omit some details of the derivation, but it can be easily verified that the following is a valid lower bound to (P).

$$\sum_{t=0}^{K_j - 1} v_t + \sum_{j=1}^{n} (\alpha_j^* + u_j), \tag{17}$$

where $\alpha_j^*, j = 1, \ldots, n$ are from the pricing problem.

We suggest another early-termination method that entails more computational effort, but hopefully, will yield stronger bounds. For the dual problem (DP), tailing-off means that the dual values (u, v) from the RMP remain infeasible for (DP). Fortunately, for (DP), a feasible solution (u^0, v^0) is easy to obtain. We therefore can determine a maximum stepsize λ $(0 \leq \lambda \leq 1)$ such that $(u^0, v^0) + \lambda \cdot (u - u^0, v - v^0)$ is feasible; the associated objective value of (DP) is a valid lower bound to (P). In fact, we have

$$\lambda^* = \min \left\{ \frac{f_{jk} - u_j^0 - \sum_{s \in \mathcal{S}_{jk}} v_s^0}{u_j - u_j^0 + \sum_{s \in \mathcal{S}_{jk}} (v_s - v_s^0)} : u_j - u_j^0 + \sum_{s \in \mathcal{S}_{jk}} (v_s - v_s^0) > 0 \right\}. \tag{18}$$

Although λ^* cannot be evaluated directly using (18), we can compute it using a fast dichotomy procedure that repeatedly calls the above separation algorithm. Assuming that (u, v), as output from column generation, is reasonably close to the dual-feasible polyhedron, the initial λ perhaps should be chosen closer to 1 than to 0.

5 Primal heuristics

Solutions to relaxations of schedule problems can help guide us in finding high-quality feasible integer solutions.

5.1 Representations of feasible solutions

Due to the enormous dimensions of the y-space, it is better suited to analyze relaxation solutions in the more concise x-space, where

$$x_{jt} \equiv \sum_{k:\mathcal{S}_{jk}\ni t} y_{jk}, \ \forall t = r_j, \ldots, T-1; j = 1, \ldots, n. \tag{19}$$

Under assumptions (A1) and (A2) (see Section 4.1), the set of all feasible integer solutions can be expressed as

$$\Phi^0 = \left\{ x : \sum_{t=r_j}^{T-1} x_{jt} = p_j, \forall j; \sum_{j=1, r_j \leq t}^{n} x_{jt} = 1, \forall t; x_{jt} \in \{0,1\}, \forall j, t \right\} \tag{20}$$

This representation of the feasible integer solution set should work nicely with the column generation for (P) in a branch-and-price algorithm, because branching decisions on x_{jt} can be naturally incorporated into the pricing problem.

It turns out that it suffices to restrict ourselves to the subset of Φ^0 that is formed by all *preemptive list schedules*. Such schedules can be represented by partial orders that define the priorities between jobs. When more than one job compete for a time period, the job with the highest priority among the competing jobs gets assigned to the period. The following proposition is a straightforward observation (folklore); see, also, Lemma 2.1 and the proof thereof in [17].

Proposition 1 *Among all the optimal integer solutions to the preemptive scheduling problem with a regular objective, there exists one solution that is a preemptive list schedule.*

Preemptive list schedules are conducive to enumerative schemes. They can be enumerated as follows: Initially, the partial order is null; i.e., $\pi = \emptyset$. Starting from period r_1 and for each period $t = r_1, \ldots, T-1$, we assign the job with the highest priority in the set of competing jobs (i.e., those with $r_j \leq t$). At a *decision point*, when either the job in service finishes or a new job arrives, a determination is required as to which job will be assigned to the next period. All determinations will be examined, provided that they are consistent with the partial order π, which will then be augmented accordingly. This process results in an enumeration tree whose leaves are not necessarily total orders, but nevertheless are sufficient to uniquely determine a preemptive list schedule.

In the remainder of this section, we describe three primal heuristics. The simple interchange heuristic routine mentioned above may be used as a post-processing measure to ensure that primal solutions obtained are always preemptive list schedules.

5.2 Rounding

Rounding is a common technique to obtain an integer solution from a fractional solution as output by LP relaxation. Given a fractional solution y^{LP}, rounding can be applied either to y variables, or to x variables by the transformation (19). We find it is less cumbersome to work with x variables, and we solve a transportation problem in the from of (TP) with transportation costs defined as $c_{jt} \equiv x_{jt}^{LP}$, where x_{jt}^{LP} is the factional solution. Due to the total unimodularity of (TP), the solution of (TP) will be integral, and by the definition of costs, will to some extent reflect our preference in assigning jobs to periods. A variant of this approach would be to add a small time-dependent weighting term to each c_{jt}; e.g., for the total weighted completion time problem, it is natural to define $c_{jt} \equiv x_{jt}^{LP} + \omega \cdot t \cdot w_j / p_j$ with say, $\omega = 0.01$.

5.3 Best-α

The notion of α-points was investigated in [8,6,9,10,17] and furthermore developed in [9,17], for the purposes of converting preemptive schedules to nonpreemptive schedules. It is relatively less known for its usage to obtain approximate solutions to preemptive problems. Indeed, it has been applied to the transportation problem solution of [7,4,10] to obtain a 1.47-approximation [9] for $1|r_j, pmtn|\sum w_j C_j$. Moreover, it has recently been applied to the preemptive list schedule obtained with Smith's ratio rule, and the resulting preemptive best-α schedule is a 4/3-approximation for the same problem [17].

In our present context, we can compute the best-α preemptive schedule from the factional solution x_{jt}^{LP}. Since x_{jt}^{LP} is computed from y_{jt}^{LP}, the LP solution of the strong formulation, we are hopeful that the resulting best-α preemptive schedule would be of high quality.

5.4 Solutions to transportation problems

As mentioned before, the particular (TP) problem in [7,4,10] is a relaxation of $1|r_j, pmtn|\sum w_j C_j$. Let x^{TP} denote the optimal solution to this (TP), then the objective value of x^{TP} as a solution to (TP) is a lower bound for $1|r_j, pmtn|\sum w_j C_j$. Meanwhile, x^{TP} is automatically integral and hence is a feasible solution to the preemptive problem (in fact, a 2-approximation [9]).

Let (u^*, v^*) be the optimal dual solution to (P). Using the primal-dual argument of [14], it can be seen that a (TP) with costs defined as $c_{jt} \equiv u_j^*/p_j + v_t^*$ for all j, t yields the same optimal value as (P). Being the strongest (TP) relaxation as it is, the question, however, remains whether the corresponding transportation problem solution is also a high quality primal solution. Our hope is that there is a correlation.

6 Discussion and conclusion

It is a classical method to obtain relaxations of scheduling problems by allowing preemption. When the relaxed problems are polynomially solvable (e.g., $1|r_j, pmtn|\sum C_j$), they provide lower bounds that can be used in branch-and-bound. But, the relaxation gap can be substantial; indeed, for $1|r_j|\sum C_j$, the ratio of the optimal nonpreemptive value to the optimal preemptive value can be made arbitrarily close to an upper bound of $e/(e-1)$ [6,10].

Even in cases when preemptive problems are NP-hard, heuristic solutions of preemptive problems may still offer valuable insight. For example, in [17] the notion of α-points is applied to the preemptive list schedule based on Smith's ratio rule (the schedule itself is a 2-approximation for $1|r_j, pmtn|\sum w_j C_j$), and gets a 4/3-approximation for $1|r_j, pmtn|\sum w_j C_j$.

The majority of the present paper is devoted to solution approaches for preemptive problems. Our main result is a new IP formulation whose LP relaxation provides strong lower bounds. Previously, the best relaxation of $1|r_j, pmtn|\sum w_j C_j$ is based on a transportation problem. The ratio of the optimal preemptive value to the lower bound from the transportation problem relaxation never exceeds 4/3, but can be arbitrarily close to 8/7 in the worst case [17]. As we have shown, our new relaxation dominates the transportation relaxation, and therefore, is automatically a 4/3-relaxation. However, this worst-case ratio is too pessimistic.

The combinatorial auctions view of scheduling problems has benefited us in deriving strong IP formulations for scheduling problems. Conversely, the knowledge that we have gained in tackling the IP formulations bears great promises in aiding the solution of the WDP in combinatorial auctions. The "=" constraints (1) that make each job win exactly one bundle (as required by our scheduling problems) could be relaxed to "$\leq$"; our WDP formulation would then be of the same form as two formulations studied in [21,11]. Furthermore, there are no fundamental changes to the WDP that would prohibit us from applying the same method.

There are some open questions that we have not fully explored in this paper, but believe of interest to the scheduling community. Computational experiments are under way to gain experience in the formulations and methods. Also, we are convinced that the proposed IP formulation for the preemptive problem, motivated by combinatorial auctions, will bring new thrust to the advancement of theoretical research in approximation algorithms. For example, it would be interesting to know the worst-case competitive ratios of the primal heuristics (in particular, best-α) discussed in Section 5.

References

1. van den Akker, J.M., van Hoesel, C.P.M., Savelsbergh, M.W.P.: A polyhedral approach to single-machine scheduling problems. Math. Program. **85**, 541–572 (1999)
2. Avella, P., Boccia, M., D'Auria, B.: Near-optimal solutions of large-scale single-machine scheduling problems. INFORMS J. Comput. **17**(2), 183–191 (2005)
3. Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W.P., Vance, P.H.: Branch-and-price: Column generation for solving huge integer programs. Oper. Res. **46**(3), 316–329 (1998)
4. Belouadah, H., Posner, M.E., Potts, C.N.: Scheduling with release dates on a single machine to minimize total weighted completion time. Discrete Appl. Math. **36**(3), 213–231 (1992)
5. Bigras, L., Gamache, M., Savard, G.: Time-indexed formulations and the total weighted tardiness problem (2005). G-2005-30, Les Cahiers du GERAD
6. Chekuri, C., Motwani, R., Natarajan, B., Stein, C.: Approximation techniques for average completion time scheduling. SIAM J. Comput. **31**, 146–166 (2001)
7. Dyer, M.E., Wolsey, L.A.: Formulating the single machine sequencing problem with release dates as a mixed integer program. Discrete Appl. Math. **26**, 255–270 (1990)
8. Goemans, M.X.: Improved approximation algorithms for scheduling with release dates. In: Proc. 8th ACM-SIAM Symposium on Discrete Algorithms, vol. 25, pp. 591–598. New Orleans, Louisiana (1997)
9. Goemans, M.X., M., W.J., Williamson, D.P.: A 1.47-approximation algorithm for a preemptive single-machine scheduling problem. Oper. Res. Lett. **26**(4), 149–154 (2000)
10. Goemans, M.X., Queyranne, M., Schulz, A.S., Skutella, M., Wang, Y.: Single machine scheduling with release dates. SIAM J. Discrete Math. **15**(2), 165–192 (2002)
11. Günlük, O., Ladányi, L.: A branch-and-price algorithm and new test problems for spectrum auctions. Management Sci. **51**(3), 391–406 (2005)
12. Lasdon, L.S.: Optimization Theory for Large Systems. Macmillan, New York, NY (1970)
13. Lübbecke, M.E., Desrosiers, J.: Selected topics in column generation. Oper. Res. **53**(6), 1007–1023 (2005)
14. Pan, Y., Shi, L.: On the equivalence of the max-min transportation lower bound and the time-indexed lower bound for single-machine scheduling problems. Math. Program., Ser. A (2006). Forthcoming. Available online: `http://www.optimization-online.org/DB_FILE/2005/03/1100.pdf`
15. Sandholm, T., Suri, S., Gilpin, A., Levine, D.: CABOB: A fast optimal algorithm for winner determination in combinatorial auctions. Management Sci. **51**(3), 374–390 (2005)
16. Schrage, L.: A proof of the optimality of the shortest remaining processing time discipline. Oper. Res. **16**(3), 687–690 (1968)
17. Schulz, A.S., Skutella, M.: The power of α-points in preemptive single machine scheduling. J Sched. **5**, 121–133 (2002)
18. Smith, D.R.: A new proof of the optimality of the shortest remaining processing time discipline. Oper. Res. **26**(1), 197–199 (1978)
19. Sousa, J.P., Wolsey, L.A.: A time indexed formulation of non-preemptive single machine scheduling problems. Math. Program. **54**, 353–367 (1992)
20. Vanderbeck, F., Wolsey, L.A.: An exact algorithm for IP column generation. Oper. Res. Lett. **19**(4), 151–159 (1996)
21. de Vries, S., Vohra, R.V.: Combinatorial auctions: A survey. INFORMS J. Comput. **15**(3), 284–309 (2003)