37

A Generic Primal-Dual Approximation Algorithm for an Interval Packing and Stabbing Problem

Sofia Kovaleva
University of Maastricht

Frits C.R. Spieksma
Katholieke Universiteit Leuven

37.1 Introduction

Packing and stabbing (or covering) problems are two basic problems in combinatorial optimization. Apart from the fact that they arise in many practical applications, investigating these problems improves our understanding of fundamental issues in combinatorial optimization. Admittedly, the generality of a packing or a stabbing problem has its price: When it comes to solving such a problem, and when one insists that an optimal solution is produced for all instances, one needs to accept that, for some instances, large running times are unavoidable. Moreover, when one restricts oneself to polynomial-time algorithms, only very weak statements concerning the quality of the solutions found can be made (see, e.g., Chapter 1, Ausiello et al. [1] or Vazirani [2] for an introduction and terminology).

Here, we focus on a special case of these two problems. On the one hand, by studying a (geometric) special case, we enter the world of polynomial-time algorithms that admit a constant performance guarantee; on the other, this special case is still general enough to admit a variety of applications.

As an appetizer, consider the following two questions dealing with intervals on the line. Imagine that n intervals $(l_i, r_i]$, $i = 1, \ldots, n$, etc. on the line are given; further, we say that two intervals are *disjoint* when their intersection is empty, interval i *contains* point p when $l_i < p \leq r_i$, and interval i is *stabbed* by point p when interval i contains p.

Question 1. Select as many pairwise disjoint intervals as possible.

Question 2. Stab every interval at least once using as few points as possible.

These two questions are, in fact, easy to answer. Indeed, it is well known that the following rule does the job. After sorting the intervals in nondecreasing order with respect to the r_i's, repeat the following instructions iteratively: (i) select the remaining interval with the smallest r_i, (ii) discard all intervals containing r_i, and (iii) add point r_i to the set of points. Essentially, this rule solves an independent set problem, and a clique cover problem, in an interval graph.

This chapter deals with generalizations of this setting. We consider the case when the set of intervals is partitioned into, say m, subsets such that at most one interval of a subset can be selected (while still requiring disjointness). Such a subset of intervals is sometimes referred to as a *job*. Even more generally, we consider the problems that arise when a given number of intervals from a subset can be selected, and when the disjointness restriction is replaced by allowing a given number of intervals to intersect each other. Also, weights for the intervals are considered.

In the sequel we formulate precisely the problems we consider, describe previous research, and explain the setup of this chapter. Our main focus is on the description of a generic primal-dual algorithm for a weighted set packing problem (WSP), and how this method works for our particular geometric setting. This chapter is based on a part of the Ph.D. thesis of Kovaleva [3].

Preliminaries

We use an underlying grid to formulate our problems. Given is a grid consisting of t columns, numbered consecutively from left to right, and m numbered rows. Furthermore, given is a set of intervals $I = \{1, 2, \ldots, n\}$ lying on the rows of the grid. An interval $i \in I$ is specified by the triple l_i, r_i, ρ_i, where l_i, r_i are the indices of the left- and the rightmost columns intersecting interval i and ρ_i is the index of the row where interval i lies. For each column c, row r, and interval i we are given positive integral parameters v_c, u_r, and p_i, respectively, referred to as the column, row, and interval capacities. For each interval i we are given a positive integral parameter w_i referred to as the interval demand. We assume that the intervals are ordered according to nondecreasing r_i. We refer to the family of inputs that arise in this way as JI. Thus, AQ1 an instance $\mathcal{I}$ of JI can be seen as a collection of numbers t, m, for each $i \in I : l_i, r_i, \rho_i, p_i, w_i$, for each $c = 1, \ldots, t : v_c$, and for each $r = 1, \ldots, m : u_r$.

We use the following terminology in the sequel: Column c is said to *stab* interval i if column c intersects interval i, that is, $l_i \leq c \leq r_i$. Also, row ρ_i is said to *stab* interval i if interval i lies on row ρ_i.

For each instance $\mathcal{I}$ of JI we formulate the following two problems:

Job Interval Packing Problem (JIP)

For each interval $i \in I$ specify an integral multiplicity x_i such that

- *it does not exceed the interval capacity p_i,*
- *for each column c the sum of multiplicities of the intervals stabbed by column c does not exceed the column capacity v_c,*
- *for each row r the sum of multiplicities of the intervals stabbed by row r does not exceed the row capacity u_r,*
- *the total demand $\sum w_i x_i$ is maximized.*

Job Interval Stabbing Problem (JIS)

For each column c, row r, and interval i specify integral multiplicities y_c, z_r, and s_i respectively, such that

- *for each interval i the sum of the multiplicities of the columns stabbing interval i plus the multiplicity of the row stabbing interval i plus the multiplicity of interval i equals at least its demand w_i,*
- *the total capacity $\sum_{c=1}^{t} v_c y_c + \sum_{r=1}^{m} u_r z_r + \sum_{i=1}^{n} p_i s_i$ is minimized.*

One may interpret the interval demands w_i as interval weights in JIP and the column, row and interval capacities v_c, u_r, p_i as column, row, and interval weights in JIS. However, in this chapter we refer to these parameters as demands and capacities to keep the terminology uniform for both problems.

In this chapter we describe a generic primal-dual algorithm and show how it can be applied to two special cases of JIP and JIS. The first special case deals with instances where all the column, row, and interval capacities are unit, that is, $v_c = u_r = p_i = 1, \ \forall c, r, i$. We refer to the family of inputs satisfying this condition as *JI with unit capacities*. The second special case assumes that all the interval demands are unit, that is, $w_i = 1, \ \forall i = 1, \ldots, n$. The family of these inputs is referred to as *JI with unit demands*.

AQ2 Below we give natural ILP formulations of JIP and JIS:

$$\text{JIP:}\quad \text{Maximize} \quad \sum_{i=1}^{n} w_i x_i \tag{37.1}$$

$$\text{subject to} \quad \sum_{i:\rho_i=r} x_i \leq u_r \quad \forall r = 1, \ldots, m \tag{37.2}$$

$$\sum_{i:c\in[l_i,r_i]} x_i \leq v_c \quad \forall c = 1, \ldots, t \tag{37.3}$$

$$x_i \leq p_i \quad \forall i = 1, \ldots, n \tag{37.4}$$

$$x_i \in \mathbb{Z}_+^1 \quad \forall i = 1, \ldots, n \tag{37.5}$$

$$\text{JIS:}\quad \text{Minimize} \quad \sum_{c=1}^{t} v_c y_c + \sum_{r=1}^{m} u_r z_r + \sum_{i=1}^{n} p_i s_i \tag{37.6}$$

$$\text{subject to} \quad z_{\rho_i} + \sum_{c\in[l_i,r_i]} y_c + s_i \geq w_i \quad \forall i = 1, \ldots, n \tag{37.7}$$

$$z_r, y_c, s_i \in \mathbb{Z}_+^1 \quad \forall r, c, i \tag{37.8}$$

Observe that for any instance $\mathcal{I}$ of JI the LP relaxations of these ILP formulations (which arise when we substitute the integrality constraints [37.5] and [37.8] by nonnegativity constraints $x_i \geq 0, \ \forall i = 1, \ldots, n$ and $z_r, y_c, s_i \geq 0, \forall r, c, i$, respectively) constitute a primal-dual pair of LP problems. It follows then from the strong duality theorem for linear programming that for any instance $\mathcal{I}$, the LP relaxations of JIP and JIS have the same optimum value, which we denote by $LP(\mathcal{I})$, and thus the following holds:

$$JIP(\mathcal{I}) \leq LP(\mathcal{I}) \leq JIS(\mathcal{I}) \tag{37.9}$$

where $JIP(\mathcal{I})$ and $JIS(\mathcal{I})$ are the optimum values of JIP and JIS for $\mathcal{I}$, respectively.

We say that problems JIP and JIS are weakly dually related and establish the following result:

Lemma 37.1 (Weak duality lemma for JIP and JIS)
For any instance $\mathcal{I}$ of JI, for any feasible solution to JIP with value $JIP^{feas}(\mathcal{I})$ and any feasible solution to JIS with value $JIS^{feas}(\mathcal{I})$ holds:

$$JIP^{feas}(\mathcal{I}) \leq JIS^{feas}(\mathcal{I}).$$

Proof
Follows from (37.9). □

Previous Research
JIP and its special cases have received a great deal of attention in the literature. Its applications, mentioned by different authors, include printed circuit board assembly, combinatorial auctions, satellite photography, computational biology, throughput scheduling [4–7]. JIP is MAX SNP-hard even if all the parameters are unit ([7]; this implies that there is no polynomial-time approximation scheme for JIP unless $\mathcal{P} = \mathcal{NP}$). A greedy 1/2-approximation algorithm for JIP with unit capacities and demands (i.e., $u_r = v_c = p_i = w_i = 1, \ \forall c, r, i$) is described in Spieksma [7]. A better approximation guarantee for this case has been found by Chuzhoy et al. [8]. They present an algorithm with performance guarantee $(e-1)/e - \varepsilon \approx 0.63 - \varepsilon$, for any $\varepsilon > 0$, exploiting a sophisticated technique involving randomized LP-rounding. Bar-Noy et al. [6] describe a $(1/2 - \varepsilon)$-approximation algorithm for JIP with unit capacities (i.e., $u_r = v_c = p_i = 1, \ \forall c, r, i$) using an LP-rounding technique. One can easily generalize this algorithm to JIP (with arbitrary capacities and demands) and, using the ideas introduced by Calinescu et al. [9], improve the approximation factor

to 1/2. The resulting 1/2-approximation algorithm (which involves solving an LP problem) has to our knowledge so far the highest approximation factor for JIP. Combinatorial 1/2-approximation algorithms for JIP with unit capacities are described by Berman and DasGupta [4] and Bar-Noy et al. [10]. In Ref. [5] the performance of greedy algorithms for JIP with constant column capacity and unit row capacity (i.e., $v_c = v, \ \forall c, \ u_r = 1, \ \forall r$) is investigated using competitive analysis.

So far JIS has not been as extensively studied as JIP. Applications of JIS that have been mentioned in the literature include military and medical applications [11]. JIS is also MAX SNP-hard even if all the parameters are unit [12]. Recently, Kovaleva and Spieksma [13] describe a $\frac{e}{e-1}$-approximation algorithm for JIS with unit demand. This algorithm is based on solving the linear programming relaxation of constraints (37.6)–(37.8), and rounding this solution to an integral one. Earlier work is described in Gaur et al. [14] who consider the so-called *rectangle stabbing* problem, which generalizes JIS with unit demands to the case where rectangles intersecting several rows of the grid are given instead of intervals; their work implies a 2-approximation for JIS with unit demand. Hassin and Megiddo [11] describe a combinatorial $2 - \frac{1}{K+1}$-approximation algorithm for the case of JIS with unit capacities and demands, where each interval is intersected by exactly K columns, and a 2-approximation for a slightly more general case, when each interval is intersected by the same number of columns.

Outline of the Chapter

Section 37.2 is devoted to JI with unit capacities. We first describe a generic primal-dual algorithm, called Local Covering, for the WSP (Section 37.2.1) and then specify a setup of Local Covering for JIP with unit capacities, yielding algorithm ALG1 (Section 37.2.2). We also give conditions which guarantee that a particular setup of Local Covering yields a constant factor approximation algorithm. This algorithm is in fact a primal-dual interpretation of the *Opportunity Cost* algorithm described by Akcoglu et al. in [15], which is in turn based on the local-ratio technique, introduced by Bar-Yehuda and Even [16] and later elaborated upon by Bar-Noy et al. [10].

A distinctive feature of the primal-dual algorithm ALG1 described in Section 37.2.2 is that it simultaneously delivers two feasible solutions, one for JIP and one for JIS with unit capacities. Moreover, we show that their values are within a factor of 2 of each other. The weak duality relation between JIP and JIS implies then that these solutions are respectively a 1/2-approximation for JIP with unit capacities and a 2-approximation for JIS with unit capacities. This analysis is tight, that is, the approximation guarantees cannot be improved by carrying out a better analysis of the algorithm.

Note that, when viewed as a 1/2-approximation algorithm for JIP, ALG1 behaves similar to the algorithms described previously by Berman and DasGupta [4] and Bar-Noy et al. [10].

In Section 37.3 we consider JI with unit demands. It is easy to see that JIS with unit demands is a special case of the the weighted hitting set problem. Following the framework of the generic *primal-dual algorithm with reverse delete step* described in Ref. [17] for the weighted hitting set problem, we develop a primal-dual approximation algorithm ALG2. Similar to ALG1 it returns two feasible solutions, one for JIS and the other for JIP with unit demands. Again, we show that their values are within a factor of 2 of each other. This makes ALG2 a 1/2-approximation algorithm for JIP and a 2-approximation algorithm to JIS with unit demands.

Note that the algorithms described here are combinatorial and do not require solving a linear program.

37.2 The Case of Unit Capacities

37.2.1 The Local Covering Algorithm for the Weighted Set Packing Problem

Consider the WSP: *given is a collection $\mathcal{S}$ of subsets of a finite ground set $\mathcal{E}$. For each subset $s \in \mathcal{S}$ a nonnegative weight w_s is given. Find a maximum-weight collection A of disjoint subsets from $\mathcal{S}$.*

Below we give a natural ILP formulation of WSP, where we use the characteristic vector of A, $\chi \in \{0, 1\}^{|\mathcal{S}|}$, as a vector of decision variables:

$$\text{WSP:} \quad \text{Maximize} \quad \sum_{s \in \mathcal{S}} w_s \chi_s \tag{37.10}$$

$$\text{subject to} \quad \sum_{s: e \in s} \chi_s \le 1 \quad \forall e \in \mathcal{E} \tag{37.11}$$

$$\chi_s \in \{0, 1\} \quad \forall s \in \mathcal{S} \tag{37.12}$$

By substituting nonnegativity constraints $\chi_s \ge 0 \ \forall s \in \mathcal{S}$ for integrality constraints (37.12), we obtain the LP relaxation of this formulation.

Let us also introduce the dual linear problem to this LP relaxation (here we use $\gamma \in \mathbb{R}_+^{|\mathcal{E}|}$ for a vector of dual variables):

$$\text{Minimize} \quad \sum_{e \in \mathcal{E}} \gamma_e \tag{37.13}$$

$$\text{subject to} \quad \sum_{e \in s} \gamma_e \ge w_s \quad \forall s \in \mathcal{S} \tag{37.14}$$

$$\gamma_e \ge 0 \quad \forall e \in \mathcal{E} \tag{37.15}$$

For ease of discussion let us introduce some terminology. We say that

- an element $e \in \mathcal{E}$ and a subset $s \in \mathcal{S}$ are *incident* to each other if $e \in s$;
- the *coverage* of a subset $s \in \mathcal{S}$ by a vector $\gamma \in \mathbb{R}_+^{|\mathcal{E}|}$ is a value equal to the sum of the elements of γ corresponding to the elements e belonging to s, that is, $\sum_{e \in s} \gamma_e$;
- a subset $s \in \mathcal{S}$ is *covered* if its coverage equals at least w_s. Otherwise, we say that s is *violated*;
- a subset s_1 is a *neighbor* of a subset s_2 if they share a common element;
- $N(s)$ is the set of all the neighbors of s. Note that $s \in N(s)$;
- a feasible solution χ to the ILP formulation (37.10)–(37.12) as well as the corresponding set $A \in \mathcal{S}$ is a *feasible packing*;
- a feasible solution γ to the LP formulation (37.13)–(37.15) is a *feasible covering*.

Definition 37.1

For a given vector $\gamma \in \mathbb{R}_+^{|\mathcal{E}|}$ *and a subset* s^* *let us call a vector* $\delta \in \mathbb{R}_+^{|\mathcal{E}|}$ *satisfying:* $\sum_{e \in s} \delta_e \ge \min(w_s - \sum_{e \in s} \gamma_e, w_{s^*} - \sum_{e \in s^*} \gamma_e)$, $\forall s \in N(s^*)$, *a local covering for* γ *in* s^*.

Let us now describe a generic primal-dual algorithm local covering for WSP. The framework of the algorithm is the following: Initially, all the dual variables γ_e are zero and A is empty. Until γ becomes a feasible covering do

- select (some) subset s, violated by the current γ, and push it on a stack,
- construct a local covering δ for γ in s,
- increment vector γ by the the values of vector δ.

When γ becomes a feasible covering, pop the subsets from the stack iteratively and each time add a subset to A if this does not violate the feasibility of A.

Figure 37.1 gives a formal description of local covering (notice that $\delta \in \mathbb{R}_+^{|\mathcal{E}|}$, $\Delta \in \mathbb{R}_+$, $l \in \mathbb{N}$).

Observe that it is not exactly specified in the algorithm how s^l and δ are selected. The description of these selection procedures is left to a particular setup. It does not make sense to discuss implementation and efficiency of the algorithm in such a general setting. Let us only establish the following result:

1. $\gamma \leftarrow \mathbf{0}$
2. $A \leftarrow \emptyset$
3. $l \leftarrow 0$
4. Until γ is a feasible covering
    $l \leftarrow l + 1$
    Select s^l such that $\sum_{e \in s^l} \gamma_e < w_{s^l}$
    $\Delta \leftarrow w_{s^l} - \sum_{e \in s^l} \gamma_e$
    Find $\delta \in \mathbb{R}_+^{|\mathcal{E}|}$ s. t.: $\sum_{e \in s} \delta_e \geq \min(w_s - \sum_{e \in s} \gamma_e, \Delta)$, $\forall s \in N(s^l)$
    $\gamma \leftarrow \gamma + \delta$
5. For l downto 1
    if $A \cup \{s^l\}$ is feasible then $A \leftarrow A \cup \{s^l\}$
6. Return A (and γ)

FIGURE 37.1 Algorithm Local Covering.

Theorem 37.1

For any instance $\mathcal{I}$ of WSP, the set A and the vector γ returned by local covering are a feasible packing and a feasible covering respectively. Moreover, if Δ and δ found by the algorithm at each iteration satisfy:

$$\sum_{e \in \mathcal{E}} \delta_e \leq \beta \cdot \Delta$$

then

$$\sum_{e \in \mathcal{E}} \gamma_e \leq \beta \sum_{s \in A} w_s$$

Proof

The feasibility of A and γ is obvious. Let us establish the relation between their values. Let p be the number of iterations made by the algorithm at step 4. Observe that p is at most $|\mathcal{S}|$ since at each iteration the number of violated subsets decreases by at least 1 and the iterations stop when there is no more violated subset. Let Δ^l, δ^l and γ^l ($l = 1, \ldots, p$) be the values of Δ, vector δ, and vector γ respectively at the end of the lth iteration of step 4. Let γ^0 be a zero vector. So we have $\gamma^l = \gamma^{l-1} + \delta^l$, $\forall l = 1, \ldots, p$. The condition of the theorem can be written as

$$\sum_{e \in \mathcal{E}} \delta_e^l \leq \beta \cdot \Delta^l \quad \forall l = 1, \ldots, p$$

Further, let A^q be the state of set A at the end of the loop of the cycle at step 5, corresponding to $l = q$. Let A^{p+1} be $\emptyset$. Then we have $\emptyset = A^{p+1} \subseteq A^p \subseteq \cdots \subseteq A^1$.

We show that for each $l = 1, \ldots, p+1$

$$\sum_{e \in \mathcal{E}} \left(\gamma_e^p - \gamma_e^{l-1}\right) \leq \beta \sum_{s \in A^l} \left(w_s - \sum_{e \in s} \gamma_e^{l-1}\right) \tag{37.16}$$

Then for $l = 1$, since γ^0 is a zero vector and A^1 is the set A returned by the algorithm, we obtain the result of the theorem.

We use induction on $l = p+1, \ldots, 1$. The basis of induction is $l = p+1$. Then inequality (37.16) trivially holds, since $A^{p+1} = \emptyset$. Suppose the inequality is proven for $l = j+1$. Let us prove it for $l = j$. So, we have

$$\sum_{e \in \mathcal{E}} \left(\gamma_e^p - \gamma_e^j\right) \leq \beta \sum_{s \in A^{j+1}} \left(w_s - \sum_{e \in s} \gamma_e^j\right) \tag{37.17}$$

and the theorem will be established if we show that

$$\sum_{e\in\mathcal{E}} \left(\gamma_e^p - \gamma_e^{j-1}\right) \le \beta \sum_{s\in A^j} \left(w_s - \sum_{e\in s} \gamma_e^{j-1}\right) \tag{37.18}$$

First, let us show that the following inequality holds:

$$\sum_{s\in A^j} \left(w_s - \sum_{e\in s} \gamma_e^{j-1}\right) \ge \sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j}\right) + \Delta^j \tag{37.19}$$

There are two cases possible: either $A^j = A^{j+1}$ or $A^j = A^{j+1} \cup \{s^j\}$.

Consider the first case. Suppose $A^j = A^{j+1}$. Clearly, this case is only possible if A^{j+1} contains a neighbor of subset s^j, say subset s^q, that is, $s^q \in N(s^j)$. Consider γ^j, it is equal to $\gamma^{j-1} + \delta^j$, where δ^j satisfies

$$\sum_{e\in s} \delta_e^j \ge \min\left(w_s - \sum_{e\in s} \gamma_e^{j-1}, \Delta^j\right), \quad \text{for all } s \in N(s^j)$$

and in particular for s^q, that is, $\sum_{e\in s^q} \delta_e^j \ge \min\left(w_{s^q} - \sum_{e\in s^q} \gamma_e^{j-1},\ \Delta^j\right)$.

Assume

$$\sum_{e\in s^q} \delta_e^j \ge \min\left(w_{s^q} - \sum_{e\in s^q} \gamma_e^{j-1}, \Delta^j\right) = w_{s^q} - \sum_{e\in s^q} \gamma_e^{j-1}$$

Then

$$\sum_{e\in s^q} \gamma_e^j = \sum_{e\in s^q} \gamma_e^{j-1} + \sum_{e\in s^q} \delta_e^j \ge w_{s^q}$$

This means that subset s^q is covered by γ^j and therefore also by $\gamma^{j+1}, \ldots, \gamma^p$, since $\gamma^j \le \gamma^{j+1} \le \cdots \le \gamma^p$. Therefore, subset s^q is not violated at iteration j or later and cannot be selected and pushed on the stack during iterations $j+1, \ldots, p$ of step 4, which contradicts with $s^q \in A^{j+1}$. Therefore, the only possible case is that

$$\sum_{e\in s^q} \delta_e^j \ge \min\left(w_{s^q} - \sum_{e\in s^q} \gamma_e^{j-1}, \Delta^j\right) = \Delta^j$$

Now, using the assumption $A^j = A^{j+1}$, the facts that $\gamma^{j-1} = \gamma^j - \delta^j$ and that there exists $s^q \in A^{j+1}$ such that the above inequality holds, we can rewrite the left-hand side of inequality (37.19) as

$$\begin{aligned}
\sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j-1}\right) &= \sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j} + \sum_{e\in s} \delta_e^j\right) \\
&= \sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j}\right) + \sum_{s\in A^{j+1}} \sum_{e\in s} \delta_e^j \ge \sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j}\right) + \Delta^j
\end{aligned}$$

which proves inequality (37.19) in the case $A^j = A^{j+1}$.

Consider now the case $A^j = A^{j+1} \cup \{s^j\}$.

Using the fact that $\Delta^j = w_{s^j} - \sum_{e\in s^j} \gamma_e^{j-1}$ and $\gamma^{j-1} \le \gamma^j$, rewrite the left-hand side of inequality (37.19) as

$$\sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j-1}\right) + \left(w_{s^j} - \sum_{e\in s^j} \gamma_e^{j-1}\right) \ge \sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j}\right) + \Delta^j$$

which proves inequality (37.19) in the case $A^j = A^{j+1} \cup \{s^j\}$.

Now, with inequality (37.19) established, we rewrite it by multiplying both sides by β:

$$\beta \sum_{s\in A^j} \left(w_s - \sum_{e\in s} \gamma_e^{j-1} \right) \geq \beta \sum_{s\in A^{j+1}} \left(w_s - \sum_{e\in s} \gamma_e^{j} \right) + \beta \Delta^j$$

Next, using the condition of the theorem, that is, $\sum_{e\in\mathcal{E}} \delta_e^j \leq \beta\Delta^j$, the induction hypothesis (37.17) and the fact $\gamma^j - \delta^j = \gamma^{j-1}$, we can bound the last expression from below by

$$\sum_{e\in\mathcal{E}} (\gamma_e^p - \gamma_e^j) + \sum_{e\in\mathcal{E}} \delta_e^j = \sum_{e\in\mathcal{E}} (\gamma_e^p - \gamma_e^{j-1})$$

Thus, we establish inequality (37.18), which proves the theorem. □

The weak duality theorem for linear programming implies $OPT(\mathcal{I}) \leq \sum_{e\in\mathcal{E}} \gamma_e(\mathcal{I})$, where $OPT(\mathcal{I})$ is the optimum value of WSP for instance $\mathcal{I}$ and $\gamma(\mathcal{I})$ is feasible covering γ returned by local covering for $\mathcal{I}$. Let $A(\mathcal{I})$ be the set A returned by the algorithm for $\mathcal{I}$, then from Theorem 37.1 we have

$$OPT(\mathcal{I}) \leq \sum_{e\in\mathcal{E}} \gamma_e(\mathcal{I}) \leq \beta \sum_{s\in A(\mathcal{I})} w_s$$

Corollary 37.1

If (a particular setup of) the local covering algorithm runs in polynomial time and the condition of Theorem 37.1 is satisfied, then the local covering is a $\frac{1}{\beta}$-approximation algorithm.

37.2.2 Algorithm ALG1

Suppose we are given an instance $\mathcal{I}$ of JI with unit capacities. Recall that this is a grid consisting of t columns, numbered consecutively from left to right, and m numbered rows together with a set of intervals $I = \{1, 2, \ldots, n\}$ lying on the rows of the grid. An interval $i \in I$ is specified by the triple (l_i, r_i, ρ_i), where l_i, r_i are the indices of the left- and the rightmost columns intersecting the interval and ρ_i is the index of the row where it lies. For each interval $i \in I$ we are given a positive integral parameter w_i referred to as the interval demand. We assume that the intervals are ordered according to nondecreasing r_i.

Recall that the job interval packing problem (JIP) with unit capacities, can be formulated as follows: *select a maximum-weight subset of intervals A such that no two intervals share a column or row.* Observe that that this problem is a special case of the WSP, considered in the previous section. Indeed, let the ground set $\mathcal{E}$ be the set of all the columns and rows of the grid and the collection $\mathcal{S}$ be $\{s_1, \ldots, s_n\}$, where s_i is the subset of columns and the row stabbing interval i, that is, $s_i = \{column\ l_i, \ldots, column\ r_i, row\ \rho_i\}$. The weights of the subsets are equal to the corresponding interval weights: $w_{s_i} = w_i, \forall i \in I$. It is easy to see that any feasible packing corresponds to a feasible solution to JIP with unit capacities of the same value.

Below we give an ILP formulation of JIP with unit capacities, where we use a characteristic vector x of A as a vector of decision variables:

$$\text{JIP:} \quad \text{Maximize} \quad \sum_{i=1}^{n} w_i x_i \tag{37.20}$$

$$\text{subject to} \quad \sum_{i:\rho_i = r} x_i \leq 1 \quad \forall r = 1, \ldots, m \tag{37.21}$$

$$\sum_{i:c\in[l_i, r_i]} x_i \leq 1 \quad \forall c = 1, \ldots, t \tag{37.22}$$

$$x_i \in \mathbb{Z}_+^1 \quad \forall i = 1, \ldots, n \tag{37.23}$$

The dual to its LP relaxation (the dual variables $z \in \mathbb{R}^m$ and $y \in \mathbb{R}^t$ correspond to the constraints [37.21] and [37.22] respectively) looks as follows:

$$\text{Dual:} \quad \text{Minimize} \quad \sum_{c=1}^{t} y_c + \sum_{r=1}^{m} z_r \tag{37.24}$$

$$\text{subject to} \quad z_{\rho_i} + \sum_{c \in [l_i, r_i]} y_c \geq w_i \quad \forall i = 1, \ldots, n \tag{37.25}$$

$$z_r, \, y_c \geq 0 \qquad \forall r, \, c \tag{37.26}$$

Note that by replacing in this formulation the nonnegativity constraints (37.26) with integrality constraints

$$z_r, \, y_c \in \mathbb{Z}_+, \; \forall r, \, c, \tag{37.27}$$

we obtain an ILP formulation of the job interval stabbing JIS problem with unit capacities: *For each column c and row r specify integral multiplicities y_c and z_r, respectively such that*

- *for each interval i the sum of multiplicities of the columns and the row stabbing interval i is at least the demand w_i,*
- *the sum of the multiplicities is minimum.*

Note that in the case of JIS with unit capacities the interval multiplicities can be fixed to zero without loss in the optimum value, since, given any feasible solution, one can obtain a feasible solution of the same value by decreasing interval multiplicities to zero and increasing row multiplicities so as to preserve the feasibility.

We now describe a setup of the generic algorithm Local Covering for JIP with unit capacities, yielding a primal-dual approximation algorithm called ALG1.

Let us reproduce step 4 of local covering (see Figure 37.2) and translate it into the terms of JIP with unit capacities.

Line (1), that is, selecting a violated subset $s^l = s_i$ corresponds in our context to selecting interval i such that $\sum_{c \in [l_i, r_i]} y_c + z_{\rho_i} < w_i$. When more then one index i satisfies this condition we select the smallest of them.

Line (2) should be translated as $\Delta \leftarrow w_i - \sum_{c \in [l_i, r_i]} y_c - z_{\rho_i}$.

In line (3) we have to find $\delta = (\delta_{col\,1}, \ldots, \delta_{col\,t}, \; \delta_{row\,1}, \ldots, \delta_{row\,m})$ s.t.

$$\sum_{e \in s_j} \delta_e \geq \min\left(w_j - \sum_{c \in [l_j, r_j]} y_c - z_{\rho_j}, \; \Delta \right), \quad \forall j \text{ s.t. } s_j \in N(s_i) \tag{37.28}$$

where i is the number selected in line (1). We assign the value of Δ to the elements of δ corresponding to the right-most column stabbing interval i and to its row, and 0 to all the other elements.

```
Until γ is a feasible covering
   l ← l + 1
(1) Select s^l such that Σ_{e∈s^l} γ_e < w_{s^l}
(2) Δ ← w_{s^l} − Σ_{e∈s^l} γ_e
(3) Find δ ∈ R_+^{|E|} s. t.: Σ_{e∈s} δ_e ≥ min(w_s − Σ_{e∈s} γ_e, Δ), ∀s ∈ N(s^l)
(4) γ ← γ + δ
```

FIGURE 37.2 Step 4 of Local Covering.

1. $y \leftarrow \mathbf{0}$, $z \leftarrow \mathbf{0}$
2. $A \leftarrow \emptyset$
3. $l \leftarrow 0$
4. For $i \leftarrow 1$ to n
   $\Delta \leftarrow w_i - \sum_{c\in[l_i,\, r_i]} y_c - z_{\rho_i}$,
   if $\Delta > 0$ then:
   $l \leftarrow l+1$, $y_{r_i} \leftarrow y_{r_i} + \Delta$, $z_{\rho_i} \leftarrow z_{\rho_i} + \Delta$, $i_l \leftarrow i$
5. For l downto 1
   if $A \cup \{i_l\}$ is feasible to *JIP* then $A \leftarrow A \cup \{i_l\}$
6. Return A (and y, z)

FIGURE 37.3 Algorithm ALG1.

Lemma 37.2

If vectors z and y and index i are such that $w_j - \sum_{c\in[l_j,r_j]} y_c - z_{\rho_j} \leq 0$ *for all* $j = 1, \ldots, i-1$, *then vector* δ *defined as*

$$\delta_e = \begin{cases} \Delta, & \text{if } e = col\, r_i \\ \Delta, & \text{if } e = row\, \rho_i \\ 0, & \text{otherwise} \end{cases}$$

satisfies (37.28) for any $\Delta > 0$.

Proof

The condition of the lemma guarantees that Eq. (37.28) is satisfied for all $s_j \in N(s_i)$ such that $j < i$.

Consider the other neighbors of s_i, that is, all $s_j \in N(s_i)$ such that $j \geq i$. These subsets correspond to the intervals that either lie on the same row or share a column with interval i and whose right-most stabbing column has index at least equal to r_i. Then sharing a column with interval i implies sharing the column r_i. This means that each subset $s_j \in N(s_i)$, $j \geq i$, includes either row ρ_i or column r_i. Thus,

$$\forall s_j \in N(s_i),\ \text{s.t. } j \geq i: \sum_{e\in s_j} \delta_e \geq \min(\delta_{col\ r_i},\ \delta_{row\ \rho_i}) = \Delta$$

This implies the lemma. □

Figure 37.3 shows the formal description the algorithm ALG1 (see also Ref. [12]). Here A is a set of interval indices.

Theorem 37.2

For any instance $\mathcal{I}$ *of JI with unit capacities, the set A and the vectors* (y, z) *returned by the algorithm ALG1 describe feasible solutions to JIP and JIS respectively, and their values (denoted by* $val(y, z)$ *and* $val(A)$*, respectively) are related as*

$$val(y, z) \leq 2 \cdot val(A)$$

Proof

Obviously A is a feasible solution to JIP with unit capacities. Consider (y, z). According to Theorem 37.1 it is a feasible covering, that is, it is a feasible solution to the LP formulation (37.24)–(37.26). Obviously the vectors y and z are integral since w_i is integral for all $i \in I$. Thus (y, z) is a feasible solution to the ILP formulation (37.24),(37.25),(37.27).

Let us establish the ratio of 2 between the values of the solutions. Observe that the conditions of Theorem 37.1 are satisfied with $\beta = 2$. Indeed, at each iteration of step 4 of the algorithm $\sum_{e\in\mathcal{E}} \delta_e = 2\Delta$. Theorem 37.1 implies that

$$\sum_{r=1}^{m} z_r + \sum_{c=1}^{t} y_c \le 2 \cdot \sum_{i\in A} w_i \qquad \square$$

Algorithm ALG1 can be implemented to run in $O(n \log n)$ time [3].

From the weak duality relation between JIP and JIS, and from Theorem 37.2, we have that for any instance $\mathcal{I}$ of JI with unit capacities

$$JIP(\mathcal{I}) \le \text{val}(y(\mathcal{I}), z(\mathcal{I})) \le 2 \cdot \text{val}(A(\mathcal{I})) \le 2JIS(\mathcal{I})$$

where $JIP(\mathcal{I})$ and $JIS(\mathcal{I})$ are the optimal values of JIP and JIS for $\mathcal{I}$, and $(y(\mathcal{I}), z(\mathcal{I}))$, and $A(\mathcal{I})$ are the values of the solutions returned by algorithm ALG1 applied to $\mathcal{I}$.

AQ5

Corollary 37.1

Algorithm ALG1 is a 1/2-approximation algorithm for JIP with unit capacities and 2-approximation for JIS with unit capacities.

The results stated in Corollary 37.1 are tight, that is, the analysis of the algorithm's performance cannot be improved to provide a better factor [12].

37.3 The Case of Unit Demands: Algorithm ALG2

In this section we focus on JI with unit demands. This special case of JI can be described as follows: given is a grid consisting of t columns, numbered consecutively from left to right, and m numbered rows together with a set of intervals $I = \{1, 2, \ldots, n\}$ lying on the rows of the grid. An interval i is specified by the triple (l_i, r_i, ρ_i), where l_i, r_i are the indices of the leftmost and the rightmost columns intersecting the interval and ρ_i is the index of the row where it lies. For each column c, row r, and interval i, we are given positive integral parameters v_c, u_r, and p_i respectively, referred to as the column, row, and interval capacities. We assume that the intervals are ordered according to nondecreasing r_i.

The objective of the JIS with unit demands is: *find a collection H of columns, rows, and intervals of minimum total capacity, such that for each interval $i \in I$, H contains either a column stabbing interval i, or the row stabbing interval i, or the interval i itself.*

Note that JIS with unit demands is a special case of the well-known *weighted hitting set* problem (WHS): *given is a finite weighted ground set $\mathcal{E}$, each element $e \in \mathcal{E}$ having a nonnegative weight w_e, and a collection of its subsets $\mathcal{S}$. Find a minimum-weight subset $H \subseteq \mathcal{E}$ such that $H \cap s \ne \emptyset$ for any $s \in \mathcal{S}$.*

Indeed, let the set of all the columns, rows, and intervals with their weights constitute a weighted ground set $\mathcal{E}$. Let $\mathcal{S}$ be $\{s_1, s_2, \ldots, s_n\}$, where subset s_i contains the interval i together with the subset of the columns and the row stabbing it. Then any feasible hitting set corresponds to a feasible solution to JIS with unit demands of the same value.

In the spirit of WHS we say that an interval i is *hit* by a subset H of columns, rows, and intervals if H contains either column or row stabbing interval i, or interval i itself.

Below we give an ILP formulation of JIS with unit demands (for the decision variables we use here characteristic vector (y, z, s) of H, where $y \in \mathbb{Z}^t$ is associated with the set of columns, $z \in \mathbb{Z}^m$ with the set of rows and $s \in \mathbb{Z}^n$ with the set of intervals).

$$\text{JIS: Minimize} \quad \sum_{c=1}^{t} v_c y_c + \sum_{r=1}^{m} u_r z_r + \sum_{i=1}^{n} p_i s_i \tag{37.29}$$

$$\text{subject to} \quad z_{\rho_i} + \sum_{c\in[l_i,r_i]} y_c + s_i \ge 1 \quad \forall i \tag{37.30}$$

$$z_r, y_c, s_i \in \{0, 1\} \qquad \forall r, c, i \tag{37.31}$$

The dual to its LP relaxation is

$$\text{Dual:}\quad \text{Maximize} \quad \sum_{i=1}^{n} x_i \tag{37.32}$$

$$\text{subject to} \quad \sum_{i:\rho_i=r} x_i \le u_r \qquad \forall r = 1, \ldots, m \tag{37.33}$$

$$\sum_{i:c\in[l_i,r_i]} x_i \le v_c \qquad \forall c = 1, \ldots, t \tag{37.34}$$

$$x_i \le p_i \qquad \forall i = 1, \ldots, n \tag{37.35}$$

$$x_i \ge 0 \qquad \forall i = 1, \ldots, n \tag{37.36}$$

Note that when replacing the nonnegativity constraints (37.36) with integrality constraints $x_i \in \mathbb{Z}\ \forall i = 1, \ldots, n$, we obtain an ILP formulation of the JIP with unit demands.

Specify an integral multiplicity for each interval i*, not exceeding its capacities* p_i*, such that*

- *for each column* c *or row* r *the sum of the multiplicities of the intervals sharing it does not exceed the capacity* v_c *or* u_r *respectively,*
- *the sum of the multiplicities is minimum.*

Following the framework of the generic *primal-dual algorithm with reverse delete step* described for WHS by Goemans and Williamson [17], we develop a primal-dual algorithm for JIS with unit demands, called ALG2.

We use auxiliary variables $\hat{v} \in \mathbb{R}^t$, $\hat{u} \in \mathbb{R}^m$, and $\hat{p} \in \mathbb{R}^n$ which are in fact slack variables for the constraints (37.33)–(37.35), that is, at each moment in time:

$$\hat{v}_c = v_c - \sum_{i:c\in[l_i,r_i]} x_i \ \ \forall c, \qquad \hat{u}_r = u_r - \sum_{i:\rho_i=r} x_i \ \ \forall r, \qquad \hat{p}_i = p_i - x_i \ \ \forall i \tag{37.37}$$

for some current values of x_i, $i = 1, \ldots, n$.

Initially, the dual vector x is zero, the set H is empty and the slack variables $\hat{v}$, $\hat{u}$, $\hat{p}$ are equal to v, u, and p respectively. For each interval $i \in I$ we check whether it is already hit by H, if not we do the following:

- assign to the dual variable x_i the minimum of the values of slack variables corresponding to the following elements: the columns stabbing interval i, its row and interval i itself, that is, $x_i \leftarrow \min\{\hat{v}_{l_i}, \ldots, \hat{v}_{r_i}, \hat{u}_{\rho_i}, \hat{p}_i\}$;
- update the slack variables (since the value of x_i changed). Because of the way x_i was updated, at least one of the slack variables $\hat{v}_{l_i}, \ldots, \hat{v}_{r_i}, \hat{u}_{\rho_i}, \hat{p}_i$ has to be zero now;
- add to H the elements (columns $l_i, \ldots, r_i$, row ρ_i, or interval i) whose corresponding slack variables are zero. Note that at least one of these element has to be added to H and thus interval i becomes hit.

Clearly, after all the n intervals are processed as above, H is a feasible solution to JIS with unit demands, since all the intervals are hit. At the next stage we try to remove elements from H. For that we consider each element in H and remove it if feasibility of H is preserved. The order of considerations plays a role here. First we check the columns in the order reverse to the order they were added to H. Then all the other elements, that is, rows and intervals, in an arbitrary order.

The formal description of algorithm ALG2 is shown in Figure 37.4. We use three index sets to represent set H, the set of column, row, and interval indices H^{col}, H^{row}, and H^{int}, respectively.

1. $x \leftarrow \mathbf{0}$
2. $H^{col} \leftarrow \emptyset$, $H^{row} \leftarrow \emptyset$, $H^{int} \leftarrow \emptyset$
3. $\hat{v} \leftarrow v$, $\hat{u} \leftarrow u$, $\hat{p} \leftarrow p$
4. $l \leftarrow 0$
5. For $i \leftarrow 1$ to n
   If $([l_i, r_i] \cap H^{col} = \emptyset)$ *AND* $(\{\rho_i\} \cap H^{row} = \emptyset)$ then
   $x_i \leftarrow \min\{\hat{v}_{l_i}, ..., \hat{v}_{r_i}, \hat{u}_{\rho_i}, \hat{p}_i\}$;
   For $c \leftarrow l_i$ to r_i if $(\hat{v}_c \leftarrow \hat{v}_c - x_i) = 0$ then
   $l \leftarrow l + 1$, $c_l \leftarrow c$, $H^{col} \leftarrow H^{col} \cup \{c\}$
   If $(\hat{u}_{\rho_i} \leftarrow \hat{u}_{\rho_i} - x_i) = 0$ then $H^{row} \leftarrow H^{row} \cup \{\rho_i\}$
   If $(\hat{p}_i \leftarrow \hat{p}_i - x_i) = 0$ then $H^{int} \leftarrow H^{int} \cup \{i\}$
6. For $j \leftarrow l$ downto 1
   if $H^{col} - \{c_j\}$, together with H^{row} and H^{int}, is feasible
   then $H^{col} \leftarrow H^{col} - \{c_j\}$
7. For all $i \in H^{int}$
   if $H^{int} - \{i\}$ together with H^{col} and H^{row} is feasible
   then $H^{row} \leftarrow H^{row} - \{i\}$
8. For all $r \in H^{row}$
   if $H^{row} - \{r\}$, together with H^{col} and H^{int}, is feasible
   then $H^{row} \leftarrow H^{row} - \{r\}$
9. Return H^{col}, H^{row}, H^{int} (and x)

FIGURE 37.4 Algorithm ALG2.

Theorem 37.1

For any instance $\mathcal{I}$ of JI with unit demands the sets $(H^{col}, H^{row}, H^{int})$ and vector x returned by the algorithm ALG2 describe feasible solutions to JIS and JIP respectively, and their values are related as follows:

$$val(H^{col}, H^{row}, H^{int}) \leq 2\, val(x)$$

To prove this we need a preliminary lemma. This is a result for the WHS that can be also found in Ref. [17]:

Lemma 37.1

Consider an instance of WHS. If a set $H \subset \mathcal{E}$, vector $\chi \in \mathbb{R}^{|\mathcal{S}|}$ and $\beta \geq 0$ satisfy

$$\forall e \in H : \sum_{s \ni e} \chi_s = w_e \quad \textit{and} \quad \forall s \in \mathcal{S}, \ \textit{such that}\ \chi_s > 0 : \ |s \cap H| \leq \beta$$

then

$$\sum_{e \in H} w_e \leq \beta \sum_{s \in \mathcal{S}} \chi_s$$

Proof

Using the conditions of the lemma we have

$$\sum_{e \in H} w_e = \sum_{e \in H} \sum_{s \ni e} \chi_s = \sum_{s \in \mathcal{S}} |s \cap H| \chi_s \leq \beta \sum_{s \in \mathcal{S}} \chi_s$$ □

Proof (of the Theorem)
Obviously, by construction the sets $(H^{col}, H^{row}, H^{int})$ describe a feasible solution to JIS with unit demands and the vector x is feasible to the dual LP formulation (37.33)–(37.35). Note, that the integrality of the input data implies integrality of x. Thus x is feasible to JIP with unit demands.

Let us establish the relation between the solution values. Recall the representation of JIS with unit demands as a special case of WHS, described earlier in this section. We use the result of the lemma with $(H^{col}, H^{row}, H^{int})$ representing H, x representing χ, β equal to 2.

Let us show that the first condition of the lemma is satisfied. Recall that an index of an element (which can be a column, row, or interval) is added to one of $(H^{col}, H^{row}, H^{int})$ only when the corresponding slack variable (37.37) becomes zero. After a slack variable becomes zero, it is not changed by the algorithm anymore. Thus at the end of the algorithm we have all the slack variables (37.37) corresponding to the elements in the solution to be zero, and thus the first condition of the lemma follows.

Let us establish the second condition, that is, show that for $(H^{col}, H^{row}, H^{int})$ returned by the algorithm and for any $i \in I$, such that $x_i > 0$ holds:

$$|\{l_i, \ldots, r_i\} \cap H^{col}| + |\{\rho_i\} \cap H^{row}| + |\{i\} \cap H^{int}| \leq 2$$

Take i such that $x_i > 0$. If we show that $|\{l_i, \ldots, r_i\} \cap H^{col}| \leq 1$, the above bound follows easily from the fact that, owing to the minimality of solution $(H^{col}, H^{row}, H^{int})$, accomplished in steps 6–8. $|\{i\} \cap H^{int}| = 0$ for any i, for which $|\{l_i, \ldots, r_i\} \cap H^{col}| + |\{\rho_i\} \cap H^{row}| \geq 1$.

Suppose $|\{l_i, \ldots, r_i\} \cap H^{col}| \geq 2$, that is, H^{col} contains at least two columns incident to the interval i, say, columns c_1 and c_2, $c_1 < c_2$. Consider the moment right before x_i became positive, that is, the beginning of the ith iteration at step 5 of the algorithm. All the previously considered intervals j, $j < i$, are already covered by this moment and neither c_1, nor c_2 are yet added to H^{col} (otherwise, $\{l_i, l_i + 1, \ldots, r_i\} \cap H^{col} = \emptyset$ would not hold). Look now at step 6, the moment when we are considering column c_1 as a candidate for removal from H^{col}. We claim that all intervals j, $j < i$, are currently covered by other elements (columns, rows, or intervals). This is because of the fact that no element, added to the solution during step 5 before column c_1, can be considered for removal in step 6 before c_1. Further, it is not difficult to see that all intervals l, $l \geq i$, stabbed by column c_1, have to be stabbed by column c_2 as well by the ordering of the intervals (see Figure 37.5). Therefore, nothing can prevent us from removing c_1 from H^{col} in step 6. □

Algorithm ALG2 can be implemented to run in $O(nt)$ time [3].

Theorem 37.1 together with the weak duality relation between JIP and JIS has the following consequence.

Corollary 37.2

Algorithm ALG2 is a 1/2-approximation algorithm for JIP with unit demands and a 2-approximation for JIS with unit demands.

Again, these performance guarantees are tight (see Kovaleva and Spieksma [12]). Finally, note that both algorithms ALG1 and ALG2 can be applied to instances of JI with unit capacities and demands. It is not difficult to verify that the solutions for JIP (subsets of intervals) returned by the two algorithms coincide, while the solutions to JIS (subsets of columns and rows) may be different.

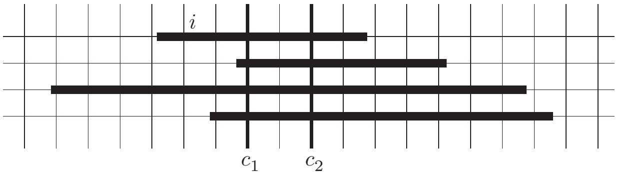

FIGURE 37.5 All the intervals l, $l \geq i$, incident to column c_1 have to be incident to column c_2 as well.

References

[1] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., and Protasi, M., *Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties*, Springer, Berlin, 1999.

[2] Vazirani, V.V., *Approximation Algorithms*, Springer, Berlin, 2001.

[3] Kovaleva, S., Approximation of Geometric Set Packing and Hitting Set Problems, Ph.D. thesis, Maastricht University, 2003.

[4] Berman, P. and DasGupta, B., Multi-phase algorithms for throughput maximization for real-time scheduling, *J. Comb. Opt.*, 4, 307, 2000.

[5] Erlebach, T. and Spieksma, F.C.R., Interval selection: applications, algorithms, and lower bounds, *J. Algorithms*, 46, 27, 2003.

[6] Bar-Noy, A., Guha, S., Naor, J., and Schieber, B., Approximating the throughput of multiple machines in real-time scheduling, *SIAM J. Comput.*, 31, 331, 2001.

[7] Spieksma, F.C.R., On the approximability of an interval scheduling problem, *J. Scheduling*, 2, 215, 1999.

[8] Chuzhoy, J., Ostrovsky, R., and Rabani, Y., Approximation algorithms for the job interval selection problem and related scheduling problems, *Proc. FOCS*, 2001.

[9] Calinescu, G., Chakrabarti, A., Karloff, H., and Rabani, Y., Improved approximation algorithms for resource allocation, *Proc. IPCO Conf.*, Lecture Notes in Computer Science, Vol. 2337, Springer, Berlin, 2002, p. 401.

[10] Bar-Noy, A., Bar-Yehuda, R, Freund, A., Naor, J., and Schieber, B., A unified approach to approximating resource allocation and scheduling, *JACM*, 48, 1069, 2001.

[11] Hassin, R. and Megiddo, N., Approximation algorithms for hitting objects with straight lines, *Disc. Appl. Math.*, 30, 29, 1991.

[12] Kovaleva, S. and Spieksma, F.C.R., Primal-dual approximation algorithms for a packing-covering pair of problems, *RAIRO-Oper. Res.*, 36, 53, 2002.

[13] Kovaleva, S. and Spieksma, F.C.R., Approximation algorithms for rectangle stabbing and interval stabbing problems, conditionally accepted for *SIAM J. Disc. Math.*, 2005.

AQ3

[14] Gaur, D., Ibaraki, T., and Krishnamurti, R., Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem, *J. Algorithms*, 43, 138, 2002.

[15] Akcoglu, K., Aspres, J., DasGupta, B., and Kao, M.-Y., Opportunity cost algorithms for combinatorial auctions, in *Applied Optimization: Computational Methods in Decision-Making, Economics, and Finance*, Kontoghiorghes, E.J., Rustem, B., and Siokos, S., Eds., *Applied Optimization: Computational Methods in Decision-Making, Economics, and Finance*, Kluwer Academic, Dordrecht, 2002.

[16] Bar-Yehuda, R. and Even, S., A local-ratio theorem for approximating the weighted set cover problem, *Ann. Disc. Math.*, 25, 27, 1985.

[17] Goemans, M.X. and Williamson, D.P., The primal-dual method for approximation algorithms and its application to network design problems, in *Approximation Algorithms for NP-Hard Problems*, Hochbaum, D.S., Ed., PWC Publishing Company, Boston.

AQ4

Query Form

CRC Press

The C5505 Handbook, Second Edition

JOURNAL TITLE:	**84002**
ARTICLE NO:	**CH037**

Queries and / or remarks

Query No.	Details required	Author's response
AQ1	Please provide the full form for "JI"	
AQ2	Please provide the full form for "ILP" and "LP".	
AQ3	Please provide the volume no. and page range for ref. [13].	
AQ4	Please provide the year of publication for ref. [17].	
AQ5	Renumber the sequences of Corollary, Theorem, and Lemma.	