ARRIVAL
Algorithms for Robust and online Railway optimization: Improving the Validity and reliability of Large scale systems

STREP
Member of the FET Open Scheme

ARRIVAL – TR – 0192

On Fault-Tolerant Facility Location

Jaroslaw Byrka, Aravind Srinivasan, Chaitanya Swamy

February 2009
On Fault-Tolerant Facility Location

Jaroslaw Byrka\textsuperscript{1,2}, Aravind Srinivasan\textsuperscript{3}, and Chaitanya Swamy\textsuperscript{4}

\textsuperscript{1} Centrum Wiskunde & Informatica, Kruislaan 413, NL-1098 SJ Amsterdam, Netherlands. J.Byrka@cwi.nl
\textsuperscript{2} Dep. of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, Netherlands.
\textsuperscript{3} Dept. of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA. srin@cs.umd.edu
\textsuperscript{4} Dept. of Combinatorics & Optimization, Faculty of Mathematics, University of Waterloo, ON N2L 3G1 Waterloo, CANADA. cswamy@math.uwaterloo.ca

Abstract. We give a new LP-rounding 1.724-approximation algorithm for the metric Fault-Tolerant Uncapacitated Facility Location problem. This improves on the previously best known 2.076-approximation algorithm of Shmoys & Swamy. Our work applies a dependent-rounding technique in the domain of facility location. The analysis of our algorithm benefits from, and extends, methods developed for Uncapacitated Facility Location; it also helps uncover new properties of the dependent-rounding approach.

1 Introduction

In Facility Location problems we are given a set of clients $\mathcal{C}$ who require a certain service. To provide such service we need to open a subset of a given set of facilities $\mathcal{F}$. Opening each facility $i \in \mathcal{F}$ costs $f_i$ and servicing a client $j$ by facility $i$ costs $c_{ij}$; the standard assumption is that the $c_{ij}$ are symmetric and constitute a metric. (The non-metric case is much harder to approximate.) In this paper, we follow Shmoys & Swamy [10] and study the Fault-Tolerant Facility Location (FTFL) problem, where each client has a positive integer specified as its coverage requirement $r_j$. The task is to find a minimum-cost solution which opens some facilities from $\mathcal{F}$ and connects each client $j$ to $r_j$ different open facilities.

The FTFL problem was introduced by Jain & Vazirani [6]. Guha et al. [5] gave the first constant factor approximation algorithm with approximation ratio 2.408. This was later improved by Swamy & Shmoys [10] who gave a 2.076-approximation algorithm.

The FTFL problem generalizes the standard Uncapacitated Facility Location (UFL) problem wherein $r_j = 1$ for all $j$, for which Guha & Khuller [4] proved...
an approximation lower bound at 1.463. The currently best approximation ratio for UFL is achieved by the 1.5-approximation algorithm of Byrka [2].

1.1 Our contribution

In this paper we give a new LP-rounding 1.724-approximation algorithm for the FTFL problem. We give the first application of the dependent rounding technique of [9] to a facility location problem.

Our algorithm uses a novel clustering method, which allows clusters not to be disjoint, but rather to form a laminar family of subsets of facilities. The hierarchical structure of the obtained clustering allows for a proper usage of properties of dependent rounding.

Finally, one of our main technical contributions is Theorem 2, which states a new property of the dependent rounding technique itself.

2 Dependent rounding

Given a fractional vector $y = (y_1, y_2, \ldots, y_N) \in [0,1]^N$ we often seek to round it to an integral vector $\hat{y} \in \{0,1\}^N$ that is in a problem-specific sense very “close to” $y$. The dependent-randomized-rounding technique of [9] is one such approach known for preserving the sum of the entries deterministically, along with concentration bounds for any linear combination of the entries; we will generalize a known property of this technique in order to apply it to the FTFL problem. The very useful pipage rounding technique of [1] was developed prior to [9], and can be viewed as a derandomization (deterministic analog) of [9] via the method of conditional probabilities. Indeed, the results of [1] were applied in the work of [10]; the probabilistic intuition, as well as our generalization of the analysis of [9], help obtain our results.

Define $[t] = \{1,2,\ldots,t\}$. Given a fractional vector $y = (y_1, y_2, \ldots, y_N) \in [0,1]^N$, the rounding technique of [9] (henceforth just referred to as “dependent rounding”) is a polynomial-time randomized algorithm to produce a random vector $\hat{y} \in \{0,1\}^N$ with the following three properties:

(P1): marginals. $\forall i$, $\Pr[\hat{y}_i = 1] = y_i$;

(P2): sum-preservation. With probability one, $\sum_{i=1}^N \hat{y}_i$ equals either $\lfloor \sum_{i=1}^N y_i \rfloor$ or $\lceil \sum_{i=1}^N y_i \rceil$; and

(P3): negative correlation. $\forall S \subseteq [N]$, $\Pr[\bigwedge_{i \in S}(\hat{y}_i = 0)] \leq \prod_{i \in S}(1 - p_i)$, and $\Pr[\bigwedge_{i \in S}(\hat{y}_i = 1)] \leq \prod_{i \in S} p_i$.

The dependent-rounding algorithm is described in Appendix A. In this paper, we also consider a version of this technique (henceforth just referred to as “partial dependent rounding”), which potentially leaves one of the entries fractional, i.e., the resulting $\hat{y}$ vector has at most one entry $\hat{y}_i$ with $0 < \hat{y}_i < 1$ and all the other entries are in $\{0,1\}$. Partial dependent rounding, when applied to a subset of entries of $y$ indexed by $S \subseteq [N]$ has the following variant of property (P2):
(P2'): sum-preservation. With probability one, \( \sum_{i \in S} \hat{y}_i = \sum_{i \in S} y_i \)
and \( |\{i \in S : \hat{y}_i = 1\}| = |\sum_{i \in S} y_i| \).

Dependent rounding may be implemented to incorporate partial dependent rounding for a given family of subsets \( S \subset 2^{[N]} \), provided that \( S \) is a laminar family.

Now, let \( S \subseteq [N] \) be any subset, not necessarily form from \( S \). In order to present our results, we need two functions, \( \text{Sum}_S \) and \( g_{\lambda,S} \). For any vector \( x \in \{0,1\}^n \), let \( \text{Sum}_S(x) = \sum_{i \in S} x_i \) be the sum of the elements of \( x \) indexed by elements of \( S \); in particular, if \( x \) is a (possibly random) vector with all entries either 0 or 1, then \( \text{Sum}_S(x) \) counts the number of entries in \( S \) that are 1. Next, given \( s = |S| \) and a real vector \( \lambda = (\lambda_0, \lambda_1, \ldots, \lambda_s) \), we define, for any \( x \in \{0,1\}^n \),
\[
g_{\lambda,S}(x) = \sum_{i=0}^s \lambda_i \cdot I(\text{Sum}_S(x) = i),
\]
where \( I(\cdot) \) denotes the indicator function. Thus, \( g_{\lambda,S}(x) = \lambda_i \) if \( \text{Sum}_S(x) = i \).

Let \( \mathcal{R}(y) \) be a random vector in \( \{0,1\}^N \) obtained by independently rounding each \( y_i \) to 1 with probability \( y_i \), and to 0 with the complementary probability of \( 1 - y_i \). Suppose, as above, that \( \hat{y} \) is a random vector in \( \{0,1\}^N \) obtained by applying the dependent rounding technique to \( y \). We start with a general theorem and then specialize it to Theorem 2 that will be very useful for us:

**Theorem 1.** Suppose we conduct dependent rounding on \( y = (y_1, y_2, \ldots, y_N) \). Let \( S \subseteq [N] \) be any subset with cardinality \( s \geq 2 \), and let \( \lambda = (\lambda_0, \lambda_1, \ldots, \lambda_s) \) be any vector such that for all \( r \) such that \( 0 \leq r \leq s - 2 \), \( \lambda_r - 2\lambda_{r+1} + \lambda_{r+2} \leq 0 \). Then, \( \mathbb{E}[g_{\lambda,S}(\hat{y})] \geq \mathbb{E}[g_{\lambda,S}(\mathcal{R}(y))] \).

**Theorem 2.** For any \( y \in \{0,1\}^N \), \( S \subseteq [N] \), and \( k = 1, 2, \ldots, \), we have
\[
\mathbb{E}[\min\{k, \text{Sum}_S(\hat{y})\}] \geq \mathbb{E}[\min\{k, \text{Sum}_S(\mathcal{R}(y))\}].
\]

Using the notation \( \exp(t) = e^t \), our next key result is:

**Theorem 3.** For any \( y \in \{0,1\}^N \), \( S \subseteq [N] \), and \( k = 1, 2, \ldots, \), we have
\[
\mathbb{E}[\min\{k, \text{Sum}_S(\mathcal{R}(y))\}] \geq k \cdot (1 - \exp(-\text{Sum}_S(y)/k)).
\]

The above two theorems yield a key corollary that we will use:

**Corollary 1.**
\[
\mathbb{E}[\min\{k, \text{Sum}_S(\hat{y})\}] \geq k \cdot (1 - \exp(-\text{Sum}_S(y)/k)).
\]

Let us introduce one more piece of notation. For any \( w \in \mathcal{R}^N_{\geq 0}, k \in \{1, 2, \ldots, \}, \ S \subseteq [N] \), define \( X_0 = \{i \in S : \hat{y}_i = 1\} \) and
\[
X(S, \hat{y}, w, k) = \begin{cases} X_0 & \text{if } |X_0| \leq k \\
\arg\min_{X' \subseteq X_0, |X'| = k} \sum_{i \in X'} w_i & \text{if } |X_0| > k
\end{cases}
\]
Note that the above corollary is a bound on \( \mathbb{E}[|X(S, \hat{y}, w, k)|] \). We will also use the following bound.
Theorem 4. Let \( X = X(S, \hat{y}, w, k) \), then

\[
E[\sum_{i \in X} w_i] \leq E[|X|] \cdot \frac{\sum_{i \in S} y_i w_i}{\sum_{i \in S} y_i}.
\]

3 Algorithm

3.1 LP-relaxation

The FTFL problem is defined by the following Integer Program (IP).

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in F} f_i y_i + \sum_{j \in C} \sum_{i \in F} c_{ij} x_{ij} & (1) \\
\text{subject to:} & \quad \sum_{i} x_{ij} \geq r_j & \forall j \in C & (2) \\
& x_{ij} \leq y_i & \forall j \in C \forall i \in F & (3) \\
& y_i \leq 1 & \forall i \in F & (4) \\
& x_{ij}, y_i \in \mathbb{Z}^0 & \forall j \in C \forall i \in F, & (5)
\end{align*}
\]

where \( C \) is the set of clients, \( F \) is the set of possible locations of facilities, \( f_i \) is a cost of opening a facility at location \( i \), \( c_{ij} \) is a cost of serving client \( j \) from a facility at location \( i \), and \( r_j \) is the amount of facilities client \( j \) needs to be connected to.

If we relax constraint (5) to \( x_{ij}, y_i \geq 0 \) we obtain the standard LP-relaxation of the problem. Let \((x^*, y^*)\) be an optimal solution to this LP relaxation. We will give an algorithm that rounds this solution to an integral solution \((\tilde{x}, \tilde{y})\) with cost at most \( \gamma \approx 1.7245 \) times the cost of \((x^*, y^*)\).

3.2 Scaling

We may assume, without loss of generality, that for any client \( j \in C \) there exists at most one facility \( i \in F \) such that \( x_{ij} < y_i \). Moreover, this facility may be assumed to have the highest distance to client \( j \) among the facilities that fractionally serve \( j \) in \((x^*, y^*)\).

We first set \( \hat{x}_{ij} = \hat{y}_i = 0 \) for all \( i \in F, j \in C \). Then we scale up the fractional solution by the constant \( \gamma \approx 1.7245 \) to obtain a fractional solution \((\hat{x}, \hat{y})\). To be precise: we set \( \hat{x}_{ij} = \min\{1, \gamma \cdot x^*_{ij}\}, \hat{y}_i = \min\{1, \gamma \cdot y^*_i\} \).

We open each facility \( i \) with \( \hat{y}_i = 1 \) and connect each client-facility pair with \( \hat{x}_{ij} = 1 \). To be more precise, we modify \( \hat{y}, \hat{x}, \hat{\tilde{x}} \) and service requirements \( r \) as follows. For each facility \( i \) with \( \hat{y}_i = 1 \), set \( \hat{y}_i = 0 \) and \( \hat{y}_i = 1 \). Then, for every pair \((i, j)\) such that \( \hat{x}_{ij} = 1 \), set \( \hat{x}_{ij} = 0, \hat{\tilde{x}}_{ij} = 1 \) and decrease \( r_j \) by one. When this process is finished we call the resulting \( r, \hat{y} \) and \( \hat{x} \) by \( \tilde{\mathbf{7}}, \tilde{\mathbf{y}} \) and \( \tilde{\mathbf{x}} \). Note that the connections that we made in this phase may be paid for by a difference in the connection cost between \( \tilde{x} \) and \( \tilde{x} \). We will show that the remaining connection cost of the solution of the algorithm is expected to be at most the cost of \( \tilde{\mathbf{7}} \).

For the feasibility of the final solution, it is essential that if we connected client \( j \) to facility \( i \) in this initial phase, we will not connect it again to \( i \) in the
rest of the algorithm. There will be two ways of connecting clients in the process of rounding $\bar{x}$. The first one connects client $j$ to a subset of facilities serving $j$ in $\bar{x}$. Recall that if $j$ was connected to facility $i$ in this initial phase, then $\bar{x}_{ij} = 0$, and no additional $i - j$ connection will be created.

The connections of the second type will be created in a process of clustering. The clustering that we will use is a generalization of the clustering used by Chudak & Shmoys for the UFL problem [3]. As a result of this clustering process, client $j$ will be allowed to connect itself via a different client $j'$ to a facility open around $j'$. $j'$ will be called a cluster center for a subset of facilities, and it will make sure that at least some guaranteed number of these facilities will get opened.

To be certain that client $j$ does not get again connected to facility $i$ with a path via client $j'$, facility $i$ will never be a member of the set of facilities clustered by client $j'$. We call a facility $i$ special for client $j$ iff $\tilde{y}_i = 1$ and $0 < x_{ij} < 1$. Note that, by our earlier assumption, there is at most one special facility for each client $j$, and that a special facility must be at maximal distance among facilities serving $j$ in $\bar{x}$. When rounding the fractional solution in Section 3.5, we take care that special facilities are not members of the formed clusters.

### 3.3 Close and distant facilities

Before we describe how do we cluster facilities, we specify the facilities that are interesting for a particular client in the clustering process. The following can be bought of as a version of a filtering technique of Lin and Vitter [7], first applied to facility location by Shmoys et al. [8]. The analysis that we use here is a version of the argument of Byrka [2].

As a result of the scaling that was described in the previous section, the connection variables $\bar{x}$ amount for a total connectivity that potentially exceeds the requirement $\bar{r}$. More precisely, we have $\sum_{i \in F} \bar{x}_{ij} \geq \gamma \cdot \bar{r}_j$ for every client $j \in C$. We will consider for each client $j$ a subset of facilities that are just enough to provide it a fractional connection of $\bar{r}_j$. Such a subset is called a set of close facilities of client $j$ and is defined as follows.

For every client $j$ consider the following construction. Let $i_1, i_2, \ldots, i_{|F|}$ be the ordering of facilities in $F$ in a nondecreasing order of distances $c_{ij}$ to client $j$. Let $i_k$ be the facility in this ordering, such that $\sum_{l=1}^{k-1} \bar{x}_{ilj} < \bar{r}_j$ and $\sum_{l=1}^k \bar{x}_{ilj} \geq \bar{r}_j$. Define

$$
\bar{x}^{(c)}_{ij} = \begin{cases} 
\bar{x}_{ij} & \text{for } l < k, \\
\bar{r}_j - \sum_{l=1}^{k-1} \bar{x}_{ilj} & \text{for } l = k, \\
0 & \text{for } l > k
\end{cases}
$$

Define $\bar{x}^{(d)}_{ij} = \bar{x}_{ij} - \bar{x}^{(c)}_{ij}$ for all $i \in F, j \in C$.

We will call the set of facilities $i \in F$ such that $\bar{x}^{(c)}_{ij} > 0$ the set of close facilities of client $j$ and we denote it by $C_j$. By analogy, we will call the set of facilities $i \in F$ such that $\bar{x}^{(d)}_{ij} > 0$ the set of distant facilities of client $j$ and denote it $D_j$. Observe that for a client $j$ the intersection of $C_j$ and $D_j$ is either
empty, or contains exactly one facility. In the latter case, we will say that this facility is both distant and close. Note that, unlike in the UFL problem, we may not simply split this facility to the close and the distant part, because it is essential that we make at most one connection to this facility in the final integral solution. Let \( d_j^{(\text{max})} = c_{ik} \) be the distance from client \( j \) to the farthest of its close facilities.

### 3.4 Clustering

We now construct a family of subsets of facilities \( S \in 2^F \). These subsets \( S \in S \) will be called clusters and they will guide to rounding procedure described next. There will be a client related to each cluster with a single client \( j \) related to at most one cluster, which we will call \( S_j \).

Clients \( j \) with \( r_j = 1 \) and a special facility \( i' \in C_j \) (recall that a special facility is a facility that is fully open in \( y \) but only partially used by \( j \) in \( \overline{y} \)) will be called special and will not take part in the clustering process. Let \( C' \) denote the set of all other, non-special clients. Observe that clients \( j \) with \( r_j \geq 2 \) do not have any special facilities among their close facilities. As a consequence, there are no special facilities among the close facilities of clients from \( C' \) involved in the clustering.

Let us define \( y(i) = 0 \) for all facilities \( i \in F \) and \( y(S) = \lceil \sum_{i \in S} y_i \rceil \). For each client \( j \in C \) we will keep two sets \( A_j \) and \( B_j \). At the start of the clustering process \( A_j = C_j \) and \( B_j = \emptyset \); \( A_j \) will be used to store close facilities and clusters of close facilities, \( B_j \) will be used to store only clusters. We will slightly abuse the set notation and write that clusters contain both facilities and smaller clusters. More formally we could write that clusters are just subsets of facilities, but we would like to emphasize the hierarchical structure of the formed system of clusters. One may imagine facilities being leaves and clusters being internal nodes of a forest that eventually becomes a tree, when all the clusters are added.

Note that our clustering is related to, but more complex then the one of Chudak and Shmoys [3] for UFL and of Swamy and Shmoys [10] for FTFL, where clusters are pairwise disjoint and each contains facilities whose fractional opening sums up to or slightly exceeds the value of 1.

We use the following procedure to compute clusters. While there exists a client \( j \in C \), such that \( rr_j = \sum_{S \in (A_j \cup B_j)} y(S) > 0 \), take such \( j \) with minimal \( d_j^{(\text{max})} \) and do the following:

1. Take \( S_j \) to be an inclusion-wise minimal subset of \( A_j \), such that \( \sum_{i \in S_j} (y_i - y(i)) \geq rr_j \). Note, that the summation is over both facilities and clusters in \( S_j \); a cluster form \( A_j \) may either be included in \( S_j \) or be disjoint form \( S_j \).
2. Make \( S_j \) a new cluster by setting \( S \leftarrow S \cup \{S_j\} \).
3. Update \( A_j \leftarrow (A_j \setminus S_j) \cup \{S_j\} \).
4. For each client \( j' \) with \( rr_{j'} > 0 \) do
   - If \( S_j \subseteq A_{j'} \), then set \( A_{j'} \leftarrow (A_{j'} \setminus S_j) \cup \{S_j\} \).
   - If \( S_j \cap A_{j'} \neq \emptyset \) and \( S_j \setminus A_{j'} \neq \emptyset \),
     then set \( A_{j'} \leftarrow A_{j'} \setminus S_j \) and \( B_{j'} \leftarrow (B_{j'} \setminus A_{j'}) \cup \{S_j\} \).
Eventually, add a cluster \( S_r = \mathcal{F} \) containing all the facilities to the family \( \mathcal{S} \).

We call a client \( j' \) active in a particular iteration, if before this iteration its residual requirement \( \text{rr}_j = \bar{\gamma}_j - \sum_{i \in (A_j \cup B_j)} y(i) \) was positive. During the above procedure, all active clients \( j \) have in their sets \( A_j \) and \( B_j \) only facilities and clusters that are not members of any other clusters (i.e., they are roots of their trees in the current forest). Therefore, when a new cluster \( S_j \) is created, it contains all the other clusters with which it has nonempty intersections (i.e., the new cluster \( S_j \) becomes a root of a new tree).

We shall now argue that there is enough fractional opening in facilities and clusters in \( A_j \) to cover the residual requirement \( \text{rr}_j \) when cluster \( S_j \) is to be formed. Consider a fixed client \( j \in C' \). At the start of the clustering we have \( A_j = C_j \), and therefore \( \sum_{i \in A_j} (\bar{\gamma}_i - y(i)) = \sum_{i \in C_j} \bar{\gamma}_i \geq \gamma_j = \text{rr}_j \). It remains to show, that \( \sum_{i \in A_j} (\bar{\gamma}_i - y(i)) - \text{rr}_j \) does not decrease over time. When a client \( j' \) is considered and cluster \( S_{j'} \) is created, the following cases are possible:

1. \( S_{j'} \cap A_j = \emptyset \), then \( A_j \) and \( \text{rr}_j \) do not change;
2. \( S_{j'} \subseteq A_j \), then \( A_j \) changes its structure, but \( \sum_{i \in A_j} \bar{\gamma}_i \) and \( \sum_{i \in B_j} y(i) \) do not change; hence \( \sum_{i \in A_j} (\bar{\gamma}_i - y(i)) - \text{rr}_j \) also does not change;
3. \( S_{j'} \cap A_j \neq \emptyset \) and \( S_{j'} \setminus A_j \neq \emptyset \), then, by inclusion-wise minimality of set \( S_{j'} \), we have \( [\bar{\gamma}_{S_{j'}}] - \sum_{S \in (S_{j'} \cup B_j)} [\bar{\gamma}_S] - \sum_{S \in (S_{j'} \cup A_j)} [\bar{\gamma}_S] \geq 0 \); hence, \( \sum_{i \in A_j} (\bar{\gamma}_i - y(i)) - \text{rr}_j \) cannot decrease.

Let \( A'_j = A_j \setminus \mathcal{F} \) be the set clusters in \( A_j \). Recall that all facilities in clusters in \( A'_j \) are close facilities of \( j \). Note, that each cluster \( S_{j'} \in B_j \) was created from close facilities of a client \( j' \) with \( d^{(\max)}_{j'} \leq d^{(\max)}_j \). We also have \( S_{j'} \cap C_j \neq \emptyset \), hence, by the triangle inequality, all facilities in \( S_{j'} \) are at distance at most \( 3 \cdot d^{(max)}_j \) from \( j \). We may conclude it with the following

**Corollary 2.** The computed family of clusters \( \mathcal{S} \) contains for each client \( j \in C' \) a collection of disjoint clusters \( A'_j \cup B_j \) containing only facilities within distance \( 3 \cdot d^{(max)}_j \), and \( \sum_{S \in A'_j \cup B_j} [\sum_{i \in S} \bar{\gamma}_i] \geq \gamma_j \).

### 3.5 Opening of facilities by dependent rounding

Given the family of subsets \( \mathcal{S} \in 2^\mathcal{F} \) computed by the clustering procedure from Section 3.4, we may proceed with rounding the fractional opening vector \( \bar{\gamma} \) into an integral vector \( y^R \). This is done by applying the dependent rounding technique of Section 2, guided by the family \( \mathcal{S} \), which is done as follows.

While \( \mathcal{S} \) is not empty:

1. take \( S \in \mathcal{S} \) such that there is no \( S' \in \mathcal{S} \) with \( S' \subset S \),
2. apply partial dependent rounding to entries of \( \bar{\gamma} \) indexed by elements of \( S \),
3. remove \( S \) from \( \mathcal{S} \).
If there remains a fractional entry, round it independently and let \( y^R \) be the resulting vector.

Observe that the above process is one of the possible implementations of dependent rounding applied to \( \hat{y} \). As a result, the random integral vector \( y^R \) satisfies properties (P1),(P2), and (P3). Additionally, property (P2') of the partial dependent rounding holds for each cluster \( S \in \mathcal{S} \). Hence, at least \( |\sum_{i \in S} y_i| \) entries in each \( S \in \mathcal{S} \) are rounded to 1. Therefore, by Corollary 2, we get

**Corollary 3.** For each client \( j \in C' \).

\[
|\{i \in \mathcal{F} | y_i^R = 1 \text{ and } c_{ij} \leq 3 \cdot d_j^{(\text{max})}\}| \geq \tau_j.
\]

Next, we combine the facilities opened by rounding \( y^R \) with facilities opened already when scaling which are recorded in \( \hat{y} \), i.e., we update \( \hat{y} \leftarrow \hat{y} + y^R \).

Eventually, we connect each client \( j \in C \) to \( r_j \) are closest opened facilities and code it in \( \hat{x} \). With the notation introduced at the end Section 2, this corresponds to setting \( \hat{x}_{ij} = 1 \) iff \( i \in X(S, \hat{y}, w, k) \), where \( S \leftarrow \mathcal{F} \), \( \hat{y} \leftarrow \hat{y}, w_i \leftarrow c_{ij} \), and \( k \leftarrow r_j \).

## 4 Analysis

We will now estimate the expected cost of the solution \((\hat{x}, \hat{y})\). The tricky part is to bound the connection cost, which we do as follows. We argue that a certain fraction of the demand of client \( j \) may be satisfied from its close facilities, then some part of the remaining demand can be satisfied from its distant facilities.

### 4.1 Average distances

Let us consider weighted average distances from a client \( j \) to sets of facilities fractionally serving it. Let \( d_j \) be the average connection cost in \( \mathcal{F} \) defined as

\[
d_j = \frac{\sum_{i \in \mathcal{F}} c_{ij} \cdot x_{ij}}{\sum_{i \in \mathcal{F}} x_{ij}}.
\]

Let \( d_j^{(c)}, d_j^{(d)} \) be the average connection costs in \( \mathcal{F}^{(c)} \) and \( \mathcal{F}^{(d)} \) defined as

\[
d_j^{(c)} = \frac{\sum_{i \in \mathcal{F}^{(c)}} c_{ij} \cdot x_{ij}^{(c)}}{\sum_{i \in \mathcal{F}^{(c)}} x_{ij}^{(c)}},
\]

\[
d_j^{(d)} = \frac{\sum_{i \in \mathcal{F}^{(d)}} c_{ij} \cdot x_{ij}^{(d)}}{\sum_{i \in \mathcal{F}^{(d)}} x_{ij}^{(d)}}.
\]

Let \( R_j \) be a parameter defined as

\[
R_j = \frac{d_j - d_j^{(c)}}{d_j}
\]

8
if \( d_j > 0 \) and \( R_j = 0 \) otherwise. Observe that \( R_j \) takes value between 0 and 1. \( R_j = 0 \) implies \( d^{(c)}_j = d_j = d^{(d)}_j \), and \( R_j = 1 \) occurs only when \( d^{(c)}_j = 0 \). The role played by \( R_j \) is that it measures a certain parameter of the instance, big values are good for one part of the analysis, small values are good for the other.

**Lemma 1.** \( d^{(d)}_j \leq d_j(1 + \frac{R_j}{\gamma - 1}) \).

**Proof.** Recall that \( \sum_{i \in F} x^{(c)}_{ij} = \tau_j \) and \( \sum_{i \in F} x^{(d)}_{ij} \geq (\gamma - 1) \cdot \tau_j \). Therefore, we have \( (d^{(d)}_j - d_j) \cdot (\gamma - 1) \leq (d_j - d^{(c)}_j) \cdot 1 = R_j \cdot d_j \), which is equivalent to \( d^{(d)}_j \leq d_j(1 + \frac{R_j}{\gamma - 1}) \).

Finally, observe that the average distance from \( j \) to the distant facilities of \( j \) gives an upper bound on the maximal distance to any of the close facilities of \( j \). Namely, \( d^{(max)}_j \leq d^{(d)}_j \).

### 4.2 Amount of service from close and distant facilities

We will now argue, that in the solution \( (\tilde{x}, \tilde{y}) \) a certain portion of the demand is expected to be served by the close and distant facilities of each client.
Recall that for a client $j$ it is possible, that there is a facility that is both its close and its distant facility. Once we have a solution that opens such a facility, we would like to say what fraction of the demand is served from the close facilities. To make our analysis simpler we will toss a properly biased coin to decide if using this facility counts as using a close facility. With this trick we, in a sense, split such a facility into a close and a distant part. Note that we may only do it for this part of the analysis, but not for the actual rounding algorithm from Section 3.5.

Applying the above described split of the undecided facility, we get that the total fractional opening of close facilities of client $j$ is exactly $\tau_j$, and the total fractional opening of both close and distant facilities is at least $\gamma \cdot \tau_j$. Therefore, Corollary 1 yields the following:

**Corollary 4.** The amount of close facilities used by client $j$ in a solution described in Section 3.5 is expected to be at least $(1 - \frac{1}{e}) \cdot \tau_j$.

**Corollary 5.** The amount of close and distant facilities used by client $j$ in a solution described in Section 3.5 is expected to be at least $(1 - \frac{1}{2e}) \cdot \tau_j$.

### 4.3 Calculation

We may now combine the pieces into the following algorithm ALG:

1. solve LP-relaxation of (1)-(5),
2. scale the fractional solution as described in Section 3.2,
3. create a family of clusters as described in Section 3.4,
4. round the fractional openings as described in Section 3.5,
5. connect each client $j$ to $r_j$ closest open facilities,
6. output the solution as $(\bar{x}, \bar{y})$.

**Theorem 5.** ALG is an 1.724-approximation algorithm for the Fault-Tolerant Facility Location problem.

**Proof.** First observe, that the solution produced by ALG is trivially feasible to the original problem (1)-(5), as we simply choose different $r_j$ facilities for client $j$ in step 5. What is less trivial is that all the $r_j$ facilities used by $j$ are within a certain small distance. Let us now bound the expected connection cost of the obtained solution.

For each client $j \in C$ we get $r_j - \tau_j$ facilities opened in Step 2. As we already argued in Section 3.2, we may afford to connect $j$ to these facilities and pay the connection cost from the difference between $\sum_i c_{ij} \bar{x}_{ij}$ and $\sum_i c_{ij} \tau_{ij}$. We will now argue, that client $j$ may connect to the remaining $\tau_j$ with the expected connection cost bounded by $\sum_i c_{ij} \bar{x}_{ij}$.

For a special client $j \in (C \setminus C')$ we have $\tau_j = 1$ and already in Step 2 one special facility at distance $d_{j}^{(max)}$ from $j$ is opened. We cannot blindly connect $j$ to this facility, since $d_{j}^{(max)}$ may potentially be bigger then $\gamma \cdot d_j$. What we do instead, is that we first look at close facilities of $j$ that, as a result of the rounding
in Step 4, with a certain probability, give one open facility at a small distance. By Corollary 4 this probability is at least $1 - 1/e$. By Theorem 4, the expected connection cost to this open facility is at most $d_j^{(c)}$. Only if no close facility is open, we use the distant facility, which results in a low enough expectation of the connection cost of client $j$.

In the remaining, we only look at non-special clients $j \in C'$. By Corollary 4 $j$ may expect to use $(1 - 1/e) \cdot \mathbf{\tau}_j$ of its close facilities, which are at average distance $d_j^{(c)}$. By Theorem 4, the expected connection cost this incurs is bounded by $((1 - 1/e) \cdot \mathbf{\tau}_j) \cdot d_j^{(c)}$. Client $j$ may also expect to use some of its distant facilities to satisfy a fraction of its demand at the average distance of $d_j^{(d)}$. By Corollary 5, $j$ may expect to get at least $(1 - 1/e) \cdot \mathbf{\tau}_j$ open facilities from both these groups altogether. Hence $j$ may expect at least $((1 - 1/e) - (1 - 1/e)) \cdot \mathbf{\tau}_j$ more open facilities at average distance at most $d_j^{(d)}$. Again, by Theorem 4, the expected connection cost this incurs is bounded by $((1 - 1/e) - (1 - 1/e)) \cdot \mathbf{\tau}_j \cdot d_j^{(d)}$. All the remaining facilities client $j$ gets deterministically within the distance of at most $3 \cdot d_j^{(max)}$, which is possible by the properties of the rounding procedure described in Section 3.5, see Corollary 3.

Concluding, the expected connection cost of $j$ may be bounded by

$$
((1 - 1/e) \cdot \mathbf{\tau}_j) \cdot d_j^{(c)} + ((1 - 1/e) - (1 - 1/e)) \cdot \mathbf{\tau}_j \cdot d_j^{(d)} + \left(\frac{1}{e^\gamma} \cdot \mathbf{\tau}_j \right) \cdot (3 \cdot d_j^{(max)})
$$

$$
\leq \mathbf{\tau}_j \cdot \left( (1 - 1/e) \cdot d_j^{(c)} + ((1 - \frac{1}{e^\gamma}) - (1 - 1/e)) \cdot d_j^{(d)} + \frac{1}{e^\gamma} \cdot 3d_j^{(d)} \right)
$$

$$
= \mathbf{\tau}_j \cdot \left( (1 - 1/e) \cdot d_j^{(c)} + ((1 + \frac{2}{e^\gamma}) - (1 - 1/e)) \cdot d_j^{(d)} \right)
$$

$$
\leq \mathbf{\tau}_j \cdot \left( (1 - 1/e) \cdot (1 - R_j) \cdot d_j + ((1 + \frac{2}{e^\gamma}) - (1 - 1/e)) \cdot (1 + \frac{R_j}{\gamma - 1}) \cdot d_j \right)
$$

$$
= \mathbf{\tau}_j \cdot d_j \cdot \left( (1 - 1/e) \cdot (1 - R_j) + \left( \frac{2}{e^\gamma} + 1/e \right) \cdot (1 + \frac{R_j}{\gamma - 1}) \right)
$$

$$
= \mathbf{\tau}_j \cdot d_j \cdot \left( (1 - 1/e) + \left( \frac{2}{e^\gamma} + 1/e \right) + R_j \cdot \left( \frac{2}{e^\gamma} + 1/e \right) \cdot \frac{1}{\gamma - 1} - (1 - 1/e) \right)
$$

$$
= \mathbf{\tau}_j \cdot d_j \cdot \left( 1 + \frac{2}{e^\gamma} + R_j \cdot \left( \frac{\frac{2}{e^\gamma} + 1/e}{\gamma - 1} - (1 - 1/e) \right) \right),
$$

where the second inequality follows from Lemma 1 and the definition of $R_j$.

---

5 To be more precise, Corollary 4 gives only a lower bound on the expected amount of facilities in this case. However, we may correct for it by counting only that many facilities in expectation. Suppose that the real expectation of the amount of opened facilities is $E > (1 - 1/e) \cdot \mathbf{\tau}_j$, then we may count each opened facility with probability $(1 - 1/e) \cdot \mathbf{\tau}_j / E$ for this case, and with the remaining probability to the next case with a higher expected distance.
Observe that for $1 < \gamma < 2$, we have \((\frac{2}{e\gamma} + 1/e) - (1 - 1/e) > 0\). Recall that by definition, $R_j \leq 1$; so, $R_j = 1$ is the worst case for our estimate, and therefore
\[
\bar{\pi}_j \cdot d_j \left( 1 + \frac{2}{e\gamma} + R_j \cdot \left( \frac{2}{e\gamma} + 1/e - (1 - 1/e) \right) \right) \leq \bar{\pi}_j \cdot d_j \cdot \left( 1/e + \frac{2}{e\gamma} \right)(1 + \frac{1}{\gamma - 1}).
\]

Recall that $\bar{\pi}$ incurs for each client $j$ a connection cost $\sum_{i \in F} c_{ij} \bar{x}_{ij} \geq \gamma \cdot \bar{\pi}_j \cdot d_j$.

We fix $\gamma = \gamma_0$, such that $\gamma_0 = (1/e + \frac{2}{e\gamma})(1 + \frac{1}{\gamma - 1}) \leq 1.724$.

To conclude, the expected connection cost of $j$ to facilities opened during the rounding procedure is at most the fractional connection cost of $\bar{\pi}$. The total connection cost is, therefore at most the connection cost of $\hat{x}$, which is at most $\gamma$ times the connection cost of $x^*$.

By property (P1) of dependent rounding, every single facility $i$ is opened with the probability $\hat{y}_i$, which is at most $\gamma$ times $y_i^*$. Therefore, the total expected cost of the solution produced by ALG is at most $\gamma \approx 1.724$ times the cost of the fractional optimal solution $(x^*, y^*)$. \hfill \square

5 Concluding remarks

We have presented improved approximation algorithms for the metric Fault-Tolerant Uncapacitated Facility Location problem. The main technical innovation is the usage and analysis of dependent rounding in this context. We believe that variants of dependent rounding will also be fruitful in other location problems. Finally, we conjecture that the approximation threshold for both UFL and FTFL is the value $1.46 \cdots$ suggested by [4]; it would be very interesting to prove or refute this.

References


Appendix

A  The rounding approach of [9]

The dependent-rounding approach of [9] to round a given $y = (y_1, y_2, \ldots, y_N) \in [0,1]^N$ is as follows. Suppose the current version of the rounded vector is $v = (v_1, v_2, \ldots, v_N) \in [0,1]^N$; $v$ is initially $y$. When we describe the random choice made in a step below, this choice is made independent of all such choices made thus far. If all the $v_i$ lie in $\{0, 1\}$, we are done, so let us assume that there is at least one $v_i \in (0, 1)$. The first (simple) case is that there is exactly one $v_i$ that lies in $(0, 1)$; we round $v_i$ in the natural way – to 1 with probability $v_i$, and to 0 with complementary probability of $1 - v_i$; letting $V_i$ denote the rounded version of $v_i$, we note that 

$$E[V_i] = v_i. \quad (6)$$

This simple step is called a Type I iteration, and it completes the rounding process. The remaining case is that of a Type II iteration: there are at least two components of $v$ that lie in $(0, 1)$. In this case we choose two such components $v_i$ and $v_j$ with $i \neq j$, arbitrarily. Let $\epsilon$ and $\delta$ be the positive constants such that:

(i) $v_i + \epsilon$ and $v_j - \epsilon$ lie in $[0,1]$, with at least one of these two quantities lying in $\{0, 1\}$, and (ii) $v_i - \delta$ and $v_j + \delta$ lie in $[0,1]$, with at least one of these two quantities lying in $\{0, 1\}$. It is easily seen that such strictly-positive $\epsilon$ and $\delta$ exist and can be easily computed. We then update $(v_i, v_j)$ to a random pair $(V_i, V_j)$ as follows:

- with probability $\delta/(\epsilon + \delta)$, set $(V_i, V_j) := (v_i + \epsilon, v_j - \epsilon)$;
- with the complementary probability of $\epsilon/(\epsilon + \delta)$, set $(V_i, V_j) := (v_i - \delta, v_j + \delta)$.

The main properties of the above that we will need are:

$$\Pr[V_i + V_j = v_i + v_j] = 1; \quad (7)$$

$$E[V_i] = v_i \quad \text{and} \quad E[V_j] = v_j; \quad (8)$$

$$E[V_i V_j] \leq v_i v_j. \quad (9)$$

We iterate the above iteration until all we get a rounded vector. Since each iteration rounds at least one further variables, we need at most $N$ iterations.

B  Proofs of some results

Proof. (For Theorem 1) Recall that in the dependent-rounding approach, we begin with the vector $v^{(0)} = (y_1, y_2, \ldots, y_N)$; in each iteration $t \geq 1$, we start
with a vector \( v^{(t-1)} \) and probabilistically modify at most two of its entries, to produce the vector \( v^{(t)} \). We define a potential function \( \Phi(v^{(t)}) \), which is a random variable that is fully determined by \( v^{(t)} \), i.e., determined by the random choices made in iterations 1, 2, \ldots, \( t \):

\[
\Phi(v^{(t)}) = \sum_{\ell=0}^s \lambda_\ell \sum_{A \subseteq S: |A| = \ell} \left( \prod_{a \in A} v_a^{(t)} \cdot \prod_{b \in (S-A)} (1 - v_b^{(t)}) \right). \tag{10}
\]

Recall that dependent rounding terminates in some \( m \leq N \) iterations. A moment’s reflection shows that:

\[
\Phi(v^{(0)}) = \mathbb{E}[g_{\lambda,S}(R(y))]; \quad \mathbb{E}[\Phi(v^{(m)})] = \mathbb{E}[g_{\lambda,S}(\hat{y})]. \tag{11}
\]

Our main inequality will be the following:

\[
\forall t \in [m], \quad \mathbb{E}[\Phi(v^{(t)})] \geq \mathbb{E}[\Phi(v^{(t-1)})]. \tag{12}
\]

This implies that

\[
\mathbb{E}[\Phi(v^{(m)})] \geq \mathbb{E}[\Phi(v^{(0)})] = \Phi(v^{(0)}),
\]

which, in conjunction with (11) will complete our proof.

Fix any \( t \in [m] \), and fix any choice for the vector \( v^{(t-1)} \) that happens with positive probability. Conditional on this choice, we will next prove that

\[
\mathbb{E}[\Phi(v^{(t)})] \geq \Phi(v^{(t-1)}); \tag{13}
\]

note that the expectation in the l.h.s. is only w.r.t. the random choice made in iteration \( t \), since \( v^{(t-1)} \) is now fixed. Once we have (13), (12) follows from Bayes’ Theorem by a routine conditioning on the value of \( v^{(t-1)} \).

Let us show (13). We first dispose of two simple cases. Suppose iteration \( t \) is a Type I iteration, and that \( v_i^{(t-1)} \) is the only component of \( v^{(t-1)} \) that lies in \((0, 1)\). Since \( \Phi(v^{(t)}) \) is a linear function of the random variable \( v_i^{(t)} \), (13) holds with equality, by (6). A similar argument holds if iteration \( t \) is a Type II iteration in which the components \( v_i^{(t-1)} \) and \( v_j^{(t-1)} \) are probabilistically altered in this iteration, if at most one of \( i \) and \( j \) lies in \( S \).

So suppose iteration \( t \) is a Type II iteration, and that both \( i \) and \( j \) lie in \( S \) (again, \( v_i^{(t-1)} \) and \( v_j^{(t-1)} \) are the components altered in this iteration). Let \( v_i = v_i^{(t-1)} \) and \( v_j = v_j^{(t-1)} \) for notational simplicity, and let \( V_i \) and \( V_j \) denote their respective altered values. Note that there are deterministic reals \( u_0, u_1, u_2, u_3 \) which depend only on the components of \( v^{(t-1)} \) other than \( v_i^{(t-1)} \) and \( v_j^{(t-1)} \), such that

\[
\Phi(v^{(t-1)}) = u_0 + u_1 v_i + u_2 v_j + u_3 v_i v_j;
\]

\[
\Phi(v^{(t)}) = u_0 + u_1 V_i + u_2 V_j + u_3 V_i V_j.
\]

Therefore, in order to prove our desired bound (13), we have from (8) and (9) that is it is sufficient to show

\[
u_3 \leq 0, \tag{14}
\]
which we proceed to do next.

Let us analyze (10), the definition of \( \Phi \), to calculate \( u_3 \). Let, for \( 0 \leq \ell \leq s \), \( \alpha_\ell \) denote the contribution of the term

\[
\lambda_\ell \cdot \sum_{A \subseteq S: |A|=\ell} \left( \prod_{a \in A} v^{(t)}_a \cdot \prod_{b \in (S-A)} (1 - v^{(t)}_b) \right)
\]

(15)

to \( u_3 \); note that

\[
u_3 = \sum_{\ell=0}^s \alpha_\ell.
\]

In order to compute the values \( \alpha_\ell \), it is convenient to define certain quantities \( \beta_r \), which we do next. Define \( T = S - \{i, j\} \), and note that \( |T| = s - 2 \). For \( 0 \leq r \leq s - 2 \), define

\[
\beta_r = \sum_{B \subseteq T: |B|=r} \left( \prod_{p \in B} v^{(t)}_p \cdot \prod_{q \in (T-B)} (1 - v^{(t)}_q) \right).
\]

Now, as a warmup, note that \( \alpha_0 = \beta_0 \) and \( \alpha_s = \beta_{s-2} \). Let us next compute \( \alpha_\ell \) for \( 1 \leq \ell \leq s - 1 \). The sum (15) can contribute a \( v^{(t)}_i \cdot v^{(t)}_j \) term in three ways:

- by taking both \( i \) and \( j \) in the set \( A \) in (15) – this is possible only if \( \ell \geq 2 \) – with a coefficient of \( \lambda_\ell \beta_{\ell-2} \) for the \( v^{(t)}_i \cdot v^{(t)}_j \) term;
- by taking both \( i \) and \( j \) in the set \( S - A \) in (15) – this is possible only if \( \ell \leq s - 2 \) – with a coefficient of \( \lambda_\ell \beta_{\ell} \) for the \( v^{(t)}_i \cdot v^{(t)}_j \) term; and
- by taking exactly one of \( i \) and \( j \) in the set \( A \) – this is possible for any \( \ell \in [s-1] \) – with a coefficient of \( -2\lambda_\ell \beta_{\ell-1} \) for the \( v^{(t)}_i \cdot v^{(t)}_j \) term (with the factor of 2 arising from the choice of \( i \) or \( j \) to put in \( A \)).

Rearranging the above three items, the contribution of \( \beta_r \) to \( u_3 \), for \( 0 \leq r \leq s - 2 \), is \( \lambda_r - 2\lambda_{r+1} + \lambda_{r+2} \). That is,

\[
u_3 = \sum_{r=0}^{s-2} (\lambda_r - 2\lambda_{r+1} + \lambda_{r+2}) \cdot \beta_r.
\]

Thus, the hypothesis of the theorem and the fact that all the values \( \beta_r \) are non-negative, together show that \( u_3 \leq 0 \) as required by (14).

**Proof.** (For Theorem 2) Let \( s = |S| \). The theorem directly follows from property (P1) if either \( s \leq 1 \) or \( k \geq s \), so we may assume that \( s \geq 2 \) and that \( k \leq s - 1 \). Of course, we may also assume that \( k \geq 1 \). Note that for any \( x \in \{0, 1\}^N \),

\[
\min\{k, \text{Sum}_S(x)\} = (\sum_{\ell \leq k} \ell \cdot I(\text{Sum}_S(x) = \ell)) + (\sum_{\ell > k} \ell \cdot I(\text{Sum}_S(x) = \ell))
\]

\[
= g_{\lambda,S}(x),
\]

15
where
\[ \lambda = (0, 1, 2, \ldots, k, k, \ldots, k). \]
It is easy to verify that for all \(0 \leq r \leq s - 2\), \(\lambda_r - 2\lambda_{r+1} + \lambda_{r+2} \leq 0\). (Recall that \(1 \leq k \leq s - 1\). The sum in the l.h.s. is zero for all \(r \neq k - 1\), and equals \(-1\) for \(r = k - 1\). Thus we have the theorem, from Theorem 1. \(\Box\)

**Proof.** (For Theorem 3) We prove the theorem by induction on \(N + k\). The base case is where \(k = 1\), in which case

\[ E[\min\{1, \text{Sum}_S(\mathcal{R}(y))\}] = \Pr[\exists i \in S : \mathcal{R}(y) = 1] = 1 - \prod_{i \in S} (1 - y_i) \geq 1 - \prod_{i \in S} \exp(-y_i) = 1 - \exp(-\text{Sum}_S(y)), \]
completing the proof of the base case.

**Remark.** Note that if \(k \geq |S|\), then the theorem follows easily:

\[ E[\min\{k, \text{Sum}_S(\mathcal{R}(y))\}] = E[\text{Sum}_S(\mathcal{R}(y))] = \text{Sum}_S(y) = k \cdot (\text{Sum}_S(y)/k) \geq k \cdot (1 - \exp(-\text{Sum}_S(y)/k)), \]

since
\[ \exp(-t) \geq 1 - t \text{ for all real } t. \tag{16} \]

For the inductive case, we assume \(k \geq 2\), and hence also that \(N \geq 2\) (the case \(k \geq N\) is easy as shown by the remark above). For notational convenience, we may also assume without loss of generality that \(S = [N]\), since the entries of \(\mathcal{R}(y)\) have been rounded independently. Let \(z\) denote the \((N-1)\)-dimensional vector \((y_2, y_3, \ldots, y_N)\), \(p = y_1\), and \(t = \sum_i y_i\); note that \(t \geq p\). A simple conditioning on the rounding of \(y_1\) shows that \(E[\min\{k, \text{Sum}_S(\mathcal{R}(y))\}]\) equals

\[ p \cdot \min\{k, 1 + \text{Sum}_S(\mathcal{R}(z))\} + (1 - p) \cdot \min\{k, \text{Sum}_S(\mathcal{R}(z))\}; \]
i.e., it equals

\[ p \cdot (1 + \min\{k - 1, \text{Sum}_S(\mathcal{R}(z))\}) + (1 - p) \cdot \min\{k, \text{Sum}_S(\mathcal{R}(z))\}. \tag{17} \]

Applying the induction hypothesis to both terms in (17) and simplifying, we see that our desired quantity \(E[\min\{k, \text{Sum}_S(\mathcal{R}(y))\}]\) is at least

\[ p \cdot (1 + (k - 1) \cdot (1 - \exp(-(t - p)/k))) + (1 - p) \cdot k \cdot (1 - \exp(-(t - p)/(k - 1))), \]
which in turn equals

\[ k - p(k - 1) \cdot \exp(-(t - p)/k)) - k(1 - p) \cdot \exp(-(t - p)/(k - 1)). \]
Thus, in order to complete the induction proof, it suffices to show that
\[ p(k - 1) \cdot \exp(-(t - p)/k)) + k(1 - p) \cdot \exp(-(t - p)/(k - 1)) \leq k \cdot \exp(-t/k). \]

Multiplying both sides by \( \exp(t/k) \), this reduces to proving
\[ p(k - 1) \cdot \exp(p/k) + k(1 - p) \cdot \exp(t/k - t/(k - 1)) \cdot \exp(p/(k - 1)) \leq k. \quad (18) \]

Note that \( t \geq p \) and that the l.h.s. of (18) is a decreasing function of \( t \) while the r.h.s. is independent of \( t \); thus, it suffices to prove (18) for the case where \( t = p \).

That is, we aim to prove
\[ p(k - 1) \cdot \exp(p/k) + k(1 - p) \cdot \exp(p/k) \leq k, \]
i.e., that \( \exp(-p/k) \geq \frac{p(k - 1) + k(1 - p)}{k} = 1 - p/k \), which is true by (16). This completes the induction proof. \( \square \)