

**A 3-approximation Algorithm for  
the k-level Uncapacitated Facility  
Location Problem**

*Karen Aardal, Fabian A. Chudak, and  
David B. Shmoys*

**CRPC-TR99784  
January 1999**

Center for Research on Parallel Computation  
Rice University  
6100 South Main Street  
CRPC - MS 41  
Houston, TX 77005

# A 3-approximation algorithm for the $k$ -level uncapacitated facility location problem

Karen Aardal\*

Fabián A. Chudak<sup>†</sup>

David B. Shmoys<sup>‡</sup>

## Abstract

In the  $k$ -level uncapacitated facility location problem, we have a set of demand points where clients are located. The demand of each client is known. Facilities have to be located at given sites in order to service the clients, and each client is to be serviced by a sequence of  $k$  different facilities, each of which belongs to a distinct level. There are no capacity restrictions on the facilities. There is a positive fixed cost of setting up a facility, and a per unit cost of shipping goods between each pair of locations. We assume that these distances are all nonnegative and satisfy the triangle inequality. The problem is to find an assignment of each client to a sequence of  $k$  facilities, one at each level, so that the demand of each client is satisfied, for which the sum of the setup costs and the service costs is minimized.

We develop a randomized algorithm for the  $k$ -level facility location problem that is guaranteed to find a feasible solution of expected cost within a factor of 3 of the optimum cost. The algorithm is a randomized rounding procedure that uses an optimal solution of a linear programming relaxation and its dual to make a random choice of facilities to be opened. We show how this algorithm can be derandomized to yield a 3-approximation algorithm.

## 1 Introduction

In the classical single-level uncapacitated facility location problem, we want to decide where to locate facilities, and to which open facility each client should be assigned, so as to minimize the total cost of setting up the facilities and of servicing the clients. In the  $k$ -level uncapacitated facility location problem, each client must be serviced by a sequence of  $k$  different facilities. This problem has applications, for instance, in more complex logistical systems where central depots ship goods through smaller depots, and further down the hierarchy, to the clients. We refer to Kaufman, Vanden Eede & Hansen [10], Van Roy & Erlenkotter [14], Tcha & Lee [13], Barros [2], and Aardal, Labbé, Leung & Queyranne [1] for further applications of multi-level location problems, as well as optimization algorithms and heuristic approaches. Since the  $k$ -level problem is  $\mathcal{NP}$ -hard even if  $k = 1$ , an approximation algorithm is useful in order to quickly find feasible solutions of good quality.

It was observed already in the 1970's that the linear relaxation of the so-called "strong" integer programming formulation of the 1-level problem provides values close to the integer optimum value,

---

\*aardal@cs.uu.nl. Department of Computer Science, Utrecht University. Research partially supported by the ESPRIT Long Term Research Project nr. 20244 (Project ALCOM-IT: *Algorithms and Complexity in Information Technology*), by the project TMR-DONET nr. ERB FMRX-CT98-0202, both of the European Community, by NSF grant CCR-9307391 through David B. Shmoys, Cornell University, and by NSF through the Center for Research on Parallel Computation, Rice University, under Cooperative Agreement No. CCR-9120008.

<sup>†</sup>chudak@cs.cornell.edu. IBM TJ Watson Research Center, Yorktown Heights, NY 10598. This research was done while the author was a graduate student at the School of Operations Research & Industrial Engineering, Cornell University, Ithaca, NY 14853. Research partially supported by NSF grants DMS-9505155 and CCR-9700029 and by ONR grant N00014-96-1-00500.

<sup>‡</sup>shmoys@cs.cornell.edu. School of Operations Research & Industrial Engineering, Cornell University. Research partially sponsored by NSF grants CCR-9700029 & DMS-9505155 and ONR grant N00014-96-1-00500.

and so it was natural to hope to find an approximation algorithm with a good performance guarantee, based on this relaxation. In practice, the dual-ascent based heuristic by Erlenkotter [7] produces excellent solutions, but in the worst case this algorithm performs arbitrarily poorly. We shall say that an algorithm for a particular optimization problem is a  $\rho$ -approximation algorithm if it is a polynomial-time algorithm that is guaranteed to find a feasible solution of objective function value within a factor of  $\rho$  of the optimum. For the 1-level problem where the objective is to maximize the “profit” of servicing the clients, a  $(1 - \frac{1}{e})$ -approximation algorithm was developed by Cornuéjols, Fisher & Nemhauser in 1977 [5], but for the version where we want to minimize the total cost, a constant performance guarantee remained elusive until recently. In 1997, Shmoys, Tardos & Aardal [12] considered the special case in which the service costs satisfy the triangle inequality, and gave a 3.16-approximation algorithm based on the linear relaxation for the 1-level problem. This performance guarantee was improved by Guha & Khuller [9] to 2.408, and later by Chudak & Shmoys [3, 4], who presented a 1.736-approximation algorithm. These algorithms are based on the linear relaxation as well. We give a 3-approximation algorithm for the  $k$ -level facility location problem, which is the first constant performance guarantee for any  $k \geq 2$ . (This improves on a preliminary version of this result, a 3.16-approximation algorithm for just the 2-level case [12]; the other results of this conference paper [12] will be published in a separate journal paper.)

For a more extensive overview of algorithmic results for various facility location problems we refer to Cornuéjols, Nemhauser & Wolsey [6], Shmoys et al. [12], and Chudak [3].

## 2 The $k$ -level uncapacitated facility location problem

Let  $D$  be the set of demand points. Each client  $j \in D$  must be assigned to precisely one facility at each of the  $k$  levels. Let  $F^l$  be the set of sites where facilities on level  $l$ ,  $1 \leq l \leq k$  may be located. We denote the set of all such sites,  $\cup_{l=1}^k F^l$ , by  $F$ . The cost of setting up facility at site  $i$  is  $f_i$ . We assume that  $f_i > 0$  for each  $i \in F$ . Whenever we use the expression “facility  $i$ ” we mean “facility at site  $i$ ”. The cost of shipping between points  $i, j \in F \cup D$  is equal to  $c_{ij}$ . Throughout this paper, we make the following assumptions on the service costs:

- $c_{ij} \geq 0$ , for each  $i, j \in F \cup D$ ;
- $c_{ij} = c_{ji}$ , for each  $i, j \in F \cup D$ , i.e., the service costs are *symmetric*,
- $c_{ik} \leq c_{ij} + c_{jk}$ , for each  $i, j, k \in F \cup D$ , i.e., the service costs satisfy the *triangle inequality*.

We shall use  $p$  to denote a sequence of facilities  $i_l \in F^l$ ,  $l = 1, \dots, k$ , and shall refer to  $p$  as a *path* of facilities. The set of all possible paths is denoted by  $P$ . Each client must be assigned to precisely one path  $p \in P$ . The total service cost incurred by assigning client  $j$  to path  $p = (i_1, i_2, \dots, i_k)$  is equal to  $c_{pj} = c_{i_1, i_2} + c_{i_2, i_3} + \dots + c_{i_{k-1}, i_k} + c_{i_k, j}$ .

Let  $x_{pj}$  be equal to 1 if client  $j$  is assigned to path  $p$ , and 0 otherwise. Furthermore, let  $y_{i_l}$  be equal to 1 if facility  $i_l$  is open at level  $l$ . If we relax the constraints that  $x_{pj} \in \{0, 1\}$ , for each  $p \in P$  and  $j \in D$ , and  $y_{i_l} \in \{0, 1\}$ , for each  $i_l \in F^l$ ,  $l = 1, \dots, k$ , we obtain the following linear programming relaxation of the  $k$ -level uncapacitated facility location problem:

$$z_{LP} = \min \sum_{l=1}^k \sum_{i_l \in F^l} f_{i_l} y_{i_l} + \sum_{p \in P} \sum_{j \in D} c_{pj} x_{pj} \quad (1)$$

subject to

$$\sum_{p \in P} x_{pj} = 1, \quad \text{for each } j \in D, \quad (2)$$

$$\sum_{p: p \ni i_l} x_{pj} - y_{i_l} \leq 0, \quad \text{for each } j \in D \text{ and } i_l \in F^l, l = 1, \dots, k, \quad (3)$$

$$x_{pj} \geq 0, \quad \text{for each } p \in P, \text{ and } j \in D, \quad (4)$$

$$y_{i_l} \geq 0, \quad \text{for each } i_l \in F^l, \ l = 1, \dots, k. \quad (5)$$

In the integer programming formulation, constraints (2) impose that each client is assigned to precisely one path. To interpret constraints (3), fix a client  $j$ , and a facility  $i_l$  on level  $l$ . No assignment of client  $j$  to a path using facility  $i_l$  is possible unless facility  $i_l$  is open. If  $y_{i_l}$  is equal to 0, then the sum of all assignment variables for client  $j$  to use paths containing facility  $i_l$  must also be 0. The upper bounds  $x_{pj} \leq 1$  are not necessary in the model due to the combination of constraints (2) and (4). Similarly, we do not need to state the constraints  $y_{i_l} \leq 1$  explicitly since the sum in constraints (3),  $\sum_{p: p \ni i_l} x_{pj}$ , is bounded from above by the sum in constraints (2),  $\sum_{p \in P} x_{pj} = 1$ , and since we assume that the fixed cost  $f_{i_l} > 0$ , for each  $i_l \in F^l$ .

Let  $v_j$  and  $w_{i_l, j}$  denote the dual variables corresponding to the primal constraints (2) and (3), respectively. The dual problem corresponding to the linear programming relaxation is given below:

$$z_{LP} = \max \sum_{j \in D} v_j \quad (6)$$

subject to

$$v_j - \sum_{i_l \in p} w_{i_l, j} \leq c_{pj}, \quad \text{for each } p \in P, \text{ and } j \in D, \quad (7)$$

$$\sum_{j \in D} w_{i_l, j} \leq f_{i_l}, \quad \text{for each } 1 \leq l \leq k, \text{ and } i_l \in F^l, \quad (8)$$

$$w_{i_l, j} \geq 0, \quad \text{for each } 1 \leq l \leq k, i_l \in F^l, \text{ and } j \in D. \quad (9)$$

Since the dual has a polynomial number of variables, we can use the ellipsoid algorithm to solve this linear program in polynomial time, given a polynomial-time separation algorithm [8]. In other words, we must design a polynomial-time algorithm that, given  $(v, w)$ , either concludes that  $(v, w)$  is feasible, or else outputs a constraint that is violated by it. Constraints (8) can be checked explicitly (since there are only a polynomial number of them). For each  $j \in D$ , the constraints (7) can be checked by a straightforward shortest path computation. Since the dual linear program can be solved in polynomial time, the primal can also be solved in polynomial time.

In the analysis of our algorithm, we will use the following lemma in order to bound the value of the service cost of a feasible integer solution.

**Lemma 1** *Let  $(\bar{x}, \bar{y})$  and  $(\bar{v}, \bar{w})$ , respectively, be optimal solutions to the primal and dual linear programs. Then  $\bar{x}_{pj} > 0$  implies that  $c_{pj} \leq \bar{v}_j$ , for each  $p \in P$  and  $j \in D$ .*

*Proof:* Consider the following set of complementary slackness conditions:

$$\bar{x}_{pj}(c_{pj} - \bar{v}_j + \sum_{i_l \in p} \bar{w}_{i_l, j}) = 0, \quad \text{for each } p \in P, \text{ and } j \in D, \quad (10)$$

which must hold for any pair of optimal solutions to the linear program and its dual. If  $\bar{x}_{pj} > 0$ , then  $c_{pj} - \bar{v}_j + \sum_{i_l \in p} \bar{w}_{i_l, j} = 0$ , or equivalently,  $c_{pj} + \sum_{i_l \in p} \bar{w}_{i_l, j} = \bar{v}_j$ . Since  $\bar{w}_{i_l, j} \geq 0$ , we have that  $c_{pj} \leq \bar{v}_j$ . ■

### 3 The algorithm

We next present a randomized algorithm that produces feasible solutions of expected value no more than 3 times the optimum value. After the description of the algorithm, we prove the claimed performance guarantee, and then describe how the algorithm can be derandomized.

**Description of the algorithm.** We shall refer to the algorithm described below as *Random Path*. We start by solving the linear programs (1) – (5) and (6) – (9) to optimality. We denote the optimal solutions by  $(\bar{x}, \bar{y})$  and  $(\bar{v}, \bar{w})$ , respectively. The aim of our algorithm is to construct so-called *clusters* of facility and demand locations. Each cluster consists of a path of facility locations, and a set of clients that are serviced uniquely by this path. Each cluster is *centered* at a certain demand point, as described below. During the execution of our algorithm we maintain:

1. A feasible fractional solution  $(x, y)$ . Initially, we set  $(x, y) := (\bar{x}, \bar{y})$ .
2. A set  $\mathcal{C}$  of clusters. Initially, we set  $\mathcal{C} := \emptyset$ .
3. A set  $\bar{D}$  of demand points that are not part of any cluster in  $\mathcal{C}$ . Initially,  $\bar{D} := D$ .

In each iteration, we choose a client  $j \in \bar{D}$  with minimum value of  $\bar{v}_j + \bar{c}_j$ , where  $\bar{c}_j = \sum_{p \in P} c_{pj} \bar{x}_{pj}$ . Let  $j_t$  denote the client chosen in iteration  $t$ . Client  $j_t$  is the *center* of the cluster  $C_t$ , which is computed in the following way. Let  $P_t$  be the set of paths of facilities that service client  $j_t$  by any positive fraction, i.e.,  $P_t = \{p \in P : x_{p, j_t} > 0\}$ . Also, let  $F_t^l = \{i_l \in F^l : i_l \text{ belongs to at least one path in } P_t\}$ , for each  $l = 1, \dots, k$ , and  $F_t = \cup_{l=1}^k F_t^l$ . Let  $T_t = \{j \in D : \exists x_{pj} > 0 \text{ such that } p \text{ contains at least one facility in } F_t\}$ . Notice that  $j_t \in T_t$ . For an illustration of the sets  $P_t$  and  $T_t$ , we refer to Figure 1. Let  $C_t$  be the set of all clients in  $T_t$  and all facilities in  $F_t$ , i.e.,  $C_t = T_t \cup F_t$ .

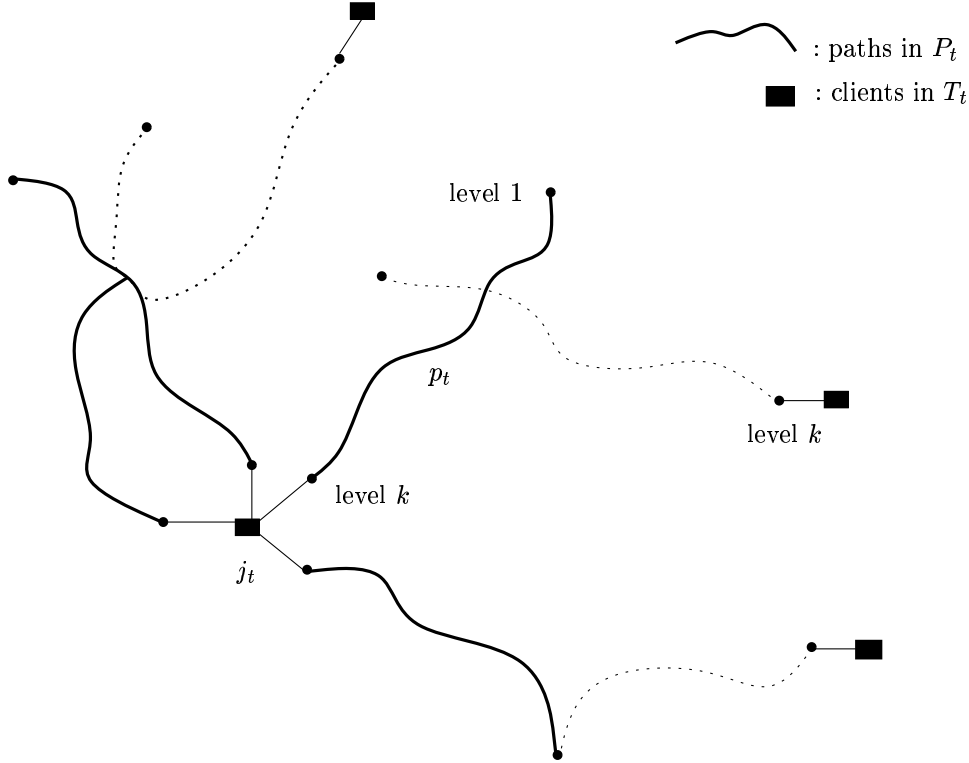


Figure 1: The paths in  $P_t$ , the client  $j_t$ , and the other clients in  $T_t$ .

Next, choose a path  $p \in P_t$  with probability  $x_{p, j_t}$ . Call the chosen path  $p_t$ . Round all variables  $\{y_{i_l}\}_{i_l \in p_t}$  to 1 and all variables  $\{y_{i_l}\}_{i_l \in F_t - p_t}$  to 0. We assign each client in  $T_t$  to the path  $p_t$ ; that is, we set  $x_{p_t, j} = 1$  for each  $j \in T_t$  and  $x_{pj} = 0$  for each  $j \in T_t$  and  $p \in P_t - \{p_t\}$ . The situation after rounding is illustrated in Figure 2. We now set  $\bar{D} := \bar{D} - T_t$  and  $\mathcal{C} := \mathcal{C} \cup \{C_t\}$ . We iterate this

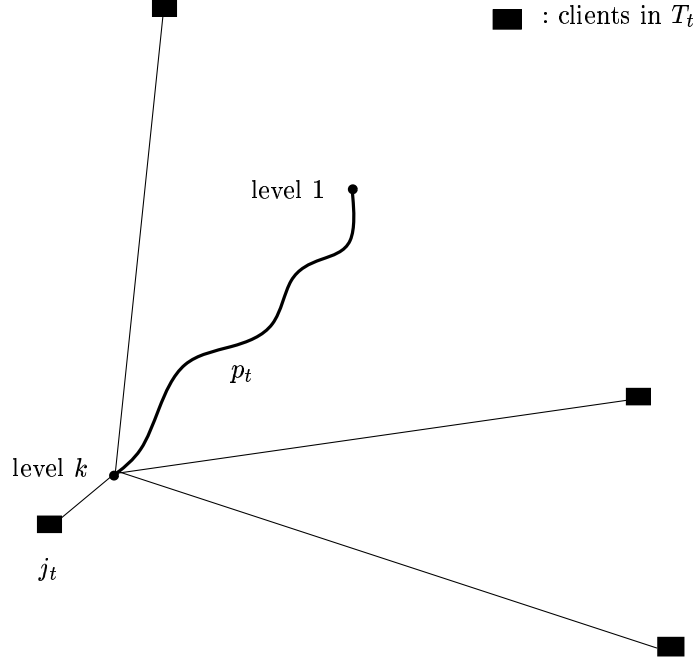


Figure 2: The solution created by rounding within the cluster  $C_t$ .

process until  $\bar{D} = \emptyset$ . Notice that each client belongs to precisely one cluster in  $\mathcal{C}$  when the algorithm terminates.

We are now ready to prove the following theorem.

**Theorem 2** *The algorithm Random Path runs in polynomial time, and produces a feasible integer solution whose expected total cost is no more than 3 times the optimum cost.*

*Proof:* We first show that our algorithm produces a feasible integer solution. In the rounding step of iteration  $t$  we create a cluster by first choosing a certain client  $j_t$  as a center. Then we randomly choose one of the fractional paths  $p_t \in P_t$ . Each variable  $y_{i_l} : i_l \in p_t$  is set to 1, and so all facilities in path  $p_t$  are open. Finally, we assign all clients in the set  $T_t$  to the path  $p_t$ . The assignment of the clients in  $T_t$  to the path  $p_t$  is integer and feasible. No client in  $\bar{D} - T_t$  is fractionally serviced by any of the facilities in  $F_t$ , by definition, and so they are unaffected by the current rounding step. Hence, a feasible fractional solution  $(x, y)$  is maintained during the execution of the algorithm. Upon termination of the algorithm the solution  $(x, y)$  is integral.

We now need to prove that the algorithm has the claimed performance guarantee. Consider the cluster  $C_t$  created in iteration  $t$ . Although the algorithm updates the solution  $(x, y)$  throughout its execution, at the start of iteration  $t$ , we have that  $x_{pj} = \bar{x}_{pj}$ , for each  $p \in P_t$  and  $j \in \bar{D}$ , and  $y_{i_l} = \bar{y}_{i_l}$ , for each  $i_l \in F_t^l$ ,  $l = 1, \dots, k$ . Hence, the expected fixed cost incurred in iteration  $t$  is equal to

$$\sum_{p \in P_t} \left( \sum_{i_l \in p} f_{i_l} \right) \bar{x}_{p, j_t} = \sum_{l=1}^k \sum_{i_l \in F_t^l} f_{i_l} \left( \sum_{p: p \ni i_l} \bar{x}_{p, j_t} \right) \leq \sum_{l=1}^k \sum_{i_l \in F_t^l} f_{i_l} \bar{y}_{i_l}, \quad (11)$$

where the inequality holds due to constraints (3).

Furthermore, the expected service cost for client  $j_t$  is equal to

$$\sum_{p \in P_t} c_{p, j_t} \bar{x}_{p, j_t} = \bar{c}_{j_t}. \quad (12)$$

To obtain a bound on the expected service cost for each client  $j \in C_t$ ,  $j \neq j_t$ , we need to observe what happens in the rounding step. Since client  $j$  belongs to the set  $T_t$ , it is fractionally serviced through at least one facility that is also fractionally servicing the center  $j_t$  of cluster  $C_t$ . Let  $\bar{p}$  be the path that fractionally services  $j$  and that has at least one facility in common with a path  $p' \in P_t$ . Let facility  $i_q$  be the facility with largest level index that belongs to both paths  $\bar{p}$  and  $p'$ ; see Figure 3. After rounding, client  $j$  is completely serviced by path  $p_t$ . Due to the triangle inequality we can

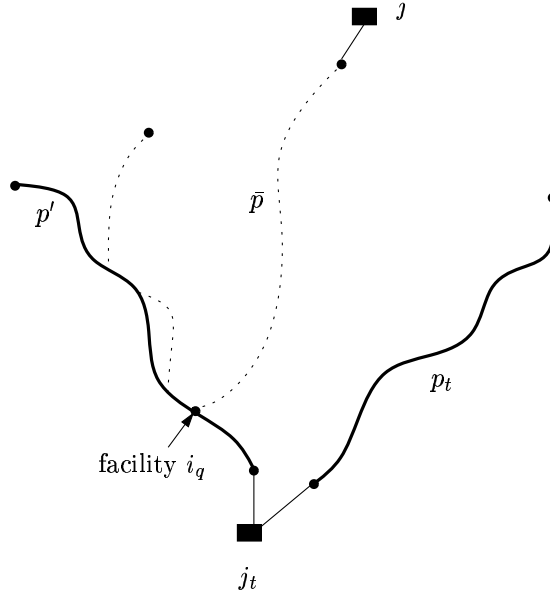


Figure 3:

bound the cost  $c_{p_t, j}$  from above by the sum of the cost of traversing  $\bar{p}$  from  $j$  to  $i_q$ , and the cost of traversing  $p'$  from  $i_q$  to  $j_t$ , and the cost of traversing all of  $p_t$  from  $j_t$ . Hence, by linearity of expectation and the fact that the costs are nonnegative and symmetric, we can bound from above the expected service cost for  $j$  by

$$c_{\bar{p}j} + c_{p', j_t} + \bar{c}_{j_t} \leq \bar{v}_j + \bar{v}_{j_t} + \bar{c}_{j_t} \leq 2\bar{v}_j + \bar{c}_j, \quad (13)$$

where the first inequality follows from Lemma 1, and the second inequality follows from the fact that the choice of client  $j_t$  ensures that  $\bar{c}_{j_t} + \bar{v}_{j_t} \leq \bar{c}_j + \bar{v}_j$ .

If we combine the expressions (11)–(13), then we conclude that the total expected cost for all clusters in  $\mathcal{C}$  is bounded by

$$\sum_{l=1}^k \sum_{i_l \in F^l} f_{i_l} \bar{y}_{i_l} + \sum_{j \in D} (\bar{c}_j + 2\bar{v}_j) = 2 \sum_{j \in D} \bar{v}_j + \sum_{l=1}^k \sum_{i_l \in F^l} f_{i_l} \bar{y}_{i_l} + \sum_{j \in D} \sum_{p \in P} c_{pj} \bar{x}_{pj} = 3z_{LP}.$$

The running time of the algorithm is dominated by the time required to solve the linear programming relaxation and its dual. As noted above, these linear programs can be solved in polynomial time by the ellipsoid algorithm.  $\blacksquare$

**Derandomizing the algorithm.** Since the *expected cost* of cluster  $C_t$  created in iteration  $t$  by the algorithm *Random Path* is bounded by

$$\sum_{l=1}^k \sum_{i_l \in F_t^l} f_{i_l} \bar{y}_{i_l} + \sum_{j \in T_t} (\bar{c}_j + 2\bar{v}_j), \quad (14)$$

there must exist a path in  $p'_t \in P_t$  such that the cost of cluster  $C_t$  obtained by choosing path  $p'_t$  is no more than the bound on the expected cost (14). Hence, in a derandomized version of our algorithm, we can choose any path in iteration  $t$  that yields a cost of cluster  $C_t$  that is less than or equal to the bound (14). One specific possibility is to evaluate the cluster cost for each path in  $P_t$ , and take the path that yields the minimum cost. If, for example, we assume that the solution  $(\bar{x}, \bar{y})$  is an extreme point solution of this linear program, then the number of positive variables  $\bar{x}_{pj}$  is at most the number of constraints in this linear program. Thus, there are only a polynomial number of paths to consider in each iteration.

**Theorem 3** *The derandomized version of Random Path is a 3-approximation algorithm.*

## 4 Discussion

An interesting open question is whether the techniques used by Chudak & Shmoys to improve the performance guarantee for the 1-level problem can be extended to the  $k$ -level problem as well. In our algorithm, as well as in the algorithm of Shmoys, Tardos & Aardal [12], the value of the fixed costs of the integer feasible solution does not increase, compared to the linear programming solution, so the increase in total cost over  $z_{LP}$  is all due to the bound on the service cost. Chudak & Shmoys gave a more refined rounding procedure so that the service cost and the facility cost are better balanced.

A related question is to obtain a tight bound on the *duality gap*, the worst-case ratio between the integer and fractional optimal values, for the  $k$ -level problem (and also the 1-level problem), given different assumptions on the objective function.

It would also be interesting to investigate how our algorithm performs in practice. The most substantial obstacle is the solution of the linear programming relaxation, but it would be interesting to see if a cutting-plane approach to solving the dual linear program (and hence the primal) would be possible with reasonable computation times. Such an approach might be further enhanced by preprocessing the input to reflect the fact that most (expensive) paths are unlikely to be used in an optimal primal solution. This approach has been successful, for example, in computational work on the traveling salesman problem, where one typically includes only the ten shortest edges from each vertex of the underlying graph in the initial formulation. Once this initial linear program has been solved to optimality, one can compute the reduced cost of every excluded variable to verify whether the current solution is optimal with respect to the complete set of variables.

## Acknowledgments

We wish to thank J.K. Lenstra and A.M. Verweij for useful comments on a preliminary version of this manuscript.

## References

- [1] K. Aardal, M. Labbé, J. Leung, and M. Queyranne. On the two-level uncapacitated facility location problem. *INFORMS J. Comput.*, 8:289–301, 1996.
- [2] A. I. M. B. de Barros. *Discrete and Fractional Programming Techniques for Location Problems*. Ph.D. Thesis, Erasmus Universiteit Rotterdam. Tinbergen Institute Research Series No. 89. Thesis Publishers, Amsterdam, 1995.
- [3] F. A. Chudák. Improved approximation algorithms for uncapacitated facility location. In: R.E. Bixby, E.A. Boyd, R. Z. Ríos-Mercado (eds.), *Integer Programming and Combinatorial Optimization, 6th International IPCO Conference*. Lecture Notes in Computer Science 1412, pages 180–194, Springer-Verlag, Berlin Heidelberg, 1998.



- [4] F. A. Chudák and D. B. Shmoys. Improved approximation algorithms for the uncapacitated facility location problem. Manuscript, 1999.
- [5] G. Cornuéjols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Sci.*, 8:789–810, 1977.
- [6] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. In: P. Mirchandani and R. Francis (eds.), *Discrete Location Theory*, pages 119–171. John Wiley and Sons, Inc., New York, 1990.
- [7] D. E. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research* 26:992–1009, 1978.
- [8] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.
- [9] S. Guha, and S. Khuller. Greedy strikes back: improved facility location algorithms. In: *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 649–657, 1998.
- [10] L. Kaufman, M. vanden Eede, and P. Hansen. A plant and warehouse location problem. *Operational Research Quarterly*, 28:547–557, 1977.
- [11] P. B. Mirchandani and R. L. Francis (eds.). *Discrete Location Theory*. John Wiley and Sons, Inc., New York, 1990.
- [12] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *29th ACM Symposium on Theory of Computing*, pages 265–274, 1997.
- [13] D. Tcha and B. Lee. A branch-and-bound algorithm for the multi-level uncapacitated location problem. *European J. Oper. Res.*, 18:35–43, 1984.
- [14] T. J. Van Roy and D. Erlenkotter. A dual based procedure for dynamic facility location. *Management Sci.*, 28:1091–1105, 1982.