Capacitated Domination: Constant Factor Approximations for Planar Graphs*

Mong-Jen Kao** and D.T. Lee***

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
d97021@csie.ntu.edu.tw, dtlee@nchu.edu.tw

Abstract. We consider the capacitated domination problem, which models a service-requirement assigning scenario and which is also a generalization of the dominating set problem. In this problem, we are given a graph with three parameters defined on the vertex set, which are cost, capacity, and demand. The objective of this problem is to compute a demand assignment of least cost, such that the demand of each vertex is fully-assigned to some of its closed neighbours without exceeding the amount of capacity they provide. In this paper, we provide the first constant factor approximation for this problem on planar graphs, based on a new perspective on the hierarchical structure of outer-planar graphs. We believe that this new perspective and technique can be applied to other capacitated covering problems to help tackle vertices of large degrees.

1 Introduction

For decades, *Dominating Set* problem has been one of the most fundamental and well-known problems in both graph theory and combinatorial optimization. Given a graph $G = (V, E)$ and an integer k, *Dominating Set* asks for a subset $D \subseteq V$ whose cardinality does not exceed k such that every vertex in the graph either belongs to this set or has a neighbour which does. As this problem is known to be NP-hard, approximation algorithms have been proposed in the literature [1,10,11].

A series of study on capacitated covering problem was initiated by Guha et al., [9], which addressed the capacitated vertex cover problem from a scenario of Glycomolecule ID (GMID) placement. Several follow-up papers have appeared since then, studying both this topic and related variations [4,7,8]. These problems are also closely related to work on the capacitated facility location problem, which has drawn a lot of attention since 1990s. See [3,16].

* Supported in part by the National Science Council, Taiwan, under Grants NSC99-2911-I-002-055-2, NSC98-2221-E-001-007-MY3, and NSC98-2221-E-001-008-MY3.

** This work was done when the author was with Karlsruhe Institute of Technology (KIT), Germany, as a visiting scholar under the NSC-DAAD-sponsored sandwich program (grant number NSC99-2911-I-002-055-2).

*** The author's present address is Department of Computer Science and Engineering, National Chung-Hsing University, Taichung, Taiwan.

T. Asano et al. (Eds.): ISAAC 2011, LNCS 7074, pp. 494–503, 2011.

Motivated by a general service-requirement assignment scenario, Kao et al., [12,14] considered a generalization of the dominating set problem called *Capacitated Domination*, which is defined as follows. Let $G = (V, E)$ be a graph with three non-negative parameters defined on each vertex $u \in V$, referred to as the cost, the capacity, and the demand, further denoted by $w(u)$, $c(u)$, and $d(u)$, respectively. The demand of a vertex stands for the amount of service it requires from its adjacent vertices, including the vertex itself, while the capacity of a vertex represents the amount of service each multiplicity (copy) of that vertex can provide.

By a demand assignment function f we mean a function which maps pairs of vertices to non-negative real numbers. Intuitively, $f(u, v)$ denotes the amount of demand of u that is assigned to v. We use $N_G(v)$ to denote the set of neighbours of a vertex $v \in V$.

Definition 1 (feasible demand assignment function). *A demand assignment function f is said to be feasible if $\sum_{u \in N_G[v]} f(v, u) \geq d(v)$, for each $v \in V$, where $N_G[v] = N_G(v) \cup \{v\}$ denotes the neighbours of v unions v itself.*

Given a demand assignment function f, the corresponding capacitated dominating multi-set $\mathcal{D}(f)$ is defined as follows. For each vertex $v \in V$, the *multiplicity* of v in $\mathcal{D}(f)$ is defined to be $x_f(v) = \left\lceil \frac{\sum_{u \in N_G[v]} f(u,v)}{c(v)} \right\rceil$. The cost of the assignment function f, denoted $w(f)$, is defined to be $w(f) = \sum_{u \in V} w(u) \cdot x_f(u)$.

Definition 2 (Capacitated Domination Problem). *Given a graph $G = (V, E)$ with cost, capacity, and demand defined on each vertex, the capacitated domination problem asks for a feasible demand assignment function f such that $w(f)$ is minimized.*

For this problem, Kao et al., [14], presented a $(\Delta + 1)$-approximation for general graphs, where Δ is the maximum vertex degree of the graph, and a polynomial time approximation scheme for trees, which they proved to be NP-hard. In a following work [12], they provided more approximation algorithms and complexity results for this problem. On the other hand, Dom et al., [6] considered a variation of this problem where the number of multiplicities available at each vertex is limited and proved the *W[1]*-hardness when parameterized by treewidth and solution size. Cygan et al., [5], made an attempt toward the exact solution and presented an $O(1.89^n)$ algorithm when each vertex has unit demand. This result was further improved by Liedloff et al., [15].

Our Contributions. We provide the first constant factor approximation algorithms for the capacitated domination problem on planar graphs. This result can be considered a break-through with respect to the pseudo-polynomial time approximations given in [12], which is based on a dynamic programming on graphs of bounded treewidth. The approach used in [12] stems from the fact that vertices of large degrees will fail most of the techniques that transform a pseudo-polynomial time dynamic programming algorithm into approximations, i.e., the error accumulated at vertices of large degrees could not be bounded.

In this work, we tackle this problem using a new approach. Specifically, we give a new perspective toward the hierarchical structure of outer-planar graphs, which enables us to further tackle vertices of large degrees. Then we analyse both the primal and the dual linear programs of this problem to obtain the claimed result. We believe that the approach we provided in this paper can be applied to other capacitated covering problems to help tackle vertices of large degrees as well. Due to the space limit, the proofs as well as certain technical details are omitted. Please refer to our full article [13] for further reference.

2 Preliminary

We assume that all the graphs considered in this paper are simple and undirected. Let $G = (V, E)$ be a graph. We denote the number of vertices, $|V|$, by n. The set of neighbors of a vertex $v \in V$ is denoted by $N_G(v) = \{u : (u, v) \in E\}$. The closed neighborhood of $v \in V$ is denoted by $N_G[v] = N_G(v) \cup \{v\}$. We use $deg_G(v)$ and $deg_G[v]$ to denote the cardinality of $N_G(v)$ and $N_G[v]$, respectively. The subscript G in $N_G[v]$ and $deg_G[v]$ will be omitted when there is no confusion.

A planar embedding of a graph G is a drawing of G in the plane such that the edges intersect only at their endpoints. A graph is said to be planar if it has a planar embedding. An outer-planar graph is a graph which adopts a planar embedding such that all the vertices lie on a fixed circle, and all the edges are straight lines drawn inside the circle. For $k \geq 1$, k-outerplanar graphs are defined as follows. A graph is 1-outerplanar if and only if it is outer-planar. For $k > 1$, a graph is called k-outerplanar if it has a planar embedding such that the removal of the vertices on the unbounded face results in a $(k-1)$-outerplanar graph.

An integer linear program (ILP) for capacitated domination is given in (1). The first inequality ensures the feasibility of the demand assignment function f required in Definition 1. In the second inequality, we model the multiplicity function x as defined. The third constraint, $d(v)x(u) - f(v, u) \geq 0$, which seems unnecessary in the problem formulation, is required to bound the integrality gap between the optimal solution of this ILP and that of its relaxation. To see that this additional constraint does not alter the optimality of any optimal solution, we have the following lemma.

$$
\begin{aligned}
&\text{Minimize} \quad \sum_{u \in V} w(u)x(u) && (1)\\
&\text{subject to} &&\\
&\quad \sum_{v \in N[u]} f(u, v) - d(u) \geq 0, && u \in V\\
&\quad c(u)x(u) - \sum_{v \in N[u]} f(v, u) \geq 0, && u \in V\\
&\quad d(v)x(u) - f(v, u) \geq 0, && v \in N[u], u \in V\\
&\quad f(u, v) \geq 0, \;\; x(u) \in \mathbb{Z}^+ \cup \{0\}, && u, v \in V
\end{aligned}
$$

Lemma 1. *Let f be an arbitrary optimal demand assignment function. We have $d(v) \cdot x_f(u) - f(v, u) \geq 0$ for all $u \in V$ and $v \in N[u]$.*

However, without this constraint, the integrality gap can be arbitrarily large. This is illustrated by the following example. Let $\alpha > 1$ be an arbitrary constant, and $\mathcal{T}(\alpha)$ be an n-vertex star, where each vertex has unit demand and unit cost.

The capacity of the central vertex is set to be n, which is sufficient to cover the demand of the entire graph, while the capacity of each of remaining $n-1$ petal vertices is set to be αn.

Lemma 2. *Without the additional constraint $d(v)x(u) - f(v,u) \geq 0$, the integrality gap of the ILP (1) on $\mathcal{T}(\alpha)$ is α, where $\alpha > 1$ is an arbitrary constant.*

Indeed, with the additional constraint applied, we can refrain from unreasonably assigning a small amount of demand to any vertex in any fractional solution. Take a petal vertex, say v, from $\mathcal{T}(\alpha)$ as example, given that $d(v) = 1$ and $f(v,v) = 1$, this constraint would force $x(v)$ to be at least 1, which prevents the aforementioned situation from being optimal.

For the rest of this paper, for any graph G, we denote the optimal values to the integer linear program (1) and to its relaxation by $OPT(G)$ and $OPT_f(G)$, respectively. Note that $OPT_f(G) \leq OPT(G)$.

3 Constant Approximation for Outer-Planar Graphs

Without loss of generality, we assume that the graphs are connected. Otherwise we simply apply the algorithm to each of the connected component separately. In the following, we first classify the outer-planar graphs into a class of graphs called general-ladders and show how the corresponding general-ladder representation can be extracted in $O(n \log^3 n)$ time in §3.1. Then we consider in §3.2 and §3.3 both the primal and the dual programs of the relaxation of (1) to further reduce a given general-ladder and obtain a constant factor approximation. We analyse the algorithm in §3.4 and extend our result to planar graphs in §3.5.

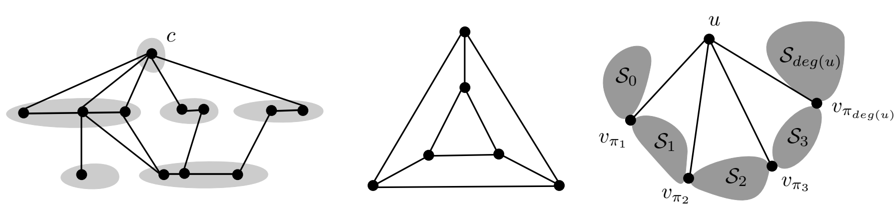

Fig. 1. (a) A general-ladder with anchor c. (b) A 2-outerplanar graph which fails to be a general-ladder. (c) The subdivision formed by a vertex u in an outer-planar embedding.

3.1 The Structure

First we define the notation which we will use later on. By a total order of a set we mean that each pair of elements in the set can be compared, and therefore an ascending order of the elements is well-defined. Let $P = (v_1, v_2, \ldots, v_k)$ be a path. We say that P is an *ordered path* if a total order $v_1 \prec v_2 \prec \ldots \prec v_k$ or $v_k \prec v_{k-1} \prec \ldots \prec v_1$ is defined on the set of vertices.

Definition 3 (General-Ladder). *A graph $G = (V, E)$ is said to be a general-ladder if a total order on the set of vertices is defined, and G is composed of* *a set of layers* $\{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_k\}$, *where each layer is a collection of subpaths of an ordered path such that the following holds. The top layer,* $\mathcal{L}_1$, *consists of a single vertex, which is referred to as the anchor, and for each* $1 < j < k$ *and* $u, v \in \mathcal{L}_j$, *we have (1)* $N[u] \subseteq \mathcal{L}_{j-1} \cup \mathcal{L}_j \cup \mathcal{L}_{j+1}$, *and (2)* $u \prec v$ *implies* $\max_{p \in N[u] \cap \mathcal{L}_{j+1}} p \preccurlyeq \min_{q \in N[v] \cap \mathcal{L}_{j+1}} q$.

Note that each layer in a general-ladder consists of a set of ordered paths which are possibly connected only to vertices in the neighbouring layers. See Fig. 1 (a). Although the definition of general-ladders captures the essence and simplicity of an ordered hierarchical structure, there are planar graphs which fall outside this framework. See also Fig. 1 (b).

In the following, we state and argue that every outerplanar graph meets the requirements of a general-ladder. We assume that an outer-planar embedding for any outer-planar graph is given as well. Otherwise we apply the $O(n \log^3 n)$ algorithm provided by Bose [2] to compute such an embedding.

Let $G = (V, E)$ be an outer-planar graph, $u \in V$ be an arbitrary vertex, and $\mathcal{E}$ be an outer-planar embedding of G. We fix u to be the smallest element and define a total order on the vertices of G according to their orders of appearances on the outer face of $\mathcal{E}$ in a counter-clockwise order. For convenience, we label the vertices such that $u = v_1$ and $v_1 \prec v_2 \prec v_3 \prec \ldots \prec v_n$.

Let $N(u) = \left\{v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_{deg(u)}}\right\}$ denote the neighbours of u such that $v_{\pi_1} \prec v_{\pi_2} \prec \ldots \prec v_{\pi_{deg(u)}}$. $N(u)$ divides the set of vertices except u into $deg(u) + 1$ subsets, namely, $\mathcal{S}_0 = \{v_2, v_3, \ldots, v_{\pi_1}\}$, $\mathcal{S}_i = \{v_{\pi_i}, v_{\pi_i+1}, \ldots, v_{\pi_{i+1}}\}$ for $1 \leq i < deg(u)$, and $\mathcal{S}_{deg(u)} = \{v_{\pi_{deg(u)}}, v_{\pi_{deg(u)}+1}, \ldots, v_n\}$. See Fig. 1 (c) for an illustration. For $1 \leq i < deg(u)$, we partition $\mathcal{S}_i$ into two sets L_i and R_i as follows. Let $d_{\mathcal{S}_i}$ denote the distance function defined on the induced subgraph of $\mathcal{S}_i$. Let $L_i = \left\{v : v \in \mathcal{S}_i, d_{\mathcal{S}_i}(v_{\pi_i}, v) \leq d_{\mathcal{S}_i}\left(v, v_{\pi_{i+1}}\right)\right\}$ and $R_i = \mathcal{S}_i \backslash L_i$.

Denote $\ell(v) \equiv d_G(u, v)$. Now consider the set of the edges connecting L_i and R_i. Note that, this is exactly the set of edges connecting vertices on the shortest path between v_{π_i} and $\max_{a \in L_i} a$ and vertices on the shortest path between $v_{\pi_{i+1}}$ and $\min_{b \in R_i} b$. See also Fig. 2. Below we present our structural lemma, which states that, when the vertices are classified by their distances to u, these edges can only connect vertices between neighbouring sets and do not form any crossing.

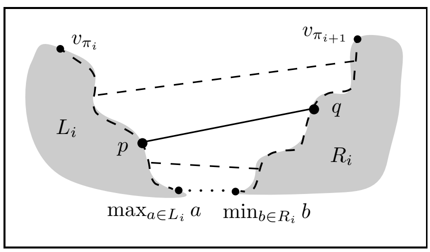

Fig. 2. Partition of $\mathcal{S}_i$ into L_i and R_i

Lemma 3. *Any outer-planar graph* $G = (V, E)$ *together with an arbitrary vertex* $u \in V$ *is a general-ladder anchored at* u, *where the set of vertices in each layer are classified by their distances to the anchor* u.

Extracting the General-ladder. Let $\mathcal{G} = (V, E)$ be the input outer-planar graph and $u \in V$ be an arbitrary vertex. We can extract the corresponding general-ladder as stated below.

Theorem 1. *Given an outer-planar graph $\mathcal{G}$ and its outer-planar embedding, we can compute in linear time a general-ladder representation for G.*

For the rest of this paper we will denote the layers of this particular general-ladder representation by $\mathcal{L}_0, \mathcal{L}_1, \ldots, \mathcal{L}_M$. The following additional structural property comes from the outer-planarity of $\mathcal{G}$ and our construction scheme.

Lemma 4. *For any $0 < i \leq M$ and $v \in \mathcal{L}_i$, we have $|N(v) \cap \mathcal{L}_{i-1}| \leq 2$. Moreover, if v has two neighbours in $\mathcal{L}_i$, say, v_1 and v_2 with $v_1 \prec v \prec v_2$, then there is an edge joining v_1 (and v_2, respectively) and each neighbouring vertex of v in $\mathcal{L}_{i-1}$ that is smaller (larger) than v.*

The Decomposition. The idea behind this decomposition is to help reduce the dependency between vertices of large degrees and their neighbours such that further techniques can be applied. To this end, we tackle the demands of vertices from every three layers separately.

For each $0 \leq i < 3$, let $\mathcal{R}_i = \bigcup_{j \geq 0} \mathcal{L}_{3j+i}$. Let $\mathcal{G}_i = (V_i, E_i)$ consist of the induced subgraph of $\mathcal{R}_i$ and the set of edges connecting vertices in $\mathcal{R}_i$ to their neighbours. Formally, $V_i = \bigcup_{v \in \mathcal{R}_i} N[v]$ and $E_i = \bigcup_{v \in \mathcal{R}_i} \bigcup_{u \in N[v]} e(u,v)$. In addition, we set $d(v) = 0$ for all $v \in \mathcal{G}_i \backslash \mathcal{R}_i$. Other parameters remain unchanged.

Lemma 5. *Let f_i, $0 \leq i < 3$, be an optimal demand assignment function for $\mathcal{G}_i$. The assignment function $f = \sum_{0 \leq i < 3} f_i$ is a 3-approximation of $\mathcal{G}$.*

3.2 Removing More Edges

We describe an approach to further simplifying the graphs $\mathcal{G}_i$, for $0 \leq i < 3$. Given any feasible demand assignment for $\mathcal{G}_i$, we can properly reassign the demand of a vertex to a constant number of neighbours while the increase in terms of fractional cost remains bounded.

For each $v \in \mathcal{R}_i$, we sort the closed neighbours of v according to their cost in ascending order such that $w(\pi_v(1)) \leq w(\pi_v(2)) \leq \ldots \leq w(\pi_v(deg[v]))$, where $\pi_v : \{1, 2, \ldots, deg[v]\} \rightarrow N[v]$ is an injective function. For convenience, we set $\pi_v(deg[v]+1) = \phi$. Suppose that $v \in \mathcal{L}_\ell$. We identify the following four vertices.

- Let j_v, $1 \leq j_v \leq deg[v]$, be the smallest integer such that $c(\pi_v(j_v)) > d(v)$. If $c(\pi_v(j_v)) \leq d(v)$ for all $1 \leq j \leq deg[v]$, then we let $j_v = deg[v]+1$.
- Let k_v, $1 \leq k_v < j_v$, be the integer such that and $w(\pi_v(k_v))/c(\pi_v(k_v))$ is minimized. k_v is defined only when $j_v > 1$.
- Let $p_v = \max_{u \in N[v] \cap \mathcal{L}_{\ell-1}} u$ and $q_v = \max_{u \in N[v] \cap \mathcal{L}_{\ell+1}} u$.

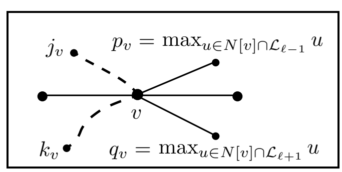

Fig. 3. Incident edges of a vertex $v \in \mathcal{L}_\ell$ to be kept

Intuitively, $\pi_v(j_v)$ is the first vertex in the sorted list whose capacity is greater than $d(v)$, and $\pi_v(k_v)$ is the vertex with best cost-capacity ratio among the first $j_v - 1$ vertices. p_v and q_v are the rightmost neighbour of v in layer $\mathcal{L}_{\ell-1}$ and $\mathcal{L}_{\ell+1}$, respectively.

We will omit the function π_v and use j_v, k_v to denote $\pi_v(j_v)$, $\pi_v(k_v)$ without confusion. The reduced graph $\mathcal{H}_i$ is defined as follows. Denote the set of neighbours to be disconnected from v by $R(v) = N[v] \setminus (\mathcal{L}_\ell \cup \{j_v \cup k_v \cup p_v \cup q_v\})$, and let $\mathcal{H}_i = \mathcal{G}_i \setminus \bigcup_{v \in \mathcal{R}_i} \bigcup_{u \in R(v)} \{e(u,v)\}$. Roughly speaking, in graph $\mathcal{H}_i$ we remove the edges which connect vertices in $\mathcal{R}_i$, say v, to vertices not in $\mathcal{R}_i$, except possibly for j_v, k_v, p_v, and q_v. See Fig. 3. Note that, although our reassigning argument applies to arbitrary graphs, only when two vertices are unimportant to each other can we remove the edge between them.

Lemma 6. *In the subgraph $\mathcal{H}_i$, we have*

- *For each $v \notin \mathcal{R}_i$, at most one incident edge of v which was previously in $\mathcal{G}_i$ will be removed.*
- *For each $v \in \mathcal{R}_i$, the degree of v in $\mathcal{H}_i$ is upper-bounded by* 6.
- $OPT_f(\mathcal{H}_i) \le 2 \cdot OPT_f(\mathcal{G}_i)$.

We also remark that, although $OPT_f(\mathcal{H}_i)$ is bounded in terms of $OPT_f(\mathcal{G}_i)$, an α-approximation for $\mathcal{H}_i$ is not necessarily a 2α-approximation for $\mathcal{G}_i$. That is, having an approximation $\mathcal{A}$ with $OPT(\mathcal{A}) \le \alpha \cdot OPT(\mathcal{H}_i)$ does not imply that $OPT(\mathcal{A}) \le 2\alpha \cdot OPT(\mathcal{G}_i)$, for $OPT(\mathcal{H}_i)$ could be strictly larger than $OPT_f(\mathcal{H}_i)$. Instead, to obtain our claimed result, an approximation with a stronger bound, in terms of $OPT_f(\mathcal{H}_i)$, is desired.

3.3 Greedy Charging Scheme

We show how we can further approximate the optimal solution for the reduced graph $\mathcal{H}_i$ by a primal-dual charging argument. We apply a technique from [14] to obtain a feasible solution for the dual program of the relaxation of (1), which is given in (2). By Lemma 4, we can further tighten the approximation ratio.

$$\begin{aligned}
&\text{Maximize} \quad \sum_{u \in V} d(u) y_u && (2)\\
&\text{subject to}\\
&\quad c(u) z_u + \sum_{v \in N[u]} d(v) g_{u,v} \le w(u), && u \in V\\
&\quad y_u \le z_v + g_{v,u}, && v \in N[u],\ u \in V\\
&\quad y_u \ge 0,\ \ z_u \ge 0,\ \ g_{v,u} \ge 0, && v \in N[u],\ u \in V
\end{aligned}$$

We first describe an approach to obtaining a feasible solution to (2) and how a corresponding feasible demand assignment can be found. Note that any feasible solution to (2) will serve as a lower bound to any feasible solution of (1) by the linear program duality.

During the process, we will maintain a vertex subset, V^ϕ, which contains the set of vertices with non-zero unassigned demand. For each $u \in V$, let $d^\phi(u) = \sum_{v \in N[u] \cap V^\phi} d(v)$ denote the amount of unassigned demand from the closed neighbours of u. We distinguish between two cases. If $c(u) < d^\phi(u)$, then we say that u is heavily-loaded. Otherwise, u is lightly-loaded. During the process, some heavily-loaded vertices might turn into lightly-loaded due to the demand assignments of its closed neighbours. For each of these vertices, say v, we will maintain a vertex subset $D^*(v)$, which contains the set of unassigned vertices in $N[v] \cap V^\phi$ when v is about to fall into lightly-loaded. For other vertices, $D^*(v)$ is defined to be an empty set.

Initially, $V^\phi \equiv \{u : u \in \mathcal{L}_i, d(u) \neq 0\}$ and all the dual variables are set to be zero. We increase the dual variable y_u simultaneously, for each $u \in V^\phi$. To maintain the dual feasibility, as we increase y_u, we have to raise either z_v or $g_{v,u}$, for each $v \in N[u]$. If v is heavily-loaded, then we raise z_v. Otherwise, we raise $g_{v,u}$. Note that, during this process, for each vertex u that has a closed neighbour in V^ϕ, the left-hand side of the inequality $c(u)z_u + \sum_{v\in N[u]} d(v)g_{u,v} \leq w(u)$ is constantly raising. As soon as one of the inequalities $c(u)z_u + \sum_{v\in N[u]} d(v)g_{u,v} \leq w(u)$ is met with equality (saturated) for some vertex $u \in V$, we perform the following operations.

If u is lightly-loaded, we assign all the unassigned demand from $N[u] \cap V^\phi$ to u. In this case, there are still $c(u) - d^\phi(u)$ units of capacity free at u. We assign the unassigned demand from $D^*(u)$, if there is any, to u until either all the demand from $D^*(u)$ is assigned or all the free capacity in u is used. On the other hand, if u is heavily-loaded, we mark it as heavy and delay the demand assignment from its closed neighbours.

Then we set $\mathcal{Q}_u \equiv N[u] \cap V^\phi$ and remove $N[u]$ from V^ϕ. Note that, due to the definition of d^ϕ, even when u is heavily-loaded, we still update $d^\phi(p)$ for each $p \in V$ with $N[p] \cap N[u] \neq \phi$, if needed, as if the demand was assigned. During the above operation, some heavily-loaded vertices might turn into lightly-loaded due to the demand assignments (or simply due to the update of d^ϕ). For each of these vertices, say v, we set $D^*(v) \equiv N[v] \cap (V^\phi \cup \mathcal{Q}_u)$. Intuitively, $D^*(v)$ contains the set of unassigned vertices from $N[v] \cap V^\phi$ when v is about to fall into lightly-loaded.

This process is continued until $V^\phi = \phi$. For those vertices which are marked as heavy, we iterate over them according to their chronological order of being saturated and assign at this moment all the remaining unassigned demand from their closed neighbours to them.

Let $f^* : V \times V \to R^+ \cup \{0\}$ denote the resulting demand assignment function, and $x^* : V \to Z^+ \cup \{0\}$ denotes the corresponding multiplicity function. The following lemma bounds the cost of the solution produced by our algorithm.

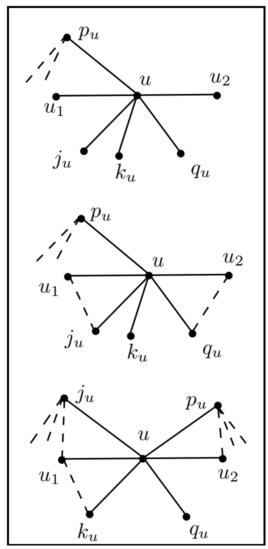

Fig. 4. Situations when a unit demand of u is fully-charged

Lemma 7. *For any $\mathcal{H}_i$ obtained from a general-ladder $\mathcal{G}_i$, we have $w(f^*) \leq 7 \cdot OPT_f(\mathcal{H}_i)$.*

Thanks to the structural property provided in Lemma 4, given the fact that the input graph is outer-planar, we can modify the algorithm slightly and further improve the bound given in the previous lemma. To this end, we consider the situations when a unit demand from a vertex u with $deg[u] = 7$ and argue that, either it is not fully-charged by all its closed neighbours, or we can modify the demand assignment, without raising the cost, to make it so.

Lemma 8. *Given the fact that $\mathcal{H}_i$ comes from an outerplanar graph, we can modify the algorithm to obtain a demand assignment function f^* such that $w(f^*) \leq 6 \cdot OPT_f(\mathcal{H}_i)$.*

3.4 Overall Analysis

We summarize the whole algorithm and our main theorem. Given an outer-planar graph $G = (V, E)$, we use the algorithm described in §3.1 to compute a general-ladder representation of G, followed by applying the decomposition to obtain three subproblems, $\mathcal{G}_0$, $\mathcal{G}_1$, and $\mathcal{G}_2$. For each $\mathcal{G}_i$, we use the approach described in §3.2 to further remove more edges and obtain the reduced subgraph $\mathcal{H}_i$, for which we apply the algorithm described in §3.3 to obtain an approximation, which is a demand assignment function f_i for $\mathcal{H}_i$. The overall approximation, e.g., the demand assignment function f, for G is defined as $f = \sum_{0 \leq i < 3} f_i$.

Theorem 2. *Given an outerplanar graph G as an instance of capacitated domination, we can compute a constant factor approximation for G in $O(n^2)$ time.*

3.5 Extension to Planar Graphs

In this section we extend our outer-planar result to a constant factor approximation for planar graphs under a general framework due to [1]. As our algorithm is designed mainly for outerplanar graphs, to meet the minimum requirement of this framework, which is the ability to deal with planar graphs of at least three levels, we have to modify our algorithm to undertake this difference.

In the following, we assume that the input graph, G, is 3-outerplanar and sketch only the key changes we made on our algorithm. Let L_0, L_1, and L_2 be the sets of vertices from the three levels of G. In addition, we also have $d(v) = 0$ for each $v \notin L_1$. See Fig. 5 (a).

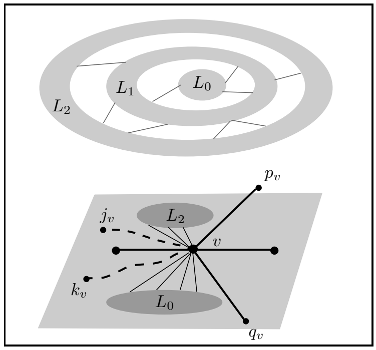

Fig. 5. (a) 3-outerplanar graph. (b) Local connections w.r.t. a vertex v. Bold edges represent links in the ladder extracted from L_1. Thin edges represent links between L_1 and L_0, L_2.

Obtaining the General Ladders. For each level L_i, $0 \leq i < 3$, we define a total order according to the counter-clockwise order of appearances of the vertices. The general-ladder is extracted from L_1 as we did before. Furthermore, for each vertex in the ladder, its incident edges to vertices in L_0 and L_2 are also included.

Removing Redundant Edges. In addition to the four vertices we identified for each vertex v with non-zero demand, we identify two more vertices, which literally corresponds to the rightmost neighbours of v in levels L_0 and L_2, respectively. See also Fig. 5 (b).

Theorem 3. *Given a planar graph G as an instance of capacitated domination, we can compute a constant factor approximation for the G in polynomial time.*

4 Conclusion

Due to the flexibility of the ways the demand can be assigned, the results we provided here seem to have room for further improvements. However, when the demand cannot be split, it is not difficult to prove a constant approximation threshold. Therefore, it would be very interesting to investigate the problem complexity on planar graphs.

Second, as we have shown in §3.1, the concept of general-ladders does not extend directly to k-outerplanar graphs for $k \geq 2$. It would be interesting to formalize and extend this concept to k-outerplanar graphs, for it seems helpful not only to our problem, but also to most capacitated covering problems as well.

Acknowledgements. The author would like to thank the anonymous referees for their very helpful comments on the layout of this work.

References

1. Baker, B.S.: Approximation algorithms for np-complete problems on planar graphs. J. ACM 41, 153–180 (1994)
2. Bose, P.: On embedding an outer-planar graph in a point set. CGTA: Computational Geometry: Theory and Applications 23, 2002 (1997)
3. Chudak, F.A., Williamson, D.P.: Improved approximation algorithms for capacitated facility location problems. Math. Program. 102, 207–222 (2005)
4. Chuzhoy, J.: Covering problems with hard capacities. SIAM J. Comput. 36, 498–515 (2006)
5. Cygan, M., Pilipczuk, M., Wojtaszczyk, J.O.: Capacitated Domination Faster Than $O(2^n)$. In: Kaplan, H. (ed.) SWAT 2010. LNCS, vol. 6139, pp. 74–80. Springer, Heidelberg (2010), doi:10.1007/978-3-642-13731-0_8
6. Dom, M., Lokshtanov, D., Saurabh, S., Villanger, Y.: Capacitated Domination and Covering: A Parameterized Perspective. In: Grohe, M., Niedermeier, R. (eds.) IWPEC 2008. LNCS, vol. 5018, pp. 78–90. Springer, Heidelberg (2008)
7. Gandhi, R., Halperin, E., Khuller, S., Kortsarz, G., Srinivasan, A.: An improved approximation algorithm for vertex cover with hard capacities. J. Comput. Syst. Sci. 72, 16–33 (2006)
8. Gandhi, R., Khuller, S., Parthasarathy, S., Srinivasan, A.: Dependent rounding in bipartite graphs. In: FOCS 2002, pp. 323–332 (2002)
9. Guha, S., Hassin, R., Khuller, S., Or, E.: Capacitated vertex covering. J. Algorithms 48(1), 257–270 (2003)
10. Hochbaum, D.: Approximation algorithms for the set covering and vertex cover problems. SIAM Journal on Computing 11, 555–556 (1982)
11. Johnson, D.S.: Approximation algorithms for combinatorial problems. In: The 5th Annual ACM Symposium on Theory of Computing, pp. 38–49 (1973)
12. Kao, M.-J., Chen, H.-L.: Approximation Algorithms for the Capacitated Domination Problem. In: Lee, D.-T., Chen, D.Z., Ying, S. (eds.) FAW 2010. LNCS, vol. 6213, pp. 185–196. Springer, Heidelberg (2010)
13. Kao, M.-J., Lee, D.: Capacitated domination: Constant factor approximation for planar graphs (manuscript, 2011), `http://arxiv.org/abs/1108.4606`
14. Kao, M.-J., Liao, C.-S., Lee, D.T.: Capacitated domination problem. Algorithmica 60, 274–300 (2011), doi:10.1007/s00453-009-9336-x
15. Liedloff, M., Todinca, I., Villanger, Y.: Solving Capacitated Dominating Set by Using Covering by Subsets and Maximum Matching. In: Thilikos, D.M. (ed.) WG 2010. LNCS, vol. 6410, pp. 88–99. Springer, Heidelberg (2010)
16. Shmoys, D.B., Tardos, E., Aardal, K.: Approximation algorithms for facility location problems (extended abstract). In: STOC 1997, pp. 265–274 (1997)