Approximating the Reliable Resource Allocation Problem Using Inverse Dual Fitting

Kewen Liao[1] **Hong Shen**[1]

[1] School of Computer Science
The University of Adelaide, SA 5005, Australia
Email: {kewen, hong}@cs.adelaide.edu.au

Abstract

We initiate the study of the Reliable Resource Allocation (RRA) problem. In this problem, we are given a set of sites equipped with an unbounded number of facilities as resources. Each facility has an opening cost and an estimated reliability. There is also a set of clients to be allocated to facilities with corresponding connection costs. Each client has a reliability requirement (RR) for accessing resources. The objective is to open a subset of facilities from sites to satisfy all clients' RRs at a minimum total cost. The Unconstrained Fault-Tolerant Resource Allocation (UFTRA) problem studied in (Liao & Shen 2011) is a special case of RRA.

In this paper, we present two equivalent primal-dual algorithms for the RRA problem, where the second one is an acceleration of the first and runs in quasi-linear time. If all clients have the same RR above the threshold that a single facility can provide, our analysis of the algorithm yields an approximation factor of $2+2\sqrt{2}$ and later a reduced ratio of 3.722 using a factor revealing program. The analysis further elaborates and generalizes the generic inverse dual fitting technique introduced in (Xu & Shen 2009). As a by-product, we also formalize this technique for the classical minimum set cover problem.

Keywords: Reliable Resource Allocation, Approximation Algorithms, Time Complexity, Inverse Dual Fitting Technique.

1 Introduction

Fault-tolerant design is essential in many industrial applications and network optimization problems like resource allocation. In the Unconstrained Fault-Tolerant Resource Allocation (UFTRA) problem studied in (Liao & Shen 2011), we are given a set of sites $\mathcal{F}$ and a set of clients $\mathcal{C}$. At each site $i \in \mathcal{F}$, an unbounded number of facilities with f_i as costs can be opened to serve as resources. There is also a connection cost c_{ij} between each client $j \in \mathcal{C}$ and all facilities of i. The objective is to optimally allocate a certain number of facilities from each i to serve every client j with $r_j \in \mathcal{R}$ requests while minimizing the sum of facility opening and client connection costs. This problem can be formulated by the following integer linear program (ILP) with variable y_i denoting in the solution the number of facilities to open at site i, and x_{ij} the number of connections between site i and client j.

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}} c_{ij}x_{ij} \\ \text{subject to} & \forall j \in \mathcal{C} : \sum_{i\in\mathcal{F}} x_{ij} \geq r_j \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : y_i - x_{ij} \geq 0 \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : x_{ij} \in \mathbb{Z}^+ \\ & \forall i \in \mathcal{F} : y_i \in \mathbb{Z}^+ \end{array} \quad (1)$$

UFTRA forms a relaxation of the Fault-Tolerant Facility Location (FTFL) problem (Jain & Vazirani 2000) by allowing domains of y_i's and x_{ij}'s to be non-negative rather than 0-1 integers. In addition, by setting $\forall j \in \mathcal{C} : r_j = 1$ these problems become the classical Uncapacitated Facility Location (UFL) problem. Both FTFL and UFTRA measure the fault-tolerance only by the number of connections each client makes. We observe this measurement is not sufficient in many applications like the VLSI design, and the resource allocation we considered here. For instance, a client j in UFTRA may connect to r_j facilities that are all susceptible to failure (with very low reliability) and therefore j is still very likely to encounter faults. This observation motivates us to study an alternative model called Reliable Resource Allocation (RRA) that provides more solid fault-tolerance. In particular, RRA assumes all facilities of a site possess an estimated probability (between 0 and 1) of being reliable (with no fault). Also, the fault-tolerance level of the clients is ensured by their fractional reliability requirement (RR) values to be provided by facilities. In this paper, we only consider the case where client-facility connection costs c_{ij}'s form a metric, i.e. they are non-negative, symmetric and satisfy triangle inequality. This is because even the non-metric UFL can be easily reduced from the set cover problem (Feige 1998) that is hard to approximate better than $O(\log n)$ unless $NP \subseteq DTIME\left[n^{O(\log\log n)}\right]$.

Related Work: Two important techniques in designing good approximation algorithms for facility location problems are primal-dual and LP-rounding. For the non-uniform FTFL, the existing primal-dual method in (Jain & Vazirani 2000) yields a non-constant factor. Constant results were only for the special case where r_j's are equal. In particular, Jain et al. (Jain et al. 2003) showed their MMS and JMS algorithms for UFL can be adapted to the special case of FTFL while preserving approximation

This work was partially supported by Australian Research Council Discovery Project grant #DP0985063.

ratios of 1.861 and 1.61 respectively. Swamy and Shmoys (Swamy & Shmoys 2008) improved the result to 1.52 with cost scaling and greedy augmentation techniques. On the other hand, LP-rounding approach have met more successes in dealing with the general case of FTFL. Guha et al. (Guha et al. 2001, 2003) obtained the first constant factor algorithm with ratio 2.408. Later, this was improved to 2.076 by Swamy and Shmoys (Swamy & Shmoys 2008) with more sophisticated rounding techniques. Recently, Byrka et al. (Byrka et al. 2010) applied the dependent rounding technique and achieved the current best ratio of 1.7245.

UFTRA was first introduced by Xu and Shen (Xu & Shen 2009). They used a phase-greedy algorithm to obtain approximation ratio of 1.861, but their algorithm runs in pseudo-polynomial time. The ratio was later improved to 1.5186 by Liao and Shen (Liao & Shen 2011) using a star-greedy algorithm. This problem was also studied by Yan and Chrobak (Yan & Chrobak 2011) who gave a rounding algorithm that achieved 3.16-approximation. However, none of these studies provide efficient strongly polynomial time algorithms and consider the reliability issue.

In contrast to FTFL and UFTRA, UFL has been studied extensively with mature results. For the primal-dual methods, JV (Jain & Vazirani 2001), MMS (Mahdian et al. 2001) and JMS (Jain et al. 2002) algorithms achieved approximation ratios of 3, 1.861 and 1.61 respectively. Charikar and Guha (Charikar & Guha 2005) improved the result of JV algorithm to 1.853 and Mahdian et al. (Mahdian et al. 2006) improved that of JMS algorithm to 1.52, both using the standard cost scaling and greedy augmentation techniques. For the rounding approaches, Shmoys et al. (Shmoys et al. 1997) first gave a ratio of 3.16 based on the filtering and rounding technique of Lin and Vitter (Lin & Vitter 1992). Later, Guha and Khuller (Guha & Khuller April 1999) improved the factor to 2.41 by combing Shmoys's result with a simple greedy phase. Chudak and Shmoys (Chudak & Shmoys 2003) again presented an improvement with ratio of 1.736 using clustered randomized rounding. Sviridenko (Sviridenko 2002) combined this solution with the pipage rounding to obtain 1.582-approximation. Afterwards, Byrka (Byrka 2007) achieved the ratio of 1.5 by combining rounding with a bi-factor result of JMS algorithm. Based on his work, recently Li's more careful analysis in (Li 2011) obtained the current best ratio of 1.488. For the lower bound, Guha and Khuller (Guha & Khuller April 1999) proved it is 1.463 for UFL. This holds unless $P = NP$ (Chudak & Williamson 2005). The ratio also bounds FTFL and UFTRA since UFL is a special case of them.

Our Contributions: We initiate the study of the RRA problem towards provision of more robust fault-tolerance in the resource allocation paradigm. To the best of our knowledge, this is the first theory work that takes into account the quality of service (QoS) requirement for resource allocation. Further, our ideas have potential to influence some classical facility location problems. For the RRA problem, we present two equivalent primal-dual algorithms inspired by the MMS algorithm (Mahdian et al. 2001) for UFL. In particular, the second algorithm is a significant improvement of the first one in runtime that is quasi-linear, which is comparable to the current best efficient algorithm for UFL (Mahdian et al. 2006). Since UFTRA is a special case of RRA, this algorithm also implies the first strongly polynomial time algorithm for UFTRA with uniform connection requirements. For the approximation ratio analysis, RRA is a harder problem than UFTRA and the main difficulty we overcome is to deal with the fractional reliabilities. We apply the inverse dual fitting technique introduced in (Xu & Shen 2009) as the central idea for analyzing the algorithm. Our analysis further elaborates and generalizes this generic technique, which naturally yields approximation factors of $2 + 2\sqrt{2}$ and 3.722 for RRA, where every client is provided with the same RR that is at least the highest reliability among all facilities. Apparently, this provided minimum threshold ensures the clients' lowest fault tolerance level. For the problem without the threshold, which is theoretically valid, we leave the approximation bound open. In the closing discussions, we also formalize the inverse dual fitting technique for analyzing the minimum set cover problem.

2 The RRA Problem

In the RRA problem, we are given a set of sites $\mathcal{F}$ and a set of clients $\mathcal{C}$, where $|\mathcal{F}| = n_f$ and $|\mathcal{C}| = n_c$. Let $n = n_f + n_c$, $m = n^2$ for convenience of runtime analysis, Each site $i \in \mathcal{F}$ has an unbounded number of facilities with f_i as the cost and p_i ($0 \le p_i \le 1$) as the reliability. Each client $j \in \mathcal{C}$ has a RR r_j that must be satisfied by facilities from sites in $\mathcal{F}$. There is also a connection cost c_{ij} between every client-facility pair. The objective is to optimally open a certain number of facilities in every site to satisfy clients' RRs while minimizing the total cost. The problem is formulated into the ILP below in which y_i denotes the number of facilities to open at site i, and x_{ij} the total number of connections/assignments between i and j.

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}} c_{ij}x_{ij} \\ \text{subject to} & \forall j \in \mathcal{C} : \sum_{i\in\mathcal{F}} p_i x_{ij} \ge r_j \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : y_i - x_{ij} \ge 0 \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : x_{ij} \in \mathbb{Z}^+ \\ & \forall i \in \mathcal{F} : y_i \in \mathbb{Z}^+ \end{array} \tag{2}$$

Its LP-relaxation and dual LP are the following:

$$\begin{array}{ll} \text{minimize} & \sum_{i\in\mathcal{F}} f_i y_i + \sum_{i\in\mathcal{F}}\sum_{j\in\mathcal{C}} c_{ij}x_{ij} \\ \text{subject to} & \forall j \in \mathcal{C} : \sum_{i\in\mathcal{F}} p_i x_{ij} \ge r_j \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : y_i - x_{ij} \ge 0 \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : x_{ij} \ge 0 \\ & \forall i \in \mathcal{F} : y_i \ge 0 \end{array} \tag{3}$$

$$\begin{array}{ll} \text{maximize} & \sum_{j\in\mathcal{C}} r_j \alpha_j \\ \text{subject to} & \forall i \in \mathcal{F} : \sum_{j\in\mathcal{C}} \beta_{ij} \le f_i \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : \alpha_j p_i - \beta_{ij} \le c_{ij} \\ & \forall j \in \mathcal{C} : \alpha_j \ge 0 \\ & \forall i \in \mathcal{F}, j \in \mathcal{C} : \beta_{ij} \ge 0 \end{array} \tag{4}$$

Compare this formulation to UFTRA, the key difference is the introduction of p_i's and r_j's that are fractional. In addition, if $\forall i \in \mathcal{F} : p_i = 1$ and $\forall j \in \mathcal{C} : r_j$ is a positive integer, RRA becomes UFTRA. Nevertheless, none of the previous algorithms (Xu & Shen 2009, Yan & Chrobak 2011, Liao & Shen 2011) for UFTRA are both efficient and easily adaptable.

2.1 The Algorithms

We present two primal-dual algorithms that incrementally build primal solutions y_i's and x_{ij}'s in

LP (2) at different rates. Initially, they are all 0s. Nonetheless, both of them terminate when all clients' RRs are satisfied, i.e. the set $\mathcal{U} = \left\{j \in \mathcal{C} \mid \sum_{i\in\mathcal{F}} p_i x_{ij} < r_j\right\}$ is empty. Our first algorithm inspired by the MMS algorithm (Mahdian et al. 2001) for UFL naively runs in pseudo-polynomial time. Without loss of generality, assuming in the solution a client j makes total d_j connections in the order from 1 to d_j and each connection is associated with a *virtual port* of j denoted by j^{vp} $(1 \le vp \le d_j)$. The algorithm can then associate every client j with d_j dual values $\alpha_j^1, \ldots, \alpha_j^{d_j}$. Denoting $\phi\left(j^{vp}\right)$ as the facility/site client j's vpth port connected with, we can now interpret the termination condition of the algorithm again as $\forall j \in \mathcal{C} : \sum_{vp=1}^{d_j} p_{\phi(j^{vp})} \ge r_j$. Note that although unlike UFTRA, the required number of connections d_j is not pre-known for each client j in RRA, the above steps are necessary since it establishes a relationship between the fractional r_j's and integral d_j's for the algorithm's analysis. In addition, a virtual port in this setting can only establish one connection with a facility of any site. Throughout the paper we do not identify facilities within a site individually (like the function ϕ we denote) because they are identical and this will not affect the solution and the analysis of the algorithm.

Algorithm 1 Primal-Dual Algorithm

Input: $\forall i, j : f_i, p_i, c_{ij}, r_j$.
Output: $\forall i, j : y_i, x_{ij}$.
Initialization: Set $\mathcal{U} = \mathcal{C}$, $\forall i, j : d_j = 1, y_i = 0, x_{ij} = 0$.

While $\mathcal{U} \neq \emptyset$, increase time t uniformly and execute the events below:

- Event 1: $\exists i \in \mathcal{F}, j \in \mathcal{U}$ s.t. $p_i t = c_{ij}$ and $x_{ij} < y_i$.
  Action 1-a: Set $x_{ij} = x_{ij} + 1$, $\alpha_j^{d_j} = t$ and $\phi\left(j^{d_j}\right) = i$;
  Action 1-b: If $\sum_{i\in\mathcal{F}} p_i x_{ij} \ge r_j$ then set $\mathcal{U} = \mathcal{U}\backslash\{j\}$, else set $d_j = d_j + 1$.

- Event 2: $\exists i \in \mathcal{F}$ s.t. $\sum_{j\in\mathcal{U}} \max\left(0, p_i t - c_{ij}\right) = f_i$.
  Action 2-a: Set $y_i = y_i + 1$ and $\mathcal{U}_i = \{j \in \mathcal{U} \mid p_i t \ge c_{ij}\}$; $\forall j \in \mathcal{U}_i$: do Action 1-a;
  Action 2-b: $\forall j \in \mathcal{U}_i$: do Action 1-b.

Remark 1. *For the convenience of runtime analysis, sequential actions of events are separated as above. If more than one event happen at the same time, the algorithm processes all of them in an arbitrary order. Also, the events themselves may repeatedly happen at any time t because unconstrained number of facilities at a site are allowed to open.*

Remark 2. *If we adopt the approach of the JMS algorithm (Jain et al. 2002) for UFL that also considers optimizing clients' total connection costs, it may render a feasible solution to RRA infeasible due to the clients' reliability constraints.*

Moreover, the primal-dual algorithm shown above is associated with a global time t that increases monotonically from 0. In this event-driven like algorithm, we use variable d_j to keep track of the ports of client j that connect in order, and the value of $\alpha_j^{d_j}$ is assigned the time at which j's port d_j establishes a connection to $\phi\left(j^{d_j}\right)$. At any t, we define the *payment* of a client $j \in \mathcal{U}$ to a site $i \in \mathcal{F}$ as $p_i t$ and the *contribution* as $\max\left(0, p_i t - c_{ij}\right)$. As t increases, we let the action that j connects to a facility of i (solution x_{ij} increased by one) happens under two events: 1) j fully pays the connection cost of an already opened facility at i that it is not connected to (implying at this time $y_i > x_{ij}$); 2) the total contribution of clients in $\mathcal{U}$ to a closed facility at i fully pays its opening cost f_i (implying at this time a new facility at i will be opened) and $p_i t \ge c_{ij}$. Note that in the algorithm's ratio analysis (Section 2.2), we will associate values of dual variables α_j's and β_{ij}'s in LP (4) with values of $\alpha_j^{d_j}$'s and the contribution defined here.

Lemma 1. *The Primal-Dual Algorithm computes a feasible primal solution to RRA and its runtime complexity is* $O\left(n^2 \lceil \frac{\max_j r_j}{\min_i p_i} \rceil\right)$.

Proof. The feasibility of the solution is obvious since the output of the algorithm obeys the constraints and the variable domains of ILP (2). For runtime, we use two binary heaps (both sorted by time t) to store anticipated times of Event 1 and Event 2 respectively. For Event 1, t is computed as $\frac{c_{ij}}{p_i}$ according to the algorithm, whereas t is $\frac{f_i + \sum_{j\in\mathcal{U}_i} c_{ij}}{p_i \cdot |\mathcal{U}_i|}$ for Event 2. Therefore, detecting the next event (with smallest t) to process from two heaps takes time $O(1)$ and updating the heaps takes $O(\log m)$ in each iteration. Similar to the JV (Jain & Vazirani 2001) and MMS (Mahdian et al. 2001) algorithms for UFL, it actually takes $O(n_f \log m)$ to process every Action 1-b occurred, $O(1)$ for Action 1-a and $O(n_c)$ for Action 2-a. In addition, it is easy to see that Action 1-b is triggered totally n_c times, and Action 1-a and 2-a both at most $\sum_{j\in\mathcal{C}} d_j$ times. Since $\sum_{j\in\mathcal{C}} d_j \le n_c \max_{j\in\mathcal{C}} d_j \le n_c \lceil \frac{\max_j r_j}{\min_i p_i} \rceil$, the total time complexity is $O\left(n_c^2 \lceil \frac{\max_j r_j}{\min_i p_i} \rceil\right)$. □

The previous algorithm runs in pseudo-polynomial time that depends on both p_i's and r_j's. However through a more careful look at the algorithm, we are able to speed it up to strongly polynomial time. First of all, we can combine the repeated events into a single event by growing solution y_i's and x_{ij}'s at a faster rate, and thereby reducing the total number of events to process. This is because similar to UFTRA, RRA allows multiple connections between each client-site pair. Thus once a facility of a site is opened and connected with a group of clients' ports, according to the previous algorithm, additional facilities at this site will subsequently open and connect with this group of clients' other ports until one of these clients fulfills its RR. Similarly, once a client's port starts to connect to an open facility at a site, its other ports may connect to this site's other open facilities. Formally in Algorithm 2, let FR_j denote the already fulfilled reliability of client j and ToC the total number of connections to make after combining repeated events. The incremental rate of the solution can then be determined by the value of ToC. Secondly, it is not necessary to waste computation time to explicitly record $\alpha_j^{d_j}$'s and $\phi\left(j^{d_j}\right)$'s for totally $\sum_{j\in\mathcal{C}} d_j$ connections as in Algorithm 1, because they implicitly exist only for the algorithm's ratio analysis. Therefore by making these changes, the following algorithm in fact runs in quasi-linear time in terms of m as defined.

Algorithm 2 Accelerated Primal-Dual Algorithm

Input: $\forall i,\, j:\, f_i,\, p_i,\, c_{ij},\, r_j$.
Output: $\forall i,\, j:\, y_i,\, x_{ij}$.
Initialization: Set $\mathcal{U} = \mathcal{C},\ \forall i,\, j:\, y_i = 0,\, x_{ij} = 0,\, FR_j = 0$.

While $\mathcal{U} \neq \emptyset$, increase time t uniformly and execute the events below:

- Event 1: $\exists i \in \mathcal{F},\, j \in \mathcal{U}$ s.t. $p_i t = c_{ij}$ and $x_{ij} < y_i$.
  Action 1-a: Set $ToC = \min\left(y_i - x_{ij},\, \lceil \frac{r_j - FR_j}{p_i} \rceil\right)$;
  Action 1-b: Set $x_{ij} = x_{ij} + ToC$ and $FR_j = FR_j + p_i \cdot ToC$;
  Action 1-c: If $FR_j \geq r_j$ then set $\mathcal{U} = \mathcal{U} \backslash \{j\}$.

- Event 2: $\exists i \in \mathcal{F}$ s.t. $\sum_{j \in \mathcal{U}} \max(0,\, p_i t - c_{ij}) = f_i$.
  Action 2-a: Set $\mathcal{U}_i = \{j \in \mathcal{U} \,|\, p_i t \geq c_{ij}\}$, $ToC = \min_{j \in \mathcal{U}_i} \lceil \frac{r_j - FR_j}{p_i} \rceil$ and $y_i = y_i + ToC$; $\forall j \in \mathcal{U}_i$: do Action 1-b;
  Action 2-b: $\forall j \in \mathcal{U}_i$: do Action 1-c.

Remark 3. *If more than one event happen at the same time, process all of them in an arbitrary order.*

Lemma 2. *The Accelerated Primal-Dual Algorithm computes a feasible primal solution to RRA and its runtime complexity is* $\tilde{O}(m)$.

Proof. The primal solution is feasible because the algorithm is identical to Algorithm 1 in terms of the solution y_i's and x_{ij}'s produced. The difference is it combines multiple repeated events in order to reduce the total occurrences of the actions. Therefore for runtime, we are able to bound the total number of Event 2 and Action 2-a to n_c rather than $\sum_{j \in \mathcal{C}} d_j$, since as mentioned before once a facility of a site is opened, it will trigger at least one client's RR to be satisfied and there are n_c clients in total. In addition, the total number of Event 1 is at most n_c times of Event 2 because there will be maximum n_c Event 1 following each Event 2. Thus total number of Action 1-a and 1-b is bounded by n_c^2. Finally, same as the Algorithm 1 it takes $O(1)$ for Action 1-a and 1-b, $O(n_c)$ for Action 2-a and $O(n_f \log m)$ to process each of total n_c Action 1-c, the total time is therefore $O(m \log m)$. □

2.2 The Inverse Dual Fitting Analysis

We elaborate and generalize the inverse dual fitting technique introduced in (Xu & Shen 2009) for the algorithm's analysis. We observe this technique is more generic and powerful than the dual fitting technique in (Jain et al. 2003) especially for the multi-factor analysis. In the RRA problem, there are two types of costs, so first we have the following definition.

Definition 1. *An algorithm is bi-factor* $(\rho_f,\, \rho_c)$ *or single factor* $\max(\rho_f,\, \rho_c)$*-approximation for RRA, iff for every instance* $\mathcal{I}$ *of RRA and any feasible solution SOL (possibly fractional) of* $\mathcal{I}$ *with facility cost* F_{SOL} *and connection cost* C_{SOL}*, the total cost produced from the algorithm is at most* $\rho_f F_{SOL} + \rho_c C_{SOL}$ *(*$\rho_f,\, \rho_c$ *are both positive constants greater than or equal to one).*

Inverse dual fitting then considers the scaled instance of the problem and shows that dual solution of the original instance is feasible to the scaled instance. Also, it is obvious that the original instance's primal solution is feasible to the scaled instance. As for the RRA problem, we can construct a new instance $\mathcal{I}'$ by scaling any original instance $\mathcal{I}$'s facility cost by ρ_f and connection cost by ρ_c ($\rho_f \geq 1$ and $\rho_c \geq 1$). The scaled problem will then have the following formulation.

$$\begin{array}{ll} \text{minimize} & \sum_{i \in \mathcal{F}} \rho_f f_i y'_i + \sum_{i \in \mathcal{F}} \sum_{j \in \mathcal{C}} \rho_c c_{ij} x'_{ij} \\ \text{subject to} & \forall j \in \mathcal{C}:\, \sum_{i \in \mathcal{F}} p_i x'_{ij} \geq r_j \\ & \forall i \in \mathcal{F}, j \in \mathcal{C}:\, y'_i - x'_{ij} \geq 0 \\ & \forall i \in \mathcal{F}, j \in \mathcal{C}:\, x'_{ij} \geq 0 \\ & \forall i \in \mathcal{F}:\, y'_i \geq 0 \end{array} \tag{5}$$

$$\begin{array}{ll} \text{maximize} & \sum_{j \in \mathcal{C}} r_j \alpha'_j \\ \text{subject to} & \forall i \in \mathcal{F}:\, \sum_{j \in \mathcal{C}} \beta'_{ij} \leq \rho_f f_i \\ & \forall i \in \mathcal{F}, j \in \mathcal{C}:\, \alpha'_j p_i - \beta'_{ij} \leq \rho_c c_{ij} \\ & \forall j \in \mathcal{C}:\, \alpha'_j \geq 0 \\ & \forall i \in \mathcal{F}, j \in \mathcal{C}:\, \beta'_{ij} \geq 0 \end{array} \tag{6}$$

The constant factor analysis relies on the threshold that $\forall j \in \mathcal{C}:\, r_j = r$ and $r \geq \max_i p_i$. We first denote the total solution costs of LPs (3), (4), (5) and (6) by SOL_{LP}, SOL_D, SOL'_{LP} and SOL'_D respectively. In the original problem, let $SOL_{LP} = F_{SOL} + C_{SOL}$, where F_{SOL} and C_{SOL} represent the total facility cost and connection cost (both are possibly fractional) of any solution SOL, then it is clear that $SOL'_{LP} = \rho_f \cdot F_{SOL} + \rho_c \cdot C_{SOL}$. Also, we can get the corresponding $SOL'_D = SOL_D$ by letting $\alpha'_j = \alpha_j$.

Now we denote SOL_P as the total cost of the feasible primal solution $(y_i,\, x_{ij})$ returned by the algorithm and let SOL_D represent the total cost of its corresponding constructed dual solution $(\alpha_j,\, \beta_{ij})$. We will see later how this dual is constructed. Obviously, $(y_i,\, x_{ij})$ is a feasible solution to both LPs (3) and (5). By the weak duality theorem established between LPs (5) and (6), and if the constructed solution $(\alpha_j,\, \beta_{ij})$ from the algorithm is feasible to LP (6) after letting $\alpha'_j = \alpha_j$ and $\beta'_{ij} = \beta_{ij}$, then we have $SOL_D = SOL'_D \leq SOL'_{LP} = \rho_f \cdot F_{SOL} + \rho_c \cdot C_{SOL}$. Further, if $SOL_P \leq SOL_D$ is true then it implies the algorithm is $(\rho_f,\, \rho_c)$-approximation. The following lemma is therefore immediate.

Lemma 3. *The Primal-Dual Algorithm is* $(\rho_f,\, \rho_c)$*-approximation if its constructed dual solution* $(\alpha_j,\, \beta_{ij})$ *is feasible to LP* (6) *and the corresponding* $SOL_D \geq SOL_P$.

The steps left are to construct a feasible dual $(\alpha_j,\, \beta_{ij})$ from our algorithm and show $SOL_D \geq SOL_P$. For the second step, we have $SOL_P = \sum_{j \in \mathcal{C}} \sum_{1 \leq vp \leq d_j} p_{\phi(j^{vp})} \alpha_j^{vp}$ because the total dual values fully pay client connection and facility opening costs in the algorithm. In order to bound SOL_P with SOL_D, we aim to establish a relationship between r_j's (fractional) and d_j's (integral). Without loss of generality, we can set $\forall i \in \mathcal{F}, j \in \mathcal{C}:\, \alpha_j = 2\alpha_j^{d_j},\, \beta_{ij} = \max(0,\, p_i \alpha_j - \rho_c c_{ij})$. We then have $SOL_D = \sum_{j \in \mathcal{C}} 2\alpha_j^{d_j} r_j = \sum_{j \in \mathcal{C}} \alpha_j^{d_j} (r_j + r_j)$. Next we use the threshold information $\forall j \in \mathcal{C}:\, r_j \geq \max_i p_i$ and the *key observation* that although

$\forall j \in \mathcal{C} : r_j \leq \sum_{1\leq vp\leq d_j} p_{\phi(j^{vp})}$, $r_j + \max_i p_i \geq \sum_{1\leq vp\leq d_j} p_{\phi(j^{vp})}$ because before client j makes the last connection $r_j \geq \sum_{1\leq vp\leq d_j-1} p_{\phi(j^{vp})}$ and $\sum_{1\leq vp\leq d_j-1} p_{\phi(j^{vp})} + \max_i p_i \geq \sum_{1\leq vp\leq d_j} p_{\phi(j^{vp})}$. Hence, $SOL_D \geq \sum_{j\in\mathcal{C}} \alpha_j^{d_j} (r_j + \max_i p_i) \geq \sum_{j\in\mathcal{C}} \sum_{1\leq vp\leq d_j} p_{\phi(j^{vp})} \alpha_j^{d_j} \geq SOL_P$ (since $\alpha_j^{d_j} \geq \alpha_j^{vp}$). Now the only step left is to show (α_j, β_{ij}) is a feasible solution. Obviously the second constraint of LP (6) holds from $\alpha_j = 2\alpha_j^{d_j}$ and $\beta_{ij} = \max(0, p_i\alpha_j - \rho_c c_{ij})$. The remaining is to show the first constraint also holds. Built upon Lemma 3, we have the following lemma and corollary.

Lemma 4. *The Primal-Dual Algorithm is (ρ_f, ρ_c)-approximation if $\forall i \in \mathcal{F} : \sum_{j\in\mathcal{A}} \left(2p_i\alpha_j^{d_j} - \rho_c c_{ij}\right) \leq \rho_f f_i$, where $\mathcal{A} = \left\{j \in \mathcal{C} \,|\, \alpha_j^{d_j} \geq \frac{\rho_c}{2} \cdot \frac{c_{ij}}{p_i}\right\}$.*

Corollary 1. *Without loss of generality, for every site i order the corresponding $k = |\mathcal{A}|$ clients in $\mathcal{A} = \left\{j \in \mathcal{C} \,|\, \alpha_j^{d_j} \geq \frac{\rho_c}{2} \cdot \frac{c_{ij}}{p_i}\right\}$ s.t. $\alpha_1^{d_1} \leq \alpha_2^{d_2} \leq \cdots \leq \alpha_k^{d_k}$. Then the Primal-Dual Algorithm is (ρ_f, ρ_c)-approximation if $\forall i \in \mathcal{F} : \sum_{j=1}^{k} \left(2p_i\alpha_j^{d_j} - \rho_c c_{ij}\right) \leq \rho_f f_i$.*

We proceed the proof to find ρ_f and ρ_c that bound all $\alpha_j^{d_j}$'s. The next lemma captures the metric property of the problem and Lemma 6 generates one pair of satisfying (ρ_f, ρ_c).

Lemma 5. *For any site i and clients j, j' with $r_j = r_{j'} = r$, we have $p_i\alpha_j^{d_j} \leq p_i\alpha_{j'}^{d_{j'}} + c_{ij} + c_{ij'}$.*

Proof. If $\alpha_j^{d_j} \leq \alpha_{j'}^{d_{j'}}$, the lemma obviously holds. Now consider $\alpha_j^{d_j} > \alpha_{j'}^{d_{j'}}$, it implies j' makes its final connection earlier than j in our algorithm. At time $t = \alpha_j^{d_j} - \epsilon$, client j' has already satisfied its RR $r_{j'}$ through connections with $d_{j'}$ open facilities while j has not fulfilled r_j. Thus among these $d_{j'}$ facilities there is at least one that j has not connected to, because otherwise j will have $r_{j'} = r = r_j$ fulfilled reliability which is a contradiction. Denote this facility by i', by triangle inequality we have $c_{i'j} \leq c_{ij} + c_{ij'} + c_{i'j'}$. Since i' is already open at time t, then $p_i\alpha_j^{d_j} \leq c_{i'j}$ by our algorithm; j' is connected to i', then $p_i\alpha_{j'}^{d_{j'}} \geq c_{i'j'}$. The lemma follows. □

The next lemma and the subsequent bi-factor approximation ratio are naturally generated from the inverse dual fitting analysis. On the other hand, they are difficult to establish using the traditional dual fitting technique.

Lemma 6. *For any site i with $s = |\mathcal{B}|$ clients s.t. $\mathcal{B} = \left\{j \in \mathcal{C} \,|\, \alpha_j^{d_j} \geq x \cdot \frac{c_{ij}}{p_i}\right\}$, $x > 0$ and $\alpha_1^{d_1} \leq \alpha_2^{d_2} \leq \cdots \leq \alpha_s^{d_s}$, then $\forall i \in \mathcal{F} : \sum_{j=1}^{s} \left(p_i\alpha_j^{d_j} - \left(2 + \frac{1}{x}\right) c_{ij}\right) \leq \left(1 + \frac{1}{x}\right) f_i$.*

Proof. First, we claim $\forall i \in \mathcal{F} : \sum_{j=1}^{s} \max\left(0, p_i\alpha_1^{d_1} - c_{ij}\right) \leq f_i$. This is clearly true because at time $t = \alpha_1^{d_1} - \epsilon$, all the clients in $\mathcal{B}$ are also in $\mathcal{U}$ which implies from our algorithm their total contribution should not exceed any facility's opening cost. So we also have:

$$\forall i \in \mathcal{F} : \sum_{j=1}^{s} \left(p_i\alpha_1^{d_1} - c_{ij}\right) \leq f_i \tag{7}$$

.

In Lemma 5, by letting $j' = 1$ and because in $\mathcal{B}$ $\alpha_1^{d_1} \geq \frac{x}{p_i} c_{i1}$, we get:

$$\forall i \in \mathcal{F}, j \in \mathcal{B} : p_i\alpha_j^{d_j} \leq \left(1 + \frac{1}{x}\right) p_i\alpha_1^{d_1} + c_{ij} \tag{8}$$

Therefore, after combining inequalities (7) and (8), $\forall i \in \mathcal{F}$:

$$\begin{aligned}
\sum_{j=1}^{s} p_i\alpha_j^{d_j} &\leq \sum_{j=1}^{s} \left(1 + \frac{1}{x}\right) p_i\alpha_1^{d_1} + \sum_{j=1}^{s} c_{ij} \\
&= \left(1 + \frac{1}{x}\right) \sum_{j=1}^{s} \left(p_i\alpha_1^{d_1} - c_{ij}\right) + \left(2 + \frac{1}{x}\right) \sum_{j=1}^{s} c_{ij} \\
&\leq \left(1 + \frac{1}{x}\right) f_i + \left(2 + \frac{1}{x}\right) \sum_{j=1}^{s} c_{ij}
\end{aligned}$$

The lemma then follows. □

Relating this lemma to Corollary 1, if $\mathcal{B} \supseteq \mathcal{A}$ then it implies (ρ_f, ρ_c)-approximation where $\rho_f = 2 + \frac{2}{x}$ and $\rho_c = 4 + \frac{2}{x}$. Also, $\mathcal{B} \supseteq \mathcal{A}$ iff $x \leq \frac{\rho_c}{2} = 2 + \frac{1}{x}$, i.e. $0 < x \leq 1 + \sqrt{2}$. Therefore, when $x = 1 + \sqrt{2}$, the algorithm is $\left(2\sqrt{2}, 2 + 2\sqrt{2}\right)$-approximation. However, this ratio can be reduced through the factor revealing technique in (Jain et al. 2003). Consider the following lemma that capture the execution of the primal-dual algorithm more precisely than the claim in Lemma 6.

Lemma 7. *For any site i and the corresponding k clients in $\mathcal{A}$, we have $\forall 1 \leq j \leq k : \sum_{h=j}^{k} \max\left(0, p_i\alpha_j^{d_j} - c_{ih}\right) \leq f_i$.*

Proof. At time $t = \alpha_j^{d_j} - \epsilon$, all clients ordered from j to k are in set $\mathcal{U}$ (not fulfilled) and they have the same dual value $\alpha_j^{d_j}$. The lemma then follows because at any time in the primal-dual algorithm, the total contribution of all clients in $\mathcal{U}$ will not exceed the facility's opening cost at site i. □

Now if we let $v_j = p_i\alpha_j^{d_j}$ in Lemma 5 and 7, from these lemmas it is clear v_j, f_i and c_{ij} here constitute a feasible solution to the factor revealing program (4) in (Jain et al. 2003). Also from its Lemma 3.6, we can directly get $\sum_{j=1}^{k} (v_j - 1.861c_{ij}) \leq 1.861f_i$, i.e. $\sum_{j=1}^{k} \left(2p_i\alpha_j^{d_j} - 3.722c_{ij}\right) \leq 3.722f_i$. This result together with Lemma 2 and Corollary 1 lead to the following theorem.

Theorem 1. *The Accelerated Primal-Dual Algorithm achieves 3.722-approximation for RRA in time $\tilde{O}(m)$ when all clients are provided with the same RR that is at least the highest reliability among all facilities.*

Since the UFTRA problem is a special case of RRA, we get the first strongly polynomial time algorithm for the uniform UFTRA problem.

Theorem 2. *The Accelerated Primal-Dual Algorithm achieves* 3.722*-approximation in time* $\tilde{O}(m)$ *for UFTRA with uniform connection requirements.*

In fact, by adapting our Algorithm 2, in (Liao & Shen n.d.) we are able to show that uniform UFTRA can be approximated with a factor of 1.861 in quasi-linear time. Furthermore, both Marek Chrobak (Chrobak n.d.) and the authors have observed that uniform UFTRA is approximation-preserving reducible to UFL.

3 Discussions

A majority of optimization problems target to either minimize or maximize the aggregation (either a linear combination or not) of various types of costs described in the problem instances under some constraints of the solution. However, problems like the shortest path only considers one type of cost—weights of edges, whereas the facility location problems normally have two costs—facility and connection costs. To approximate these problems involving different costs, the concept of multi-factor analysis arose naturally for balancing these costs in a solution and thereby obtaining a tighter/more precise approximation ratio. Although the inverse dual fitting technique may be seen as an extension of dual fitting, it actually occupies greater advantage by tightly coupling with the generic multi-factor analysis. Moreover in the ratio analysis of the RRA problem, we have shown this technique is able to simplify the analysis and work seamlessly with the factor revealing technique. Next, we will briefly see how the primal-dual method in (Vazirani 2001, Jain et al. 2002) together with this technique yields simpler analysis for the fundamental set cover problem.

In the minimum set cover problem, we are given a universe $\mathcal{U}$ of n elements and a collection $\mathcal{S}$ containing $s_1, \ldots, s_k$ that are subsets of $\mathcal{U}$ with corresponding non-negative costs $c_1, \ldots, c_k$. The objective is to pick a minimum cost collection from $\mathcal{S}$ whose union is $\mathcal{U}$. The problem can be easily formulated into the following LP in which the variable x_s denotes whether the set $s \in \mathcal{S}$ is selected.

$$\begin{aligned} \text{minimize} \quad & \sum_{s\in\mathcal{S}} c_s x_s \\ \text{subject to} \quad & \forall j \in \mathcal{U} : \sum_{s:j\in s} x_s \geq 1 \\ & \forall s \in \mathcal{S} : x_s \in \{0,1\} \end{aligned}$$

Its LP-relaxation and dual LP are:

$$\begin{aligned} \text{minimize} \quad & \sum_{s\in\mathcal{S}} c_s x_s \\ \text{subject to} \quad & \forall j \in \mathcal{U} : \sum_{s:j\in s} x_s \geq 1 \\ & \forall s \in \mathcal{S} : x_s \geq 0 \end{aligned}$$

$$\begin{aligned} \text{maximize} \quad & \sum_{j\in\mathcal{U}} \alpha_j \\ \text{subject to} \quad & \forall s \in \mathcal{S} : \sum_{j\in s\cap\mathcal{U}} \alpha_j \leq c_s \qquad (9) \\ & \forall j \in \mathcal{U} : \alpha_j \geq 0 \end{aligned}$$

In the primal-dual algorithm, all of the uncovered elements j's simply raise their duals α_j's until the cost of a set s in $\mathcal{S}$ is fully paid for. At this moment, s is selected (x_s is set to 1) and duals of j's in s are frozen and withdrawn from the sets other than s. The algorithm then iteratively repeat these steps until there are no uncovered elements left. Clearly at the end of algorithm, $\sum_{j\in\mathcal{U}} \alpha_j = \sum_{s\in\mathcal{S}} c_s x_s$. In the analysis that follows inverse dual fitting, we consider to scale the costs of all sets in $\mathcal{S}$ by a positive number ρ. Since the set cover problem has only one type of cost, the inverse dual fitting technique will only generate a single factor. Similar to the analysis in the RRA problem, if the solutions x_s's and α_j's produced here are feasible to the scaled problem, then we have $\sum_{j\in\mathcal{U}} \alpha_j \leq \sum_{s\in\mathcal{S}} \rho c_s x_s$ by the weak duality theorem and this implies the algorithm is ρ-approximation. Obviously, x_s's are feasible and the left to do is to show LP (9)'s scaled constraint holds, i.e. $\forall s \in \mathcal{S} : \sum_{j\in s\cap\mathcal{U}} \alpha_j \leq \rho c_s$. Without loss of generality, we can assume there are l_s elements in set s and $\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_{l_s}$. So now we need to show $\forall s \in \mathcal{S} : \sum_{j=1}^{l_s} \alpha_j \leq \rho c_s$. Also, from the primal-dual algorithm it is easy to see that at time $t = \alpha_i - \epsilon$, $\forall s \in \mathcal{S}, 1 \leq i \leq l_s : \sum_{j=i}^{l_s} \alpha_j \leq c_s$ ($\alpha_j = \alpha_i$) which implies $\forall s \in \mathcal{S} : \sum_{i=1}^{l_s} \alpha_i \leq \sum_{i=1}^{l_s} \frac{1}{l_s-i+1} c_s$. Therefore, $\rho = \max_{l_s} \sum_{i=1}^{l_s} \frac{1}{l_s-i+1} \leq \mathcal{H}_n$ (n-th harmonic number where $n = |\mathcal{U}|$) and the set cover is $\mathcal{H}_n$-approximation.

Finally, it would be very interesting to see how other techniques and problem contexts can benefit from the inverse dual fitting technique. Also, migrating the idea of reliability to some other classical problems remains theoretically challenging.

References

Byrka, J. (2007), An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem, *in* 'APPROX '07/RANDOM '07', Springer-Verlag, Berlin, Heidelberg, pp. 29–43.

Byrka, J., Srinivasan, A. & Swamy, C. (2010), Fault-tolerant facility location: A randomized dependent lp-rounding algorithm, *in* 'IPCO', pp. 244–257.

Charikar, M. & Guha, S. (2005), 'Improved combinatorial algorithms for facility location problems', *SIAM J. Comput.* **34**(4), 803–824.

Chrobak, M. (n.d.). private communication, 2011.

Chudak, F. A. & Shmoys, D. B. (2003), 'Improved approximation algorithms for the uncapacitated facility location problem', *SIAM J. Comput.* **33**(1), 1–25.

Chudak, F. & Williamson, D. (2005), 'Improved approximation algorithms for capacitated facility location problems', *Mathematical programming* **102**(2), 207–222.

Feige, U. (1998), 'A threshold of ln n for approximating set cover', *Journal of the ACM (JACM)* **45**(4), 634–652.

Guha, S. & Khuller, S. (April 1999), 'Greedy strikes back: Improved facility location algorithms', *Journal of Algorithms* **31**, 228–248(21).

Guha, S., Meyerson, A. & Munagala, K. (2001), Improved algorithms for fault tolerant facility location, *in* 'SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms', Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp. 636–641.

Guha, S., Meyerson, A. & Munagala, K. (2003), 'A constant factor approximation algorithm for the fault-tolerant facility location problem', *J. Algorithms* **48**(2), 429–440.

Jain, K., Mahdian, M., Markakis, E., Saberi, A. & Vazirani, V. V. (2003), 'Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP', *Journal of the ACM* **50**(6), 795–824.

Jain, K., Mahdian, M. & Saberi, A. (2002), A new greedy approach for facility location problems, *in* 'STOC '02: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing', ACM, New York, NY, USA, pp. 731–740.

Jain, K. & Vazirani, V. V. (2000), An approximation algorithm for the fault tolerant metric facility location problem, *in* 'APPROX '00: Proceedings of the Third International Workshop on Approximation Algorithms for Combinatorial Optimization', Springer-Verlag, London, UK, pp. 177–183.

Jain, K. & Vazirani, V. V. (2001), 'Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation', *Journal of the ACM* **48**(2), 274–296.

Li, S. (2011), A 1.488 approximation algorithm for the uncapacitated facility location problem, *in* 'ICALP (2)', pp. 77–88.

Liao, K. & Shen, H. (2011), Unconstrained and constrained fault-tolerant resource allocation, *in* 'COCOON', pp. 555–566.

Liao, K. & Shen, H. (n.d.), Fast fault-tolerant resource allocation. to appear in PDCAT 2011.

Lin, J.-H. & Vitter, J. S. (1992), e-approximations with minimum packing constraint violation, *in* 'STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing', ACM, New York, NY, USA, pp. 771–782.

Mahdian, M., Markakis, E., Saberi, A. & Vazirani, V. (2001), A greedy facility location algorithm analyzed using dual fitting, *in* 'APPROX '01/RANDOM '01: Proceedings of the 4th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 5th International Workshop on Randomization and Approximation Techniques in Computer Science', Springer-Verlag, London, UK, pp. 127–137.

Mahdian, M., Ye, Y. & Zhang, J. (2006), 'Approximation algorithms for metric facility location problems', *SIAM J. Comput.* **36**(2), 411–432.

Shmoys, D. B., Tardos, E. & Aardal, K. (1997), Approximation algorithms for facility location problems, *in* 'Proceedings of the 29th Annual ACM Symposium on Theory of Computing', pp. 265–274.

Sviridenko, M. (2002), An improved approximation algorithm for the metric uncapacitated facility location problem, *in* 'Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization', Springer-Verlag, London, UK, pp. 240–257.

Swamy, C. & Shmoys, D. B. (2008), 'Fault-tolerant facility location', *ACM Trans. Algorithms* **4**(4), 1–27.

Vazirani, V. V. (2001), *Approximation Algorithms*, Springer-Verlag, Berlin.

Xu, S. & Shen, H. (2009), The fault-tolerant facility allocation problem, *in* 'Proceedings of the 20th International Symposium on Algorithms and Computation', ISAAC '09, Springer-Verlag, Berlin, Heidelberg, pp. 689–698.

Yan, L. & Chrobak, M. (2011), 'Approximation algorithms for the Fault-Tolerant Facility Placement problem', *Information Processing Letters* .