Resource Allocation for Covering Time Varying Demands

Venkatesan T. Chakaravarthy[1], Amit Kumar[2], Sambuddha Roy[1], and Yogish Sabharwal[1]

[1] IBM Research - India, New Delhi
[2] Indian Institute of Technology, New Delhi
{vechakra, sambuddha, ysabharwal}@in.ibm.com; amitk@cse.iitd.ac.in

Abstract. We consider the problem of allocating resources to satisfy demand requirements varying over time. The input specifies a demand for each timeslot. Each resource is specified by a start-time, end-time, an associated cost and a capacity. A feasible solution is a multiset of resources such that at any point of time, the sum of the capacities offered by the resources is at least the demand requirement at that point of time. The goal is to minimize the total cost of the resources included in the solution. This problem arises naturally in many scenarios such as workforce management, sensor networks, cloud computing, energy management and distributed computing. We study this problem under the partial cover setting and the zero-one setting. In the former scenario, the input also includes a number k and the goal is to choose a minimum cost solution that satisfies the demand requirements of at least k timeslots. For this problem, we present a 16-approximation algorithm; we show that there exist "well-structured" near-optimal solutions and that such a solution can be found in polynomial time via dynamic programming. In the zero-one setting, a feasible solution is allowed to pick at most one copy of any resource. For this case, we present a 4-approximation algorithm; our algorithm uses a novel LP relaxation involving flow-cover inequalities.

1 Introduction

We consider the problem of allocating resources to satisfy demand requirements varying over time. We assume that time is uniformly divided into discrete timeslots. The input specifies a demand for each timeslot. Each resource is specified by its start-time, end-time, the capacity of the resource that it offers during this interval and its associated cost. A feasible solution is a set of resources satisfying the constraint that at any timeslot, the total sum of the capacities offered by the resources is at least the demand required at that timeslot, i.e. the selected resources must cover the demands.

The above problem is motivated by applications in workforce management. For instance, consider the problem of scheduling employees in call centers. Typically, in these settings, we have a reasonably accurate forecast of demands which

will arrive at any timeslot and for each employee, we know the time interval in which they can work. Employees have proficiencies (or capacities) determining the number of calls they can attend on an average in a timeslot. They also have cost or wages associated with them. The goal is to choose a minimum cost set of employees whose schedule meets the demand at all the timeslots. The problem framework is quite general and captures many other situations arising in sensor networks, cloud computing, energy management and distributed computing (see [12, 7, 4]).

Motivated by such applications, Chakaravarthy et al. [4] studied the above problem. They considered the version (called MultiResAll) wherein the solution is allowed to pick multiple copies of a resource by paying as many units of the associated cost. In the context of workforce management, employees can be classified based on their proficiency and the shifts they work in. Choosing multiple units of a single resource corresponds to selecting multiple employees of the same class. They presented a 4-approximation algorithm for the MultiResAll problem. In this paper, we consider two generalizations of the MultiResAll problem.

The first variant (called partial MultiResAll) considers the partial covering scenario, wherein the input also specifies a number k and a feasible solution is only required to satisfy the demand for at least k timeslots. In the workforce management setting, this corresponds to the concept of service level agreements (SLA's), which stipulates that the requirements of a large fraction of timeslots are satisfied.

The second variant considers a more natural bounded availability setting, wherein for each resource, the input specifies a bound on the maximum number of copies that can be selected. In this paper, we shall study the version (called (0,1)-ResAll) in which a solution can pick a resource at most once. Note that the bounded setting can be reduced to the zero-one setting, by duplicating the resources (albeit at an increased running time). We now formally define these problems.

Problem Definitions: We first define the MultiResAll problem. We consider time to be uniformly divided into discrete units ranging from 1 to T. We refer to each integer t in the range $[1, T]$ as a *timeslot*. The input specifies a *demand profile* $d : [1, T] \rightarrow \mathbb{Z}$; here $d(t)$ (also denoted by d_t) is the *demand* at timeslot t for $t \in [1, T]$. The input further consists of a set of *resources* (or *intervals*) $\mathcal{I}$. Each resource $i \in \mathcal{I}$ is specified by an interval $I_i = [s_i, e_i]$, where s_i and e_i are the *start-time* and the *end-time* of the resource i; we assume that s_i and e_i are integers in the range $[1, T]$. The resource i is also associated with a *capacity* (or *height*) h_i and a cost c_i. We say that the resource i is *active* at a timeslot t, if $t \in I_i$. For a timeslot t, let $A(t)$ denote the set of all resources active at timeslot t.

Let S be a multiset of resources. For a resource $i \in S$, let $f_S(i)$ denote the number of times i appears in S. The multiset S is said to *cover* a timeslot t, if the sum of the capacities of the resources from S active at the timeslot t is at least d_t, i.e., $\sum_{i \in A(t)} f_S(i) \cdot h_i \geq d_t$. The multiset S is said to be a *full cover*, if

S covers all the timeslots $t \in [1, T]$. The cost of the multiset S is defined to be $\texttt{cost}(S) = \sum_{i \in S} f_S(i) \cdot c_i$ (where the summation is over distinct intervals in S).

- MultiResAll Problem: In this problem, the goal is to find a full cover having the minimum cost.
- Partial MultiResAll Problem: In this problem, the input also includes an integer k. A multiset S is said to be a *k-partial cover*, if S covers at least k timeslots. The goal is to find a minimum cost k-partial cover.
- (0,1)-ResAll Problem: In this problem, a feasible solution is allowed to pick each resource at most once. The goal is to find a minimum cost full cover.

Prior Work: As mentioned earlier, a 4-approximation algorithm for the MultiResAll problem was presented in [4]. The algorithm was based on the primal dual approach.

The special case of the (0,1)-ResAll problem wherein there is only a single timeslot ($T = 1$) corresponds to the classical minimum knapsack cover problem (MKP). It is well known that the problem is NP-hard and that it admits a FPTAS [11]. En route to proving more general results, Carnes and Shmoys [3] gave a 2-approximation algorithm based on the primal dual approach.

The (0,1)-ResAll problem is related to the well-studied unsplittable flow problem on line (UFP). In the (0,1)-ResAll problem, we need to select a set of intervals covering a given demand profile. In the UFP problem, the goal is to select a set of intervals that can be packed within a given bandwidth profile. Thus, while UFP is a packing problem, (0,1)-ResAll is its analogous covering version. Bansal et al.[1] presented a quasi-PTAS for the restricted case of the UFP problem, when all the capacities and demands are quasi-polynomial. In a recent breakthrough, Bonsma et al. [2] designed a polynomial time $(7 + \epsilon)$-approximation algorithm. Under the so called "no-bottleneck assumption", Chekuri et al. [6] presented a $(2 + \epsilon)$-approximation algorithm. Prior work have addressed partial versions of many other covering problems, such as vertex cover [8], multicut [10] and spanning trees [9].

Chakrabarty et al.[5] studied the more general version of the (0,1)-ResAll problem called column restricted covering integer programs. Their framework yields a 40-approximation algorithm for the (0,1)-ResAll problem. Independent of our work, Korula obtained a 4-approximation algorithm for the (0,1)-ResAll problem (see [5]). To the best of our knowledge, partial versions of the MultiResAll and (0,1)-ResAll problems have not been studied.

Our Results and Techniques: The main results of this paper are as follows:

- We present a 16-approximation algorithm for the partial MultiResAll problem.
- For the special case, of the partial MultiResAll problem, where the input demands are uniform, our approach can be fine tuned to yield an improved 2-approximation algorithm. The details of this result will be presented in the full version of the paper.
- Our next result is a 4-approximation algorithm for the (0,1)-ResAll problem. This result generalizes the known 4-approximation algorithm for the MultiResAll problem [4].

Our techniques do not extend to the partial version of the (0,1)-ResAll problem. Obtaining a constant factor approximation for this problem remains open.

The 4-approximation algorithm for the MultiResAll problem [4] is based on the primal-dual approach applied to a natural LP formulation. In the case of (0,1)-ResAll, the corresponding natural LP has an unbounded integrality gap. We circumvent the issue by strengthening the LP with "flow-cover inequalities", à la the approach that Carnes and Shmoys [3] adopt for the MKP problem. An additional advantage of the new LP is that it admits a simpler primal-dual analysis, while matching the approximation factor of the MultiResAll problem.

We now discuss two approaches that have been widely successful for solving partial covering problems (such as vertex cover) and briefly outline the difficulties in applying either of them to our problem. One of the approaches typically starts with the natural LP relaxation, which may have unbounded integrality gap. It turns out that that by augmenting it with some extra information, the integrality gap can be brought down to a constant (see for instance: partial vertex cover [8], k-MST [9]). In the MultiResAll problem, it is not clear how to construct such a strengthened LP.

A second approach goes via Langrangian relaxations [13] and has been successful applied to problems such as such as multicut [10] and k-MST [9]. Under this paradigm, one first designs a primal-dual approximation algorithm for the prize collecting version with certain additional properties, which is then used to solve the partial covering problem. We do not know how to design such an algorithm for the prize collecting version of the MultiResAll problem; however, we note that our techniques discussed below yield a constant factor approximation for prize collecting version.

Our algorithm for the partial MultiResAll problem builds on two main insights regarding the MultiResAll problem. The first insight is that there always exists a near-optimal solution satisfying the following simple structural property: for every timeslot, there exists an interval in the solution whose copies are enough to cover the demand at the timeslot (without the aid of other intervals). The second main insight is that among all the solutions satisfying the above property, the optimal one can be found in polynomial time using dynamic programming. Combining the two ideas, we get an alternative constant factor approximation algorithm for the MultiResAll problem. We show that both the steps extend to the partial cover setting and thereby we get a constant factor approximation algorithm for the partial MultiResAll problem. It is interesting to note that the recent $(7+\epsilon)$-approximation algorithm for the UFP problem [2] also uses a similar strategy of first showing the existence of solutions with special properties and then invoking dynamic programming.

2 Approximating the Partial MultiResAll Problem

In this section, we present a 16-approximation algorithm for the partial MultiResAll problem. The notion of *single resource assignment* (SRA) covers, defined next, is useful for this purpose.

Single Resource Assignment (SRA) Solution: Let S be a multiset of resources. The multiset S is said to cover a timeslot $t \in [1, T]$ in an *SRA fashion*, if we can find a single resource $i \in S$ such that the demand at t can be covered by the copies of i alone i.e., there exists $i \in S$ such that $f_S(i) \cdot h_i \geq d_t$. The multiset S is said to be an *SRA full cover*, if S covers all the timeslots in an SRA fashion. A k-partial cover S is said to be an *SRA k-partial cover*, if S covers at least k timeslots in an SRA fashion.

Let $\texttt{Opt}$ and $\texttt{pOpt}$ denote the optimal full cover and k-partial cover, respectively. Similarly, let $\widehat{\texttt{Opt}}$ and $\widehat{\texttt{pOpt}}$ denote the optimal SRA full cover and SRA k-partial cover, respectively. Our constant factor approximation algorithm is based on the following two theorems and the subsequent corollaries.

Theorem 1. *There exists an SRA full cover $\widehat{S}$ such that $\texttt{cost}(\widehat{S}) \leq 16 \cdot \texttt{cost}(\texttt{Opt})$.*

Theorem 2. *Given an instance of the* MultiResAll *problem, we can find the optimal SRA full cover in polynomial time.*

Corollary 1. *There exists an SRA k-partial cover $\widehat{S}$ such that $\texttt{cost}(\widehat{S}) \leq 16 \cdot \texttt{cost}(\texttt{pOpt})$.*

Corollary 2. *Given an instance of the partial* MultiResAll *problem, we can find the optimal SRA k-partial cover $\widehat{\texttt{pOpt}}$ in polynomial time.*

We note that an alternative proof of Theorem 2 is given in [5]. However, our proof is significantly simpler. By combining the two corollaries, we obtain a constant factor approximation algorithm for the partial MultiResAll problem, as follows. Our overall algorithm simply runs the algorithm claimed in Corollary 2 and outputs the SRA k-partial cover $\widehat{\texttt{pOpt}}$. Corollary 1 implies that $\texttt{cost}(\widehat{\texttt{pOpt}}) \leq \texttt{cost}(\widehat{S}) \leq 16 \cdot \texttt{cost}(\texttt{pOpt})$. We have established the main result of the section: the partial MultiResAll problem can be approximated within a factor of 16.

2.1 Existence of Good SRA covers

In this section, we first prove Theorem 1 and then derive Corollary 1. In proving Theorem 1, even though it suffices to show only the existence of $\widehat{S}$, we shall in fact present a polynomial time algorithm for producing the claimed SRA full cover. The algorithm goes via the primal-dual approach.

Let us first consider a naive LP formulation. We associate a variable x_i with each $i \in \mathcal{I}$, which specifies the number of copies of the resource i in the solution. For each timeslot t, we will have a constraint that enforces the coverage requirement for the timeslot ($\sum_{i \in A(t)} h_i \cdot x_i \geq d_t$). However, it is not hard to

show that this LP has an unbounded integrality gap. For instance, consider a single timeslot t (i.e., $T = 1$) with $d_t = 1$ and let there be a single resource i of cost $c_i = 1$, with height $h_i = B$ (for some B). Then, the optimal integral solution will have cost 1. On the other hand, the LP can set $x_i = 1/B$ and get a cost of $1/B$. One can easily handle the issue via adjusting heights as follows. For a resource $i \in I$ and a timeslot t where i is active, denote $\widetilde{h}_i(t) = \min\{h_i, d_t\}$. We obtain a strengthened LP formulation by utilizing the adjusted heights $\widetilde{h}$.

$$\begin{aligned} \min \qquad & \sum_{i \in \mathcal{I}} x_i \cdot c_i \\ \sum_{i \in A(t)} \widetilde{h}(i,t) \cdot x_i \geq d_t \qquad & \text{for all time-slots } t \\ x_i \geq 0 \qquad & \text{for all } i \in \mathcal{I} \end{aligned}$$

Each integral primal feasible solution $\mathbf{x}$ corresponds to a full cover multiset S; similarly, every full cover S corresponds to a primal integral feasible solution $\mathbf{x}$, given by $\mathbf{x}_i = f_S(i)$. In our discussion, we shall use the viewpoint of multisets.

The dual of the LP is shown next. It has a variable y_t for each timeslot t.

$$\begin{aligned} \max \qquad & \sum_{t \in [1,T]} y_t \cdot d_t \\ \sum_{i \in A(t)} y_t \cdot \widetilde{h}(i,t) \leq c_i \qquad & \text{for all } i \in \mathcal{I} \\ y_t \geq 0 \qquad & \text{for all } t \end{aligned}$$

We will produce a primal integral solution $\widehat{S}$ and a dual feasible solution $\mathbf{y}$ such that the complementary slackness conditions are satisfied approximately: (i) Primal slackness conditions: for any $i \in \mathcal{I}$, if $f_{\widehat{S}}(i) > 0$ then the corresponding dual constraint is tight:

$$\sum_{t : i \in A(t)} y_t \cdot \widetilde{h}(i,t) = c_i. \tag{1}$$

(ii) Approximate dual slackness conditions: for any timeslot t, if $y_t > 0$ then the corresponding primal constraint is tight within a factor of 16:

$$\sum_{i : i \in A(t)} f_{\widehat{S}}(i) \cdot \widetilde{h}(i,t) \leq 16 \cdot d_t. \tag{2}$$

Using well-known arguments via complementary slackness and weak duality, we can show that the above conditions imply $\texttt{cost}(\widehat{S}) \leq 16 \cdot \texttt{cost}(\texttt{Opt})$. This would prove Theorem 1. We now present the algorithm meeting the above requirements.

The algorithm runs in three phases, a construction phase, a deletion phase and a doubling phase. In the construction phase, we shall construct a dual feasible solution $\mathbf{y}$ and also obtain an SRA full cover $\mathbf{A}$. In the deletion phase, we delete certain carefully chosen intervals from $\mathbf{A}$ and obtain a multiset S'. The

set S' may not be a full cover, but it will satisfy at least half the demand at each timeslot in an SRA fashion. In the final doubling phase, we simply double the number of copies each interval in S' and obtain an SRA full cover $\mathbf{B}$. We shall argue that the primal feasible solution $\widehat{S} = \mathbf{B}$ and the dual feasible solution $\mathbf{y}$ satisfy the slackness conditions (1) and (2).

Construction Phase: We employ an greedy procedure for constructing a dual feasible solution $\mathbf{y}$ having high objective value (i.e., $\sum_t \mathbf{y}_t \cdot d_t$). The algorithm runs in multiple iterations, wherein each iteration we greedily choose the dual variable y_t that potentially gives us the maximum benefit; thus, we choose the timeslot t having the maximum demand d_t. The algorithm is formally described next. To start with all the timeslots are marked as *alive*. Consider any iteration $j \geq 0$. Let t_j be the timeslot having the highest demand d_{t_j} among all the currently alive timeslots. We raise the dual variable y_{t_j} to the maximum possible value until some dual constraint, say corresponding to an interval i_j, becomes tight. We add $\lceil d_{t_j}/\widetilde{h}(i_j, t_j) \rceil$ copies of i_j in our solution $\mathbf{A}$. We call t_j the *main* timeslot of i_j. The interval i_j then marks as *dead* all the timeslots in its span that are currently alive. We say that the main timeslot t_j along with all the other timeslots marked dead by i_j are *associated* with i_j. We proceed in this manner until our solution $\mathbf{A}$ covers all the timeslots. This completes the construction of $\mathbf{y}$ and $\mathbf{A}$.

Each interval $i \in \mathbf{A}$ covers all the timeslots associated with it in an SRA fashion and hence, $\mathbf{A}$ is an SRA full cover. Let ℓ be the number of main timeslots, given as $t_1, t_2, \ldots, t_\ell$. For each interval i chosen by $\mathbf{A}$, let the collection of copies of i be called the *bundle* of i. For an interval i and a timeslot t, let $\widetilde{H}(i,t)$ denote the *adjusted bundle height at* t, given by $\widetilde{H}(i,t) = f_{\mathbf{A}}(i) \cdot \widetilde{h}(i,t)$. The solution $\mathbf{A}$ satisfies the following property.

Lemma 1. *Let i be any interval chosen in $\mathbf{A}$ and t^* be any main timeslot within the span of i. Then, $\widetilde{H}(i,t^*) \leq 2d_{t^*}$.*

Proof. Let t be the main timeslot associated with i. By definition, $\widetilde{H}(i,t^*) = \lceil d_t/h_i \rceil \cdot \min\{h_i, d^*_t\}$. We see that $d_t \leq d_{t^*}$ (for otherwise t^* cannot a main timeslot). A simple calculation yields the lemma. □

Deletion Phase: In the deletion phase, we delete certain redundant intervals from $\mathbf{A}$ and obtain a multiset S'. Consider the main timeslots in an arbitrary order, say the original order $t_1, t_2, \ldots, t_\ell$. Let t^* be any main timeslot in this ordering. Lemma 1 shows that for any $i \in A(t^*)$, $\widetilde{H}(i,t^*) \leq 2d_{t^*}$. We partition the set $A(t^*)$ of intervals active at t^* in a geometric fashion into a set of *bands* as follows. Let $m = \lceil \log(2d_{t^*}) \rceil$. For $1 \leq j \leq m$, define the *band* B_j to be

$$B_j = \{i \in \mathbf{A} \cap A(t^*) \ : \ 2d_{t^*}/2^j < \widetilde{H}(i,t^*) \leq 2d_{t^*}/2^{j-1}\}.$$

For each band B_j, let i_1 be the interval in B_j extending farthest to the left (i.e., having the minimum start-time); similarly, let i_2 be the interval active at t^* extending farthest to the right (i.e., having the maximum end-time). We shall

retain these two interval (bundles) and delete all the other interval (bundles) of B_j from $\mathbf{A}$. Let the constructed multiset be S'.

We note that S' need not be an SRA full cover (it may not even be a full cover). However, we can make the following claim about S' that at every timeslot t, the solution S' covers at least half the demand at t in an SRA fashion. The lemma is proved as follows. If i, the interval covering t in $\mathbf{A}$, got deleted while considering a main timeslot t^*, then either the leftmost or the rightmost interval appearing in the same band as i cover at least half the demand at t.

Lemma 2. *For any timeslot t, there exists an interval $i' \in S'$ such that i' is active at t and $f_{S'}(i')h_{i'} \geq d_t/2$.*

Proof. Let i be the interval associated with t. Clearly, if i is included in S', then $f_{S'}(i)h_i \geq d_t$ and we can take $i' = i$. Now suppose i got deleted while considering a main timeslot t^*. With respect to t^*, let B_j be the band to which i belongs and let i_1 and i_2 be the leftmost and rightmost intervals that were retained for this band. Then, at least one of these two intervals (say i_1) is also active at t. Since i_1 is in the same band as i, $\widetilde{H}(i_1, t^*) \geq \widetilde{H}(i, t^*)/2$. First consider the case where $h_i \leq d_{t^*}$. In this case, $\widetilde{H}(i, t^*) = f_{S'}(i) \cdot h_i \geq d_t$. It follows that $f_{S'}(i_1) \cdot h_{i_1} \geq \widetilde{H}(i_1, t^*) \geq d_t/2$. Now suppose $h_i \geq d_{t^*}$. This implies that $\widetilde{H}(i, t^*) \geq d_{t^*}$ and so, i belongs to the band B_1 (corresponding to the range $[d_{t^*}, 2d_{t^*}]$). Since i_1 also belongs to the same band as i, $\widetilde{H}(i_1, t^*) \geq d_{t^*}$, which implies that $f_{S'}(i_1) \cdot h_{i_1} \geq d_{t^*}$. Notice that $d_{t^*} \geq d_{t'}$, where t' is main timeslot of i; otherwise, t' would have been considered earlier than t^* and so, i would have marked t^* as dead and so, t^* could not be a main timeslot. Notice that $d_{t'} \geq d_t$; otherwise t' cannot be the main timeslot of i. It follows that $d_{t^*} \geq d_t$. This implies that that $f_{S'}(i_1) \cdot h_{i_1} \geq d_t$. Thus, we can take $i' = i_1$. □

Doubling Phase: In this phase, we transform S' into an SRA full cover $\mathbf{B}$, by simply doubling the number of copies of every interval in S'. Lemma 2 implies that the solution $\mathbf{B}$ is an SRA full cover.

Complementary Slackness Conditions: We now argue that the complementary slackness conditions (1) and (2) are satisfied by the solutions $\mathbf{y}$ and $\mathbf{B}$. First notice that the primal slackness conditions (1) are automatically satisfied after the construction phase; in the latter two phases, we do not change the dual solution $\mathbf{y}$. Let us now focus on the dual slackness condition (2). The dual variable $y_t > 0$ only for main timeslots t. The lemma below proves the slackness condition for all these main timeslots.

Lemma 3. *For any main timeslot t^*,*

$$\sum_{i \in A(t^*)} f_{\mathbf{B}}(i)\widetilde{h}(i, t^*) \leq 16 \cdot d_{t^*}.$$

Proof. We first derive a bound on the quantity $W = \sum_{i \in A(t^*)} f_{S'}(i)\widetilde{h}(i, t^*)$. Consider the bands $B_1, B_2, \ldots, B_m$ corresponding to the timeslot t^*. At the end of the deletion phase, only two intervals were retained in S' for each band

B_j: they contribute at most $2 \cdot (2d_{t^*}/2^{j-1})$ to the quantity W. So, the total contribution across all the bands can be computed as:

$$W \leq \sum_{j=1}^{m} 2 \cdot (2d_{t^*}/2^{j-1}) \leq 8 \cdot d_{t^*}.$$

The doubling procedure implies that for any interval $i \in S'$, $f_{\mathbf{B}}(i) = 2 \cdot f_{S'}(i)$. This means the LHS of the inequality in the lemma is at most $2W \leq 16 \cdot d_{t^*}$. □

This completes the proof of Theorem 1.

Proof of Corollary 1: Note that the proof of Theorem 1 generalizes to the following partial covering scenario. Given a subset of timeslots $X \subseteq [1, T]$, the primal-dual algorithm can produce a solution $\widehat{S}$ such that $\widehat{S}$ covers all the timeslots in X in an SRA fashion and $\texttt{cost}(\widehat{S}) \leq 16 \cdot \texttt{Opt(X)}$, where $\texttt{Opt(X)}$ is the optimum solution covering all the timeslots in X. The optimum partial k-cover $\texttt{pOpt}$ covers a subset of timeslots Z with $|Z| \geq k$. We focus on only the timeslots in Z and ignore the rest. By invoking the generalization (with $X = Z$), we can get an SRA k-partial cover $\widehat{S}$ such that $\texttt{cost}(\widehat{S}) \leq 16 \cdot \texttt{cost(pOpt)}$ (since $\texttt{Opt(Z)} = \texttt{pOpt}$).

2.2 Computing Optimal SRA Covers

Here, we first prove Theorem 2 and then derive Corollary 2. The algorithm goes via a reduction to a problem that we call the *layered interval covering* problem (LIC), defined next.

Layered Interval Covering Problem (LIC**):** The input consists of a set of intervals $\mathcal{I}$ over timeslots $[1, T]$. Each interval is specified by its range $[s_i, e_i]$, where s_i is the start-time and e_i is the end-time) and a cost c_i. The input includes a set of *colors* $\mathcal{L} = \{1, 2, \ldots, L\}$ and specifies a color $\chi(i)$, for each interval $i \in \mathcal{I}$ and each timeslot $t \in [1, T]$. An interval i is said to *cover* a timeslot t, if i is active at t and $\chi(i) \geq \chi(t)$. A feasible solution S is a set of intervals such that for each timeslot $t \in [1, T]$, at least one interval i covers t. The goal is to compute a feasible solution of minimum cost.

Reduction: It is not hard to reduce the problem of finding the optimal SRA full cover to the LIC problem. Let $\mathcal{A}$ be the input SRA full cover problem instance and we will produce a LIC problem instance $\mathcal{B}$. The number of timeslots for $\mathcal{B}$ is declared to be the same as that of $\mathcal{A}$ (i.e., T is retained as such). Let $\mathcal{D}$ be the set of all distinct demand values (i.e., $\mathcal{D} = \{d_t \ : t \in [1, T]\}$) and let $L = |\mathcal{D}|$. Notice that $L \leq |T|$. Let $a_1, a_2, \ldots, a_L$ be the values in $\mathcal{D}$ sorted in the increasing order. We declare the number of colors in $\mathcal{B}$ to be L, so that each a_j corresponds to a color j. For each interval $i \in \mathcal{A}$ we introduce at most i in $\mathcal{B}$ as follows. For each $1 \leq j \leq L$, let r be the smallest integer such that $a_j \leq rh_i < a_{j+1}$; if such an r does not exist, we ignore this j; otherwise, we introduce a copy i' of i in $\mathcal{B}$ with color $\chi(i') = j$ and cost $c_{i'} = rc_i$ (where c_i is the cost of i in $\mathcal{A}$). This completes the reduction. It is easy to see that any feasible solution S of $\mathcal{B}$ can be transformed into an SRA full cover of $\mathcal{A}$ with the same cost and vice versa.

Below we show that the LIC problem can be solved optimally in polynomial time. It follows that optimum SRA full covers can be found in polynomial time, establishing Theorem 2.

Solving the LIC **Problem Optimally:** Our polynomial time algorithm is based on dynamic programming and it builds on a decomposition lemma for feasible solutions. The lemma needs the notion of *time-cuts*, defined next.

Let S be a set of intervals that cover all the timeslots in some range $[a, b] \subseteq [1, T]$. Let S_1 and S_2 be a partition of the set S and t be a timeslot satisfying $a \leq t \leq b-1$. Then the triplet $\langle S_1, S_2, t\rangle$ is said to be a *time-cut* for S, if S_1 covers all the timeslots in the range $[a, t]$ and S_2 covers all the timeslots in the range $[(t+1), b]$. Intuitively, some intervals in S_1 may span (and even cover) some timeslots in $[(t+1), b]$, but S_2 is responsible for covering all the timeslots in this range (and vice versa). The decomposition lemma says that we can always find a time-cut, but for an exceptional scenario. The restriction that $t \leq b-1$ ensures that the time-cut is non-trivial.

Lemma 4. *Let S be a set of intervals covering all the timeslots in some range $[a, b]$. Then, at least one of the following conditions holds: (i) there exists a time-cut for S; (ii) there exists an interval $i \in S$ spanning the entire range $[a, b]$.*

Proof. Consider any timeslot $t \in [a, b]$. Among all the intervals in S active at t, let i be the one having the maximum color; we say that i is *responsible* for t. Let i^* be any interval active at the initial timeslot a. If i^* spans the range $[a, b]$ then the (ii) is satisfied and we are done. So, assume that $e_{i^*} \leq b-1$. Let t^* be the maximum timeslot for which i^* is responsible. Define S_1 to be the set consisting of i^* and all intervals $i \in S$ having end-time $e_i \leq t^*-1$ and let $S_2 = S - S_1$. We claim that the triple $\langle S_1, S_2, t^*\rangle$ is a time-cut. To see this first consider any timeslot $t \in [(t^*+1), b]$ and we will argue that some interval in S_2 covers t. Since i^* is not responsible for any timeslot in $[(t^*+1), b]$, there must exist an interval $i \neq i^*$ covering t. Therefore, such an interval i belongs to S_2. Now consider any timeslot $t \in [a, t^*]$ and let us argue that some interval in S_1 covers t. Let i be any interval $i \in S$ covering t; so, $\chi(i) \geq \chi(t)$. If i belongs to S_1, we are done. Hence, consider the case where $i \in S_2$. We see that i is active at t^*, because i is active at t and has end-time $e_i \geq t^*$. Since i^* is responsible for t^*, we have that $\chi(i^*) \geq \chi(i)$. Hence, i^* also covers t. Finally, notice that $t^* \leq b-1$. □

Lemma 4 can easily be generalized to the partial setting as follows.

Corollary 3. *Let $[a, b]$ be a range of timeslots and let S a set of intervals covering all the timeslots in a subset $X \subseteq [a, b]$. Then, at least one of the following is true: (i) there exists a partition of S into S_1 and S_2, and a timeslot $t \in [a, b-1]$ such that S_1 covers all the timeslots in $X \cap [a, t]$ and S_2 covers all those in $X \cap [t+1, b]$; (ii) there exists an interval $i \in S$ spanning all the timeslots in X.*

For a range of timeslots $[a, b] \subseteq [1, T]$ and a color p, let $U([a, b], p)$ denote all the timeslots $t \in [a, b]$ such that $\chi(t) \geq p$. Let $\mathrm{DP}([a, b], p)$ be the cost of

the optimum set of intervals covering all the timeslots in $U([a,b],p)$. Corollary 3 establishes the following recurrence relation: $\mathrm{DP}([a,b],p) = \min\{Q_1, Q_2\}$, where

$$Q_1 = \min_{a \le t \le b-1} \{\mathrm{DP}([a,t],p) + \mathrm{DP}([t+1,b],p)\}$$

$$Q_2 = \min_{i \text{ spans } U([a,b],p)} \{c_i + \mathrm{DP}([a,b],\chi(i)+1)\}.$$

The quantity Q_1 corresponds to the situation where (i) in Corollary 3 applies. In Q_2, we try all the possible intervals i spanning $U([a,b],p)$. Any such i can cover all the timeslots t with $p \le \chi(t) \le \chi(i)$ and so the recursive component focuses on $U([a,b],\chi(i)+1)$.

Proof of Corollary 2: We now focus on the partial version of the LIC problem; here the input also includes an integer k and the goal is to find an optimum solution that covers at least k timeslots. We can extend the recurrence relation used in solving LIC to the partial cover setting. For a range of timeslots $[a,b]$, a color p and a number r, let $\mathrm{pDP}([a,b],p,r)$ be the cost of the optimum set of intervals covering at least r of the timeslots in $U([a,b],p)$. Using Corollary 3, we can write a recurrence relation for $\mathrm{pDP}([a,b],p,r)$. For a color $q \ge p$, let $\lambda([a,b],[p,q])$ be the number of timeslots t such that $p \in [a,b]$ and $\chi(t) \in [p,q]$. Corollary 3 establishes the following recurrence relation: $\mathrm{pDP}([a,b],p,r) = \min\{Q_1, Q_2\}$, where

$$Q_1 = \min_{\substack{a \le t \le b-1 \\ r_1 \le r}} \{\mathrm{pDP}([a,t],p,r_1) + \mathrm{pDP}([t+1,b],p,r-r_1)\}$$

$$Q_2 = \min_{i \text{ spans } U([a,b],p)} \{c_i + \mathrm{pDP}(\ \ [a,b],\ \chi(i)+1,\ r - \lambda([a,b],[p,\chi(i)])\ \ )\}.$$

In Q_1, in addition iterating over all possible cuts t, we also iterate over r_1, which symbolizes the number of timeslots that would be covered in the first range. In Q_2, we try all the possible intervals i spanning $U([a,b],p)$, while keeping count of the timeslots that got covered by the choice of i (i.e., $\lambda([a,b],[p,\chi(i)])$. Based on the above recurrence relation, we can write an dynamic program based algorithm; the final solution corresponds to the entry $\mathrm{pDP}([1,T],1,k))$. The running time of the algorithm is $O(km^4)$; here $m = \max\{n,T\}$, where k is the number timeslots required to be covered, n is the total number of intervals and T is the total number of timeslots. The problem of finding the optimum SRA k-partial cover can be reduced to the partial LIC problem, thereby establishing Corollary 2.

3 Approximation algorithm for the (0,1)-ResAll problem

In this Section, we present a primal-dual based 4-approximation algorithm for the (0,1)-ResAll problem.

The IP Formulation: As indicated in the introduction, we consider the strengthened LP augmented with so called flow-cover inequalities. We associate a variable x_i with each $i \in \mathcal{I}$; this variable is an indicator variable for whether the resource i is selected in the solution. There will be a constraint for every set

$S \subseteq \mathcal{I}$ of intervals: given that the set S is already chosen, the solution must pick enough intervals from the remaining intervals in $\mathcal{I}$, such that the *residual* demand is covered. With this goal in mind, given a set S, let $d_t(S)$ denote the *residual demand*, i.e. $d_t(S) = d_t - \sum_{i \in S \cap A(t)} h_i$. We define $\widetilde{h}(i,t,S) = \min\{h_i, d_t(S)\}$. Then, the IP is as follows.

$$\min \quad \sum_{i \in \mathcal{I}} x_i \cdot c_i$$

$$\sum_{i \in A(t), i \notin S} \widetilde{h}(i,t,S) \cdot x_i \geq d_t(S) \qquad \text{for all time-slots } t \text{ and all subsets } S \subseteq \mathcal{I}$$

$$x_i \in \{0,1\} \qquad \text{for all } i \in \mathcal{I}$$

One could have written simpler IP with h_i replacing $\widetilde{h}(i,t,S)$ in the constraints above. However, it can be shown that the corresponding LP also has an unbounded integrality gap.

We will consider the LP corresponding to the above IP, formed by relaxing the integrality constraints $x_i \in \{0,1\}$ to $x_i \geq 0$. The dual of the LP is as follows: the dual has variables $z(S,t)$ corresponding to the primal constraints.

$$\max \quad \sum_{(S,t)} z(S,t) \cdot d_t(S)$$

$$\sum_{(S,t): i \in A(t), i \notin S} \widetilde{h}(i,t,S) \cdot z(S,t) \leq c_i \qquad \text{for every interval } i$$

$$z(S,t) \geq 0 \qquad \text{for all } (S,t) : S \subseteq \mathcal{I}$$

Note that the primal LP has exponentially many constraints, one for each (S,t) pair; thus the dual has exponentially many variables. However, in the primal dual algorithm that we describe in the following we shall set only polynomially many variables in the dual to non-zero values.

Our primal-dual algorithm runs in two phases, a forward phase and a delete phase. The forward phase produces a dual feasible solution $\mathbf{z}$ and a primal integral feasible solution $\mathbf{A}$. The second phase outputs an arbitrary minimal feasible solution $\mathbf{B}$ contained within $\mathbf{A}$.

Forward Phase: Our goal is to produce a dual feasible solution $\mathbf{z}$ having high objective value ($\max \sum_{(S,t)} \mathbf{z}(S,t) \cdot d_t(S)$). Towards that goal we will raise the dual variables $z(S,t)$ one by one in a greedy fashion. In each step, we shall raise the variable $z(S,t)$ that gives us the highest benefit (while maintaining feasibility of the solution).

Let $S_0 = \emptyset$. The procedure runs in multiple iterations. For $j \geq 1$, the jth iteration is as follows. Let t_j be the timeslot t having the highest residual demand $d_t(S_{j-1})$. Raise the dual variable $z(S_{j-1}, t_j)$ till some dual constraint becomes tight; let the interval in the primal corresponding to this constraint be i_j. Set $S_j = S_{j-1} \cup \{i_j\}$. Repeat this procedure until the set S_j covers all the timeslots. Let the final set obtained by the procedure be $\mathbf{A} = \{i_1, i_2, \ldots, i_\ell\}$, where ℓ is the number of iterations.

Note that the set $\mathbf{A}$ is a feasible solution to our problem and $\mathbf{z}$ is a feasible solution for the dual. Moreover, this pair satisfies the primal complementary slackness condition:

$$i \in \mathbf{A} \implies \sum_{(S,t):i\in A(t),i\notin S} \widetilde{h}(i,t,S)\cdot \mathbf{z}(S,t) = c_i$$

Delete Phase: In this phase, we keep deleting intervals (in an arbitrary order) from $\mathbf{A}$ while maintaining feasibility of the solution; the process produces a minimal feasible solution $\mathbf{B}$ contained within $\mathbf{A}$.
Analysis: We shall show that the solutions $\mathbf{B}$ and $\mathbf{z}$ satisfy the complementary slackness conditions approximately. Notice that the delete phase did not modify the dual solution $\mathbf{z}$. Thus, the primal complementary slackness conditions remain satisfied. The following lemma shows that $\mathbf{B}$ and $\mathbf{z}$ satisfy the dual complementary slackness conditions approximately.

Lemma 5. *The solution* $\mathbf{B}$ *and* $\mathbf{z}$ *satisfy:*

$$\mathbf{z}(S,t) > 0 \implies \sum_{\substack{i\in A(t)\\ i\in \mathbf{B}-S}} \widetilde{h}(i,t,S) \le 4\cdot d_t(S) \tag{3}$$

Proof. The only dual variable with non-zero values are (S_0,t_1), (S_1,t_2),..., $(S_{\ell-1},t_\ell)$. Fix any $r \ge 0$ and let us argue that the slackness condition holds for the pair (S_r,t_{r+1}). Let $t^* = t_{r+1}$ and let $X = \{i_{r+1},\ldots,i_\ell\}\cap A(t^*)$. We need to show:

$$\sum_{i\in X} \widetilde{h}(i,t^*,S_r) \le 4\cdot d_{t^*}(S_r)$$

Arrange the intervals from X in the increasing order of their start-times, say $u_1,u_2,\ldots,u_s$, where $s=|X|$. From this ordering, keep picking intervals i until the sum of their adjusted heights $\widetilde{h}(i,t^*,S_r)$ exceeds $d_{t^*}(S_r)$. Let X_1 be the set of chosen intervals. Let t_1 be the start-time of the last interval picked. Similarly, arrange the intervals from X in the decreasing order of their end-times, say $v_1,v_2,\ldots,v_s$. From this ordering, keep picking intervals i until the sum of their adjusted heights $\widetilde{h}(i,t^*,S_r)$ exceeds $d_{t^*}(S_r)$. Let X_2 be the set of chosen intervals. Let t_2 be the end-time of last interval picked. The sets X_1 and X_2 need not be disjoint. We claim that $X = X_1\cup X_2$. By contradiction, suppose there exists some other interval $i\in X$. Then, i must be contained in the range $[t_1,t_2]$. The interval i can be removed from our solution $\mathbf{B}$ without violating feasibility of $\mathbf{B}$. To see this, notice that at any timeslot $t\in[t_1,t^*]$ where i is active, the sum of heights of intervals in X_1 exceeds $d_{t^*}(S_r)$. Similarly, at any timeslot $t\in[t^*,t_2]$ where i is active, the sum of heights of intervals in X_2 exceeds $d_{t^*}(S_r)$. Also it is true that $d_{t^*}(S_r)\ge d_t(S_r)$, because of the choice of t^* in the iteration $r+1$ in the forward phase. Therefore, i is redundant and can be removed. Given that $\mathbf{B}$ is a minimal feasible solution, such an i cannot exist. This proves the claim that $X = X_1\cup X_2$.

Consider the quantity $W_1 = \sum_{i \in X_1} \widetilde{h}(i, t^*, S_r)$. The quantity W_1 is at least $d_{t^*}(S_r)$, but it cannot exceed $2d_{t^*}(S_r)$, because every interval $i \in X_1$ satisfies $\widetilde{h}(i, t^*, S_r) \leq d_{t^*}(S_r)$. Similarly, the quantity $W_2 = \sum_{i \in X_2} \widetilde{h}(i, t^*, S_r)$ satisfies $W_2 \leq 2d_{t^*}(S_r)$. It follows that

$$\sum_{i \in X} \widetilde{h}(i, t^*, S_r) \leq 4 \cdot d_{t^*}(S_r) \leq W_1 + W_2 \leq 4 \cdot d_{t^*}(S_r).$$

□

Lemma 5 and the primal slackness property together imply that the solution $\mathbf{B}$ constructed by the algorithm satisfies $\texttt{cost}(\mathbf{B}) \leq 4 \cdot \texttt{cost}(\texttt{Opt})$, where $\texttt{Opt}$ is the optimum solution.

References

1. N. Bansal, A. Chakrabarti, A. Epstein, and B. Schieber. A quasi-PTAS for unsplittable flow on line graphs. In *STOC*, 2006.
2. P. Bonsma, J. Schulz, and A. Wiese. A constant factor approximation algorithm for unsplittable flow on paths. In *FOCS*, 2011.
3. T. Carnes and D. Shmoys. Primal-dual schema for capacitated covering problems. In *IPCO*, 2008.
4. V. Chakaravarthy, A. Kumar, G. Parija, S. Roy, and Y. Sabharwal. Minimum cost resource allocation for meeting job requirements. In *IPDPS*, 2011.
5. D. Chakrabarty, E. Grant, and J. Könemann. On column-restricted and priority covering integer programs. In *IPCO*, 2010.
6. C. Chekuri, M. Mydlarz, and F. Shepherd. Multicommodity demand flow in a tree and packing integer programs. *ACM Transactions on Algorithms*, 3(3), 2007.
7. A. Dhesi, P. Gupta, A. Kumar, G. Parija, and S. Roy. Contact center scheduling with strict resource requirements. In *IPCO*, 2011.
8. R. Gandhi, S. Khuller, and A. Srinivasan. Approximation algorithms for partial covering problems. *J. Algorithms*, 53(1):55–84, 2004.
9. N. Garg. Saving an ϵ: a 2-approximation for the k-MST problem in graphs. In *STOC*, pages 396–402, 2005.
10. D. Golovin, V. Nagarajan, and M. Singh. Approximating the k-multicut problem. In *SODA*, 2006.
11. O. Ibarra and C. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22:463–468, October 1975.
12. A. Ingolfsson, F. Campello, X. Wu, and E. Cabral. Combining Integer Programming and the Randomization Method to Schedule Employees. *European J. Operations Research*, 202(1):153–163, 2010.
13. K. Jain and V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.