Set Cover with Requirements and Costs
Evolving over Time

Milena Mihail

College of Computing and Department of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta GA 30332
mihail@cc.gatech.edu

Abstract. We model certain issues of future planning by introducing time parameters to the set cover problem. For example, this model captures the scenario of optimization under projections of increasing covering demand and decreasing set cost. We obtain an efficient approximation algorithm with performance guarantee independent of time, thus achieving planning for the future with the same accuracy as optimizing in the standard static model.

From a technical point of view, the difficulty in scheduling the evolution of a (set cover) solution that is “good over time” is in quantifying the intuition that “a solution which is suboptimal for time \( t \) may be chosen, if this solution reduces substantially the additional cost required to obtain a solution for \( t' > t \)." We use the greedy set picking approach, however, we introduce a new criterion for evaluating the potential benefit of sets that addresses precisely the above difficulty.

The above extension of the set cover problem arose in a toolkit for automated design and architecture evolution of high speed networks. Further optimization problems that arise in the same context include survivable network design, facility location with demands and natural extensions of these problems under projections of increasing demands and decreasing costs; obtaining efficient approximation algorithms for the latter questions are interesting open problems.

1 Introduction: Set Cover with Time Parameters

Fluctuations in cost and demand is a natural phenomenon of free markets and the relevance of cost effective schedules under such fluctuations is fundamental for both clients and service providers. When cost and demand fluctuations are totally unpredictable the models are necessarily on-line and, by now, we have a rich theory of on-line algorithms with heuristics that have found concrete practical applications [2] [4] [14]. However, in many cases, fairly accurate projections for the evolution of cost and demand over time are known in advance and the performance measures of on-line models no longer apply. This paper focuses on the latter context.

The concern that motivated this paper involves the evolution of networks like SONET, ATM, FRAME RELAY, WDM, e.t.c. which are experiencing a sharp increase in service demand (e.g. bandwidth) together with a substantial decrease
Each pair of nodes A and B of the network is represented by an element i.

The requirement of element i, r(i), represents the point-to-point bandwidth demand between nodes A and B.

A set $S_j$ represents a specific SONET architecture that can be embedded in the network. It contains element i if and only if embedding the specific architecture can satisfy one unit of demand from A to B.

The cost $c(j)$ of set $S_j$ is the cost of the SONET architecture represented by $S_j$.

Figure 1 [6] describes a commercial SONET planning tool implementing an adaptation of the classical greedy algorithm for set cover. At a suitable level of abstraction, the heuristic of the planning tool can be viewed as follows: while there exists point-to-point bandwidth demand not covered by the SONET architectures selected so far, select a new architecture that is most efficient for the current iteration.
In Section 2 we give a reduction of set cover with time parameters to the standard set cover problem. This reduction carries over approximations and suggests a factor $\mathcal{H}(kT)$ algorithm for set cover with time parameters (e.g., following Chvatal [5]), where $k$ is the cardinality of the largest set in the original problem, $T$ is the number of points in time and $kT$ is the cardinality of the largest set in the set cover instance that arises in the reduction.

In Section 3 we improve this factor to an optimal $\mathcal{H}(k)$ by introducing a new criterion for picking sets and adapting the standard duality based performance guarantee accordingly. The new criterion captures the following intuition: (a) cost suboptimal solutions should be considered for time $t$, if such solutions reduce substantially the additional cost required to obtain a solution for $t' > t$, (b) there is benefit in postponing picking sets that are not necessary for time $t$, if the cost at time $t' > t$ drops substantially and (c) while there is potential benefit in sets whose cost drops over time, this benefit should be counter-measured against the potential redundancy of such sets, if their effectiveness will be eventually covered by other sets that are necessary at earlier times.

The survivable network design problem and versions of the facility location problem are further examples of combinatorial optimization problems that arise repeatedly in automated network design [1] [9] [12] [13] [15]; major progress for these problems has been reported recently [3] [8] [10] [11] [16]. The survivable network design problem and the facility location problem have natural extensions with time parameters which remain open. We give an outline in Section 4.

2 Reduction to Set Cover and a factor $\mathcal{H}(kT)$

Approximation

The formal definition of Set Cover with Requirements and Costs Evolving over Time is as follows. There is a universe of $n$ elements and a set system of $m$ sets denoted by $S_j$, $1 \leq j \leq m$. Let $k$ denote the maximum cardinality of a set $k = \max_{1 \leq j \leq m} |S_j|$. As in the standard set cover problem, elements have covering requirements and sets have costs. In this extended model however, requirements and costs evolve over $T$ discrete points in time. In particular, for each time $t : 1 \leq t \leq T$, each element $i$ needs to be covered by $r(i, t)$ sets that have been picked on or before time $t$, while picking one copy of set $S_j$ at time $t$ has cost $c(j, t)$. We assume that the future evolution of requirements and costs are known in advance and we consider a “buying” scenario where, if a set is picked at some point in time, it is never removed — for example, the purchase and installation of SONET architectures incurs tens or hundreds of millions of cost in equipment and management; once installed, such architectures are not removed. We wish to pick sets that satisfy the requirements at every point in time and are of minimal total cost. Formally, where $x(j, t)$ denotes the number
of copies of $S_j$ picked at time $t$, we have to solve:

$$
\min \sum_{t=1}^{T} \sum_{j=1}^{m} c(j, t)x(j, t)
$$

subject to

$$
\sum_{t'=1}^{t} \sum_{j : j \notin S_j} x(j, t') \geq r(i, t) \quad 1 \leq i \leq n, \ 1 \leq t \leq T
$$

$$
x(j, t) \in \mathbb{N}_0 \quad 1 \leq j \leq m, \ 1 \leq t \leq T
$$

We first give a reduction to the standard Set Cover problem. In particular, for a universe of $n$ elements, a set system of $m$ sets denoted by $S_j$, $1 \leq j \leq m$, and where $k = \max_{1 \leq j \leq m} |S_j|$, in the Set Cover problem each element $i$ has a covering requirement $r(i)$, picking one copy of set $S_j$ has cost $c(j)$ and we wish to find a minimum cost collection of sets that satisfy the covering requirements:

$$
\min \sum_{j=1}^{m} c(j)x(j)
$$

subject to

$$
\sum_{j : j \notin S_j} x(j) \geq r(i) \quad 1 \leq i \leq n
$$

$$
x(j) \in \mathbb{N}_0 \quad 1 \leq j \leq m
$$

Recall also the classical Greedy Algorithm for Set Cover which repeatedly picks sets that reduce the total number of requirements at minimum average cost per unit of covered requirement. More specifically, for each element $i$ with requirement $r(i)$ consider a stack of $r(i)$ chips labeled $p_{ir}$, $1 \leq r \leq r(i)$, and consider a further labeling of each chip as either covered or uncovered. Define the potential of a set $S_j$ with respect to such a labeling as the average cost at which the set covers uncovered chips: $P(S_j) = c(j)/|\{i : \exists p_{ir} \in S_j \text{ and } p_{ir} \text{ is uncovered}\}|$.

The algorithm then is:

**Greedy Algorithm for Set Cover**

$x(j) = 0, \ \forall j$;

label chip $p_{ir}$ “uncovered”, $\forall i, r$;

while there exist uncovered chips do

set $P(S_j) = |\{i : \exists p_{ir} \in S_j \text{ and } p_{ir} \text{ is uncovered}\}|, \ \forall j$;

for some $S_{j_0}$ that minimizes $c(j)/P(S_j)$ set $x(j_0) = x(j_0) + 1$;

for all $i$,

if some uncovered chip $p_{ir} \in S_{j_0}$ then

label chip $p_{ir}$ “covered” for exactly one such $r$;

set $\text{cost}(p_{ir}) = c(j_0)/P(S_{j_0})$;

Now the following performance guarantee is well known and follows by duality considerations [5]:

**Theorem 1.** [Chvatal]. The cost of the Greedy Algorithm for Set Cover is within a $H(k)$ multiplicative factor of the cost of any optimal solution: $\sum_{j=1}^{m} c(j)x(j) \leq H(k) \cdot \text{OPT}$.

We may now give the reduction from Set Cover With Parameters Evolving over Time to Set Cover. See Figure 2. For each element $i$ and each time $t$ of Set Cover With Parameters Evolving over Time we introduce an element $I_{it}$ with requirement $r(i, t)$ for Set Cover, and for each set $S_i$ and each time $t$ of Set Cover With Time Parameters we introduce a
new set $S_{jt} = \{I_{jt} : i \in S_j, t' \geq t\}$ of cost $c(j,t)$ for SET COVER. Realize that the maximum set cardinality is $kT$, thus Chvatal’s Theorem suggests that the greedy algorithm for set cover achieves a $\ln kT$ approximation factor. In the next section we will modify the criterion for picking sets and achieve approximation factor $\ln k$. This is optimal in view of Feige’s bound [7].

**Figure 2** Indicating the the reduction of SET COVER WITH PARAMETERS EVOLVING OVER TIME to SET COVER. This reduction increases the size of the problem by a factor $T$.

### 3 A factor $H(k)$ Modified Greedy Algorithm

How can we improve the GREEDY ALGORITHM of Section 2 when applied to SET COVER instances that arise from the reduction from SET COVER WITH REQUIREMENTS AND COSTS EVOLVING OVER TIME? Realize that a good heuristic for the latter set cover problem should capture the following: (a) cost suboptimal solutions must be considered for time $t$, if such solutions reduce substantially the additional cost required to obtain a solution for $t' > t$, (b) there is benefit in postponing picking sets that are not necessary for time $t$, if the cost at time $t' > t$ drops substantially and (c) while there is potential benefit in sets whose cost drops over time, this benefit should be counter-measured against the potential redundancy of such sets, if their effectiveness will be eventually covered by other sets that are necessary at earlier times. Realize further that the potential of sets $S_{jt}$ arising in the reduction indeed capture (a) and (b). In particular, for
(a), note that a set $S_{j,t}$ includes elements representing requirements for times $t' \geq t$ which may increase the potential of $S_{j,t}$, while for (b), note that a substantial drop of the cost of a set $S_j$ at time $t'$ is represented by the cost of the set $S_{j,t'}$ which must consequently become of high potential. However, the reduction does not capture (c). In particular, a set $S_{j,t'}$ of very low cost could be chosen at first to satisfy the requirement of an element at time $t'$. On the other hand, this element may also have requirements at time $t < t'$ which will eventually result in the choice of sets $S_{j,t}$, thus making the choice of $S_{j,t'}$ redundant. The **Modified Greedy Algorithm** below modifies the set picking criterion to take into account (c).

For the description of the **Modified Greedy Algorithm** we need the following notation. See Figure 3. For each element $I_{it}$ with requirement $r(i,t)$ consider a stack of $r(i,t)$ chips labeled $P_{it}$, $1 \leq r \leq r(i,t)$. Define a **line** as a set of chips where $i$ and $r$ are fixed and $t$ varies arbitrarily, and denote such a line by $L_{ir} = \{p_{it} : 1 \leq t \leq T\}$. Say that $L_{ir} \in S_j$ if and only if $t \leq \min \{t' : p_{it'} \in L_{ir}\}$. Consider a further labeling of each line as either **covered** or **uncovered**. Define the potential of a set $S_{j,t}$ with respect to such a labeling as the average cost at which the set covers uncovered lines: $\mathbb{P}(S_{j,t}) = C(j, t)/|\{i : \exists L_{ir} \in S_{j,t} \text{ and } L_{ir} \text{ is uncovered}\}|$.

![Figure 3](image-url)

**Figure 3** Indicating sets covering entire lines. For example, $S_{j,3}$ covers line $L_{i4}$, but $S_{j,3}$ does not cover lines $L_{i3}$, $L_{i2}$ and $L_{i1}$.

Now we modify the set picking criterion as follows:
Modified Greedy Algorithm for Set Cover with Parameters Evolving over Time

\( x(j, t) = 0, \forall j, t; \)

label line \( L_{ir} \) “uncovered”, \( \forall i, r; \)

while there exist uncovered lines do

- set \( P(S_{j \tau}) = |\{i : \exists L_{ir} \in S_{j \tau} \text{ and } L_{ir} \text{ is uncovered}\}|, \forall j, \tau; \)
- for some \( S_{j \tau} \) that minimizes \( c(j, t) / P(S_{j \tau}) \) set \( x(j_0, t_0) = x(j_0, t_0) + 1; \)

for all \( i, \)

- if some uncovered line \( L_{ir} \in S_{j \tau} \) then
  - label line \( L_{ir} \) “covered” for exactly one such \( r; \)
  - set \( \text{cost}(L_{ir}) = c(j_0, t_0) / P(S_{j \tau}); \)

For the performance guarantee observe:

**Lemma 2.** For all sets \( S_{j \tau}, \) \( \sum_{i \in S_{j \tau}} \max_{L_{ir} \in S_{j \tau}} \text{cost}(L_{ir}) \leq H(k) \cdot c(j, t). \)

**Proof.** Assume without loss of generality that \( S_{j} = \{1, \ldots, k'\}. \) Also assume without loss of generality that for fixed \( t \), among all lines \( L_{ir} \in S_{j \tau} \) (as \( r \) varies), \( L_{ir} \) was the last line to be covered by the Modified Greedy Algorithm. Finally assume without loss of generality that, for all \( 1 \leq i' \leq i \leq k', \) \( L_{ir} \) was covered at a previous or at the same iteration of the Modified Greedy Algorithm as line \( L_{i'r'} \). Then, since \( S_{j \tau} \) could have covered \( L_{ir}, \) at cost no more than \( c(j, t) / i \) we have:

\[
\text{cost}(L_{ir}) \leq \frac{c(j, t)}{i}, \quad 1 \leq i \leq k'
\]

Thus,

\[
\sum_{i \in S_{j}} \max_{L_{ir} \in S_{j \tau}} \text{cost}(L_{ir}) = \sum_{i \in S_{j}} \text{cost}(L_{ir}) \\
\leq \left( \frac{1}{k'} + \frac{1}{k' - 1} + \ldots + 1 \right) c(j, t) \\
\leq H(k) \cdot c(j, t).
\]

**Theorem 3.** Performance Guarantee. The cost of the solution of Modified Greedy Algorithm is within a \( \log k \) multiplicative factor of the cost of any optimal solution: \( \sum_{t=1}^{T} \sum_{j=1}^{m} c(j, t)x(j, t) \leq H(k) \cdot \text{OPT} \)

**Proof.** Follows by applying the Lemma to the sets of some optimal solution. In particular, let us fix an optimal solution, and suppose that it contains \( x^*(j, t) \) copies of set \( S_{j \tau}. \) Then

\[
H(k) \cdot \text{OPT} = H(k) \sum_{t=1}^{T} \sum_{j=1}^{m} c(j, t)x(j, t) \\
\geq \sum_{t=1}^{T} \sum_{j=1}^{m} x^*(j, t) \sum_{i \in S_{j \tau}} \max_{L_{ir} \in S_{j \tau}} \text{cost}(L_{ir}), \text{ by the Lemma}
\]
\[
\sum_{i=1}^{n} \max_{r(i, t)} \sum_{t=1}^{T} c(L_{ir}), \text{ by counting}
\]

\[
= \sum_{t=1}^{T} \sum_{j=1}^{m} c(j, t) x(j, t), \text{ also by counting}
\]

4 Survivable Network Design and Facility Location

In the survivable network design problem we are given a weighted undirected graph and a requirement function over the cuts of the graph. We wish to pick a minimum cost subgraph such that each cut is crossed by at least as many edges as its requirement. Formally, for an undirected graph \(G(V, E), n = |V|\), cost function \(c\) on its edges: \(E \xrightarrow{\mathbf{c}} \mathbb{Q}_+\), cut requirement function \(f: 2^V \xrightarrow{\mathbf{f}} \mathbb{R}_0^+\) and where \(\delta(S)\) is the set of edges in \(E\) with exactly one endpoint in \(S\), the survivable network design problem is expressed by the integer program below.

\[
\min \sum_{e \in E} c(e) x(e) \quad \sum_{e \in \delta(S)} x(e) \geq f(S) \quad \forall S \subseteq V \quad \forall e \in E
\]

The survivable network design problem models “survivability/reliability” considerations and has a long history in practice and in theory [1] [4] [8] [10] [12] [13] [16]. The extension of this problem with time parameters involves a cut requirement function which increases with time and costs of edges which decrease with time. Formally, for an undirected graph \(G(V, E), n = |V|\), \(T = \{1, 2, \ldots, T\}\) points in time, cost function \(c\) on the edges: \(E \xrightarrow{T} \mathbb{Q}_+\) and cut requirement function \(f: 2^V \times T \xrightarrow{\mathbf{f}} \mathbb{R}_0\), the network design problem with requirements and costs evolving over time is expressed by the integer program below.

\[
\min \sum_{t=1}^{T} \sum_{e \in E} c(e, t) x(e, t) \quad \sum_{t'=1}^{t} \sum_{e \in \delta(S)} x(e, t') \geq f(S, t) \quad \forall S \subseteq V, \forall t \in [T] \quad \forall e \in E
\]

Obtaining an approximation for the above problem is open. Such an approximation would be of concrete practical importance, for example, in upgrading the architectures of survivable CCSN, SONET, and WDM telecommunications networks.

The facility location problem has many variants; here we outline a representative one. There is a collection of facilities: each facility \(i\) can be opened at cost \(f(i)\) for each unit of capacity \(u(i)\). There is also a collection of cities: each city \(j\) has demand \(d(j)\) which must be routed to open facilities. One unit of demand from city \(j\) can be routed to facility \(i\) at cost \(c(i, j)\). We wish choose facilities
that minimize the total cost:

\[
\min \sum_{i \in F,j \in C} c(i,j)x(i,j) + \sum_{i \in F} y(i)f(i) \\
\sum_{i \in F} x(i,j) \geq d(j) \quad \forall j \in C \\
u(i)y(i) \geq \sum_{j \in C} x(i,j) \quad \forall i \in F, \forall j \in C \\
x(i,j), y(i) \in \mathbb{N}_0 \quad \forall i \in F, \forall j \in C
\]

Now in the facility location problem with parameters evolving over time we have the demands increasing with time, and the costs of opening facilities and routing demands decreasing with time. We may write:

\[
\min \sum_{t=1}^{T} \sum_{i \in F,j \in C} c(i,j,t)x(i,j,t) + \sum_{i \in F} y(i,t)f(i,t) \\
\sum_{t'=1}^{T} \sum_{i \in F} x(i,j,t') \geq d(j,t) \quad \forall j \in C, \forall t \in [T] \\
u(i) \sum_{t'=1}^{T} y(i,t') \geq \sum_{j \in C} x(i,j,t') \quad \forall i \in F, \forall j \in C, \forall t \in [T] \\
x(i,j,t), y(i,t) \in \mathbb{N}_0 \quad \forall i \in F, \forall j \in C, \forall t \in [T]
\]

Obtaining an approximation for the above problem (or some suitable variant) is open. Such an approximation would be of importance, for example, in upgrading the architectures of ATM and Frame Relay telecommunications networks [15].

References

11. K. Jain and V.V. Vazirani, “Primal-Dual Approximation Algorithms for Metric Facility Location and \(k\)-Median Problems”, submitted, also in http://www.cc.gatech.edu/~vijay/

This article was processed using the \LaTeX-X macro package with LLNCS style