RAIRO Operations Research
RAIRO Oper. Res. **36** (2002) 53-71
DOI: 10.1051/ro:2002005

PRIMAL-DUAL APPROXIMATION ALGORITHMS FOR A PACKING-COVERING PAIR OF PROBLEMS *

Sofia Kovaleva[1] and Frits C.R. Spieksma[2]

Communicated by Rainer E. Burkard

Abstract. We consider a special packing-covering pair of problems. The packing problem is a natural generalization of finding a (weighted) maximum independent set in an interval graph, the covering problem generalizes the problem of finding a (weighted) minimum clique cover in an interval graph. The problem pair involves weights and capacities; we consider the case of unit weights and the case of unit capacities. In each case we describe a simple algorithm that outputs a solution to the packing problem and to the covering problem that are within a factor of 2 of each other. Each of these results implies an approximative min-max result. For the general case of arbitrary weights and capacities we describe an LP-based $(2+\epsilon)$-approximation algorithm for the covering problem. Finally, we show that, unless $\mathcal{P} = \mathcal{NP}$, the covering problem cannot be approximated in polynomial time within arbitrarily good precision.

Keywords: Primal-dual, approximation algorithms, packing-covering, intervals.

Mathematics Subject Classification. 05A05.

Received December, 2000.

* *This research was partially supported by EU Grant 30027 (APPOL II).*

[1] Department of Mathematics, Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands; e-mail: `s.kovaleva@math.unimaas.nl`

[2] Department of Applied Economics, Katholieke Universiteit Leuven, Naamsestraat 69, 3000 Leuven, Belgium.

1. Introduction

Consider the following setting. Given is a grid consisting of t columns and m rows. Furthermore, n intervals are given such that each interval I_i lies on a single row (called row(i)) and it occupies columns $l_i, l_{i+1}, \ldots, r_i$ (we refer to column j *stabbing* I_i when $l_i \leq j \leq r_i$). For each interval I_i a nonnegative *weight* w_i is given, and for each column j, for each row k and for each interval I_i nonnegative *capacities* v_j, u_k and p_i are specified respectively. We assume that all data are integral. We consider two problems:

A **PACKING** problem: specify an integral multiplicity x_i for each interval I_i that is no more than p_i, such that the sum of multiplicities of intervals sharing a common row or column does not exceed the corresponding given capacity, while maximizing total weight.

We refer to this problem as a *geometric set packing problem* (GEOSP). Below we give an integer programming formulation for it. Here, the x-variables are integer and denote the multiplicities of the intervals:

$$\begin{array}{lll} \text{Maximize} & \sum_{i=1}^{n} w_i x_i & & (1) \\ \text{subject to} & \sum_{i:\text{row}(i)=k} x_i \leq u_k & \text{for all } k = 1, \ldots, m, & (2) \\ & \sum_{i:j\in[l_i,r_i]} x_i \leq v_j & \text{for all } j = 1, \ldots, t, & (3) \\ & x_i \leq p_i & \text{for all } i = 1, \ldots, n, & (4) \\ & x_i \in Z_+^1 & \text{for all } i = 1, \ldots, n. & (5) \end{array}$$

Constraints (2) state that the sum of the multiplicities of intervals on row k cannot exceed u_k; constraints (3) express the requirement that the sum of the multiplicities of intervals that are stabbed by column j is no more than v_j. Constraints (4) are the upper bound constraints and constraints (5) are the integrality constraints.

A **COVERING** problem: find a multiplicity for each column, row and interval minimizing the total capacity, while for each interval the sum of multiplicities of the columns stabbing it plus the multiplicity of its row plus its own multiplicity is not less then its weight.

We refer to this problem as a *geometric set covering problem* (GEOSC). Below we give its integer programming formulation. Here the y, z and s-variables are integer and denote the multiplicities of the columns, rows and intervals respectively.

$$\begin{array}{lll} \text{Minimize} & \sum_{j=1}^{t} v_j y_j + \sum_{k=1}^{m} u_k z_k + \sum_{i=1}^{n} p_i s_i & & (6) \\ \text{subject to} & z_{\text{row}(i)} + \sum_{j\in[l_i,r_i]} y_j + s_i \geq w_i & \text{for all } i = 1, \ldots, n & (7) \\ & z_k, y_j, s_i \in Z_+^1 & \text{for all } k, j, i. & (8) \end{array}$$

Constraints (7) state that for each interval I_i, the sum of the multiplicities of columns stabbing I_i plus the row-multiplicity plus its own multiplicity s_i must be at least w_i and constraints (8) are the integrality constraints.

Thus an instance of any of our problems is specified by specifying numbers $t, m, n \in Z^1_+$, pairs $(l_i, r_i) \in Z^2_+$, $i = 1 \dots n$, function row$(\cdot) : \{1, 2, \dots, n\} \to \{1, 2, \dots, m\}$ and vectors $\mathbf{w} \in Z^n_+, \mathbf{v} \in Z^t_+, \mathbf{u} \in Z^m_+, \mathbf{p} \in Z^n_+$.

Notice that the two integer programming problems are primal-dual related, that is, their LP-relaxations, obtained by replacing (5) and (8) by the corresponding nonnegativity constraints, form a primal-dual pair of LPs.

Applications. Although at first sight problems GEOSP and GEOSC may look rather specific, they contain a host of problems. A special case of GEOSP is the problem of finding a maximum weighted independent set in an interval graph (see *e.g.* [10]): indeed, when allowing at most one interval on a row and setting all capacities in GEOSP to 1, this problem arises. In [7] the case is investigated where $u_k = 1, v_j = v, p_i = 1$ for all k, j, i. Applications of the resulting problem can be found in scheduling [4], molecular biology [6] and PCB-assembly [14]. We refer to these references for a more elaborate description of these problems.

Applications of the covering problem seem less abundantly present in literature; however, the following type of situation leads naturally to instances of GEOSC with $w_i = 1$ for all i. Each of m items (patients to receive treatment, products to undergo chemical processes, machines subject to inspection) has to undergo treatment on a regular basis. More precisely, for each item a set of intervals is given during which a treatment must take place. The treatment itself takes one time-unit and is provided by some kind of machine with unbounded capacity (that is, it can process any number of items), and consists of "turning the machine on" at some point in time, say q (this corresponds to selecting column q). Then the items corresponding to intervals that are stabbed by column q undergo the treatment. The objective is to minimize (a weighted combination of) the number of times the machine is turned on plus the number of items not processed (an item is not processed when at least one of its intervals has not undergone the treatment (this corresponds to selecting the row corresponding to that item)). An example of such a problem is described in [1].

It is not hard to see that GEOSC with $w_i = 1$ for all i is a special case of the well-known weighted set covering problem: let the intervals correspond to elements in the ground set, and let a set of intervals that are on a same row or a set of intervals that share a column or a single-interval set correspond to a set in the collection.

Previous and related results. Concerning approximation results for general set packing and set covering problems we refer to [12]. In [14] it is proved that GEOSP is MAX SNP-hard already when $\mathbf{w} = \mathbf{1}$, $\mathbf{u} = \mathbf{1}$, and $\mathbf{v} = \mathbf{1}$. Also, in [14] it is shown that for this case the value of an optimal solution to GEOSC is bounded by two times the value of an optimal solution to GEOSP. In [5] and [6] the case of GEOSP is studied with $\mathbf{u} = \mathbf{1}$ and $v_j = v$ for all j (where v reflects the number of machines in a scheduling context). Each of these papers presents a combinatorial $\frac{1}{2}$-approximation algorithm for GEOSP.

Actually, the problem considered in [5] is more general than the one that arises from GEOSP by setting $\mathbf{u} = \mathbf{1}$ and $v_j = v$ for all j: a nonnegative "*width*" for each interval is specified and it is required that a column's capacity is not exceeded by the total width (and not necessarily by the number of the intervals which are stabbed by it). In [5] a general framework is described, based on the local ratio technique, which allows to approximate with a constant factor a variety of special cases of this problem.

Another special case of the set covering problem for which a constant approximation factor is achieved is described in [8]. They present a 2-approximation algorithm for special set cover instances which they call tree-representable. It is easy to verify that there exist instances of GEOSC2 that are not tree-representable.

Our results. Here we show that

- if either ($\mathbf{w} = \mathbf{1}$) or ($\mathbf{v} = \mathbf{1}, \mathbf{u} = \mathbf{1}$) we present a simple algorithm that outputs feasible solutions to *both* GEOSP and GEOSC, that are within a factor of 2. For the case ($\mathbf{v} = \mathbf{1}, \mathbf{u} = \mathbf{1}$) this algorithm is based on [5] and [6]. This result implies an approximative min-max result for the corresponding pair of problems (see Sect. 2);
- a $(2 + \epsilon)$-approximation algorithm for GEOSC exists (see Sect. 4);
- GEOSC is MAX SNP-hard (see Sect. 5).

2. Primal-dual approximation algorithms

Approximation algorithms based on the primal-dual method have been used successfully in obtaining performance guarantees for a number of combinatorial optimization problems that have a natural formulation as an integer program (see *e.g.* [9,15]). Based on an IP-formulation of a problem (in our context, the covering problem, or the dual problem) one typically aims to construct a good feasible solution for this problem in the following way. The basic algorithm starts with a feasible primal solution and an infeasible dual solution. Next, guided by (some of the) complementary slackness conditions, both the primal and dual solutions are iteratively modified, so that the dual solution becomes feasible. Using problem specific features one is often able to prove worst-case ratios for these algorithms (see [9]).

Note that in general the primal solution constructed need not to be integral. In this section we present two primal-dual approximation algorithms that provide simultaneously two feasible integral solutions, one for the primal problem and one for the dual problem.

We use the terminology assumed in [11], according to which an algorithm is said to be a δ-approximation algorithm for a minimization (maximization) problem if for every instance of the problem it delivers a feasible solution with a value of at most (at least) δ times the optimum.

We assume that the intervals are indexed according to non-decreasing rightmost stabbing column, that is $r_1 \leq r_2 \leq \ldots \leq r_n$.

We also assume (wlog) that there is at least one interval on each row, therefore $m \leq n$.

Given an instance $\mathcal{I}$ of GEOSP (GEOSC) let $OPT_{\text{pack}}(\mathcal{I})$ ($OPT_{\text{cover}}(\mathcal{I})$) denote the value of the optimal solution of the corresponding problem instance. Also, let v_{LP} be equal to the optimal value of the LP-relaxation. Obviously, due to weak duality we have that $OPT_{\text{pack}}(\mathcal{I}) \le OPT_{\text{cover}}(\mathcal{I})$ for all instances $\mathcal{I}$.

2.1. Unit capacities: The case u = 1, v = 1

Notice that in this setting the constraints $x_i \le p_i$ become superfluous since they will be automatically satisfied for any integral $p_i > 0$. Therefore, all the s-variables will have value 0 and we do not need to consider them further in this subsection.

Notice also that in the IP formulation of GEOSP we can disregard the constraints $\sum_{i:j\in[l_i,r_i]} x_i \le 1$, for $j \notin \{r_1, ..., r_n\}$. Indeed, the column capacity constraints corresponding to the columns other than the last stabbing columns of the intervals are implied by the other constraints. So among the y-variables we consider only $y_{r_1}, ..., y_{r_n}$, all the other y-variables are 0.

The idea of the algorithm is proposed in [6], where a combinatorial 1/2-approximation algorithm is presented for a problem, similar to GEOSP with $\mathbf{u} = \mathbf{1}$ and $v_j = v$ for all j. Here we describe a slightly modified algorithm, referred to as *ALG1*, which delivers a feasible solution to *GEOSC* as well.

The algorithm maintains a pair of solutions for the problems GEOSP and GEOSC, $\mathbf{x}$ and $(\mathbf{y}, \mathbf{z})$ respectively. Starting with a zero assignment to $\mathbf{x}, \mathbf{y}, \mathbf{z}$ (which is infeasible for GEOSC), the algorithm iteratively modifies their values. At the end of the *forward pass* we obtain a feasible solution to GEOSC and an infeasible one to GEOSP. The *backward pass* is used to restore the feasibility of $\mathbf{x}$ to GEOSP.

Let $\mathbf{x} \in Z_n^+$, $\mathbf{y} \in Z_n^+$, $\mathbf{z} \in Z_m^+$ and $\Delta \in Z_n^+$.

The algorithm $ALG1$ proceeds as follows:

1. $\mathbf{x} \leftarrow \mathbf{0}$, $\mathbf{y} \leftarrow \mathbf{0}$, $\mathbf{z} \leftarrow \mathbf{0}$;
{*forward pass*}·

2. For all i from 1 to n do:
begin

$$\Delta_i \leftarrow \max\left\{ w_i - \left(z_{\text{row}(i)} + \sum_{j\in[l_i,r_i]} y_j \right), 0 \right\}. \tag{9}$$

If $\Delta_i > 0$ *then* $x_i \leftarrow 1$;

$y_{r_i} \leftarrow y_{r_i} + \Delta_i$;

$z_{\text{row}(i)} \leftarrow z_{\text{row}(i)} + \Delta_i$;
end;
{*backward pass*}·

3. For all i from n down to 1 do:
if $x_i = 1$ and $(0,...,0,x_i,x_{i+1},...,x_n)$ is not a feasible solution to GEOSP, assign
$x_i = 0$.

Remark 2.1. *ALG*1 can be implemented to run in $O(n\log(n))$ time, see [6].

Theorem 2.2 (Min-max *result for the case of unit capacities*)**.** *For all instances $\mathcal{I}$ of the pair GEOSP-GEOSC where all columns and rows have unit capacities,*

$$OPT_{\text{cover}}(\mathcal{I}) \leq 2 \cdot OPT_{\text{pack}}(\mathcal{I}).$$

Proof. First, we establish that *ALG*1 delivers feasible solutions $\mathbf{x}$ and $(\mathbf{y},\mathbf{z})$ to GEOSP and GEOSC respectively (in the case of unit capacities). Then we argue that $(\mathbf{1}\cdot\mathbf{y}+\mathbf{1}\cdot\mathbf{z}) \leq 2\mathbf{wx}$.

Notice that after each loop i in the forward pass vector $(\mathbf{y},\mathbf{z})$ is increased so as to satisfy the constraint corresponding to interval I_i in (7). Thus at the end of the algorithm $(\mathbf{y},\mathbf{z})$ satisfies all constraints in (7) and constitutes a feasible solution. Feasibility of the solution $\mathbf{x}$ is provided by the backward pass.

Let us now argue that $\mathbf{1}\cdot\mathbf{y}+\mathbf{1}\cdot\mathbf{z} \leq \mathbf{2wx}$. Observe that $\sum_{j=1}^{t} y_j = \sum_{k=1}^{m} z_k = \sum_{i=1}^{n} \Delta_i$, since during the forward pass in every loop i the algorithm increases the values of $\sum_{j=1}^{t} y_j$ and $\sum_{k=1}^{m} z_k$ uniformly by Δ_i. Observe also that for each interval I_i with $\Delta_i > 0$, we have

$$w_i = \Delta_i + \sum_{j:j<i, r_j\in[l_i,r_i]} \Delta_j + \sum_{l:l<i,\text{row}(l)=\text{row}(i)} \Delta_l.$$

This follows from (9) and from the fact that at the moment of evaluation of Δ_i, $z_{\text{row}(i)}$ equals the sum of contributions made by the earlier processed intervals lying on the same row and $\sum_{j\in[l_i,r_i]} y_j$ is a sum of contributions of the earlier processed intervals sharing a column with I_i (for such an interval I_j we have $r_j \in [l_i, r_i]$). Thus,

$$\mathbf{xw} = \sum_{i:x_i=1} w_i = \sum_{i:x_i=1} \Delta_i + \sum_{i:x_i=1} \left(\sum_{j:j<i, r_j\in[l_i,r_i]} \Delta_j + \sum_{j:j<i,\text{row}(j)=\text{row}(i)} \Delta_j \right). \tag{10}$$

Now, consider any interval I_j such that $\Delta_j > 0$ and $x_j = 0$. Observe that after the forward pass the current value of x_j is 1 and the fact that in the backward pass the value was changed to 0 means that $(0,...,0,1,x_{j+1},...,x_n)$ is not a feasible solution. Thus, there exists and interval I_i, $i > j$, $x_i = 1$, sharing a column or a row with interval I_j. In case it shares a row: $\text{row}(j) = \text{row}(i)$ and Δ_j is included in the last term of (10). Consider the case when it shares a column. Then due to the ordering of the intervals in the instance according to non-decreasing right-most stabbing column, the fact $i > j$ implies that $r_j \in [l_i, r_i]$. So again, the interval I_j has its contribution Δ_j to the last term of (10).

Now we have

$$\mathbf{xw} \geq \sum_{i:x_i=1} \Delta_i + \sum_{i:x_i=0} \Delta_i = \sum_{i=1}^{n} \Delta_i = \sum_{j=1}^{t} y_j = \sum_{k=1}^{m} z_k.$$

Therefore

$$2\mathbf{xw} \geq \sum_{j=1}^{t} y_j + \sum_{k=1}^{m} z_k.$$

□

Corollary 2.3. *$ALG1$ is a 1/2-approximation algorithm for GEOSP with unit column and row capacities and a 2-approximation algorithm for GEOSC with unit column and row capacities.*

2.2. Unit weights: The case w = 1

Consider another special case of GEOSP and GEOSC. Now the column, row and interval capacities are arbitrary nonnegative integer numbers, and all the intervals have the same weight, or equivalently $\mathbf{w} = \mathbf{1}$. Since in this case GEOSC can be seen as a so-called hitting set problem, the high-level description of primal-dual method given in [9] apply. Similarly to the previous case we describe an algorithm $ALG2$, which in this case delivers a feasible solution $\mathbf{x}$ to GEOSP and $\mathbf{y}, \mathbf{z}, \mathbf{s}$ to GEOSC, such that $\mathbf{vy} + \mathbf{uz} + \mathbf{sp} \leq 2(\mathbf{1} \cdot \mathbf{x})$.

During its execution the algorithm maintains solutions $\mathbf{x}$ and $\mathbf{y}, \mathbf{z}, \mathbf{s}$ to GEOSP and to GEOSC respectively. As in the previous subsection all the values have initially a value of 0. While keeping the complementary slackness conditions associated to constraints (2,3) and (4) satisfied, the algorithm iteratively modifies their values, so that $\mathbf{x}$ remains feasible to GEOSP and $\mathbf{y}, \mathbf{z}, \mathbf{s}$ constitute a feasible solution to GEOSC at the end of the *forward pass.* The *backward pass* (called the *reverse delete step* in [9]) is used to get rid of redundant column multiplicities and will allow us to obtain the ratio of 2 between the solution values.

Notice that this algorithm generalizes the algorithm described in [14] for the case of unit weights and unit capacities.

Let $\mathbf{x} \in Z_n^+$, $\mathbf{y} \in Z_t^+$, $\mathbf{z} \in Z_m^+$, $\mathbf{s} \in Z_n^+$, $\mathbf{V} \in Z_t^+$ and $\mathbf{U} \in Z_m^+$.

Algorithm *$ALG2$*:

1. $\mathbf{x} \leftarrow \mathbf{0}; \mathbf{y} \leftarrow \mathbf{0}; \mathbf{z} \leftarrow \mathbf{0}; \mathbf{s} \leftarrow \mathbf{0}; \mathbf{V} \leftarrow \mathbf{v}; \mathbf{U} \leftarrow \mathbf{u}$;

{*forward pass*}·

2. For all i from 1 to n do:

begin

$x_i \leftarrow \min\{V_{l_i}, V_{l_i+1}, ..., V_{r_i}, U_{\text{row}(i)}, p_i\}$;

if $(U_{\text{row}(i)} \leftarrow U_{\text{row}(i)} - x_i) = 0$, *then* $z_{\text{row}(i)} \leftarrow 1$; *if* $(p_i - x_i) = 0$, *then* $s_i \leftarrow 1$;

for all $j \in [l_i, r_i]$: if $(V_j \leftarrow V_j - x_i) = 0$, then $y_j \leftarrow 1$; end.
{backward pass}·
3. *For all j from t down to 1 do:*
if $y_j = 1$ and $(y_1, y_2, ..., y_{j-1}, 0, y_{j+1}, ..., y_t), \mathbf{z}, \mathbf{s}$ is a feasible solution, assign $y_j \leftarrow 0$;
4. *For all i from 1 up to n do:*
if $s_i = 1$ and $(\mathbf{y}, \mathbf{z}, (s_1, ..., s_{i-1}, 0, s_{i+1}, ..., s_n))$ is a feasible solution to GEOSC, then $s_i \leftarrow 0$.

Theorem 2.4. *The time complexity of ALG2 is $O(tn)$.*

Proof. In the forward pass the algorithm makes n loops, each of which takes at most $O(t)$ time to calculate the value of x_i and to update the values of the V-variables, thus we need $O(tn)$ time to complete the forward pass.

Consider step 3. There are t loops, we will argue that the feasibility test in a single loop i can be done in $O(n)$ time.

Let $l_1 = \max\{\max\{l : l < j, y_l = 1\}, 0\}$ and $l_2 = \min\{\min\{l : l > j, y_l = 1\}, t + 1\}$. So l_1 and l_2 are the indices of the columns, closest to column j from the left and from the right respectively and such that $\mathbf{y}$ currently has ones in the corresponding positions (if they exist, otherwise l_1 (l_2) is assigned 0 ($t+1$)).

Knowing that $(\mathbf{y}, \mathbf{z}, \mathbf{s})$ is a feasible solution (see the proof of Th. 2.5) enables us to test the feasibility of $((y_1, y_2, ..., y_{j-1}, 0, y_{j+1}, ..., y_t), \mathbf{z}, \mathbf{s})$ within $O(n)$ time: it is enough to consider all the intervals stabbed by column j and to check whether each of them satisfies one of the following conditions:

- it is stabbed by column l_1;
- it is stabbed by column l_2;
- the corresponding z-variable is 1;
- the corresponding s-variable is 1.

Obviously, this takes $O(n)$ time. Therefore if at each step the numbers l_1 and l_2 can be found in a constant time, the overall time complexity of the backward pass, and thus of the whole algorithm, is $O(tn)$.

In order to find the numbers l_1 and l_2 efficiently, we maintain an additional data structure: a list $\{0\} \leftrightarrows \{i_1\} \leftrightarrows ... \leftrightarrows \{i_p\} \leftrightarrows \{t + 1\}$, where $i_1, ..., i_p$ are the indices of all the (currently) unit elements of $\mathbf{y}$, arranged in increasing order. Each internal element of the list has a reference to the previous and to the next element.

Thus, having a reference to the element j in the list, we can obtain the values of l_1 and l_2 within constant time, by taking the previous and the next elements of the list respectively.

Notice that this data structure can be created at the beginning of the backward pass in $O(t)$ time and it takes only a constant amount of time to update it when some element of $\mathbf{y}$ changes its value from 1 to 0.

Finally, consider step 4. There are n loops and in each of them, say loop i, we need to check the values of $y_{l_i}, ..., y_{r_i}$ and $z_{\text{row}(i)}$. Thus, we have at most $t + 1$ operations. So step 4 takes $O(tn)$ time as well. □

Theorem 2.5 (Min-max *result for the case of unit weights*)**.** *For all instances $\mathcal{I}$ of the pair GEOSP-GEOSC where all intervals have unit weight,*

$$OPT_{\text{cover}}(\mathcal{I}) \leq 2 \cdot OPT_{\text{pack}}(\mathcal{I}).$$

Proof. We first establish feasibility of the solutions delivered by $ALG2$. We show feasibility of $\mathbf{x}$ by induction on the number of loops in the forward pass. The initial zero assignment is feasible. Suppose that after $i-1$ loops we have a feasible solution $(x_1, x_2, \dots, x_{i-1}, 0, \dots, 0)$. Observe that at the beginning of loop i the values of $\mathbf{V}$ and $\mathbf{U}$ denote the residual capacity after assigning multiplicities $(x_1, x_2, \dots, x_{i-1}, 0, \dots, 0)$ to the intervals. Therefore the assignment $x_i \leftarrow \min\{V_{l_i}, V_{l_i+1}, ..., V_{r_i}, U_{\text{row}(i)}\}$ does not violate the capacity constraint and thus the feasibility of the solution.

Now we proof the feasibility of $(\mathbf{y}, \mathbf{z}, \mathbf{s})$ to $GEOSC$. Observe that during the forward pass for each i at least one of the variables $y_{l_i}, y_{l_i+1}, ..., y_{r_i}, z_{\text{row}(i)}, s_i$ gets a unit value. That implies the feasibility of $(\mathbf{y}, \mathbf{z}, \mathbf{s})$ at the end of the forward pass. Since the backward pass can not violate feasibility, it remains feasible after the backward pass as well.

Let us now establish the theorem.

First we show that

$$s_i(x_i - p_i) = 0 \text{ for all } i, \tag{11}$$

$$y_j \left(\sum_{i: j \in [l_i, r_i]} x_i - v_j \right) = 0 \text{ for all } j \text{ and} \tag{12}$$

$$z_k \left(\sum_{i: \text{row}(i) = k} x_i - u_k \right) = 0 \text{ for all } k. \tag{13}$$

(Notice that these are the complementary slackness conditions associated to (2, 3) and (4).)

The first equality follows straightforward from the description of the algorithm. Consider (12). Notice that $y_j > 0$ implies $V_j = 0$. That means that the capacity of the column j is exhausted completely by assigning multiplicities $x_1, x_2, ..., x_n$ to the intervals. Thus $\sum_{i: j \in [l_i, r_i]} x_i = v_j$. A similar argument can be used to justify (13).

Now we show that

$$x_i > 0 \ \text{ implies } \ z_{\text{row}(i)} + \sum_{j \in [l_i, r_i]} y_j + s_i \leq 2. \tag{14}$$

First, notice that due to step 4 of the algorithm $s_i = 1$ implies $z_{\text{row}(i)} = \sum_{j \in [l_i, r_i]} y_j = 0$. Thus, the correctness of (14) is proved if we show that for each interval I_i, such that $x_i > 0$, $\sum_{j \in [l_i, r_i]} y_j \leq 1$. We will derive this by a contradiction argument.

Assume that there exists an interval I_i such that $\sum_{j\in[l_i,r_i]} y_j \geq 2$ and $x_i > 0$. That means that at least 2 columns stabbing the interval have unit multiplicity. Let j_1 and j_2 be the right-most and the left-most of those columns respectively. Let $\hat{\mathbf{y}}$ denote the value of vector $\mathbf{y}$ right after the forward pass. The fact that in the backward pass the value of y_{j_1} was not decreased to 0 means that $(\hat{y}_1, \hat{y}_2, ..., \hat{y}_{j_1-1}, 0, y_{j_1+1}, ..., y_t)$ is not a feasible solution to GEOSC. Therefore, there exists an interval I_k, such that (a) $z_{\text{row}(k)} = s_k = 0$, (b) there is no column j stabbing I_k, such that $j < j_1$ and $\hat{y}_j > 0$, (c) I_k is not stabbed by column j_2.

Consider loop k of the forward pass. The argument that at least one of the columns stabbing interval I_k, or its row has to get a positive value in this loop, together with (a) and (b) derived above, implies that at least one of $y_{j_1}, y_{j_1+1}, ..., y_{r_k}$ becomes positive, thus at least one of $V_{j_1}, V_{j_1+1}, ..., V_{r_k}$ becomes 0 after loop k.

(c) implies that $r_k < r_i$, thus $k < i$, which means that interval I_i was considered by the algorithm after I_k.

Since all the columns $j_1, j_1+1, ..., r_k$ stab I_i, and when evaluating x_i one of $V_{j_1}, V_{j_1+1}, ..., V_{r_k}$ is 0, x_i has to be 0, we have found a contradiction.

Due to (11–14):

$\mathbf{vy} + \mathbf{uz} + \mathbf{ps} = \sum_{j=1}^{t} y_j \sum_{i:j\in[l_i,r_i]} x_i + \sum_{k=1}^{m} z_k \sum_{i:\text{row}(i)=k} x_i + \sum_{i=1}^{n} s_i x_i = \sum_{i=1}^{n} x_i \left(z_{\text{row}(i)} + \sum_{j\in[l_i,r_i]} y_i + s_i\right) \leq 2\sum_{i=1}^{n} x_i = 2(\mathbf{1}\cdot\mathbf{x})$. □

Corollary 2.6. *ALG2 is a 1/2-approximation algorithm for GEOSP with unit interval weights and a 2-approximation algorithm for GEOSC with unit interval weights.*

3. Tightness

In this section we present 3 examples. In all the examples the weights and capacities are unit, therefore we can apply both the algorithms *ALG*1 and *ALG*2. We denote the values of the solutions delivered by the algorithms as $SOL_{\text{pack}}(ALG1)$ and $SOL_{\text{cover}}(ALG1)$ ($SOL_{\text{pack}}(ALG2)$ and $SOL_{\text{cover}}(ALG2)$ respectively).

For all three examples we have $SOL_{\text{pack}}(ALG1) = SOL_{\text{pack}}(ALG2)$ and $SOL_{\text{cover}}(ALG1) = SOL_{\text{cover}}(ALG2)$.

Example 1 shows that our bound on the gap size between the optimal values of GEOSP and GEOSC is tight even in the case when all the data are unit.

Examples 2 and 3 demonstrate that the performance guarantees derived in the previous section are tight. Example 2 establishes that *ALG*1 as well as *ALG*2 is a 1/2-approximation algorithm for GEOSP, example 3 establishes that *ALG*1 as well as *ALG*2 is a 2-approximation algorithm for GEOSC.

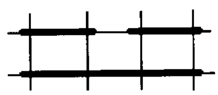

FIGURE 1. Example 1.

Example 1. Obviously for this instance $\mathcal{I}$ we have $OPT_{\text{pack}}(\mathcal{I}) = 1$ and $OPT_{\text{cover}}(\mathcal{I}) = 2$.

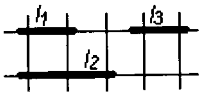

FIGURE 2. Example 2.

Example 2. For this instance both the algorithms $ALG1$ and $ALG2$ deliver solution $x_1 = 1, x_2 = x_3 = 0$ to GEOSP, with $SOL_{\text{pack}}(ALG1) = SOL_{\text{pack}}(ALG2) = 1$, while the optimal solution is $x_1 = 0, x_2 = x_3 = 1$, so that $OPT_{\text{pack}} = 2$.

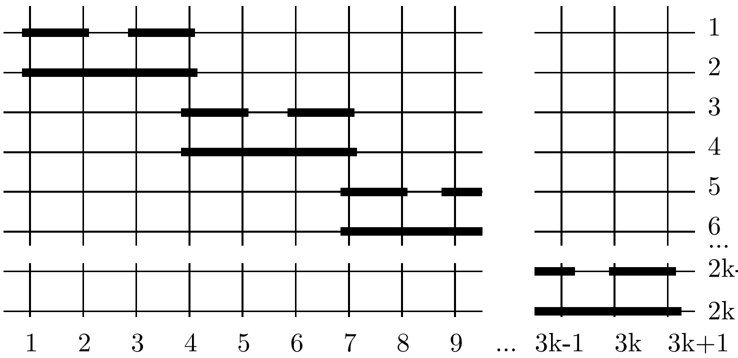

FIGURE 3. Example 3.

Example 3. The solution to GEOSC delivered by our algorithms is $y_{2+3p} = 1$, $p = 0, ..., k-1$, $z_{1+2h} = 1$, $h = 0, ..., k-1$, all the other variables are 0. So $SOL_{\text{cover}}(ALG1) = SOL_{\text{cover}}(ALG2) = 2k$, while the optimal solution is the following : $y_{1+3r} = 1$, $r = 0, ..., k$, all the other variables are 0, and thus $OPT_{\text{cover}} = k + 1$. When k increases, the ratio between $SOL_{\text{cover}}(ALG1)$ and OPT_{cover} (and between $SOL_{\text{cover}}(ALG2)$ and OPT_{cover}) tends to 2.

4. AN LP-BASED $(2 + \epsilon)$-APPROXIMATION ALGORITHM FOR GEOSC

In this section we focus on GEOSC with arbitrary weights and capacities. We present an LP-based algorithm that delivers a solution to GEOSC with a value of at most $2+\epsilon$ times the value of an optimal solution, for any $\epsilon > 0$. We use an idea described in [4] to obtain the desired result.

Informally, the algorithm works as follows. Consider an instance $\mathcal{I}$ of GEOSC and IP formulation (6–8) for it. First we solve the LP-relaxation. Let the corresponding solution of the linear program be described by $\hat{\mathbf{y}}, \hat{\mathbf{z}}, \hat{\mathbf{s}}$ and let v_{LP} be its value. In addition to $\mathcal{I}$ let us also assume that some $\epsilon > 0$ has been specified. Let

$\delta = \frac{\epsilon/4}{(n+t+m)(1+\epsilon/4)}$ and let $N = \frac{(n+t+m)(1+\epsilon/4)}{(\epsilon/4)^2}$. First we modify the values of $\hat{y}_j$, $\hat{z}_k$ and $\hat{s}_i$ so that each of them becomes 0 if it is less then δ and is multiplied by $(1+\epsilon/4)$ otherwise. Then we round each of the modified fractional values up to the nearest multiple of $1/N$, so that the rounded values are $\tilde{y}_j = a_j/N$, $\tilde{z}_k = b_k/N$ and $\tilde{s}_i = c_i/N$ for some integral a_j, b_k and c_i, $j = 1, \dots, t, k = 1, \dots, m, i = 1, \dots, n$.

In each column j we draw a_j vertical lines $j = 1, \dots, t$; similarly, we draw b_k horizontal lines in each row k, $k = 1, \dots, m$, and we create c_i distinct copies of each interval I_i.

The following question can be answered affirmatively (Th. 4.1): is it possible to color all the lines and copies of the intervals using $\frac{N}{2}$ colors (we can assume wlog that N is even), so that each line and each copy receive exactly one color, and so that each color class (that is, all lines and copies with a same color) constitutes a feasible solution? Given a set of lines and copies of a same color, a solution to GEOSC is found by setting the multiplicity of each column, row and interval equal to the number of corresponding lines or copies in the set.

Observe that if the answer is yes, for any interval, say I_i, and for any color, say blue, the number of blue lines (vertical and horizontal), stabbing I_i, plus the number of blue copies of I_i would be at least w_i.

We use the following coloring procedure. First we consider the vertical lines. Index these lines according to the index of the column they belong to in non-decreasing order. Then give line j color (j mod $\frac{N}{2}$), $j = 1, .., \sum_{h \in [1,t]} a_h$. Next consider the horizontal lines. At each row $j = 1, \dots, m$ index them consecutively from 1 to b_j and give line k color (k mod $\frac{N}{2}$), $k = 1, \dots, b_j$. At last, index the copies of each interval I_i as $1, \dots, c_i$ and give copy k color $(k + b_{\text{row}(i)})$ mod $\frac{N}{2}$, $k = 1, \dots, c_i$.

Finally, we choose a solution with the minimum value among those induced by the $N/2$ color classes.

Here is a more formal description of the algorithm, called $ALG3$.

The algorithm gets as input an instance I of GEOSC and a rational positive number ϵ.

Let $\hat{\mathbf{y}} \in \mathbf{R}^{\mathbf{t}}_+, \hat{\mathbf{z}} \in \mathbf{R}^{\mathbf{m}}_+, \hat{\mathbf{s}} \in \mathbf{R}^{\mathbf{n}}_+$, $\mathbf{a}, \mathbf{y}^{\mathbf{sol}} \in Z^t_+$, $\mathbf{b}, \mathbf{z}^{\mathbf{sol}} \in Z^m_+$, $\mathbf{c}, \mathbf{s}^{\mathbf{sol}} \in Z^n_+$, $\delta = \frac{\epsilon/4}{(n+t+m)(1+\epsilon/4)}$, $N = \frac{(n+t+m)(1+\epsilon/4)}{(\epsilon/4)^2}$, $V(i) \in Z^1_+, i = 1, \dots, N/2$.

Algorithm $ALG3$.

1. $y^{\text{sol}}_j \leftarrow 0, z^{\text{sol}}_k \leftarrow 0, s^{\text{sol}}_i \leftarrow 0$, *for all* $j = 1, \dots, t,\ k = 1, \dots, m$ *and* $i = 1, \dots, n$;
2. solve the LP-relaxation of (6–8) and store its optimal solution in $(\hat{\mathbf{y}}, \hat{\mathbf{z}}, \hat{\mathbf{s}})$;
3. if $\hat{y}_j < \delta$ *then* $\hat{y}_j \leftarrow 0$ *else* $\hat{y}_j \leftarrow (1+\epsilon/4)\hat{y}_j$,
if $\hat{z}_k < \delta$ *then* $\hat{z}_k \leftarrow 0$ *else* $\hat{z}_k \leftarrow (1+\epsilon/4)\hat{z}_k$,
if $\hat{s}_i < \delta$ *then* $\hat{s}_i \leftarrow 0$ *else* $\hat{s}_i \leftarrow (1+\epsilon/4)\hat{s}_i$, *for all* $j = 1, \dots, t,\ k = 1, \dots, m$ *and* $i = 1, \dots, n$;
4. for all j from 1 to t
for all h from $\sum_{p=1,\dots,j-1} a_p + 1$ *to* $\sum_{p=1,\dots,j} a_p$ *do* $V(h \text{ mod } N/2) \leftarrow V(h \text{ mod } N/2) + v_j$;

5. for all k from 1 to m *for all h from* 1 *to* b_k *do* $V(h \textit{ mod } N/2) \leftarrow V(h \textit{ mod } N/2) + u_k$;
6. for all i from 1 to n
for all h from $b_{\text{row}(i)}+1$ *to* $b_{\text{row}(i)}+c_p$ *do* $V(h \textit{ mod } N/2) \leftarrow V(h \textit{ mod } N/2)+p_i$;
7. $h^* \leftarrow \text{argmin}_{h=1,\dots,N/2}(V(h))$;
8. for all j from 1 to t *for all h from* $\sum_{p=1,\dots,j-1} a_p + 1$ *to* $\sum_{p=1,\dots,j} a_p$ *if* $(h \textit{ mod } N/2 = h^*)$ $y_j^{\text{sol}} \leftarrow y_j^{\text{sol}} + 1$;
9. for all k from 1 to m
for all h from $\sum_{p=1,\dots,k-1} b_p + 1$ *to* $\sum_{p=1,\dots,k} b_p$ *if* $(h \textit{ mod } N/2 = h^*)$ $z_k^{\text{sol}} \leftarrow z_k^{\text{sol}} + 1$;
10. for all i from 1 to n
for all h from $\sum_{p=1,\dots,i-1} c_p+1$ *to* $\sum_{p=1,\dots,i} c_p$ *if* $(h \textit{ mod } N/2 = h^*)$ $s_i^{\text{sol}} \leftarrow s_i^{\text{sol}}+1$.

Theorem 4.1. *ALG*3 *is a* $(2+\epsilon)$*-approximation algorithm for GEOSC.*

Proof. First we argue that *ALG*3 delivers a feasible solution to GEOSC. Consider some interval I_i. This interval is stabbed by columns of the set $l_i, \dots, r_i$, or alternatively, it is stabbed by $\sum_{j\in[l_i,r_i]} a_j$ vertical lines. If this number is equal to or larger than $w_i\frac{N}{2}$, then we are done, since the coloring approach described above ensures that interval I_i is stabbed by at least w_i vertical lines of each color.

Otherwise, as we show below, the set containing all the horizontal lines stabbing I_i and all the copies of I_i has at least $w_i\frac{N}{2}$ elements, which implies that there are at least w_i elements of each color in this set. Therefore, each color class implies a feasible solution to GEOSC on instance $\mathcal{I}$.

To show that the set of all the horizontal lines stabbing I_i and all the copies of I_i has at least $w_i\frac{N}{2}$ elements, we prove firstly that $(\tilde{\mathbf{y}}, \tilde{\mathbf{z}}, \tilde{\mathbf{s}})$ constitutes a feasible solution to the LP-relaxation of (6–8). Then, since $a_j = N\tilde{y}_j$, $b_k = N\tilde{z}_k$ and $c_i = N\tilde{s}_i$, for all j, k, i, we have:

$$b_{\text{row}(i)} + \sum_{j\in[l_i,r_i]} a_j + c_i \geq Nw_i.$$

Therefore, $\sum_{j\in[l_i,r_i]} a_j < \frac{Nw_i}{2}$ implies $b_{\text{row}(i)} + c_i \geq \frac{Nw_i}{2}$.

Now, let us consider $(\tilde{\mathbf{y}}, \tilde{\mathbf{z}}, \tilde{\mathbf{s}})$ and show that any of the constraints (7), for instance,

$$z_{\text{row}(i)} + \sum_{j\in[l_i,r_i]} y_j + s_i \geq w_i,$$

is satisfied by variables $\tilde{z}_{\text{row}(i)}, \tilde{y}_{l_i}, ..., \tilde{y}_{r_i}, \tilde{s}_i$.

For convenience of notation denote $\tilde{\mathbf{d}} = (\tilde{z}_{\text{row}(i)}, \tilde{y}_{l_i}, \dots, \tilde{y}_{r_i}, \tilde{s}_i)^T \in R^p$, $\hat{\mathbf{d}} = (\hat{z}_{\text{row}(i)}, \hat{y}_{l_i}, ..., \hat{y}_{r_i}, \hat{s}_i)^T \in R^p$ and $w = w_i$, where $(\hat{\mathbf{y}}, \hat{\mathbf{z}}, \hat{\mathbf{s}})$ is the optimal solution to the LP-relaxation and $p = l_i - r_i + 3 \leq n + t + m$. Notice, that we can assume that $w > 0$, since otherwise the inequality is trivially satisfied.

Clearly,

$$\sum_{j=1,\dots,p} \hat{d}_j \geq w. \tag{15}$$

Consider $\sum_{j=1,\dots,p} \tilde{d}_j$. Since $\tilde{d}_j = 0$ for all j such that $\hat{d}_j < \delta$, the sum can be written down as follows:

$$\sum_{j=1,\dots,p} \tilde{d}_j = \sum_{j:\hat{d}_j \geq \delta, 1 \leq j \leq p} \tilde{d}_j \geq \sum_{j:\hat{d}_j \geq \delta, 1 \leq j \leq p} (1+\epsilon/4)\hat{d}_j \geq$$

(due to (15))

$$\geq (1+\epsilon/4)\left(w - \sum_{j:\hat{d}_j < \delta, 1 \leq j \leq p} \hat{d}_j\right) \geq (1+\epsilon/4)(w - p\delta)$$
$$\geq (1+\epsilon/4)\left(w - \frac{\epsilon/4}{1+\epsilon/4}\right) \geq w.$$

Thus $(\tilde{\mathbf{y}}, \tilde{\mathbf{z}}, \tilde{\mathbf{s}})$ satisfies each of the constraints (7) and constitutes a feasible solution to the LP-relaxation.

Now we have $\frac{N}{2}$ feasible solutions, the total value of which can be bounded from above as follows:

$$\sum_{j=1}^{t} v_j a_j + \sum_{k=1}^{m} u_k b_k + \sum_{i=1}^{n} p_i c_i = N\left(\sum_{j=1}^{t} v_j \tilde{y}_j + \sum_{k=1}^{m} u_k \tilde{z}_k + \sum_{i=1}^{n} p_i \tilde{s}_i\right) \leq$$

(since $\hat{y}_j \geq \delta$ implies $\tilde{y}_j \leq (1+\epsilon/4)\hat{y}_j + 1/N$ and $\hat{y}_j < \delta$ implies $\tilde{y}_j = 0$, similarly for z_k, s_i, for all j, k, i)

$$\leq N\left((1+\epsilon/4)\left(\sum_{j=1}^{t} v_j \hat{y}_j + \sum_{k=1}^{m} u_k \hat{z}_k + \sum_{i=1}^{n} p_i \hat{s}_i\right)\right.$$
$$\left. + \frac{1}{N}\left(\sum_{j:\hat{y}_j \geq \delta, 1 \leq j \leq t} v_j + \sum_{k:\hat{z}_k \geq \delta, 1 \leq k \leq m} u_k + \sum_{i:\hat{s}_i \geq \delta, 1 \leq i \leq n} s_i\right)\right) \leq$$

(since $\hat{y}_j \geq \delta$ implies $\frac{1}{N} v_j \leq \frac{1}{N\delta}\hat{y}_j v_j = \frac{\epsilon}{4}\hat{y}_j v_j$, similarly for u_k , p_i, for all j, k, i)

$$\leq N\left((1+\epsilon/4)\left(\sum_{j=1}^{t} v_j\hat{y}_j + \sum_{k=1}^{m} u_k\hat{z}_k + \sum_{i=1}^{n} p_i\hat{s}_i\right)\right.$$
$$\left. +\epsilon/4\left(\sum_{j=1}^{t} v_j\hat{y}_j + \sum_{k=1}^{m} u_k\hat{z}_k + \sum_{i=1}^{n} p_i\hat{s}_i\right)\right) = N(1+\epsilon/2)v_{\mathrm{LP}}.$$

It is clear that among those $N/2$ feasible solutions there is at least one with value less or equal to $2(1+\epsilon/2)v_{\mathrm{LP}} = (2+\epsilon)v_{\mathrm{LP}}$. Since $ALG3$ chooses one with minimum value, it is a $(2+\epsilon)$-approximation algorithm. □

5. A non-approximability result

In this section we prove that GEOSC is hard to approximate arbitrarily closely in polynomial time (unless $\mathcal{P} = \mathcal{NP}$). We assume familiarity with some of the issues in approximation and complexity, see for instance [3] or [13].

Let us first describe a graph-theoretical interpretation of a special case of GEOSC. The special case of GEOSC of interest in this section is the case where all weights and capacities are 1 and there are at most 2 intervals per row. Then, a solution of GEOSC will have all s-variables equal to 0 and hence consist of selected rows and columns. In fact this special case of GEOSC can be formulated in a graph-theoretic context as follows. Construct a graph in which there is a node for each interval and two nodes are connected if they share a column (blue edge), or if they share a row (red edge). Thus, the graph constructed is the edge union of an interval graph and a matching. Notice that a monochromatic maximal clique in such a graph corresponds to a row or a column in GEOSC. In fact, finding a monochromatic clique cover of minimum size is exactly GEOSC.

Theorem 5.1. *GEOSC does not have a PTAS unless $\mathcal{P} = \mathcal{NP}$.*

Proof. We prove the theorem by presenting an L-reduction [13] from MAX 3-SAT-3 to GEOSC2. The result in [2] then establishes the theorem.

MAX 3-SAT-3 is the version of satisfiability in which each clause has at most 3 variables and each variable occurs at most 3 times. It is shown to be MAX SNP-hard in [3] (see also [13]).

Recall that $C = \{C_1, C_2, \ldots, C_r\}$ is a set consisting of r disjunctive clauses, each containing at most 3 literals. Let $x_1, x_2, \ldots, x_n$ denote the variables in the r clauses and, for each $i = 1, \ldots, n$, let $m(i)$ denote the number of occurrences of variable x_i (either as literal x_i or as literal $\bar{x}_i$). Arbitrarily index the occurrences of variable x_i as occurrence $1, 2, \ldots, m(i)$. Notice that without loss of generality we can assume that each variable occurs at least twice in C, thus we have $2 \leq m(i) \leq 3$ for all i and that $\sum_i m(i) \leq 3r$. Moreover, we will also assume (again wlog) that each variable occurs at least once unnegated and at least once negated in C.

We now construct an instance of GEOSC, that is a graph $G = (V, E)$ which is the edge union of an interval graph and a matching. Let I denote an instance of MAX 3-SAT-3 and $R(I)$ the corresponding instance of GEOSC with corresponding optimal values $OPT(I)$ and $OPT(R(I))$.

For each variable x_i in I, $i = 1, \ldots, n$, we have a subgraph $H1_i = (V1_i, E1_i)$ in $R(I)$, where $V1_i = \{v_{ij} | \; j = 0, \ldots, 5\}$ and $E1_i = \{\{v_{ij}, v_{i,j+1}\} | \; j = 0, \ldots, 5\}$ (indices modulo 6). So for each variable x_i in I we have a cycle consisting of 6 nodes in $R(I)$. We refer to the edges $\{v_{i0}, v_{i1}\}, \{v_{i2}, v_{i3}\}$ and $\{v_{i4}, v_{i5}\}$ as T edges, and to the edges $\{v_{i1}, v_{i2}\}, \{v_{i3}, v_{i4}\}$ and $\{v_{i5}, v_{i6}\}$ as F edges. Thus the cycle consists of alternating T and F edges.

For each clause C_j in I, $j = 1, \ldots, r$, we have a subgraph $H2_j = (V2_j, E2_j)$ in $R(I)$ depending upon the cardinality of C_j, as depicted in Figure 4.

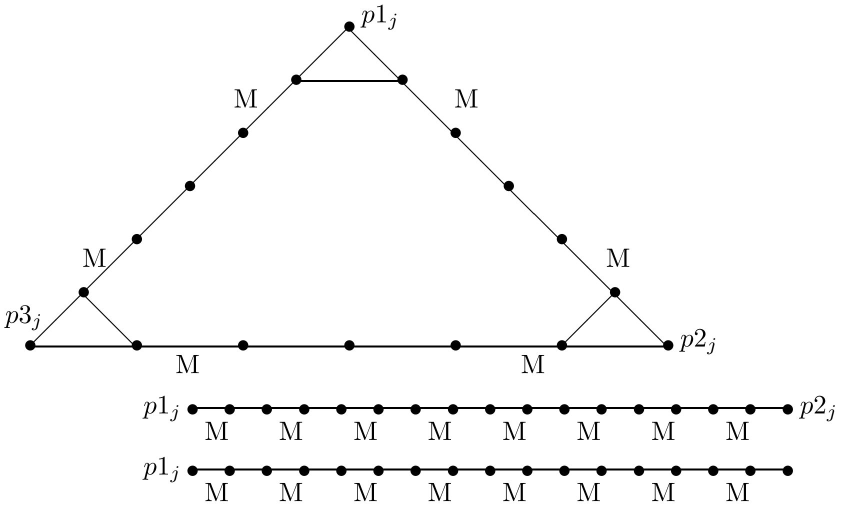

FIGURE 4. The subgraph $H2_j$ when $|C_j| = 3$ (upper figure), when $|C_j| = 2$ (middle figure: P_{17}) and when $|C_j| = 1$ (lower figure: P_{17}).

When no ambiguity is likely to occur, we refer to the nodes $p1_j$, $p2_j$ and $p3_j$ as p-nodes. Notice that each of the subgraphs has the property that a clique cover of its vertices has cardinality at least 9, whereas if 1, 2 or even 3 p-nodes need not be covered, one needs at least 8 cliques.

To connect the subgraphs introduced sofar in $R(I)$, consider some clause C_j, and consider the first variable occurring in this clause C_j, say x_i. Let this be the q-th occurrence of this variable x_i in C, $q \in \{1, \ldots, m(i)\}$. If the variable x_i occurs as literal x_i add the edges $\{p1_j, v_{i,2q-2}\}$ and $\{p1_j, v_{i,2q-1}\}$ to E. (The node $p1_j$ will then be referred to as a true p-node.) If the variable x_i occurs as literal $\bar{x}_i$ add the edges $\{p1_j, v_{i,2q-1}\}$ and $\{p1_j, v_{i,2q}\}$ to E. (The node $p1_j$ will then be referred to as a false p-node.) Consider now the second (third) variable occurring in C_j, say x_l, and let this be the q-th occurrence of this variable x_l in C, $q \in \{1, \ldots, m(l)\}$. If the variable x_l occurs as literal x_l add the edges $\{p2_j, v_{i,2q-2}\}$ and $\{p2_j, v_{i,2q-1}\}$ ($\{p3_j, v_{i,2q-2}\}$ and $\{p3_j, v_{i,2q-1}\}$) to E. If the variable x_l occurs as literal $\bar{x}_l$ add the edges $\{p2_j, v_{i,2q-1}\}$ and $\{p2_j, v_{i,2q}\}$ ($\{p3_j, v_{i,2q-1}\}$ and $\{p3_j, v_{i,2q}\}$) to E. This is done for all clauses C_j, $j = 1, \ldots, r$. See Figure 5 for a graphic representation of the way in which the subgraphs $\mathrm{H1}_i$ are connected to the p-nodes of subgraphs $\mathrm{H2}_j$.

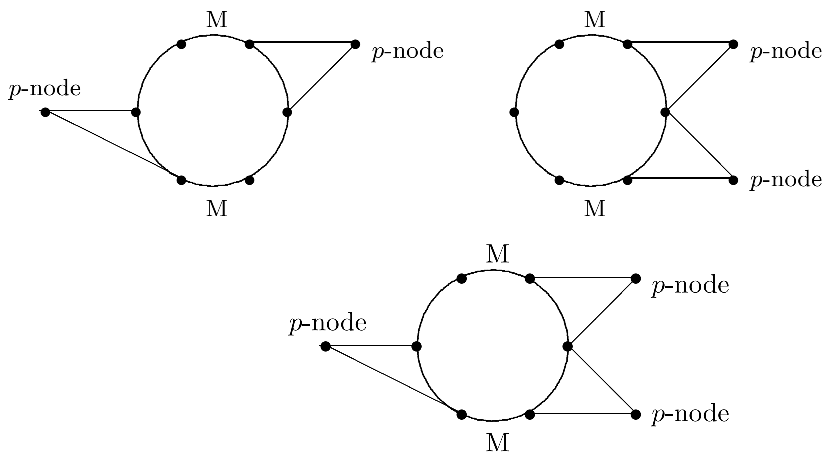

FIGURE 5. The subgraph $\mathrm{H1}_i$ and its connections when $m(i) = 2$ (upper figure, 2 possibilities) and when $m(i) = 3$ (lower figure).

Now the graph $G = (V, E)$ is completely specified.

We now exhibit a matching M in G. M consists of two parts: edges in $\cup_i H1_i$ and edges in $\cup_j H2_j$. For the first part we take the edges that are marked with an "M" in Figure 5, for the second part we take the edges that are marked with an "M" in Figure 4.

Obviously, M is indeed a matching. Also, one can verify that the remaining edges in G form an interval graph (notice that each $H2_j$ disconnects).

In the remaining part of the proof, with the word "clique cover" a monochromatic clique cover is meant, that is a clique cover with the edges of each clique having the same color.

In order to show that this reduction is an L-reduction, consider the following. Observe that $v \equiv OPT(I) \geq \frac{1}{2}r$. (Indeed, by considering the assignment: all variables true, and: all variables false, it follows that each clause is true in at least in one of both assignments.) We have:

$$OPT(R(I)) \leq 3n + 9r = \frac{9}{2}r + 9r \leq 27v = 27 \cdot OPT(I).$$

The first inequality follows from the fact that 3 cliques can be selected from each $H1_i$, $i = 1, \ldots, n$ to cover its nodes, and 9 cliques can be selected from each subgraph $H2_j$, $j = 1, \ldots, r$ to cover its nodes. Since $n \leq \frac{3}{2}r$, the inequality follows.

Consider now an arbitrary solution to $R(I)$, that is any (monochromatic) clique cover s in G with size $c(s)$. We will map this solution s using an intermediate solution s' to a solution of MAX 3-SAT-3, called $S(s)$. To do this we need the following definition. A clique cover s in G is called *consistent* iff for each $i = 1, \dots, n$, the following property holds: either the nodes from $V1_i$ are in cliques that contain only T edges and are maximal, or the nodes from $V1_i$ are in cliques that contain only F edges and are maximal.

Now we state a procedure which takes as input a clique cover s. The output of the procedure is a consistent clique cover called s' with the property that $c(s') \leq c(s)$.

Procedure. Consider s. For $i = 1, \dots, n$, consider $V1_i$. If $V1_i$ fails the property because a clique in s is not maximal, this is easily fixed by enlarging one or more cliques by adding the appropriate p-node (and perhaps adjusting the clique cover in an obvious manner). If it fails the property because it has a T edge as well as an F edge from $H1_i$ in a clique we do the following. Notice that in this case at least 4 cliques are used for the nodes in $H1_i$. Consider the p-nodes that are contained in cliques used to cover nodes of $H1_i$. Now, if among those p-nodes there are at least 2 false p-nodes take the 3 maximal F-cliques, else take the 3 maximal T-cliques. And, if necessary, take as a single clique the p-node not covered by these maximal T or F cliques (there can be at most 1, so $c(s') \leq c(s)$).

End of Procedure

After applying this procedure to any clique cover s in G, a consistent solution s' is delivered with $c(s') \leq c(s)$. Since s' is consistent, it is now straightforward to identify the corresponding solution $S(s)$ in MAX 3-SAT-3: simply set variable x_i, $i = 1, \dots, n$ true if all T-edges in subgraph $H1_i$ are in the clique cover s', else set x_i false. How many clauses in I are satisfied by this truth assignment? Observe that the construction of G implies that if for some consistent clique cover s a p-node from some $H2_j$ is contained in a clique that covers also nodes from some $H1_i$ we need 8 cliques to cover $H2_j$. Let there be l triangles for which at least one p-node is gone in this way. This implies that l clauses in I are satisfied by this truth assignment.

Again, let $v = OPT(I)$, and let $c(S(s)) = l$. The following (in)equalities are true:

- $c(s) \geq c(s')$ (by construction);
- $c(s') = 3n + 8l + 9(r - l) = 3n + 9r - l$ (by construction), and
- $OPT(R(I)) \leq 3n+8v+9(r-v) = 3n+9r-v$ (consider the truth assignment that is optimum for I; evidently, we can exhibit in $R(I))$ a corresponding clique cover of size $3n + 9r - v$).

Thus

$$c(s) - OPT(R(I)) \geq 3n + 9r - l - (3n + 9r - v) = v - l = OPT(I) - c(S(s)),$$

which finishes the proof. □

Notice that the theorem remains true when the number of intervals that share a column is bounded by 3. When formulating Theorem 5.1 in graph theoretic terms we get:

Corollary 5.2. *Finding a monochromatic clique cover in a graph that is the edge-union of an interval graph with bounded degree and a matching is MAX SNP complete.*

References

[1] J. Aerts and E.J. Marinissen, Scan chain design for test time reduction in core-based ICs, in *Proc. of the International Test Conference.* Washington DC (1998).

[2] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, Proof verification and hardness of approximation problems, in *Proc. of the 33rd IEEE Symposium on the Foundations of Computer Science* (1992) 14-23.

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

[4] A. Bar–Noy, S. Guha, J. Naor and B. Schieber, Approximating the Throughput of Multiple Machines in Real-Time Scheduling. *SIAM J. Comput.* **31** (2001) 331-352.

[5] A. Bar–Noy, R. Bar–Yehuda, A. Freund, J. Naor and B. Schieber, A Unified Approach to Approximating Resource Allocation and Scheduling. *J. ACM* **48** (2001) 1069-1090.

[6] P. Berman and B. DasGupta, Multi-phase Algorithms for Throughput Maximization for Real-Time Scheduling. *J. Combin. Optim.* **4** (2000) 307-323.

[7] T. Erlebach and F.C.R. Spieksma, Simple algorithms for a weighted interval selection problem, in *Proc. of the 11th Annual International Symposium on Algorithms and Computation (ISAAC '00). Lecture Notes in Comput. Sci.* **1969** (2000) 228-240 (see also Report M00-01, Maastricht University).

[8] N. Garg, V.V. Vazirani and M. Yannakakis, Primal-Dual Approximation Algorithms for Integral Flow and Multicut in Trees. *Algorithmica* **18** (1997) 3-20.

[9] M.X. Goemans and D.P. Williamson, *The primal-dual method for approximation algorithms and its application to network design problems*, Chap. 4 of *Approximation algorithms for NP-hard problems*, edited by D.S. Hochbaum. PWC Publishing Company, Boston (1997).

[10] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs.* Academic Press, San Diego, California (1980).

[11] D.S. Hochbaum, *Approximation algorithms for NP-hard problems.* PWC Publishing Company, Boston (1997).

[12] D.S. Hochbaum, *Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set and Related Problems*, Chap. 3 of *Approximation algorithms for NP-hard problems*, edited by D.S. Hochbaum. PWC Publishing Company, Boston (1997).

[13] C.H. Papadimitriou, *Computational Complexity.* Addison-Wesley, Reading, Massachussets (1994).

[14] F.C.R. Spieksma, On the approximability of an interval scheduling problem. *J. Schedul.* **2** (1999) 215-227.

[15] D.P. Williamson, Course notes *Primal-Dual methods*, available at `http://www.research.ibm.com/people/w/williamson/#Notes`