Case b: For each j, $c_j > c_{\beta_j}$. Then we have an algorithm that picks $\beta_j k$ sets and covers $c_j n_j$ elements each time we run the algorithm. Suppose in iteration j, the number of uncovered elements at the start of the iteration is n_j. Clearly, $n_{j+1} = n_j(1-c_j)$ and $n_1 = |X|$. Suppose after ℓ iterations $n_{\ell+1} = 1$. The total number of sets that are picked is $\sum_{j=1}^{\ell} \beta_j k$, with an approximation factor of $\sum_{j=1}^{\ell} \beta_j$.

$$n_{\ell+1} = 1 = |X| \prod_{j=1}^{\ell} (1-c_j).$$

$$\ln |X| = \sum_{j=1}^{\ell} \ln \frac{1}{1-c_j}.$$

Using the fact that $c_j > (1 - \frac{1}{e^{\frac{\beta_j}{c}}})$, we get for each j,

$$\beta_j < c \cdot \ln \frac{1}{1-c_j}.$$

$$\sum_{j=1}^{\ell} \beta_j < c \sum_{j=1}^{\ell} \ln \frac{1}{1-c_j} = c \ln |X| \text{ for } c < 1.$$

Thus Set Cover $(X, \mathcal{S})$ is a $c \ln |X|$ approximation algorithm with $c < 1$. By Feige's result [4] this implies that $NP \subseteq DTIME[n^{O(\log\log n)}]$.

□

Remark: Notice the above proof uses a distance metric where the distances are either 1 or 3. For such a distance metric of 1 or 3 and arbitrary costs we can develop an algorithm that achieves an approximation ratio of 1.463 (Corollary 4.4). For uniform costs and arbitrary distances we present an algorithm with an approximation factor of 2.225 (Subsection 4.4). It would be very interesting if we can use non-uniform costs and distances to get an improved lower bound for the general problem.

4. Improved Approximation Algorithm

4.1. Background Review

We consider the linear programming relaxation of the integer program as discussed in Section 2. We relax the requirement that the x_{ij} and y_i values have to be integers. We now discuss our modified LP formulation and show how one can derive the algorithm by Shmoys, Tardos and Aardal [16] using this LP formulation.

$$\min \quad \sum_{i \in F} \sum_{j \in D} d_j c_{ij} x_{ij}$$

$$\text{subject to} \qquad \sum_{i \in F} x_{ij} = 1 \text{ for each } j \in D$$

$$x_{ij} \le y_i \text{ for each } i \in F, j \in D$$

$$\sum_{i \in F} f_i y_i \le \lambda_f.$$

$$x_{ij} \geq 0 \text{ for each } i \in F, j \in D$$
$$y_i \geq 0 \text{ for each } i \in F$$

This formulation assumes that we know the facility cost λ_f of the optimal integral solution. Since we do not know this value, we will "guess" it within a $(1+\epsilon)$ factor by trying many different values. Assume that $f_i \geq 1$ for each $i \in F$. Let $\mathcal{F} = \sum_{i \in F} f_i$. Let $\epsilon > 0$. For each $\ell = 1 \ldots \lceil \log_{1+\epsilon} \mathcal{F} \rceil$, we will set $\lambda^\ell = (1+\epsilon)^\ell$ and solve the above LP for by setting $\lambda_f = \lambda^\ell$ for each choice of ℓ. We will run our algorithm for each choice of ℓ, and take the best solution. For some j, we have $\lambda^{j-1} < \lambda_f \leq \lambda^j$. By using this value of j, we will accurately guess λ_f within a factor of $(1+\epsilon)$ for any fixed ϵ.

We now outline the algorithm given by Shmoys, Tardos and Aardal [16]. (Our description is a slight modification of the original description given in [16].) The algorithm works by fixing a parameter α between 0 and 1. We will refer to this algorithm as STA(α). Consider the optimal (fractional) solution corresponding to this new LP. The service cost of the optimal (fractional) solution is at most the service cost of the optimal of the integer program.

For each demand node j, order all the facilities by their distance from j in non-decreasing order. For an optimal (fractional) solution, we may assume that the demand node j exhausts a facility before routing demand from another facility at the same distance. This fact is based on the observation that an optimal solution will route demand from the closest open facilities. The formal argument is as follows: Redefine $x_{i_k j}$ as the following feasible solution to the LP formulation.

$$\begin{array}{lcll} x_{i_k j} & = & \min(y_{i_k}, 1 - \sum_{s=1}^{k-1} y_{i_s}) & \text{when } \sum_{s=1}^{k-1} y_{i_s} < 1 \\ x_{i_k j} & = & 0 & \text{otherwise} \end{array}$$

From the optimality of the original solution we can also conclude that the service cost of j remains the same under the new solution.

Define $N_\alpha(j)$ as the set of facilities (in the above order) such that their y_i values (and hence x_{ij} values due to the above property) sum to at least α. That is $\sum_{s=1}^{i'} y_s \geq \alpha$ and $\sum_{s=1}^{i'-1} y_s < \alpha$. The threshold of j, denoted by $\tau(\alpha, j)$, is defined as the service distance to the farthest facility in the set $N_\alpha(j)$. Initially all demand nodes are unsatisfied and $\mathcal{D}$ is $\emptyset$. Recursively choose the unsatisfied demand node with the minimum $\tau(\alpha, j)$ value and add it to $\mathcal{D}$. Consider the cheapest facility $\min(j)$, in $N_\alpha(j)$, for this demand node. By opening a facility there, the cost goes up by at most a factor of $\frac{1}{\alpha}$. If for any node j', $N_\alpha(j) \cap N_\alpha(j')$ is nonempty, serve j' by the facility chosen by node j. Mark j and all such j' as satisfied.

It is easy to argue that every node j' gets served within service distance $3\tau(\alpha, j')$. For a node in $\mathcal{D}$ which was chosen with minimum $\tau(\alpha, j)$ value, it is trivially true. A node j' that was marked by j can be served within service distance $\tau(\alpha, j') + 2\tau(\alpha, j) \leq 3\tau(\alpha, j')$. This follows from the minimality of $\tau(\alpha, j)$.

Since each open facility corresponds to disjoint $N_\alpha(j)$, we will show that the cost of facilities is at most $\frac{1}{\alpha}\lambda_f$.

$$\sum_{j \in \mathcal{D}} f_{\min(j)} \leq \sum_{j \in \mathcal{D}} f_{\min(j)} \left(\frac{\sum_{i \in N_\alpha(j)} y_i}{\alpha}\right) \leq \frac{1}{\alpha} \sum_{j \in \mathcal{D}} \sum_{i \in N_\alpha(j)} f_i y_i \leq \frac{1}{\alpha}\lambda_f.$$

The service cost is at most $\sum_j 3 d_j \tau(\alpha, j)$.

Define

$$g(\alpha) = \frac{\sum_j d_j \tau(\alpha, j)}{\lambda_c}$$