# On Lagrangian Relaxation and Subset Selection Problems

(Extended Abstract)

Ariel Kulik* Hadas Shachnai†

**Abstract**

We prove a general result demonstrating the power of Lagrangian relaxation in solving constrained maximization problems with arbitrary objective functions. This yields a unified approach for solving a wide class of *subset selection* problems with linear constraints. Given a problem in this class and some small $\varepsilon \in (0,1)$, we show that if there exists a $\rho$-approximation algorithm for the Lagrangian relaxation of the problem, for some $\rho \in (0,1)$, then our technique achieves a ratio of $\frac{\rho}{\rho+1} - \varepsilon$ to the optimal, and this ratio is tight.

The number of calls to the $\rho$-approximation algorithm, used by our algorithms, is *linear* in the input size and in $\log(1/\varepsilon)$ for inputs with cardinality constraint, and polynomial in the input size and in $\log(1/\varepsilon)$ for inputs with arbitrary linear constraint. Using the technique we obtain approximation algorithms for natural variants of classic subset selection problems, including real-time scheduling, the *maximum generalized assignment problem (GAP)* and maximum weight independent set.

## 1 Introduction

Lagrangian relaxation is a fundamental technique in combinatorial optimization. It has been used extensively in the design of approximation algorithms for a variety of problems (see e.g., [12, 11, 18, 16, 17, 4] and a comprehensive survey in [19]). In this paper we prove a general result demonstrating the power of Lagrangian relaxation in solving constrained maximization problems of the following form. Given a universe $U$, a weight function $w : U \to \mathbb{R}^+$, a function $f : U \to \mathbb{N}$ and an integer $L \geq 1$, we want to solve

$$\Pi : \quad \max_{s \in U} \quad f(s) \qquad \text{subject to:} \quad w(s) \leq L. \tag{1}$$

We solve $\Pi$ by finding efficient solution for the Lagrangian relaxation of $\Pi$, given by

$$\Pi(\lambda) : \max_{s \in U} f(s) - \lambda \cdot w(s), \tag{2}$$

for some $\lambda \geq 0$.

A traditional approach for using Lagrangian relaxation in approximation algorithms (see, e.g., [11, 16, 4]) is based on initially finding two solutions, $SOL_1$, $SOL_2$, for $\Pi(\lambda_1), \Pi(\lambda_2)$, respectively, for some $\lambda_1, \lambda_2$, such that each of the solutions is an approximation for the corresponding

*Computer Science Dept., Technion, Haifa 32000, Israel. E-mail: kulik@cs.technion.ac.il

†Computer Science Dept., Technion, Haifa 32000, Israel. E-mail: hadas@cs.technion.ac.il. Work supported by the Technion V.P.R. Fund.

Lagrangian relaxation; while one of these solutions is feasible for $\Pi$ (i.e., satisfies the weight constraint), the other is not. A main challenge is then to find a way to *combine* $SOL_1$ and $SOL_2$ to a feasible solution which yields approximation for $\Pi$. We prove (in Theorem 1) a general result, which allows to obtain a solution for $\Pi$ based on *one* of the solutions only, namely, we show that with appropriate selection of the parameters $\lambda_1, \lambda_2$ in the Lagrangian relaxation we can obtain solutions $SOL_1$, $SOL_2$ such that one of them can be used to derive efficient approximation for our original problem $\Pi$. The resulting technique leads to fast and simple approximation algorithms for a wide class of *subset selection* problems with linear constraints.

### 1.1 Subset Selection Problems

Subset selection problems form a large class encompassing such NP-hard problems as real-time scheduling, the generalized assignment problem (GAP) and maximum weight independent set, among others. In these problems, a subset of elements satisfying certain properties needs to be selected out of a universe, so as to maximize some objective function. (We give a formal definition in Section 3.) We apply our general technique to obtain efficient approximate solutions for the following natural variants of some classic subset selection problems.

**Budgeted Real Time Scheduling (BRS):** Given is a set of *activities*, $\mathcal{A} = \{A_1, \ldots, A_m\}$, where each activity consists of a set of *instances*; an instance $\mathcal{I} \in A_i$ is defined by a half open time interval $[s(\mathcal{I}), e(\mathcal{I}))$ in which the instance can be scheduled ($s(\mathcal{I})$ is the start time, and $e(\mathcal{I})$ is the end time), some cost $c(\mathcal{I}) \in \mathbb{N}$, and a profit $p(\mathcal{I}) \in \mathbb{N}$. A schedule is *feasible* if it contains at most one instance of each activity, and for any $t \geq 0$, at most one instance is scheduled at time $t$. The goal is to find a feasible schedule, in which the total cost of all the scheduled instances is bounded by a given budget $L \in \mathbb{N}$, and the total profit of the scheduled instances is maximized. *Budgeted continuous real-time scheduling (BCRS)* is a variant of this problem where each instance is associated with a *time window* $\mathcal{I} = [s(\mathcal{I}), e(\mathcal{I}))$ and length $\ell(\mathcal{I})$. An instance $\mathcal{I}$ can be scheduled at any time interval $[\tau, \tau + \ell(\mathcal{I}))$, such that $s(\mathcal{I}) \leq \tau \leq e(\mathcal{I}) - \ell(\mathcal{I}))$. BRS and BCRS arise in many scenarios in which we need to schedule activities subject to resource constraints, e.g., storage requirements for the outputs of the activities.

**Budgeted Generalized Assignment Problem (BGAP):** Given is a set of bins with (possibly different) capacity constraints, and a set of items that have possibly different size, value and deduced cost for each bin; also, given is a budget $L \geq 0$. The goal is to pack a maximum valued subset of items into the bins subject to the capacity constraints, such that the total cost of the selected items is at most $L$. BGAP arises in many real-life scenarios, such as inventory planning with delivery costs.

**Budgeted Maximum Weight Independent Set (BWIS):** Given is a budget $L$ and a graph $G = (V, E)$, where each vertex $v \in V$ has an associated profit $p_v$ (or, *weight*) and associated cost $c_v$, choose a subset $V' \subseteq V$ such that $V'$ is an *independent set* (for any $e = (v, u) \in E$, $v \notin V'$ or $u \notin V'$), the total cost of vertices in $V'$, given by $\sum_{v \in V'} c_v$, is bounded by $L$, and the total profit of $V'$, $\sum_{v \in V'} p_v$, is maximized. BWIS is a generalization of the classical *maximum independent set (IS)* and *maximum weight independent set (WIS)* problems.

### 1.2 Contribution

We prove (in Theorem 1) a general result demonstrating the power of Lagrangian relaxation in solving constrained maximization problems with arbitrary objective functions.

We use this result to develop a unified approach for solving subset selection problems with linear constraint. Specifically, given a problem in this class and some small $\varepsilon \in (0, 1)$, we show that if there exists a $\rho$-approximation algorithm for the Lagrangian relaxation of the problem, for some $\rho \in (0, 1)$, then our technique achieves a ratio of $\frac{\rho}{\rho+1} - \varepsilon$ to the optimal. In the Appendix we give an example for a subset selection problem $\Gamma$ and show that, if there exists a $\rho$-approximation algorithm for the Lagrangian relaxation of $\Gamma$, for some $\rho \in (0, 1)$, then there exists an input $I$ for which finding the solutions $SOL_1$ and $SOL_2$ (for the Lagrangian relaxation) and combining the solutions yields at most a ratio of $\frac{\rho}{\rho+1}$ to the optimal. This shows the tightness of our bound, within additive of $\varepsilon$. The number of calls to the $\rho$-approximation algorithm, used by our algorithms, is *linear* in the input size and in $\log(1/\varepsilon)$, for inputs with cardinality constraint (i.e., where $w(s) = 1$ for all $s \in U$), and polynomial in the input size and in $\log(1/\varepsilon)$ for inputs with arbitrary linear constraint (i.e., arbitrary weights $w(s) \geq 0$).

We apply the technique to obtain efficient approximations for natural variants of some classic subset selection problems. In particular, for the budgeted variants of the real-time scheduling problem we obtain (in Section 4.1 a bound of $(1/3 - \varepsilon)$ for BRS and $(1/4 - \varepsilon)$ for BCRS. For budgeted GAP we give (in Section 4.2) an approximation ratio of $\frac{1-e^{-1}}{2-e^{-1}} - \varepsilon$.

For BWIS we show (in Section 4.3) how an approximation algorithm $\mathcal{A}$ for WIS can be used to obtain an approximation algorithm for BWIS with the same asymptotic approximation ratio. More specifically, let $\mathcal{A}$ be a polynomial time algorithm that finds in a graph $G$ an independent set whose profit is at least $f(n)$ of the optimal, where $(i)$ $f(n) = o(1)$ and $(ii)$ $\log(f(n))$ is polynomial in the input size.[1] Our technique yields an approximation algorithm which runs in polynomial time and achieves an approximation ratio of $g(n) = \Theta(f(n))$. Moreover, $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 1$. Since BWIS generalizes $WIS$, this implies that the two problems are essentially equivalent in terms of hardness of approximation.

Our technique can be applied iteratively to obtain a $(\frac{\rho}{1+d\rho} - \varepsilon)$-approximation algorithm for subset selection problems with $d$ linear constraints, when there exists a $\rho$-approximation algorithm for the non-constrained version of the problem, for some $\rho \in (0, 1)$ (we give the details in the Appendix).

It is important to note that the above results, which apply to maximization problems with *linear* constraints, do not exploit the result in Theorem 1 in its full generality. We believe that the theorem will find more uses, e.g., in deriving approximation algorithms for subset selection problems with *non-linear* constraints.

### 1.3 Related Work

Most of the approximation techniques based on Lagrangian relaxation are tailored to handle specific optimization problems. In solving the *k-median* problem through a relation to *facility location*, Jain and Vazirani developed in [16] a general framework for using Lagrangian relaxation to derive approximation algorithms (see also [11]). The framework, that is based on a primal-dual approach, finds initially two approximate solutions $SOL_1$, $SOL_2$ for the Lagrangian relaxations $\Pi(\lambda_1)$, $\Pi(\lambda_2)$ of a problem $\Pi$, for carefully selected values of $\lambda_1, \lambda_2$; a *convex combination* of these solutions yields a (fractional) solution which uses the budget $L$. This solution is then rounded to obtain an integral solution that is a good approximation for the original problem. Our approximation technique (in Section 2) differs from the technique of [16] in two ways. First, it does not require rounding a fractional solution: in fact, we do not attempt to combine the solutions $SOL_1$, $SOL_2$, but rather, examine each separately and compare the two feasible solutions which can be easily derived from $SOL_1$, $SOL_2$, using an efficient transformation of the non-feasible solution, $SOL_2$, to a feasible one. Secondly, the framework of [16] crucially depends on a primal-

[1] These two requirements hold for most approximation algorithm for the problem.

dual interpretation of the approximation algorithm for the relaxed problem, which is not required here.

Könemann et al. considered in [17] a technique for solving general partial cover problems. The technique builds on the framework of [16], namely, an instance of a problem in this class is solved by initially finding the two solutions $SOL_1$, $SOL_2$ and generating a solution $SOL$, which combined these two solutions. For a comprehensive survey of other work see, e.g., [19].[2]

There has been some earlier work on using Lagrangian relaxation to solve subset selection problems. The paper [20] considered a subclass of the class of subset selection problems that we study here. Using the framework of [16], the paper claims to obtain an approximation ratio of $\rho - \varepsilon$ for any problem in this subclass,[3] given a $\rho$-approximation algorithm for the Lagrangian relaxation of the problem (which satisfies certain properties). Unfortunately, this approximation ratio was shown to be incorrect [21]. Recently, Berget et al. considered in [4] the budgeted matching problem and the budgeted matroid intersection problem. The paper gives the first polynomial time approximation schemes for these problems. The schemes, which are based on Lagrangian relaxation, merge the two obtained solutions using some strong combinatorial properties of the problems.

The non-constrained variants of the subset selection problems that we study here are well studied. For known results on real-time scheduling and related problems see, e.g., [2, 6, 3, 7]. Surveys of known results for the generalized assignment problem are given, e.g., in [5, 8, 9, 10].

Numerous approximation algorithms have been proposed and analyzed for the maximum (weight) independent set problem. Alon at al. [1] showed that IS cannot be approximated within factor $n^{-\varepsilon}$ in polynomial time, where $n = |V|$ and $\varepsilon > 0$ is some constant, unless $P = NP$. The best known approximation ratio of $\Omega(\frac{\log^2 n}{n})$ for WIS on general graphs is due to Halldórsson [14]. A survey of other known results for a IS and WIS can be found e.g., in [13, 15].

To the best of our knowledge, approximation algorithms for the budgeted variants of the above problems are given here for the first time.

## 2 Lagrangian Relaxation Technique

Given a universe $U$, let $f : U \to \mathbb{N}$ be some objective function, and let $w : U \to \mathbb{R}^+$ be a non-negative weight function. Consider the problem $\Pi$ of maximizing $f$ subject to a budget constraint $L$ for $w$, as given in (1), and the Lagrangian relaxation of $\Pi$, as given in (2).

We assume that the value of an optimal solution $s^*$ for $\Pi$ satisfies $f(s^*) \geq 1$. For some $\varepsilon' > 0$, suppose that

$$\lambda_2 \leq \lambda_1 \leq \lambda_2 + \varepsilon'. \tag{3}$$

The heart of our approximation technique is the next result.

**Theorem 1** *For any $\varepsilon > 0$ and $\lambda_1, \lambda_2$ that satisfy (3) with $\varepsilon' = \varepsilon/L$, let $s_1 = SOL_1$ and $s_2 = SOL_2$ be $\rho$-approximate solutions for $\Pi(\lambda_1), \Pi(\lambda_2)$, such that $w(s_1) \leq L \leq w(s_2)$. Then for any $\alpha \in [1-\rho, 1]$, at least one of the following holds:*

1. $f(s_1) \geq \alpha \rho f(s^*)$
2. $f(s_2) > (1 - \alpha - \varepsilon) f(s^*) \frac{w(s_2)}{L}$.

[2]For conditions under which Lagrangian relaxation can be used to solve discrete/continuous optimization problems see e.g., [22].

[3]This subclass includes the *real-time scheduling* problem.

**Proof:** Let $L_i = w(s_i)$, $i = 1, 2$, and $L^* = w(s^*)$. From (2) we have that

$$f(s_i) - \rho f(s^*) \geq \lambda_i (L_i - \rho L^*). \tag{4}$$

Assume that, for some $\alpha \geq 1 - \rho$, $f(s_1) < \alpha\rho f(s^*)$, then

$$(\alpha - 1)\rho f(s^*) > f(s_1) - \rho f(s^*) \geq \lambda_1(L_1 - \rho L^*) \geq -\rho\lambda_1 L^* \geq -\rho\lambda_1 L.$$

The second inequality follows from (4), the third inequality from the fact that $\lambda_1 L_1 \geq 0$, and the last inequality holds due to the fact that $L^* \leq L$. Using (3), we have

$$\frac{(1-\alpha)f(s^*)}{L} < \lambda_1 < \lambda_2 + \varepsilon'. \tag{5}$$

Since $\varepsilon' = \varepsilon/L$, we have that

$$\begin{aligned} f(s_2) &\geq \lambda_2(L_2 - L^*) + \rho f(s^*) > (\frac{(1-\alpha)f(s^*)}{L} - \varepsilon')(L_2 - L) + \rho f(s^*) \\ &\geq (1-\alpha)f(s^*)\frac{L_2}{L} - \varepsilon' L_2 \geq (1 - \alpha - \varepsilon' L)\frac{L_2}{L} f(s^*) = (1 - \alpha - \varepsilon)\frac{L_2}{L} f(s^*) \end{aligned}$$

The first inequality follows from (4), by taking $i = 2$, and the second inequality is due to (5) and the fact that $L^* \leq L$. The third inequality holds since $\rho \geq 1 - \alpha$, and the last inequality follows from the fact that $f(s^*) \geq 1$. □

Theorem 1 asserts that at least one of the solutions $s_1, s_2$ is *good* in solving our original problem, $\Pi$. If $s_1$ is a good solution then we have an $\alpha\rho$-approximation for $\Pi$, otherwise we need to find a way to convert $s_2$ to a solution $s'$ such that $w(s') \leq L$ and $f(s')$ is a good approximation for $\Pi$. Such conversions are presented in Section 3 for a class of *subset selection problems with linear constraints*. Next, we show how to find two solutions which satisfy the conditions of Theorem 1.

### 2.1 Finding the Solutions $s_1, s_2$

Suppose that we have an algorithm $\mathcal{A}$ which finds a $\rho$-approximation for $\Pi(\lambda)$, for any $\lambda \geq 0$. Given an input $I$ for $\Pi$, denote the solution which $\mathcal{A}$ returns for $\Pi(\lambda)$ by $\mathcal{A}(\lambda)$, and assume that it is sufficient to consider $\Pi(\lambda)$ for $\lambda \in (0, \lambda_{max})$, where $\lambda_{max} = \lambda_{max}(I)$ and $w(\mathcal{A}(\lambda_{max})) \leq L$.

Note that if $w(\mathcal{A}(0)) \leq L$ then $\mathcal{A}(0)$ is a $\rho$-approximation for $\Pi$; otherwise, there exist $\lambda_1, \lambda_2 \in (0, \lambda_{max})$ such that $\lambda_1, \lambda_2$, and $s_1 = \mathcal{A}(\lambda_1), s_2 = \mathcal{A}(\lambda_2)$ satisfy (3) and the conditions of Theorem 1, and $\lambda_1, \lambda_2$ can be easily found using binary search. Each iteration of the binary search requires a single execution of $\mathcal{A}$ and reduces the size of the search range by half. Therefore, after $R = \lceil \log(\lambda_{max}) + \log(L) + \log(\varepsilon^{-1}) \rceil$ iterations, we have two solutions which satisfy the conditions of the theorem.

**Theorem 2** *Given an algorithm $\mathcal{A}$ which outputs a $\rho$-approximation for $\Pi(\lambda)$, and $\lambda_{max}$, such $w(\mathcal{A}(\lambda_{max})) \leq L$, a $\rho$-approximate solution or two solutions $s_1, s_2$ which satisfy the conditions of Theorem 1 can be found by using binary search. This requires $\lceil \log(\lambda_{max}) + \log(L) + \log(\varepsilon^{-1}) \rceil$ executions of $\mathcal{A}$.*

We note that when $\mathcal{A}$ is a randomized approximation algorithm whose *expected* performance ratio is $\rho$, a simple binary search may not output solutions that satisfy the conditions of Theorem 1. In this case, we repeat the executions of $\mathcal{A}$ for the same input and select the solution of maximal value. For some pre-selected values $\beta > 0$ and $\delta > 0$, we can guarantee that the probability that any of the used solutions is not a $(\rho - \beta)$-approximation is bounded by $\delta$. Thus, with appropriate selection of the values of $\beta$ and $\delta$, we get a result similar to the result in Theorem 1. We discuss this case in detail in the full version of the paper.

# 3 Approximation Algorithms for Subset Selection Problems

In this section we develop an approximation technique for subset selection problems. We start with some definitions and notation. Given a universe $U$, let $X \subseteq 2^U$ be a domain, and $f : X \to \mathbb{N}$ a set function. For a subset $S \subseteq U$, let $w(S) = \sum_{s \in S} w_s$, where $w_s \geq 0$ is the *weight* of the element $s \in U$.

**Definition 3.1** *The problem*

$$\begin{aligned} \Gamma : \quad & \max_{S \in X} \quad f(S) \quad \textit{subject to:} \\ & w(S) \leq L \end{aligned} \tag{6}$$

*is a* subset selection problem with a linear constraint *if $X$ is a lower ideal, namely, if $S \in X$ and $S' \subseteq S$ then $S' \in X$, and $f$ is a linear non-decreasing set function*[4] *with $f(\emptyset) = 0$.*

Note that subset selection problems with linear constraints are in the form of (1), and the Lagrangian relaxation of any problem $\Gamma$ in this class is $\Gamma(\lambda) = \max_{S \in X} f(S) - \lambda w(S)$. Hence, the results of Section 2 hold.

Thus, for example, BGAP can be formulated as the following subset selection problem with linear constraint. The universe $U$ consists of all pairs $(i, j)$ of item $1 \leq i \leq n$ and bin $1 \leq j \leq m$. The domain $X$ consists of all the subsets $S$ of $U$, such that each item appears at most once (i.e., for any item $1 \leq i \leq n$, $|\{(i', j') \in S : i' = i\}| \leq 1$), and the collection of items that appears with a bin $j$, i.e., $\{i : (i, j) \in S\}$ defines a feasible assignment of items to bin $j$. It is easy to see that $X$ is indeed a lower ideal. The function $f$ is $f(S) = \sum_{(i,j) \in S} f_{i,j}$, where $f_{i,j}$ is the profit from the assignment of item $i$ to bin $j$, and $w(S) = \sum_{(i,j) \in S} w_{i,j}$ where $w_{i,j}$ is the size of item $i$ when assigned to bin $j$.

The Lagrangian relaxation of BGAP is then $\max_{S \in X} f(S) - \lambda w(S) = \max_{S \in X} \sum_{(i,j) \in S} (f_{i,j} - \lambda w_{i,j})$. The latter can be interpreted as the following instance of GAP: if $f_{i,j} - \lambda w_{i,j} \geq 0$ then set $f_{i,j} - \lambda w_{i,j}$ to be the profit from assigning item $i$ to bin $j$; otherwise, make item $i$ infeasible for bin $j$ (set the size of item $i$ to be greater than the capacity of bin $j$).

We now show how the Lagrangian relaxation technique described in Section 2 can be applied to subset selection problems. Given a problem $\Gamma$ in this class, suppose that $\mathcal{A}$ is a $\rho$-approximation algorithm for $\Gamma(\lambda)$, for some $\rho \in (0, 1]$. To find $\lambda_1, \lambda_2$ and $SOL_1, SOL_2$, the binary search of Section 2.1 can be applied over the range $[0, p_{max}]$, where

$$p_{max} = \max_{s \in U} f(s) \tag{7}$$

is the maximum cost of any element in the universe $U$. To obtain the solutions $S_1, S_2$ which correspond to $\lambda_1, \lambda_2$, the number of calls to $\mathcal{A}$ in the binary search is bounded by $O(\log(\frac{L \cdot p_{max}}{\varepsilon}))$.

Given the solutions $S_1, S_2$ satisfying the conditions of Theorem 1, consider the case where, for some $\alpha \in [1 - \rho, 1]$, property 2 (in the theorem) holds. Denote the value of an optimal solution for $\Gamma$ by $OPT$. Given a solution $S_2$ such that

$$f(S_2) \geq (1 - \alpha - \varepsilon) \frac{w(S_2)}{L} \cdot OPT, \tag{8}$$

our goal is to find a solution $S'$ such that $w(S') \leq L$ (i.e., $S'$ is valid for $\Gamma$), and $f(S')$ is an approximation for $OPT$. We show below how $S'$ can be obtained from $S_2$. We first consider (in

[4]For simplicity, we assume throughout the discussion that $f(\cdot)$ is a *linear* function; however, all of the results in this section hold also for the more general case where $f(S)$ is a non-decreasing sub-modular set function, for any $S \in X$.

Section 3.1) instances with unit weights. We then describe (in Section 3.2) a scheme for general weights. Finally, we give (in Section 3.3) a scheme which yields improved approximation ratio for general instances, by applying enumeration.

### 3.1 Unit Weights

Consider first the special case where $w_s = 1$ for any $s \in U$ (i.e., $w(S) = |S|$; we refer to (6) in this case as *cardinality constraint*).

Suppose that we have solutions $S_1, S_2$ which satisfy the conditions of Theorem 1, then by taking $\alpha = \frac{1}{1+\rho}$ we get that either $f(S_1) \geq (\frac{\rho}{1+\rho} - \varepsilon)OPT$, or $f(S_2) \geq (\frac{\rho}{1+\rho} - \varepsilon)\frac{w(S_2)}{L}OPT$. If the former holds then we have a $(\frac{\rho}{1+\rho} - \varepsilon)$-approximation for the optimum; otherwise, $f(S_2) \geq (\frac{\rho}{1+\rho} - \varepsilon)\frac{w(S_2)}{L}OPT$. To obtain $S'$, select the $L$ elements in $S_2$ with the highest profits.[5] It follows from (8) that $f(S') \geq (1 - \alpha - \varepsilon) \cdot OPT = (\frac{\rho}{1+\rho} - \varepsilon)OPT$. Combining the above with the result of Theorem 2, we get the following.

**Theorem 3** *Given a subset selection problem $\Gamma$ with unit weights, an algorithm $\mathcal{A}$ which yields a $\rho$-approximation for $\Gamma(\lambda)$ and $\lambda_{max}$, such $w(\mathcal{A}(\lambda_{max})) \leq L$, a $(\frac{\rho}{\rho+1} - \varepsilon)$-approximation for $\Gamma$ can be derived by using $\mathcal{A}$ and selecting among $S_1, S'$ the set with highest profit. The number of calls for $\mathcal{A}$ is $O(\log(\frac{L \cdot p_{max}}{\varepsilon}))$, where $p_{max}$ is given in (7).*

### 3.2 Arbitrary Weights

For general element weights, we may assume w.l.o.g. that for any $s \in U$, $w_s \leq L$. We partition $S_2$ to a collection of up to $\frac{2W(S_2)}{L}$ disjoint sets $T_1, T_2, \ldots$ such that $w(T_i) \leq L$ for all $i \geq 1$. A simple way to obtain such sets is by adding elements of $S_2$ in arbitrary order to $T_i$ as long as we do not exceed the budget $L$. A slightly more efficient implementation has a running time that is linear in the size of $S_2$ (details omitted).

**Lemma 4** *Suppose that $S_2$ satisfies (8) for some $\alpha \in [1-\rho, 1]$, then there exists $i \geq 1$ such that $f(T_i) \geq \frac{1-\alpha-\varepsilon}{2} \cdot OPT$.*

**Proof:** Clearly, $f(T_1) + \ldots + f(T_N) = f(S_2)$, where $N \leq \frac{2w(S_2)}{L}$ is the number of disjoint sets. By the pigeon hole principle there is $1 \leq i \leq N$ such that $f(T_i) \geq \frac{f(S_2)}{N} \geq \frac{L \cdot f(S_2)}{2w(S_2)} \geq \frac{1-\alpha-\varepsilon}{2} \cdot OPT$. □

Assuming we have solutions $S_1, S_2$ which satisfy the conditions of Theorem 1, by taking $\alpha = \frac{1}{1+2\rho}$ we get that either $f(S_1) \geq (\frac{\rho}{1+2\rho} - \varepsilon)OPT$, or $f(S_2) \geq (\frac{2\rho}{1+2\rho} - \varepsilon)\frac{w(S_2)}{L}OPT$ and can be converted to $S'$ (by setting $S' = T_i$ for $T_i$ which maximizes $f(T_i)$), such that $f(S') \geq (\frac{\rho}{1+2\rho} - \varepsilon)OPT$, i.e., we get a $(\frac{\rho}{1+2\rho} - \varepsilon)$-approximation for $\Gamma$.

Combining the above with the result of Theorem 2, we get the following.

**Theorem 5** *Given a subset selection problem with a linear constraint $\Gamma$, an algorithm $\mathcal{A}$ which yields a $\rho$-approximation for $\Gamma(\lambda)$, and $\lambda_{max}$, such $w(\mathcal{A}(\lambda_{max})) \leq L$, a $(\frac{\rho}{2\rho+1} - \varepsilon)$-approximation for $\Gamma$ can be obtained using $\mathcal{A}$. The number of calls for $\mathcal{A}$ is $O(\log(\frac{L \cdot p_{max}}{\varepsilon}))$, where $p_{max}$ is given in (7).*

### 3.3 Improving the Bounds via Enumeration

In this section we present an algorithm which uses enumeration to obtain a new problem, for which we apply our Lagrangian relaxation technique. This enables to improve the approximation

[5]When $f$ is a submodular function, iteratively select the element $s \in S_2$ which maximizes $f(T \cup \{s\})$, where $T$ is the subset of elements chosen in the previous iterations.

ratio in Section 3.2 to match the bound obtained for unit weight inputs (in Section 3.1).[6]

For some $k \geq 1$, our algorithm 'guesses' a subset of elements $T$ of a size bounded by $k$, which appears in some optimal solution $S^*$ for $\Gamma$. The subset $T$ has the property that the profit of $T$ ($= f(T)$), is the highest in this optimal solution; then, we add to the solution elements in $U$ (whose contribution to the solution is bounded by $f(T)/|T|$), to obtain an approximate solution. Given a subset $T \subseteq U$, we define the problem $\Gamma_T$, which can be viewed as the problem that remains from $\Gamma$ after selection of $T$ to be part of the solution. Thus, we refer to this problem below as *residual problem with respect to T*. Let

$$X_T = \left\{ S |\ S \cap T = \emptyset, S \cup T \in X, \text{ and } \forall s \in S : f(\{s\}) \leq \frac{f(T)}{|T|} \right\} \tag{9}$$

Consider the residual problem $\Gamma_T$ and its Lagrangian relaxation $\Gamma_T(\lambda)$:

$$\begin{array}{ll} \Gamma_T & \text{maximize } f(S) \\ \text{subject to :} & S \in X_T \\ & w(S) \leq L - w(T) \end{array} \qquad \begin{array}{ll} \Gamma_T(\lambda) & \text{maximize } f(S) - \lambda w(S) \\ \text{subject to :} & S \in X_T \end{array}$$

In all of our examples, the residual problem $\Gamma_T$ is a smaller instance of the problem $\Gamma$, and therefore, its Lagrangian relaxation is an instance of the Lagrangian relaxation of the original problem. Assume that we have an approximation algorithm $\mathcal{A}$ which, given $\lambda$ and a pre-selected set $T \subseteq U$ of at most $k$ elements, for some constant $k > 1$, returns a $\rho$-approximation for $\Gamma_T(\lambda)$ in polynomial time (if there is a feasible solution for $\Gamma_T$). Consider the following algorithm, in which we take $k = 2$:

1. For any $T \subseteq U$ such that $|T| \leq k$, find solutions $S_1, S_2$ (for $\Gamma_T(\lambda_1), \Gamma_T(\lambda_2)$ respectively) satisfying the conditions of Theorem 1 with respect to the problem $\Gamma_T$. Evaluate the following solutions:
   (a) $T \cup S_1$
   (b) Let $S' = \emptyset$, add elements to $S'$ in the following manner:
   Find an element $x \in S_2 \backslash S'$ which maximizes the ratio $\frac{f(\{x\})}{w_x}$. If $w(S' \cup \{x\}) \leq L - w(T)$ then add $x$ to $S'$ and repeat the process, otherwise return $S' \cup T$ as a solution.
2. Return the best of the solutions found in Step 1.

Let $OPT = f(S^*)$ be an optimal solution for $\Gamma$, where $S^* = \{x_1, \ldots, x_h\}$. Order the elements in $S^*$ such that $f(\{x_1\}) \geq f(\{x_2\}) \geq \ldots \geq f(\{x_h\})$.

**Lemma 6** *Let $T_i = \{x_1, \ldots, x_i\}$, for some $1 < i \leq h$, then for any $j > i$, $f(\{x_j\}) \leq \frac{f(T_i)}{i}$.*

In analyzing our algorithm, we consider the iteration in which $T = T_k$. Then $S^* \setminus T_k$ is an optimal solution for $\Gamma_T$ (since $S^* \setminus T_k \in X_{T_k}$ as in (9)); thus, the optimal value for $\Gamma_{T_k}$ is at least $f(S^* \backslash T_k) = f(S^*) - f(T_k)$.

[6]The running time when applying enumeration depends on the size of the universe (which may be superpolynomial in the input size; we elaborate on that in Section 4.1).

**Lemma 7** *Let $S'$ be the set generated from $S_2$ by the process in Step 1(b) of the algorithm. Then* $f(S') \geq f(S_2)\frac{L-w(T)}{w(S_2)} - \frac{f(T)}{|T|}$

**Proof:** Note that the process cannot terminate when $S' = S_2$ since $w(S_2) > L - w(T)$. Consider the first element $x$ that maximized the ratio $\frac{f(\{x\})}{w_x}$, but was not added to $S'$, since $w(S' \cup \{x\}) > L - w(T)$. By the linearity of $f$, it is clear that

(i) $\frac{f(S'\cup\{x\})}{w(S'\cup\{x\})} \geq \frac{f(\{x\})}{w_x}$, and

(ii) For any $y \in S_2 \backslash (S' \cup \{x\})$, $\frac{f(\{y\}}{w_y} \leq \frac{f(\{x\})}{w_x}$.

Thus, we get that for any $y \in S_2 \backslash (S' \cup \{x\})$, $\frac{f(\{y\})}{w_y} \leq \frac{f(S'\cup\{x\})}{w(S'\cup\{x\})}$, and

$$f(S_2) = f(S' \cup \{x\}) + \sum_{y \in S_2 \backslash (S' \cup \{x\})} f(\{y\}) \leq f(S' \cup \{x\}) \frac{w(S_2)}{w(S' \cup \{x\})}.$$

By the linearity of $f$, we get $f(S') + f(\{x\}) = f(S' \cup \{x\}) \geq f(S_2)\frac{L-w(T)}{w(S_2)}$. Since $x \in S_2 \in X_T$, we get $f(\{x\}) \leq \frac{f(T)}{|T|}$. Hence $f(S') \geq f(S_2)\frac{L-w(T)}{w(S_2)} - \frac{f(T)}{|T|}$.

□

Consider the iteration of Step 1. in the above algorithm, in which $T = T_2$ (if there are at least two elements in the optimal solution; else $T = T_1$), and the values of the solutions found in this iteration. By Theorem 1, taking $\alpha = \frac{1}{1+\rho}$, one of the following holds:

1. $f(S_1) \geq \frac{\rho}{1+\rho}[f(S^*) - f(T)]$
2. $f(S_2) \geq (1 - \rho - \varepsilon)[f(S^*) - f(T)]\frac{w(S_2)}{L-w(T)}$.

If 1. holds then we get $f(S_1 \cup T) \geq f(T) + \frac{\rho}{1+\rho}[f(S^*) - f(T)] \geq (\frac{\rho}{1+\rho} - \varepsilon)f(S^*)$, else we have that $f(S_2) \geq (\frac{\rho}{1+\rho} - \varepsilon)[f(S^*) - f(T)]\frac{w(S_2)}{L-w(T)}$, and by Lemma 7,

$$f(S') \geq f(S_2)\frac{L-w(T)}{w(S_2)} - \frac{f(T)}{|T|} \geq (\frac{\rho}{1+\rho} - \varepsilon)[f(S^*) - f(T)] - \frac{f(T)}{|T|}.$$

Hence, we have

$$\begin{aligned} f(S' \cup T) &= f(S') + f(T) \geq f(T) + (\frac{\rho}{1+\rho} - \varepsilon)[f(S^*) - f(T)] - \frac{f(T)}{|T|} \\ &= (1 - \frac{1}{k})f(T) + (\frac{\rho}{1+\rho} - \varepsilon)[f(S^*) - f(T)] \geq (\frac{\rho}{1+\rho} - \varepsilon)f(S^*). \end{aligned}$$

The last inequality follows from choosing $k = 2$, and the fact that $\frac{1}{2} \geq \frac{\rho}{1+\rho} - \varepsilon$.

**Theorem 8** *The algorithm outputs a $(\frac{\rho}{1+\rho} - \varepsilon)$-approximation for $\Gamma$. The number of calls to algorithm $\mathcal{A}$ is $O((\log(p_{max}) + \log(L) + \log(\varepsilon^{-1}))n^2)$, where $n = |U|$ is the size of the universe of elements for the problem $\Gamma$.*

**Submodular objective functions:** In the more general case, where $f$ is a submodular function, we need to re-define the objective function for $\Gamma_T$ to be $f_T(S') = f(S' \cup T) - f(T)$, and the condition $f(\{s\}) \leq \frac{f(T)}{|T|}$ should be modified to $f_T(\{s\}) \leq \frac{f(T)}{|T|}$. In Step 1(b) of the algorithm, the element $x$ to be chosen in each stage is $x \in S_2 \setminus S'$ which maximizes the ratio $\frac{f_T(S'\cup\{x\}) - f_T(S')}{w_x}$.

# 4 Applications

In this section we show how the technique of Section 3 can be applied to obtain approximation algorithms for several classic subset selection problems with linear constraint.

## 4.1 Budgeted Real Time Scheduling

The budgeted real-time scheduling problem can be interpreted as the following subset selection problem with linear constraint. The universe $U$ consists of all instances associated with the activities $\{A_1, \ldots, A_m\}$. The domain $X$ is the set of all feasible schedules; $f(S)$ (where $S \in X$) is the profit from the instances in $S$, and $w(S)$ is the total cost of the instances in $S$ (note that each instance is associated with specific time interval). The Lagrangian relaxation of this problem is the classic *interval scheduling* problem discussed in [2]: the paper gives a $\frac{1}{2}$-approximation algorithm, whose running time is $O(n \log n)$, where $n$ is the total number of instances in the input. Clearly, $p_{\max}$ (as defined in (7)) can be used as $\lambda_{\max}$. By Theorem 2, we can find two solutions $S_1, S_2$ which satisfy the conditions of Theorem 1 in $O(n \log(n) \log(Lp_{\max}/\varepsilon))$ steps. Then, a straightforward implementation of the technique of Section 3.1 yields a $\left(\frac{1}{3} - \varepsilon\right)$-approximation algorithm whose running time is $O(n \log(n) \log(Lp_{\max}/\varepsilon))$ for inputs where all instances have *unit* cost. The same approximation ratio can be obtained in $O(n^3 \cdot \log(n) \log(Lp_{\max}/\varepsilon))$ steps when the instances may have *arbitrary* costs, using Theorem 8 (Note that the Lagrangian relaxation of the residual problem with respect to a subset of elements $T$ is also an instance of the interval scheduling problem.)

Consider now the continuous case, where each instance within some activity $A_i$, $1 \leq i \leq m$, is given by a time window. One way to interpret BCRS as a subset selection problem is by setting the universe to be all the pairs of an instance and a time interval in which it can be scheduled. The size of the resulting universe is unbounded: a more careful consideration of all possible start times of any instance yields a universe of super-polynomial size. The Lagrangian relaxation of this problem is known as *single machine scheduling with release time and deadlines*, for which a $(\frac{1}{2} - \varepsilon)$-approximation algorithm is given in [2]. Thus, we can apply our technique for finding two solutions $S_1, S_2$ for which Theorem 1 holds. However, the running time of the algorithm in Theorem 8 may be super-polynomial in the input size (since the number of the enumeration steps depends on the size of the universe, which may be exponentially large). Thus, we derive an approximation algorithm using the technique of Section 3.2. We summarize in the next result.

**Theorem 9** *There is a polynomial time algorithm that yields an approximation ratio of* $(\frac{1}{3} - \varepsilon)$ *for BRS and the ratio* $\left(\frac{1}{4} - \varepsilon\right)$ *for BCRS.*

Our results also hold for other budgeted variants of problems that appear in [2]. For the case where all intervals have the same (unit) profit, an approximation ratio arbitrarily close to $\frac{1-e^{-1}}{2-e^{-1}}$ can be obtained by using an algorithm of [6].

## 4.2 The Budgeted Generalized Assignment Problem

Consider the interpretation of GBAP as a subset selection problem, as given in Section 3. The Lagrangian relaxation of BGAP (and also of the deduced residual problems) is an instance of GAP, for which the paper [10] gives a $(1 - e^{-1} - \varepsilon)$-approximation algorithm. We can take in Theorem 2 $\lambda_{\max} = p_{\max}$, where $p_{\max}$ is defined by (7), and the two solutions $S_1, S_2$ that satisfy the condition of Theorem 1 can be found in polynomial time. Applying the techniques of Sections 3.1 and 3.3, we get the next result.

**Theorem 10** *There is a polynomial time algorithm that yields an approximation ratio of* $\frac{1-e^{-1}}{2-e^{-1}} - \varepsilon \approx 0.387 - \varepsilon$ *for BGAP.*

A slightly better approximation ratio can be obtained by using an algorithm of [9]. More generally, our result holds also for any constrained variant of the *separable assignment problem (SAP)* that can be solved using a technique of [10].

### 4.3 Budgeted Maximum Weight Independent Set

BWIS can be interpreted as the following subset selection problem with linear constraint. The universe $U$ is the set of all vertices in the graph, i.e., $U = V$, the domain $X$ consists of all subsets $V'$ of $V$, such that $V'$ is an independent set in the given graph $G$. The objective function $f$ is $f(V') = \sum_{v \in V'} p_v$, the weight function is $w(V') = \sum_{v \in V'} c_v$, and the weight bound is $L$. The Lagrangian relaxation of BWIS is an instance of the classic WIS problem (vertices with negative profits in the relaxation are deleted, along with their edges). Thus, given an approximation algorithm $\mathcal{A}$ for WIS with approximation ratio $f(n)$ ($n$ is the number of vertices in the graph), by Theorem 8, the technique of Section 3.3 yields an approximation algorithm $\mathcal{A}_I$ for $BWIS$. $\mathcal{A}_I$ achieves the approximation $\frac{f(n)}{1+f(n)} - \varepsilon$, and its running time is polynomial in the input size and in $\log(1/\varepsilon)$. If $\log(1/f(n))$ is polynomial, select $\varepsilon = \frac{f(n)}{n}$; the value $\log(1/\varepsilon) = \log(1/f(n)) + \log(n)$ is polynomial in the input size; thus, the algorithm remains polynomial. For this selection of $\varepsilon$, we have the following result.

**Theorem 11** *Given an $f(n)$-approximation algorithm for WIS, where $f(n) = o(n)$, for any $L \geq 1$ there exists a polynomial time algorithm that outputs a $g(n)$-approximation ratio for any instance of BWIS with the budget $L$, where $g(n) = \Theta(f(n))$, and $\lim_{n \to \infty} \frac{g(n)}{f(n)} = 1$.*

This means that the approximation ratios of $\mathcal{A}$ and $\mathcal{A}_I$ are asymptotically the same. Thus, for example, using the algorithm of [14], our technique achieves an $\Omega(\frac{\log^2 n}{n})$-approximation for BWIS.

Note that the above result holds for any constant number of linear constraints added to an input for WIS, by repeatedly applying our Lagrangian relaxation technique.

## References


[1] N. Alon, U. Feige, A. Wigderson, and D. Zuckerman. Derandomized graph products. *Computational Complexity*, 5(1):60–75, 1995.

[2] A. Bar-Noy, R. Bar-Yehuda, A. Freund, J. Naor, and B. Schieber. A unified approach to approximating resource allocation and scheduling. *Journal of the ACM*, pages 1–23, 2000.

[3] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Approximating the throughput of multiple machines in real-time scheduling. *SIAM J. Comput.*, 31(2):331–352, 2001.

[4] A. Berger, V. Bonifaci, F. Grandoni, and G. Schäfer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. In *IPCO*, 2008.

[5] C. Chekuri and S. Khanna. A ptas for the multiple knapsack problem. *SIAM J. Comput.*, 35(3):713–728, 2006.

[6] J. Chuzhoy, R. Ostrovsky, and Y. Rabani. Approximation algorithms for the job interval selection problem and related scheduling problems. *Math. Oper. Res.*, 31(4):730–738, 2006.

[7] B.-Y. R. M. B. Y. Cohen and D. Rawitz. Resource allocation in bounded degree trees. *Algorithmica*, To appear.

[8] R. Cohen, L. Katzir, and D. Raz. An efficient approximation for the generalized assignment problem. *Inf. Process. Lett.*, 100(4):162–166, 2006.

[9] U. Feige and J. Vondrák. Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. In *FOCS*, pages 667–676, 2006.

[10] L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko. Tight approximation algorithms for maximum general assignment problems. In *SODA*, pages 611–620, 2006.

[11] N. Garg. A 3-approximation for the minimum tree spanning k vertices. In *FOCS*, 1996.

[12] M. X. Goemans and R. Ravi. The constrained minimum spanning tree problem. In *SWAT*, pages 66–75, 1996.

[13] M. M. Halldórsson. Approximations of independent sets in graphs. In *APPROX*, 1998.

[14] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms Appl.*, 4(1), 2000.

[15] M. M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *Journal of Graph Algorithms and Applications*, pages 1–16, 2000.

[16] K. Jain and V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, pages 274–296, 2001.

[17] J. Könemann, O. Parekh, and D. Segev. A unified approach to approximating partial covering problems. In *ESA'06: Proceedings of the 14th conference on Annual European Symposium*, pages 468–479, London, UK, 2006. Springer-Verlag.

[18] J. Könemann and R. Ravi. A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. *SIAM J. Comput.*, 31(6):1783–1793, 2002.

[19] J. Mestre. *Primal-Dual Algorithms for Combinatorial Optimization Problems.* PhD thesis, CS Dept, Univ. of Maryland, 2007.

[20] J. Naor, H. Shachnai, and T. Tamir. Real-time scheduling with a budget. *Algorithmica*, pages 343–364, 2007.

[21] G. Schäfer. *Private communication*, 2007.

[22] L. A. Wolsey and G. L. Nemhauser. *Integer and Combinatorial Optimization.* Wiley-Interscience, 1999.

# A Lagrangian Relaxation – An Example

Consider the following problem. Given is a base set of elements $A$, where each element $a \in A$ has a (non-negative integral) profit $p(a)$; also, given are three subsets of elements $A_1, A_2, A_3 \subseteq A$, and a bound $k > 1$. We need to select a subset $S \subseteq A$ of size at most $k$, such that $S \subseteq A_1$, or $S \subseteq A_2$, or $S \subseteq A_3$, and the total profit ($\sum_{a \in S} p(a)$) from elements in $S$ is maximized. The problem can be easily interpreted as a subset selection problem, by taking the universe to be

$U = A$, the domain $X$ consists of all the subsets $S$ of $U$, such that $S \subseteq A_1$ or $S \subseteq A_2$, or $S \subseteq A_3$. The weight function is $w(S) = |S|$, with the weight bound $L = k$, and the profit of a subset $S$ is given by $f(S) = \sum_{a \in S} p(a)$.

The Lagrangian relaxation of the problem with parameter $\lambda$ is $\max_{S \in X} f(S) - \lambda w(S)$. Assume that we have an algorithm $\mathcal{A}$ which returns a $\rho$-approximation for the Lagrangian relaxation of the problem.

For any $\frac{1}{2} > \delta > 0$ and an integer $k > \frac{1}{\rho} + 4$, consider the following input:

- $A_1 = \{a_1, \ldots, a_{k-1}, b\}$, where $p(a_i) = \frac{1}{\rho}$ for $1 \leq i \leq k-1$, and $p(b) = k-1$.
- $A_2 = \{c\}$ where $p(c) = k + \delta$.
- $A_3 = \{d_1, \ldots, d_\ell\}$ where $\ell = \lceil \frac{(1+\rho)(k-1)}{\delta\rho} \rceil$, and $p(d_i) = 1 + \delta$ for $1 \leq i \leq \ell$.
- $U = A = A_1 \cup A_2 \cup A_3$, and the set $S$ to be chosen is of size at most $k$.

Denote the profit from a subset $S \subseteq U$ by $p(S)$, and the profit in the Lagrangian relaxation with parameter $\lambda$ by $p_\lambda(S)$. Clearly, the subset $S = A_1$ is an optimal solution for the problem, with the profit $p(A_1) = (k-1)\frac{1+\rho}{\rho}$. Consider the possible solutions the algorithm $\mathcal{A}$ returns for different values of $\lambda$:

- For $\lambda < 1$: the profit from any subset of $A_1$ is bounded by the original profit of $A_1$, given by $p(A_1) = (k-1)\frac{1+\rho}{\rho}$; the profit from the set $S = A_3$ is equal to $p(A_3) = (1 + \delta - \lambda)\ell \geq \delta\ell \geq (k-1)\frac{(1+\rho)}{\rho}$, i.e., $A_3$ has higher profit than $A_1$.
- For $1 \leq \lambda \leq \frac{1}{\rho}$: the profit from any subset of $A_1$ is bounded by the total profit of $A_1$ (all elements are of non-negative profit in the relaxation). Consider the difference:

$$\begin{aligned}
\rho \cdot p_\lambda(A_1) - p_\lambda(A_2) &= \rho \left( k - 1 - \lambda + (k-1)(\frac{1}{\rho} - \lambda) \right) - (k - \lambda) = \\
&= \rho k - \rho - \rho\lambda + k - -\rho\lambda k + \rho k - k + \lambda = (1-\lambda)(\rho k - 1) - \rho \leq 0
\end{aligned}$$

  This implies that in case the optimal set is $A_1$ (or a subset of $A_1$), the algorithm $\mathcal{A}$ may choose the set $A_2$.
- For $\lambda > \frac{1}{\rho}$: the maximal profit from any subset of $A_1$ is bounded in this case by $\max\{k - 1 - \lambda, 0\}$, whereas the profit from $A_2$ is $\max\{k - \lambda, 0\}$.

From the above, we get that $\mathcal{A}$ may return a subset of $A_2$ or $A_3$ for any value of $\lambda$. However, no combination of elements of $A_2$ and $A_3$ yields a solution for the original problem with profit greater than $k(1+\delta)$. This means that, by combining the solutions returned by the Lagrangian relaxation, one cannot achieve approximation ratio better than $\frac{k(1+\delta)}{(k-1)(1+\frac{1}{\rho})} = \frac{\rho}{1+\rho} \cdot \frac{k(1+\delta)}{k-1}$. Since $\frac{\rho}{1+\rho} \cdot \frac{k(1+\delta)}{k-1} \to \frac{\rho}{1+\rho}$ for $(k, \delta) \to (\infty, 0)$, one cannot achieve approximation ratio better than $\frac{\rho}{1+\rho}$.

# B Solving Multi-budgeted Subset Selection Problems

In the following we extend our technique as given in Section 3 to handle subset selection problems with $d$ linear constraints, for some $d > 1$. More formally, consider the problem:

$$\begin{aligned}
\max_{S \in X} \quad & f(S) \quad \text{subject to:} \\
\forall_{1 \leq i \leq d}: \quad & w_i(S) \leq L_i,
\end{aligned} \tag{10}$$

where $X$ is a lower ideal, and the functions $f$ and $w_i$ for every $1 \leq i \leq d$ are non-decreasing linear set function, such that $f(\emptyset) = w_i(\emptyset) = 0$. This problem can be interpreted as the following subset selection problem with (a single) linear constraint. Let $X' = \{S \in X | \forall_{1 \leq i \leq d-1} : \ w_i(S) \leq L_i\}$; the linear constraint is $w_d(S) \leq L_d$, the function $f$ remains as defined above. The Lagrangian relaxation of (10) has the same form (after removing in the relaxation elements with negative profits), but the number of constraints is now $d - 1$. This implies that, by repeatedly applying the technique in Section 3.3, we can obtain an approximation algorithm for (10) from an approximation algorithm for the non-constrained problem (in which we want to find $\max_{S \in X} f'(X)$, where $f'$ is some linear function). Thus, given a $\rho$-approximation algorithm for the problem after "relaxing" $d$ constraints, we derive a $(\frac{\rho}{1+d\rho} - \varepsilon)$-approximation algorithm for (10).

Note that there is a simple reduction[7] of the problem in (10) to the same problem with $d = 1$, which yields a $\frac{\rho}{d}$-approximation for (10), given a $\rho$-approximation algorithm $\mathcal{A}$ for the problem with single constraint. For sufficiently small $\varepsilon > 0$, the ratio of $\frac{\rho}{1+(d-1)\rho} - \varepsilon$ obtained by repeatedly applying Lagrangian relaxation and using the approximation algorithm $\mathcal{A}$ is better, for any $\rho \in (0, 1)$.

[7]Assume w.l.o.g that $L_i = 1$ for every $1 \leq i \leq d$, and set the weight of an element $e$ to be $w_e = \max_{1 \leq i \leq d} w_i(\{e\})$