Primal-Dual Schema for Capacitated Covering Problems*

Tim Carnes and David Shmoys

Cornell University, Ithaca NY 14853, USA

Abstract. Primal-dual algorithms have played an integral role in recent developments in approximation algorithms, and yet there has been little work on these algorithms in the context of LP relaxations that have been strengthened by the addition of more sophisticated valid inequalities. We introduce primal-dual schema based on the LP relaxations devised by Carr, Fleischer, Leung & Phillips for the minimum knapsack problem as well as for the single-demand capacitated facility location problem. Our primal-dual algorithms achieve the same performance guarantees as the LP-rounding algorithms of Carr et al., which rely on applying the ellipsoid algorithm to an exponentially-sized LP. Furthermore, we introduce new flow-cover inequalities to strengthen the LP relaxation of the more general capacitated single-item lot-sizing problem; using just these inequalities as the LP relaxation, we obtain a primal-dual algorithm that achieves a performance guarantee of 2.

1 Introduction

Primal-dual algorithms have played an integral role in recent developments in approximation algorithms, and yet there has been little work on these algorithms in the context of LP relaxations that have been strengthened by the addition of more sophisticated valid inequalities. We introduce primal-dual schema based on the LP relaxations devised by Carr, Fleischer, Leung & Phillips [5] for the minimum knapsack problem as well as for the single-demand capacitated facility location problem. Our primal-dual algorithms achieve the same performance guarantees as the LP-rounding algorithms of Carr et al., which rely on applying the ellipsoid algorithm to an exponentially-sized LP. Furthermore, we introduce new flow-cover inequalities to strengthen the LP relaxation of the more general capacitated single-item lot-sizing problem; using just these inequalities as the LP relaxation, we obtain a primal-dual algorithm that achieves a performance guarantee of 2.

We say an algorithm is an approximation algorithm with a performance guarantee of α when the algorithm runs in polynomial time and always produces a solution with cost within a factor of α of optimal. Primal-dual algorithms are able to gain the benefits of LP-based techniques, such as automatically generating a new lower bound for each problem instance, but without having to solve

* Research supported partially by NSF grants CCR-0635121, CCR-0430682 & DMI-0500263.

an LP. Primal-dual approximation algorithms were developed by Bar-Yehuda & Even and Chvátal for the weighted vertex cover and set cover problems, respectively [3, 7]. Subsequently, this approach has been applied to many other combinatorial problems, such as results of Agrawal, Klein & Ravi [2], Goemans & Williamson [10], Bertsimas & Teo [4] and Levi, Roundy & Shmoys [12] for other related covering problems. Other recent work has been done on covering problems with capacity constraints by Even, Levi, Rawitz, Schieber, Shahar & Sviridenko [8], Chuzhoy & Naor [6] and Gandhi, Halperin, Khuller, Kortsarz & Srinivasan [9].

In developing primal-dual (or any LP-based) approximation algorithms, it is important to have a strong LP formulation for the problem. However, there are cases when the LP relaxation of the natural IP formulation for a problem has a large integrality gap. When this happens, one can mend the situation by introducing extra *valid* inequalities that hold for any solution to the problem, but restrict the feasible region of the LP. One class of valid inequalities that has proved useful for a variety of problem are called *flow-cover inequalities*. A large class of flow-cover inequalities was developed by Aardal, Pochet & Wolsey [1] for the capacitated facility location problem. Carr et al. [5] developed a different style of flow-cover inequalities for simpler capacitated covering problems which we use in developing our primal-dual algorithms. Levi, Lodi & Sviridenko [11] used a subset of the flow-cover inequalities of Aardal et al. [1] to develop a LP-rounding algorithm for the multiple-item lot-sizing problem with monotone holding costs. Our model is not a special case of theirs, however, since our result allows time-dependent order capacitites whereas their result assumes constant order capacities across all periods. Following this work and the development of our own flow-cover inequalities for lot-sizing problems, Sharma & Williamson [13] demonstrated that the result of Levi et al. [11] has an analogue based on our flow-cover inequalities as well. Finally, it is worth noting that Van Hoesel & Wagelmans [14] have a FPTAS for the single-item lot-sizing problem that makes use of dynamic programming and input data rounding. The disadvantage of this result is that the running time of the algorithm can be quite slow for particular performance guarantees. Although the primal-dual algorithm we develop is a 2-approximation algorithm, this is a worst-case bound and we would expect it to perform much better on average in practice. Also this is the first LP-based result for the case of time-dependent capacities.

The three models studied in this paper are generalizations of one another. That is to say, the minimum knapsack problem is a special case of the single-demand capacitated facility location problem, which is a special case of the single-item lot-sizing problem. We present the result in order of generality with the aim of explaining our approach in the simplest setting first. The *minimum knapsack problem* gives a set of items, each with a weight and a value. The objective is to find a minimum-weight subset of items such that the total value of the items in the subset meets some specified demand. In the *single-demand facility location problem* there is a set of facilities, each with an opening cost and a capacity, as well as a per-unit serving cost that must be paid for each unit of demand a facility serves. The goal is to open facilities to completely serve a specified amount of demand, while minimizing the total service and facility opening costs. Finally the *single-item lot-sizing problem* considers a finite planning period of consecutive time periods. In each time period there is a specified level of demand, as well as a potential order with a given capacity and order opening cost. A feasible solution must open enough orders and order enough inventory so that in each time period there is enough inventory to satisfy the demand of that period. The inventory is simply the inventory of the previous time period plus however much is ordered in the current time period, if any. However, a per-unit holding cost is incurred for each unit of inventory held over a given time period.

The straightforward LP relaxations for these problems have a bad integrality gap, but can be strengthened by introducing valid flow-cover inequalities. The inequalities Carr et al. [5] developed for the minimum knapsack problem are as follows

$$\sum_{i \in F \setminus A} u_i(A) y_i \geq D - u(A) \qquad \forall A \subseteq F,$$

where the y_i are the binary decision variables indicating if item i is chosen, $u(A)$ is the total value of the subset of items A, and the $u_i(A)$ can be thought of as the effective value of item i with respect to A, which is the minimum of the actual value and the right-hand-side of the inequality. These inequalities arise by considering that if we did choose all items in the set A, then we still have an induced subproblem on all of the remaining items, and the values can be truncated since we are only concerned with integer solutions. Our primal-dual algorithm works essentially as a modified greedy algorithm, where at each stage the item is selected that has the largest value per cost. Instead of the actual values and costs, however, we use the effective values and the slacks of the dual constraints as costs. Similar to the greedy algorithm for the traditional maximum-value knapsack problem, the last item selected and everything selected beforehand, can each be bounded in cost by the dual LP value, yielding a 2-approximation.

The remainder of this paper is organized as follows. In Section 2 we go over the minimum knapsack result in more detail. In Section 3 we generalize this result to apply to the single-demand capacitated facility location problem. Finally in Section 4 we generalize the flow-cover inequalities to handle the lot-sizing problem, and then present and analyze a primal-dual algorithm for the single-item lot-sizing problem.

2 Minimum Knapsack

In the *minimum knapsack problem* one is given a set of items F, and each item $i \in F$ has a value u_i and a weight f_i. The goal is to select a minimum weight subset of items, $S \subseteq F$, such that the value of S, $u(S)$, is at least as big as a specified demand, D. The natural IP formulation for this problem is

$$\begin{aligned} \text{opt}_{MK} := \min & \sum_{i \in F} f_i y_i && && \text{(MK-IP)} \\ \text{s.t.} & \sum_{i \in F} u_i y_i \geq D && && (1) \\ & y_i \in \{0,1\} && \forall i \in F, \end{aligned}$$

where the y_i variables indicate if item i is chosen. The following example from [5] demonstrates that the integrality gap between this IP and the LP relaxation is at least as bad as D. Consider just 2 items where $u_1 = D - 1$, $f_1 = 0$, $u_2 = D$ and $f_2 = 1$. The only feasible integer solution chooses both items and has a cost of 1, whereas the LP solution can set $y_1 = 1$ and $y_2 = 1/D$ and incurs a cost of only $1/D$. To remedy this situation we consider using the flow-cover inequalities introduced in [5].

The idea is to consider a subset of items $A \subseteq F$ such that $u(A) < D$, and let $D(A) = D - u(A)$. This means that even if all of the items in the set A are chosen, we must choose enough items in $F \setminus A$ such that the demand $D(A)$ is met. This is just another minimum knapsack problem where the items are restricted to $F \setminus A$ and the demand is now $D(A)$. The value of every item can be restricted to be no greater than the demand without changing the set of feasible integer solutions, so let $u_i(A) = \min\{u_i, D(A)\}$. This motivates the following LP

$$\begin{aligned} \text{opt}_{MKP} := \min & \sum_{i \in F} f_i y_i && && \text{(MK-P)} \\ \text{s.t.} & \sum_{i \in F \setminus A} u_i(A) y_i \geq D(A) && \forall A \subseteq F && (2) \\ & y_i \geq 0 && \forall i \in F, \end{aligned}$$

and by the validity of the flow-cover inequalities argued above we have that every feasible integer solution to (MK-IP) is a feasible solution to (MK-P). The dual of this LP is

$$\begin{aligned} \text{opt}_{MKD} := \max & \sum_{A \subseteq F} D(A) v(A) && && \text{(MK-D)} \\ \text{s.t.} & \sum_{A \subseteq F : i \notin A} u_i(A) v(A) \leq f_i && \forall i \in F && (3) \\ & v(A) \geq 0 && \forall A \subseteq F. \end{aligned}$$

Our primal-dual algorithm begins by initializing all of the primal and dual variables to zero, which produces a feasible dual solution and an infeasible primal integer solution. Taking our initial subset of items, A, to be the empty set, we increase the dual variable $v(A)$. Once a dual constraint becomes tight, the item corresponding to that constraint is added to the set A, and we now increase the new variable $v(A)$. Note that increasing $v(A)$ does not increase the left-hand-sides of dual constraints corresponding to items in A, so dual feasibility will not be violated. This process is repeated as long as $D(A) > 0$, and once we finish we call our final set of items S, which is our integer solution. This is a feasible solution to (MK-IP) since $D(S) \leq 0$, which implies $u(S) \geq D$.

Algorithm 1: Primal-Dual for Minimum Knapsack

```
y, v ← 0
A ← ∅
while D(A) > 0 do
    Increase v(A) until a dual constraint becomes tight for item i
    y_i ← 1
    A ← A ∪ {i}
S ← A
```

Theorem 1. *Algorithm 1 terminates with a solution of cost no greater than* $2 \cdot opt_{MK}$.

Proof. Let ℓ denote the final item selected by Algorithm 1. Then because the algorithm only continues running as long as $D(A) > 0$ we have that

$$D(S \setminus \{\ell\}) > 0 \Rightarrow D - u(S \setminus \{\ell\}) > 0 \Rightarrow u(S \setminus \{\ell\}) < D.$$

Also, the variable $v(A)$ is positive only if $A \subseteq S \setminus \{\ell\}$. Thus, since the algorithm only selects items for which the constraint (3) has become tight, we have that the cost of the solution produced is

$$\sum_{i \in F} f_i y_i = \sum_{i \in S} f_i = \sum_{i \in S} \sum_{A \subseteq F : i \notin A} u_i(A) v(A).$$

The expression on the right-hand side is summing over all items, i, in the final solution S, and all subsets of items, A, that do not contain item i. This is the same as summing over all subsets of items, A, and all items in the final solution that are not in A. Thus we can reverse the order of the summations to obtain

$$\begin{aligned}
\sum_{i \in F} f_i y_i &= \sum_{A \subseteq F} v(A) \sum_{i \in S \setminus A} u_i(A) \\
&= \sum_{A \subseteq F} v(A)(u(S \setminus \{\ell\}) - u(A) + u_\ell(A)) \\
&< \sum_{A \subseteq F} v(A)(D - u(A) + u_\ell(A)),
\end{aligned}$$

where the inequality follows by making use of our observation above that $u(S \setminus \{\ell\}) < D$. But we also have $u_\ell(A) \leq D(A)$ by definition, hence

$$\sum_{i \in F} f_i y_i \leq \sum_{A \subseteq F} 2D(A)v(A) \leq 2 \cdot \text{opt}_{MKD}.$$

□

3 Single-Demand Facility Location

In the *single-demand facility location problem*, one is given a set of facilities F, where each facility $i \in F$ has capacity u_i, opening cost f_i, and there is a per-unit cost c_i to serve the demand, which requires D units of the commodity. The goal is to select a subset of facilities to open, $S \subseteq F$, such that the combined cost of opening the facilities and serving the demand is minimized. The natural IP formulation for this problem is

$$\begin{aligned}
\text{opt}_{FL} := \min \sum_{i \in F} (f_i y_i + D c_i x_i) & & \text{(FL-IP)} \\
\text{s.t.} \sum_{i \in F} x_i = 1 & & (4) \\
u_i y_i \geq D x_i & & (5) \\
y_i \geq x_i & \quad \forall i \in F & (6) \\
y_i \in \{0, 1\} & \quad \forall i \in F & \\
x_i \geq 0 & \quad \forall i \in F, &
\end{aligned}$$

where each y_i indicates if facility $i \in F$ is open and each x_i indicates the fraction of D being served by facility $i \in F$. The same example from the minimum knapsack problem also demonstrates the large integrality gap of this IP. We once again turn to the flow-cover inequalities introduced by Carr et al. [5].

For these inequalities, we once again consider a subset of facilities $A \subseteq F$ such that $u(A) < D$, and let $D(A) = D - u(A)$. This means that even if all of the facilities in the set A are opened, we must open enough facilities in $F \setminus A$ such that we will be able to assign the remaining demand $D(A)$. But certainly for any feasible integer solution, a facility $i \in F \setminus A$ cannot contribute more than $\min\{Dx_i, u_i(A)y_i\}$ towards the demand $D(A)$. So if we partition the remaining orders of $F \setminus A$ into two sets F_1 and F_2, then for each $i \in F_1$ we will consider its contribution as Dx_i, and for each $i \in F_2$ we will consider its contribution as $u_i(A)y_i$. The total contribution of these facilities must be at least $D(A)$, so if we let $\mathcal{F}$ be the set of all 3-tuples that partition F into three sets, we obtain the following LP

$$\begin{aligned}
\text{opt}_{FLP} := \min \sum_{i \in F} (f_i y_i + D c_i x_i) & & \text{(FL-P)} \\
\text{s.t.} \sum_{i \in F_1} D x_i + \sum_{i \in F_2} u_i(A) y_i \geq D(A) & \quad \forall (F_1, F_2, A) \in \mathcal{F} & (7) \\
x_i, y_i \geq 0 & \quad \forall i \in F, &
\end{aligned}$$

and by the validity of the flow-cover inequalities argued above we have that every feasible integer solution to (FL-IP) is a feasible solution to (FL-P). The dual of this LP is

$$\begin{aligned} \text{opt}_{FLD} := \max & \sum_{(F_1,F_2,A)\in\mathcal{F}} D(A)v(F_1,F_2,A) && \text{(FL-D)} \\ \text{s.t.} & \sum_{(F_1,F_2,A)\in\mathcal{F}:i\in F_1} Dv(F_1,F_2,A) \le Dc_i && \forall i \in F \quad (8) \\ & \sum_{(F_1,F_2,A)\in\mathcal{F}:i\in F_2} u_i(A)v(F_1,F_2,A) \le f_i && \forall i \in F \quad (9) \\ & v(F_1,F_2,A) \ge 0 && \forall (F_1,F_2,A) \in \mathcal{F}. \end{aligned}$$

As in Section 2 the primal-dual algorithm begins with all variables at zero and an empty subset of facilities, A. Before a facility is added to A, we will require that it become tight on both types of dual constraints. To achieve this we will leave each facility in F_1 until it becomes tight on constraint (8), move it into F_2 until it is also tight on constraint (9), and only then move it into A. As before the algorithm terminates once the set A has enough capacity to satisfy the demand, at which point we label our final solution S.

Algorithm 2: Primal-Dual for Single-Demand Facility Location

```
x, y, v ← 0
F_1 ← F
F_2, A ← ∅
while D(A) > 0 do
    Increase v(F_1, F_2, A) until a dual constraint becomes tight for facility i
    if i ∈ F_1 then                          /* i tight on (8) but not on (9) */
        Move i from F_1 into F_2
    else                                      /* else i tight on (8) and (9) */
        x_i ← u_i(A)/D
        y_i ← 1
        Move i from F_2 into A
S ← A
```

Clearly Algorithm 2 terminates with a feasible solution to (FL-IP) since all of the demand is assigned to facilities that are fully opened.

Theorem 2. *Algorithm 2 terminates with a solution of cost no greater than* $2 \cdot opt_{FL}$.

Proof. Let ℓ denote the final facility selected by Algorithm 2. By the same reasoning as in Section 2 we have

$$D(S \setminus \{\ell\}) > 0 \Rightarrow D - u(S \setminus \{\ell\}) > 0 \Rightarrow u(S \setminus \{\ell\}) < D.$$

The variable $v(F_1, A)$ is positive only if $A \subseteq S \setminus \{\ell\}$. If a facility is in S then it must be tight on both constraints (8) and (9) so

$$\sum_{i\in F}(f_i y_i + Dc_i x_i) = \sum_{i\in S}(f_i + Dc_i x_i)$$
$$= \sum_{i\in S}\left[\sum_{(F_1,F_2,A)\in\mathcal{F}:i\in F_2} u_i(A)v(F_1,F_2,A) + x_i \sum_{(F_1,F_2,A)\in\mathcal{F}:i\in F_1} Dv(F_1,F_2,A)\right],$$

as in Section 2 we can simply reverse the order of summation to get

$$\sum_{i\in F}(f_i y_i + Dc_i x_i) = \sum_{(F_1,F_2,A)\in\mathcal{F}} v(F_1,F_2,A)\left[\sum_{i\in S\cap F_2} u_i(A) + \sum_{i\in S\cap F_1} Dx_i\right].$$

Recall that at the last step of Algorithm 2, facility ℓ was assigned $D(S \setminus \{\ell\})$ amount of demand. Since $D(A)$ only gets smaller as the algorithm progresses, we have that regardless of what summation above the facility ℓ is in, it contributes no more than $D(A)$. All of the other terms can be upper bounded by the actual capacities and hence

$$\sum_{i\in F}(f_i y_i + Dc_i x_i) = \sum_{(F_1,F_2,A)\in\mathcal{F}} v(F_1,F_2,A)\left[u(S\setminus\{\ell\}) - u(A) + D(A)\right]$$
$$< \sum_{(F_1,F_2,A)\in\mathcal{F}} 2D(A)v(F_1,F_2,A) \le 2\cdot \mathrm{opt}_{FLD},$$

where the strict inequality above follows from the observation made earlier. □

4 Single-Item Lot-Sizing with Linear Holding Costs

In the single-item lot-sizing problem, one is given a planning period consisting of time periods $F := \{1,\ldots,T\}$. For each time period $t \in F$, there is a demand, d_t, and a potential order with capacity u_t, which costs f_t to place, regardless of the amount of product ordered. At each period, the total amount of product left over from the previous period plus the amount of product ordered during this period must be enough to satisfy the demand of this period. Any remaining product is held over to the next period, but incurs a cost of h_t per unit of product stored. If we let

$$h_{st} = \sum_{r=s}^{t-1} h_r$$

and set $h_{tt} = 0$ for all $t \in F$, then we obtain a standard IP formulation for this problem as follows

$$\text{opt}_{LS} := \min \sum_{s=1}^{T} f_s y_s + \sum_{s=1}^{T} \sum_{t=s}^{T} h_{st} d_t x_{st} \tag{LS-IP}$$

$$\text{s.t.} \sum_{s=1}^{t} x_{st} = 1 \qquad \forall t \tag{10}$$

$$\sum_{t=s}^{T} d_t x_{st} \leq u_s y_s \qquad \forall s \tag{11}$$

$$x_{st} \leq y_s \qquad \forall s \leq t \tag{12}$$

$$y_s \in \{0,1\} \qquad \forall s$$

$$x_{st} \geq 0 \qquad \forall s \leq t.$$

where the y_s variables indicate if an order has been placed at time period s, and the x_{st} variables indicate what fraction of the demand d_t is being satisfied from product ordered during time period s. This formulation once again suffers from a bad integrality gap, which can be demonstrated by the same example as in the previous two sections. We introduce new flow-cover inequalities to strengthen this formulation.

The basic idea is similar to the inequalities used in sections 2 and 3. We would like to consider a subset of orders, A, where even if we place all the orders in A and use these orders to their full potential, there is still unmet demand. In the previous cases, the amount of unmet demand was $D(A) = D - u(A)$. Now, however, that is not quite true, since each order s is capable of serving only the demand points t where $t \geq s$. Instead, we now also consider a subset of demand points B, and define $d(A, B)$ to be the total unmet demand in B, when the orders in A serve as much of the demand in B as possible. More formally

$$d(A, B) := \min\ d(B) - \sum_{s \in A} \sum_{t \geq s : t \in B} d_t x_{st} \tag{RHS-LP}$$

$$\text{s.t.} \sum_{s=1}^{t} x_{st} \leq 1 \qquad \forall t \tag{13}$$

$$\sum_{t=s}^{T} d_t x_{st} \leq u_s \qquad \forall s \tag{14}$$

$$x_{st} \geq 0 \qquad \forall s \leq t.$$

As before, we would also like to restrict the capacities of the orders not in A. To do this, we define

$$u_s(A, B) := d(A, B) - d(A \cup \{s\}, B), \tag{15}$$

which is the decrease in remaining demand that would result if order s were added to A. (This reduces to the same $u_s(A)$ as defined in the previous sections when considered in the framework of the earlier problems.) We once again partition the remaining orders in $F \setminus A$ into two sets, F_1 and F_2, and count the contribution of orders in F_1 as $\sum_t d_t x_{st}$ and orders in F_2 as $u_s(A,B)y_s$. This leads to the following LP, where once again $\mathcal{F}$ is the set of all 3-tuples that partition F into three sets.

$$\begin{aligned} \text{opt}_{LSP} := \min & \sum_{s=1}^{T} f_s y_s + \sum_{s=1}^{T}\sum_{t=s}^{T} h_{st} d_t x_{st} && \text{(LS-P)} \\ \text{s.t.} & \sum_{\substack{s\in F_1,\\ t\in B}} d_t x_{st} + \sum_{s\in F_2} u_s(A,B) y_s \geq d(A,B) && (16) \\ & && \forall (F_1,F_2,A)\in\mathcal{F}, B\subseteq F \\ & x_{st}, y_s \geq 0 && \forall s,t. \end{aligned}$$

Lemma 1. *Any feasible solution to* (LS-IP) *is a feasible solution to* (LS-P).

Proof. Consider a feasible integer solution (x,y) to (LS-IP) and let $S := \{s : y_s = 1\}$. Now for any $(F_1,F_2,A)\in\mathcal{F}$ and $B\subseteq F$ we know

$$\sum_{\substack{s\in F_1,\\ t\in B}} d_t x_{st} \geq d((F_2\cap S)\cup A, B),$$

since there is no way to assign demand from B to orders in $(F_2\cap S)\cup A$ without leaving at least $d((F_2\cap S)\cup A, B)$ amount of demand unfulfilled. Thus at least that amount of demand must be served by the other orders in S, namely those in F_1. Let $k := |F_2\cap S|$ and let $s_1,\dots,s_k$ denote the elements of that set in some order. Furthermore let $S_i := \{s_1,\dots,s_i\}$ for each $1\leq i\leq k$, so $S_k = F_2\cap S$. Then by repeated use of (15) we have

$$\begin{aligned} \sum_{\substack{s\in F_1,\\ t\in B}} d_t x_{st} &\geq d((F_2\cap S)\cup A, B) \\ &= d(((F_2\cap S)\cup A)\setminus S_1, B) - u_{s_1}(((F_2\cap S)\cup A)\setminus S_1, B) \\ &= d(((F_2\cap S)\cup A)\setminus S_k, B) - \sum_{i=1}^{k} u_{s_i}(((F_2\cap S)\cup A)\setminus S_i, B) \\ &= d(A,B) - \sum_{i=1}^{k} u_{s_i}(((F_2\cap S)\cup A)\setminus S_i, B) \\ &\geq d(A,B) - \sum_{s\in F_2\cap S} u_s(A,B) \\ &\geq d(A,B) - \sum_{s\in F_2} u_s(A,B) y_s, \end{aligned}$$

where the inequalities follow since $u_s(A,B)$ is increasing as elements are removed from A. Thus (x,y) satisfies all of the flow-cover inequalities (16). □

The dual of (LS-P) is

$$\text{opt}_{LSD} := \max \sum_{(F_1,F_2,A)\in\mathcal{F}} d(A,B)v(F_1,F_2,A,B) \tag{LS-D}$$

$$\text{s.t.} \sum_{\substack{(F_1,F_2,A)\in\mathcal{F},B\subseteq F:\\ s\in F_1,t\in B}} v(F_1,F_2,A,B) \leq h_{st} \qquad \forall s \leq t \tag{17}$$

$$\sum_{\substack{(F_1,F_2,A)\in\mathcal{F},B\subseteq F:\\ s\in F_2}} u_s(A,B)v(F_1,F_2,A,B) \leq f_s \qquad \forall s \tag{18}$$

$$v(F_1,F_2,A,B) \geq 0 \qquad \forall (F_1,F_2,A)\in\mathcal{F}, B\subseteq F,$$

where we simply divided constraint (17) by d_t.

Before we get to a primal-dual algorithm, we must first introduce some notation and associated machinery.

$$e_t := d_t \left[1 - \sum_{r=1}^{t} x_{rt}\right] \quad \text{- amount of demand currently unsatisfied in period } t$$

We define `Fill`(A,B) to be the following procedure that describes how to assign demand from B to orders in A. We consider the orders in A in arbitrary order, and for each order we serve as much demand as possible, processing demands from earliest to latest.

In the previous two sections, there was effectively only one demand, and so we never had to be concerned about how demand is assigned once an item or facility is chosen. Now there are many demand points, and so we must be careful that as we maintain our solution set A, we are serving as much demand as possible from the orders in A. The way the primal-dual algorithm will assign demand will correspond with how the `Fill` procedure works, and we show that this is a maximal assignment.

Lemma 2. *If we start from an empty demand assignment and run `Fill`(A,B), then we obtain a demand assignment such that $e(B) = d(A,B)$. Thus, `Fill` produces an assignment that is optimal for* (RHS-LP)*.*

Proof. Consider the latest time period $t \in B$ where $e(t) > 0$. All orders at time periods at or before t must be serving up to capacity, since otherwise they could have served more of demand d_t. All orders after time period t could not have served any more demand in B since all demand points in B after t are fully served. □

Just as in the previous two sections, the primal-dual algorithm initializes the variables to zero and the set A to the empty set. As in Section 3, we initialize F_1 to be the set of all orders, and an order will become tight first on constraint (17), when it will be moved to F_2, and then tight on (18), when it will be moved to A. Unlike in Section 3, however, constraint (17) consists of many different inequalities for the same order. This difficulty is averted since all the constraints (17) for a particular order will become tight at the same time, as is proved below in Lemma 3. This is achieved by slowly introducing demand points into the set B. Initially, B will consist only of the last demand point, T. Every time an order becomes tight on all constraints (17), it is moved from F_1 into F_2, and the demand point of that time period is added to B. In this way we always maintain that F_1 is a prefix of F, and B is the complementary suffix. When an order s becomes tight on constraint (18), we move it to A and assign demand to it by running the procedure `Fill`(s, B). Additionally we create a *reserve set* of orders, R_s, for order s, that consists of all orders earlier than s that are not in F_1 at the time s was added to A. Finally, once all of the demand has been assigned to orders, we label the set of orders in A as our current solution, S^*, and now enter a clean-up phase. We consider the orders in the reverse order in which they were added to A, and for each order, s, we check to see if there is enough remaining capacity of the orders that are in both the reserve set and our current solution, $S^* \cap R_s$, to take on the demand being served by s. If there is, then we reassign that demand to the orders in $S^* \cap R_s$ arbitrarily and remove s from our solution S^*. When the clean-up phase is finished we label the nodes in S^* as our final solution, S.

Algorithm 3: Primal-Dual for Single-Item Lot-Sizing

```
x_st, y_s ← 0
F_1 ← F
F_2, A ← ∅
B ← {T}
while d(A, F) > 0 do
    Increase v(F_1, F_2, A, B) until dual constraint becomes tight for order s
    if s ∈ F_1 then                          /* s tight on (17) but not on (18) */
        Move s from F_1 into F_2
        B ← F \ F_1
    else                                     /* else s tight on (17) and (18) */
        y_s ← 1
        Fill(s, B)
        Move s from F_2 into A
        R_s ← {r ∈ F \ F_1 : r < s}
S* ← A                                       /* start clean-up phase */
for s ← last order added to A to first order added to A do
    if remaining capacity of orders in S* ∩ R_s is enough to serve demand of s
    then
        Remove s from solution S*
        y_s, x_st ← 0                        /* unassign demand of s */
        Fill(S* ∩ R_s, F)                    /* reassign to reserve orders */
S ← S*
```

Lemma 3. *All of the constraints* (17) *for a particular order become tight at the same time, during the execution of Algorithm 3.*

Proof. We instead prove an equivalent statement: when demand t is added to B, then for any order $s \le t$ the slack of the constraint (17) corresponding to s and demand t' is h_{st} for any $t' \ge t$. This statement implies the lemma by considering $s = t$, which implies all constraints (17) for s become tight at the same time. We prove the above statement by (backwards) induction on the demand points. The case where $t = T$ clearly holds, since this demand point is in B before any dual variable is increased, and hence the slack of constraint (17) for order s and demand T is h_{sT}. Now assume the statement holds for some $t \le T$. If we consider order $s = t - 1$ then by the inductive hypothesis the slack of all constraints (17) for s and demand $t' \ge t$ is h_{st}. Hence the slack of all constraints (17) for orders $s' \le s$ decreases by h_{st} between the time t is added to B and when $t-1$ is added to B. Then by the inductive hypothesis again, we have that for any order $s' \le s$ and any demand $t' \ge t$, when $t - 1$ is added to B the slack of the corresponding constraint (17) is

$$h_{s't} - h_{st} = \sum_{r=s'}^{t-1} h_r - \sum_{r=s}^{t-1} h_r = \sum_{r=s'}^{t-1} h_r - h_{t-1} = \sum_{r=s'}^{t-2} = h_{s',t-1}.$$

Hence the statement also holds for $t - 1$. □

Define ℓ to be

$$\ell = \ell(F_1, F_2, A) := \max\{s : s \in S \cap F_2\} \cup \{0\},$$

so ℓ is the latest order in the final solution that is also in F_2, for a given partition, or if there is no such order then ℓ is 0, which is a dummy order with no capacity.

Lemma 4. *Upon completion of Algorithm 3, for any* F_1, F_2, A, B *such that* $v(F_1, F_2, A, B) > 0$*, we have*

$$\sum_{s \in S \cap F_2} u_s(A, B) + \sum_{s \in S \cap F_1} \sum_{t \in B : t \ge s} d_t x_{st} < d(A, B) + u_\ell(A, B).$$

Proof. First we consider the case when $S \cap F_2 \neq \emptyset$. Here the first summation is empty, so we just need to show the bound holds for the second summation. We know that any order $s \in S \cap F_1$ is not in the reserve set for any order in A. This follows since F_1 decreases throughout the course of the algorithm, so since s is in F_1 at this point, then clearly s was in F_1 at any point previously, in particular when any order in A was originally added to A. Hence no demand that was originally assigned to an order in A was ever reassigned to an order in F_1. But from Lemma 2 we know that only an amount $d(A, B)$ of demand from demand points in B is not assigned to orders in A, thus orders in F_1 can serve at most this amount of demand from demand points in B and we are done.

Otherwise it must be the case that $S \cap F_2 \neq \emptyset$, hence ℓ corresponds to a real order that was not deleted in the clean-up phase. By the way ℓ was chosen we know that any other order in $S \cap F_2$ is in an earlier time period than ℓ, and since these orders were moved out of F_1 before ℓ was added to A, they must be in the reserve set R_ℓ. However, since ℓ is in the final solution, it must be the case that when ℓ was being considered for deletion the orders in the reserve set that were still in the solution did not have sufficient capacity to take on the demand assigned to ℓ. Thus if we let x' denote the demand assignment during the clean-up phase when ℓ was being considered, then

$$\sum_{s \in (S \cap F_2) \setminus \{\ell\}} u_s(A,B) \leq u((S \cap F_2) \setminus \{\ell\}) < \sum_{\substack{s \in S \cap F_2 \\ t \in B : t \geq s}} d_t x'_{st}.$$

None of the orders in A had been deleted when ℓ was being considered for deletion, so only an amount $d(A,B)$ of the demand in B was being served by orders outside of A. As argued previously, none of the orders in F_1 took on demand being served by orders in A, but they also did not take on demand being served by orders in $S \cap F_2$, since these orders were never deleted. Thus we can upper bound the amount of demand that orders from $S \cap F_2$ could have been serving at the time order ℓ was being considered for deletion as follows

$$\sum_{\substack{s \in S \cap F_2 \\ t \in B : t \geq s}} d_t x'_{st} \leq d(A,B) - \sum_{\substack{s \in S \cap F_1 \\ t \in B : t \geq s}} d_t x_{st}.$$

The desired inequality is obtained by rearranging terms and adding $u_\ell(A,B)$ to both sides. □

We are now ready to analyze the cost of the solution.

Theorem 3. *Algorithm 3 terminates with a solution of cost no greater than* $2 \cdot opt_{LS}$.

Proof. As in the previous two sections, we can use the fact that all of the orders in the solution are tight on all constraints (17) and (18).

$$\begin{aligned}
&\sum_{s \in S} \left[f_s + \sum_{t=s}^{T} d_t h_{st} x_{st} \right] \\
&= \sum_{s \in S} \left[\sum_{\substack{(F_1,F_2,A) \in \mathcal{F}, B \subseteq F: \\ s \in F_2}} u_s(A,B) v(F_1,F_2,A,B) \right. \\
&\qquad \left. + \sum_{t=s}^{T} d_t x_{st} \sum_{\substack{(F_1,F_2,A) \in \mathcal{F}, B \subseteq F: \\ s \in F_1, t \in B}} v(F_1,F_2,A,B) \right] \\
&= \sum_{(F_1,F_2,A) \in \mathcal{F}, B \subseteq F} v(F_1,F_2,A,B) \left[\sum_{s \in S \cap F_2} u_s(A,B) + \sum_{s \in S \cap F_1} \sum_{t \in B : t \geq s} d_t x_{st} \right].
\end{aligned}$$

Now we can apply Lemma 4 and achieve the desired result:

$$\begin{aligned}\sum_{s\in S}\left[f_s+\sum_{t=s}^{T}d_t h_{st}x_{st}\right] &< \sum_{(F_1,F_2,A)\in\mathcal{F},B\subseteq F} v(F_1,F_2,A,B)\left[d(A,B)+u_\ell(A,B)\right]\\ &\leq \sum_{(F_1,F_2,A)\in\mathcal{F},B\subseteq F} 2d(A,B)v(F_1,F_2,A,B)\\ &\leq 2\cdot \mathrm{opt}_{LSD}. \qquad \square\end{aligned}$$

Acknowledgement We would like to thank Retsef Levi for many helpful discussions.

References

1. K. Aardal, Y. Pochet, and L. A. Wolsey. Capacitated facility location: valid inequalities and facets. *Math. Oper. Res.*, 20(3):562–582, 1995.
2. A. Agrawal, P. Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.
3. R. Bar-Yehuda and S. Even. A linear-time approximation algorithm for the weighted vertex cover problem. *J. Algorithms*, 2(2):198–203, 1981.
4. D. Bertsimas and C. P. Teo. From valid inequalities to heuristics: a unified view of primal-dual approximation algorithms in covering problems. *Operations Research*, 46(4):503–514, 2003.
5. R. D. Carr, L. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 106–115, 2000.
6. J. Chuzhoy and J. Naor. Covering problems with hard capacities. *SIAM J. Comput.*, 36(2):498–515, 2006.
7. V. Chvátal. A greedy heuristic for the set-covering problem. *Math. Oper. Res.*, 4(3):233–235, 1979.
8. G. Even, R. Levi, D. Rawitz, B. Schieber, S. Shahar, and M. Sviridenko. Algorithms for capacitated rectangle stabbing and lot sizing with joint set-up costs. Submitted to ACM Trans. on Algorithms, 2007.
9. R. Gandhi, E. Halperin, S. Khuller, G. Kortsarz, and A. Srinivasan. An improved approximation algorithm for vertex cover with hard capacities. *Journal of Computer and System Sciences*, 72(1):16–33, 2006.
10. M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995.
11. R. Levi, A. Lodi, and M. Sviridenko. Approximation algorithms for the multi-item capacitated lot-sizing problem via flow-cover inequalities. In *Proceedings of the 12th International IPCO Conference*, volume 4513 of *Lecture Notes in Computer Science*, pages 454–468. Springer, 2007.
12. R. Levi, R. Roundy, and D. B. Shmoys. Primal-dual algorithms for deterministic inventory problems. *Math. Oper. Res.*, 31(2):267–284, 2006.
13. Y. Sharma and D. Williamson. Personal communication, 2007.
14. C. P. M. van Hoesel and A. P. M. Wagelmans. Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems. *Math. Oper. Res.*, 26(2):339–357, 2001.