www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume – 5 Issue -03 March, 2016 Page No. 16013-16019

Variants Of Resource Allocation Problem

Vijaya Goel

University of Delhi, India.
Email : vgoel.du@gmail.com

Abstract
Consider the problem of allocating resources such as machines, memory or bandwidth to meet the demands of a given set of jobs. Resource allocation should be done in such a manner that the available resources are assigned to the given jobs in an economic way. The Resource allocation (RESALL) problem is motivated by its applications in many real-life scenarios such as interval scheduling, workforce management. This is an NP-hard problem. The input consists of a set of resources and a demand profile (corresponding to the set of jobs). Each resource has a capacity, a cost, a start time and a finish time. The cost of solution is the sum of the costs of the resources included in the solution. The goal is to find a minimum cost feasible solution such that multiple units of resources can be included in the solution and at any timeslot, the sum of capacities of these resources should be at least the demand requirement at that timeslot. This problem definition corresponds to the MULTIRESALL version of RESALL problem.

The two generalizations of the problem stated above are partial MULTIRESALL and (0,1)-RESALL. In the former variant, the input also specifies a number k and the goal is to find the minimum cost k-partial cover where the solution satisfies the demand requirements of at least k timeslots. In the latter one, the feasible solution can include at most one copy of each resource. The goal is to find minimum cost full cover. These variants have 16-approximation algorithm and 4-approximation algorithm respectively (see [3]). In this paper, the approximation algorithms for some variants of (0,1)-RESALL problem are analyzed.

Introduction

Many real-life scenarios can be mapped to this problem and one of them is illustrated here. Let us say we have an education portal "learnMaths.com" which delivers online lectures for the subject Mathematics. They need to cover all chapters for class XII. They have a database of teachers who can teach the required chapters. Each teacher is available for some dates known in advance to the portal. These are *consecutive* dates. Portal also has a demand profile which specifies the number of chapters they want to get covered on each day. Each teacher has capacity to teach some fixed number of chapters and a cost which the portal has to pay if they want lectures from him. Now, portal has to form a minimum cost schedule of teachers for n number of consecutive days. The teachers are classified based on their proficiency in the subject and the fees they charge. In the zero-one setting, allowing one copy of a resource in the solution corresponds to selecting one teacher from a defined class.

Problem Definition

The input is described below :

• Let the time is specified by the interval $[1, T]$. Assuming time is divided into discrete timeslots, let t denotes a timeslot specified by an integer in the range $[1, T]$.

• Let the demand profile is specified by $d : [1, T] \rightarrow Z$ where d_t denotes the demand at every timeslot t.

• Let I denotes the set of resources. Each resource $i \in I$ has a start time s_i, an end time e_i, a capacity (or height) h_i and a cost c_i. s_i and e_i are integers in the range [1, T]. The interval $I_i = [s_i, e_i]$ corresponding to resource $i \in I$ spans over the timeslots in the range $[s_i , e_i]$. 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.

A variable x_i is associated with each $i \in I$. x_i is an indicator variable. $x_i = 1$ if i^{th} interval is picked in the solution and 0 otherwise.

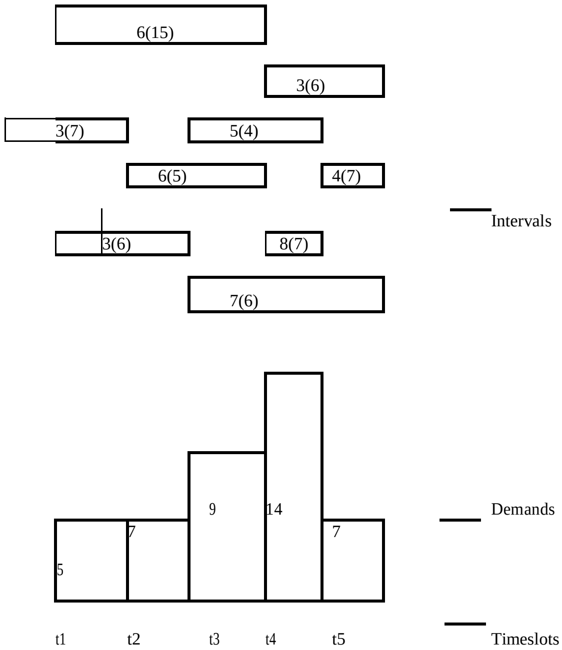


Figure 1: An instance of RESALL Problem.

Let S denotes a multiset of resources. For every set $S \subseteq I$, given that the set S is already chosen in the solution, the *residual* demand must be covered by choosing enough intervals from the remaining intervals in I. The X residual demand is denoted by $d_t(S)$ such that $d_t(S) = d_t - \sum_{i \in S \cap A(t)} h_i$.

An instance of RESALL problem is shown in figure 1 where the capacity and cost for a resource $i \in I$ is specified as $h_i(c_i)$. In (0,1)-RESAll problem, the goal is to find a minimum cost feasible solution such that the solution picks at most one copy of any resource and it satisfies the demand of all timeslots.

1.2 Prior Work

Different variants of the Resource Allocation problem are obtained by changing inputs or constraints to the problem. In *activity scheduling problem*, there is a set of activities competing for a reusable resource. Given a resource of fixed size, an algorithm is presented in [1] to approximate resource allocation and scheduling problem by a constant factor. The solution is the subset of activities such that the total amount of resources allocated never exceeds the available resources. Another variant of RESALL problem is *Varying bandwidth resource allocation problem with bag constraints(BAGVBRAP)* where the bandwidth refers to the resource offered by the system. The input consists of a set of jobs and each job consists of a bag of job instances. For each job instance, the input includes start time, finish time, bandwidth requirement and profit. Input also has bandwidths present at different points of time. The goal is to find a maximum profit feasible solution which chooses a subset of instances such that bandwidth requirements does not exceed the bandwidth availability at any point of time. Also, at most one instance can be selected in the solution from a job. A constant factor

DOI: 10.18535/ijecs/v5i3.25

algorithm for this problem is discussed in [2]. The MULTIRESALL version allows multiple copies of any resource to be picked in the solution and the goal is to find a minimum cost full cover. A 4-approximation algorithm for MULTIRESALL version uses primal dual approach as explained in [4]. (0,1)-RESALL problem is a generalization of MULTIRESALL version. A 4-approximation algorithm for (0,1)-RESALL version generalizes the 4-approximation algorithm for MULTIRESALL problem. It is also based on primal-dual method (see [3])

2. Approximation Algorithms for variants of (0,1)-RESALL problem

In this section, the IP for primal is formulated followed by its LP relaxation and the corresponding dual formulation. Then, some variants of (0,1)-RESALL problem along with their approximation algorithms are discussed.

2.1 IP Formulation

$$min \sum_{i \in I} x_i \,.\, c_i \tag{1}$$

$$\sum_{i \in A(t), i \notin S} \bar{h}(i,t,S) \,.\, x_i \geq d_t(S) \qquad \forall t \;\; and \;\; \forall S \subseteq I \tag{2}$$

$$x_i \in \{0,1\} \tag{3}$$

Where

$$\bar{h}(i,t,S) = min\{h_i, d_t(S)\} \tag{4}$$

Above IP can be relaxed by changing $x_i \in \{0, 1\}$ to $0 \leq x_i \leq 1$.

Dual for the above LP is :

$$max \sum_{(S,t)} z(S,t) \,.\, d_t(S) \tag{5}$$

$$\sum_{S: i \in A(t), i \notin S} \bar{h}(i,t,S) \,.\, z(S,t) \leq c_i \qquad \forall i \tag{6}$$

$$z(S,t) \geq 0 \qquad \forall (S,t) \tag{7}$$

The primal-dual algorithm runs in two phases, forward phase and reverse delete phase, giving 4-approximate solution as discussed in [3].

2.2 Variants of (0,1)-RESALL

Three variants of (0,1)-RESALL discussed here are *prize collecting version having penalties on discrete timeslots, prize collecting version having penalties with jobs* and *the version with bag constraint(at least one).*

1.Prize Collecting Version of (0,1)-RESALL having penalties on discrete timeslots

In (0,1)-RESALL problem, the goal is to find a minimum cost solution which covers all the timeslots. Here, there is an additional input, i.e., penalty(p_t) for each timeslot *t*. There are certain timeslots for which one

needs to pay high cost to satisfy their demands. So, those timeslots can be left out by paying the penalty corresponding
to them. The goal is to find a feasible solution which minimizes the sum of cost of intervals picked in the solution and the penalties for uncovered timeslots. Let y_t be an indicator variable whether timeslot t has been covered or not. $y_t = 1$ if that timeslot has been covered and 0 otherwise.

- IP formulation

X

- **IP formulation**

$$min \quad \sum_{i\in I} x_i \cdot c_i + \sum_{t\in T} p_t \cdot (1 - y_t) \tag{8}$$

$$\sum_{i\in A(t), i\notin S} \bar{h}(i,t,S) \cdot x_i \geq d_t(S) \cdot y_t \qquad \forall\, t \text{ and } \forall S \subseteq I \tag{9}$$

$$x_i \in \{0,1\} \tag{10}$$

$$y_t \in \{0,1\} \tag{11}$$

Above IP can be relaxed by changing $x_i \in \{0, 1\}$ to $0 \leq x_i \leq 1$ and $y_t \in \{0, 1\}$ to $0 \leq y_t \leq 1$.

This problem can easily be reduced to the original (0,1)-RESALL problem with same objective value. So, this problem also has a 4-factor approximation.

- **Reduction**

– **Construction**

Keep the same set of intervals I, T timeslots and their demands as they are in prize collecting version.

Introduce T new intervals spanning exactly 1 distinct timeslot with height d_t and cost p_t corre-sponding to that timeslot. Let these intervals form a set J.

– **Solution to the above (0,1)-RESALL would give the solution to the prize collecting ver-sion**

Solution of (0,1)-RESALL would contain intervals either from I or from J. If for any timeslot t, an

interval $i \in J$ is picked in the solution then it will not be covered in prize collecting problem because it is more cheaper to pay its penalty instead of covering it. Thus, a feasible solution to the standard version will yield a feasible solution to the prize collecting version. Also, value of the objective function remains the same because for each timeslot covered by an interval $i \in J$, the penalty p_t is paid which is nothing else but the cost of i.

– **Solution to prize collecting version gives the solution to (0,1)-RESALL**

Covered timeslots must have some intervals associated with them in the solution. As the set of intervals after reduction is same, so these timeslots will also be covered in standard (0,1)-RESALL with same set of intervals.

DOI: 10.18535/ijecs/v5i3.25

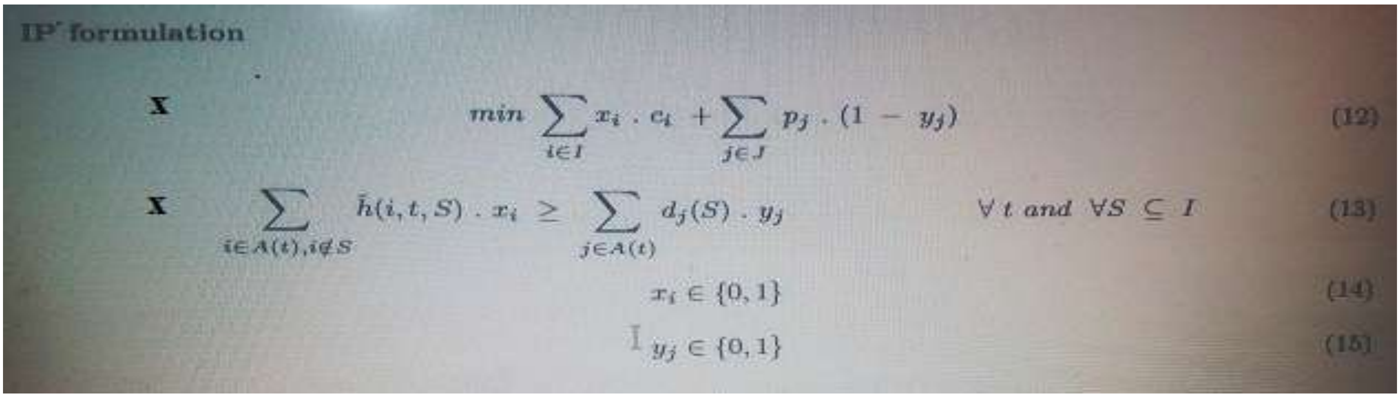
Solution contains penalties for the uncovered timeslots. Therefore, in standard (0,1)-RESALL these timeslots will be covered by the set of intervals from J. Thus, a feasible solution of prize collecting version will yield a feasible solution to the standard version because all the timeslots get satisfied in

the end. Also, value of the objective function will be same because cost of the intervals from J are nothing else than the penalties p_t.

2. Prize Collecting Version of (0,1)-RESALL having penalties with jobs

Here, there is a demand d_j and penalty p_j for each job having s_j and e_j as its starting and ending time respectively. Say, J be the number of jobs. The goal is to find a feasible solution which minimizes the sum of cost of intervals picked in the solution and the penalties for uncovered jobs. Let y_j be an indicator variable whether job j has been covered or not. $y_j = 1$ if that job has been covered and 0 otherwise.

- **IP formulation**



IP formulation

$$\min \sum_{i \in I} x_i \cdot c_i + \sum_{j \in J} p_j \cdot (1 - y_j) \qquad (12)$$

$$\sum_{i \in A(t), i \notin S} \bar{h}(i,t,S) \cdot x_i \geq \sum_{j \in A(t)} d_j(S) \cdot y_j \qquad \forall t \text{ and } \forall S \subseteq I \qquad (13)$$

$$x_i \in \{0,1\} \qquad (14)$$

$$y_j \in \{0,1\} \qquad (15)$$

Above IP can be relaxed by changing $x_i \in \{0, 1\}$ to $0 \leq x_i \leq 1$ and $y_j \in \{0, 1\}$ to $0 \leq y_j \leq 1$.

This problem can easily be reduced to the standard (0,1)-RESALL problem with same objective value. So, this problem also has a 4-factor approximation.

- **Reduction**

– **Construction**

$*$ Keep the same set of intervals I, T timeslots and their demands would be $\sum_{j \in A(t)} d_j$.

For each job, introduce a new interval having same starting and ending time with height d_j and cost p_j. Let these intervals form a set M.

– **Solution to the above (0,1)-RESALL would give the solution to the prize collecting ver-sion with jobs**

Solution of (0,1)-RESALL would contain intervals either from I or from M. If for any timeslot t, an interval $i \in M$ is picked in the solution then the job for which i is included will not be covered in prize collecting problem. Thus, a feasible solution to the standard version will yield a feasible solution to

the prize collecting version. Also, value of the objective function remains the same because for each timeslot covered by an interval $i \in M$ the penalty p_j is paid which is nothing but the cost of i.

– **Solution to prize collecting version with jobs gives the solution to (0,1)-RESALL**

For each covered job j there will be some intervals associated with it in the solution. Since the set of intervals after reduction is same, so for all timeslots t for which $j \in A(t)$, d_j demand will be satisfied in standard (0,1)-RESALL with same set of intervals.

Solution contains penalties for the uncovered jobs. Therefore, in standard (0,1)-RESALL for all timeslots t for which $j \in A(t)$, d_j demand will be satisfied by the penalty interval corresponding to the job. Thus, a feasible solution of prize collecting version will yield a feasible solution to the standard version

because all the timeslots gets satisfied in the end. Also, value of the objective function will be same because cost of the intervals from M are nothing else than the penalties p_j.

3.(0,1)-RESALL with bag constraint(at least one)

Here, input consists of set of T timeslots and their demands $(d_t)^0s$, set of bags $B_1, B_2, ...B_N$ where each bag has some intervals in it. Each interval i has some starting time s_i, ending time e_i, height h_i and cost c_i. The goal is to find a minimum cost set of intervals such that they satisfy the demand of T timeslots and contains at least one interval from each bag. Let OPT be the optimal cost of our solution.

- **IP formulation**

Its Integer Program is same as that of (0,1)-RESALL with an additional constraint for bags.

X

$$\sum_{i \in B_j} x_i \geq 1 \quad \forall j \in \{1, 2, \ldots, N\} \qquad (16)$$

- **5-factor Approximation Algorithm**

– There exists a 4-factor approximation algorithm for (0,1)-RESALL without bag constraints using primal-dual approach. Firstly, the same algorithm can be applied on (0,1)-RESALL with bag constraints as well.

– Now, only the unsatisfied bag constraints need to be satisfied. So, for each unsatisfied bag constraint, the minimum cost interval from that bag is picked.

– Total cost of intervals picked to satisfy the bag constraints ≤ the optimal cost.

– Solution obtained will be feasible since all the bag constraints and demand constraints are satisfied.

– Total cost of our algorithm = Cost of Primal-Dual Algorithm + cost of intervals picked to satisfy the bag constraints

$\leq 4.OPT + OPT \leq 5.OPT$

Hence, this algorithm is a 5-factor approximation algorithm for (0,1)-RESALL with bag constraints.

References

DOI: 10.18535/ijecs/v5i3.25

1. Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph (Seffi) Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. *J. ACM*, 48(5):1069–1090, September 2001.

2. Venkatesan T. Chakaravarthy, Anamitra R. Choudhury, and Yogish Sabharwal. A Near-linear Time Constant Factor Algorithm for Unsplittable Flow Problem on Line with Bag Constraints. In Kamal Lodaya and Meena Mahajan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*, volume 8 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 181–191, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

3. Venkatesan T. Chakaravarthy, Amit Kumar, Sambuddha Roy, and Yogish Sabharwal. Resource allocation for covering time varying demands. In *Proceedings of the 19th European conference on Algorithms*, ESA'11, pages 543–554, Berlin, Heidelberg, 2011. Springer-Verlag.

4. Venkatesan T. Chakaravarthy, Gyana R. Parija, Sambuddha Roy, Yogish Sabharwal, and Amit Kumar. Minimum cost resource allocation for meeting job requirements. In *Proceedings of the 2011 IEEE International Parallel & Distributed Processing Symposium*, IPDPS '11, pages 14–23, Washington, DC, USA, 2011. IEEE Computer Society.