OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0030-364X | EISSN 1526-5463 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000

Dynamic evolution of economically preferred facilities

Dorit S. Hochbaum
Department of Industrial Engineering and Operations Research and Walter A. Haas School of Business, University of California, Berkeley hochbaum@ieor.berkeley.edu, http://www.ieor.berkeley.edu/ hochbaum/

In a sustained development scenario it is often the case that an investment is to be made over time in facilities that generate benefits. The benefits result from joint synergies between the facilities expressed as positive utilities specific to some subsets of facilities. As incremental budgets to finance fixed facility costs become available over time, additional facilities can be opened. The question is which facilities should be opened in order to guarantee that the overall benefit return over time is on the highest possible trajectory. This problem is common in situations such as ramping up a communication or transportation network where the facilities are hubs or service stations, or when introducing new technologies such as alternative fuels for cars and the facilities are fueling stations, or when expanding the production capacity with new machines, or when facilities are functions in a developing organization that is forced to make choices of where to invest limited funding.

An intuitive strategy frequently used to evolve the set of facilities is a greedy approach that picks additional facilities which provide a highest rate of return on each budget increment. Such strategy is shown to have adverse impact by locking the system into future suboptimal solutions with total benefit that can be arbitrarily small compared to the optimal evolution sequence. It is shown here that the degree of suboptimality of such greedy strategy is extreme in that it can lead to the loss of the majority of future benefits.

The main result here is the generation of the entire efficient frontier of optimal solution sets for different budget levels. This frontier contains information that guides the schedule of the optimal evolution of the set of facilities in future expansions. It is demonstrated that the efficient frontier forms a concave envelope, and the configurations on the frontier's breakpoints are nested. The efficient frontier is generated by efficient and strongly polynomial algorithms.

For n potential facility locations and for subsets of positive benefits of total size m we find the breakpoints of the efficient frontier using a parametric minimum cut procedure in time $O(nm \log \frac{n^2}{m})$, and all N configurations on the efficient frontier in time $O(mn \log n + N)$.

Subject classifications: Combinatorial optimization; facilities planning; minimum cut, dense subgraphs; location theory; evolution of facilities

1. Introduction

The problem studied here is encountered in the development of a facility system where funds for opening facilities become available over time. Potential facilities manifest networking effect in the form of joint synergies expressed as positive benefits for some subsets of facilities. As incremental budgets to finance fixed facility costs become available, additional facilities can be opened and the overall benefit from the system grows. The problem is to determine which facilities should be opened in order to guarantee that the overall return over time is on the highest possible trajectory. This problem is common in situations such as ramping up a communication of transportation network where the facilities are hubs or service stations, or when introducing new technologies such as alternative fuels for cars and the facilities are fueling stations, or in a newly evolving organization that is forced to make choices of where to invest limited funding.

An intuitive strategy frequently used to evolve the set of facilities is a greedy approach that selects facilities to open as those that provide a highest rate of return on each budget increment. The determination of the set of facilities to be added subject to a fixed budget on the total facility costs is referred to as the *maximum benefit* problem with the goal of maximizing the utility, or benefit, of connections between all facilities in the set and subject to the limited budget constraint. The maximum benefit problem is NP-hard as it is equivalent to several well studied NP-hard problems including the problems of finding a maximum clique in a graph, and finding densest k-subgraph.

Witzgall and Saunders (1988) were the first to address the problem of evolving a set of facilities in a project of deciding on the locations of electronic mail facilities for the postal service during the 80's. In that project, there is a positive benefit for each pair of facilities, and with added number of facilities the utility for all participants increases. This is similar to the effect of having a larger number of users connected to a telephone network increasing the utility for all users, at a quadratic rate in the number of users, as each of them can communicate with all users already on the network.

Beyond the objective of finding the best configuration of facilities for each given budget we are interested in the practical question of how to evolve the facility configurations over time, adding facilities as additional budget becomes available. Once a facility has been included, its fixed cost cannot be recouped by closing the facility, and therefore an open facility remains open for the entire planning horizon. Consequently, each selected set of facilities for a given budget is required to contain the previously selected set of facilities. We call this property the *nestedness* property. Our interest is in expanding the set of facilities over time, adding facilities as additional budget becomes available, while maintaining optimal return on investment in the form of maximum benefit-to-cost ratio.

Even if it were possible to compute the maximum benefit facility set for each given budget value, these optimal configurations for increasing budgets are not necessarily nested. Witzgall and Saunders introduced the concept of a *preferred configuration*. A preferred configuration is defined recursively as follows: a configuration is preferred if its marginal (or incremental) ratio of benefit-to-cost with respect to the previous preferred configuration is at least as great as that of all configurations for equal or greater budgets. To initialize this recursive definition, the empty set of cost 0 and benefit 0 is considered to be the first preferred configuration. The concave curve on which all preferred configurations reside is called the *efficient frontier*. There are no configurations that have a higher rate of return per unit budget than the efficient frontier configurations. We refer to the problem of finding the efficient frontier and the preferred configurations lying on it as the *dynamic evolution* problem.

Surprisingly, although evaluating the maximum benefit solution for a given specific budget is an NP-hard problem, computing the entire efficient frontier and the optimal sequence of nested facility sets can be executed in polynomial time.

1.1. Applications of the dynamic evolution problem

The structure of the problem of locating electronic mail facilities, studied by Witzgall and Saunders (1988), is commonplace in sustainable development projects. The common thread in these applications is the need to ramp up the infrastructure (of facilities) in order to ensure the impact of the project in terms of sustained and effective benefit level.

- One major hurdle to introducing *hydrogen fuel* operated cars is the accessibility of refueling stations. Presently the availability of funds for opening refueling facilities is limited, leading to a restricted availability of such facilities, which in turn will inhibit the acceptance of the technology. While it may appear attractive to open those facilities that generate the largest return at a given point in time, this may not be the optimal strategy. There is a synergy between the locations of refueling stations as they determine the range of travel for hydrogen cars. In locating the current network of refueling facilities it is important to understand the benefit structure of those facilities and use an optimal evolution process for the planned growth of the network.
- The problem of evolving a *product portfolio* over time at Hewlett Packard (HP) has recently been studied by Feng et al. (2004) and reported in Zhang et al. (2005). HP business unit has thousands of active products. This large number has adverse effects in confusing customers and

causing high product management costs. Yet, while a small subset of products generate most revenue, there are low revenue products appearing in customers' orders jointly with high revenue ones. So the low revenue products cannot be removed from the portfolio as they provide benefit in terms of their synergy with high revenue products. The goal is to generate a highest revenue portfolio that does not exceed a prescribed limit on the number of product offerings and that will take into account the joint synergy between products. The efficient frontier in this case provides insight to the trade-offs involved by limiting the number of products in the portfolio. As management capabilities and support systems improve it becomes possible to increase the size of the offering while maintaining good order fulfillment performance. The question is then which products should be added to the portfolio offered, without removing currently offered products (as such removal may adversely affect customer service.)

- A *global mobile telephone network* employs satellites and ground networks as part of its facilities. The key to the success of the system is in exploiting the synergy between the satellite facilities and the existing ground networks. The ability to connect users (telephone, fax), wherever they may be in the world, depends on the availability of ground networks or satellite positioning. With limited resources restricting the purchase of satellite facilities and ground networks, an issue of major concern is how schedule the future evolution of the network and which markets to focus on, so as to provide the best service to markets that guarantee the highest rate of return.

1.2. The suboptimality of the greedy approach to dynamic evolution

When confronting the problem of adding facilities subject to a given additional budget, the intuitive approach is to add those facilities that contribute most to the benefit of the current configuration. This greedy approach can lead to seriously adverse outcomes in the future, as seen in a simple example illustrated in Figure 1 below. In this example all subsets that generate positive benefits are pairs of facilities. Therefore the problem can be presented as a graph with nodes as potential facilities, and edges for each pair that can generate positive benefit. The two facilities in the center numbered 1 and 2 have pairwise benefit of value M, where M represents a very large number. Facility 3 adds 1 unit of benefit to the first three. The same added benefit of 1 unit applies when adding facilities $4,\dots,k$ in this order, whereas adding any other facility contributes $1-\epsilon$ at most to any subset of $\{1,\dots,k\}$. (For the purpose of illustration in Figure 1 $k=5$ and n the number of potential facilities is equal to 10.) On the left side, adding any single facility to the first two from the set $\{k+1,\dots,n\}$ contributes $1-\epsilon$ to the total benefit, where ϵ represents a small positive number. Therefore none of these will be chosen in an effort to maximize the total immediate return whenever an incremental unit of budget is available.

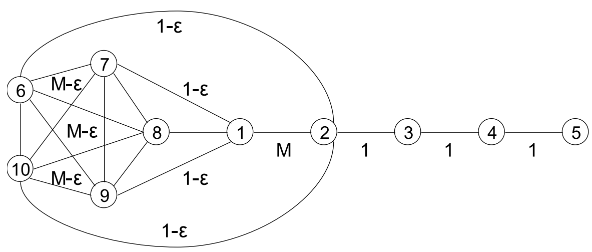

Figure 1 An example where greedy fails

We note that facilities $k+1,\dots,n$ form a clique (complete graph) where all pairwise benefits are $M-\epsilon$ each. In Figure 1 these are facilities $6,7,8,9,10$. Choosing these facilities as the first five facilities gives benefit of $10(M-\epsilon)$, whereas maximizing the immediate benefit will lead to the inferior configuration of facilities $\{1,2,3,4,5\}$ with total benefit of $M+3$. So the latter greedy solution yields total benefit almost 10 times smaller than the benefit generated by following the optimal evolution of the facility configurations. This optimal solution, consisting of the set $\{6,7,8,9,10\}$ is the first preferred configuration indicated on the efficient frontier, thus indicating that it is optimal to exclude facilities $\{1,\ldots,k\}$ for budgets that allow up to $n-k$ facilities. For general value of k the greedy and optimal choices of configurations lead to benefits of $M+(k-2)$ and $\frac{1}{2}k(k-1)(M-\epsilon)$, respectively. The ratio of the total benefit generated for k facilities by using the suboptimal greedy approach to the optimal benefit is then $O(\frac{1}{k^2})$ resulting in the loss of the majority of the potential benefits.

We note briefly that the ratio of $O(\frac{1}{k^2})$ achieved by the greedy is also the worst possible among all algorithms that choose to include the highest weight benefit pair. This is easy to see as any graph that has M as the highest benefit of a pair, can have total benefit on a subset of size k not exceeding $M\binom{k}{2}$.

1.3. The density graph and preferred configurations

We formalize the notions of preferred and nested configurations using graph terminology. The input to the problem is a complete undirected hypergraph $G=(V,E)$ where each edge in E, $e\subset E$, and with nonnegative edge weight, w_e, corresponding to the respective benefit of subset e, and with nonnegative fixed node costs, f_i, $i\in V$. (When each $|e|=2$ then the hypergraph is referred to as *graph*.) For a subset $V_H\subseteq V$, let hypergraph $H=(V_H,E_H)$ be the complete (clique) subgraph induced by V_H. The *benefit-cost ratio*, or the *density of* H, is defined as $\frac{W(E_H)}{C(V_H)}$ where $C(V_H)=\sum_{v\in V_H} f_v$ and $W(E_H)=\sum_{e\in E_H} w_e$. A commonly considered case is when all node weights are 1. In that case the "cost" represents the number of nodes in the clique and $C(V_H)=|V_H|$.

For a given *budget* B, the *maximum benefit problem* is to determine $D(B)=\max\{W(E_H): C(V_H)\le B\}$. We call the function $D(B)$ the *benefit-budget graph* or the *density graph*, and the ratio $\frac{D(B)}{B}$ is the corresponding *density*.

The efficient frontier is a concave envelope formed of the intersection of the set of all lines with the property that the density graph resides below the line. For any budget B the point $(B,D(B))$ lies on or below the concave piecewise linear function which forms the concave envelope of the density graph. A point on the envelope that has left derivative different from a right derivative is called a *breakpoint* (alternatively, breakpoints lie at the intersection of two line segments of the piecewise linear function.) The points corresponding to feasible configurations on the concave envelope are all preferred configurations.

For integer budgets and unit cost per location all preferred configurations $(B,D(B))$ satisfy $D(B)=\max\{W(E_H):C(V_H)=B\}$. This is the case since for any configuration of cost $<B$ there is another configuration of cost B with at least as high total benefit. Figure 2 depicts a schematic description of a density graph. All points on the concave envelope are preferred configurations, and some of them are breakpoints. In the instance illustrated in Figure 2, all the points of the density graph are preferred, except for the one corresponding to $B=1$.

With the formalism above we define several related problems:

1. **Maximum benefit problem**: Find $D(B)$ for a given B.
2. **Densest k-subgraph problem**: For G a graph, $f_i=1$ for all $i\in V$, and $w_{ij}=1$ for all edges $[i,j]\in E$, find $D(k)$ for a given k.
3. **Maximum clique problem**: Given an (incomplete) graph $G=(V,E')$ set $w_{ij}=1$ for all edges $[i,j]\in E'$ and $w_{ij}=0$ for $[i,j]\notin E'$, and let $f_i=1$ for all nodes i. The maximum clique problem is to find the largest k so that $D(k)=\binom{k}{2}$.
4. **Maximum density subgraph, or densest subgraph**: For $f_i=1$ for all nodes $i\in V$, find k so that $\frac{D(k)}{k}$ is maximum.
5. **The dynamic evolution problem**: Find the efficient frontier in the form of all preferred configurations and breakpoints.

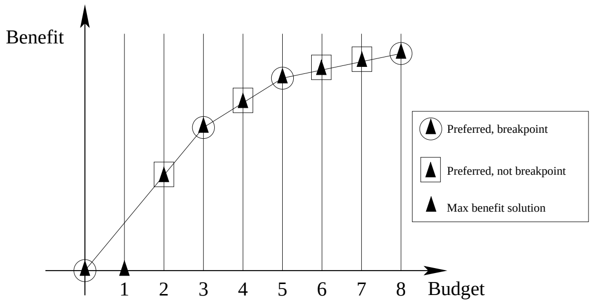

Figure 2 The concave envelope (or efficient frontier) of the density graph, the breakpoints and other preferred configurations.

The maximum benefit problem on a graph and the densest k-subgraph problem are identical except that the maximum benefit problem allows for arbitrary costs f_i. The maximum density subgraph problem does not specify the size of the sub-hypergraph and is polynomially solvable for graphs (see e.g. Goldberg (1984)) and for hypergraphs. The essence of our results is to demonstrate that the dynamic evolution problem is solvable efficiently and in the same complexity as solving the maximum density subgraph.

The breakpoints on the concave envelope have been used in other contexts previously. Barahona (2000) considered the problem of computing the concave envelope of the values of the minimum k-cut problem in a graph, as a function of k, and Ravi and Sinha (2002) used this concave envelope to get a 2-approximation algorithm for the minimum k-cut problem.

Our contributions here include the generation of the *entire* efficient frontier in polynomial time. That is, not only the breakpoints of the envelope, but also all configurations that lie on the envelope. Our work generalizes Witzgall and Saunders' model in several ways. Witzgall and Saunders proved that when all edge weights are positive then the preferred configurations set is identical to the set of breakpoints which are nested, and can be found using a minimum cost network flow algorithm. We extend that in several ways: We allow pairs that have 0 pairwise benefits; we show that the set of preferred configurations *contains* (possibly strictly) the set of breakpoints; we generalize the subsets that generate benefits from pairs to arbitrary sized subsets; and we show that the breakpoints of the concave envelope of the benefit-budget graph are a set of nested preferred configurations, but there are other preferred configurations that lie on the efficient frontier; finally, we prove that all the breakpoints as well as all other preferred configurations can be enumerated using a parametric minimum cut procedure in polynomial time.

We further prove that the efficient frontier provides a sequence of nested preferred configurations that determines the optimal sequence of facility configurations' expansion. The efficient frontier also provides useful upper bounds on the optimal value $D(k)$ that can be used in devising approximation algorithms for the NP-hard problem of finding densest k-subgraph, as shown in Hochbaum (2006).

The complexity of our algorithm for finding all the breakpoints on the efficient frontier is $O(mn \log \frac{n^2}{m})$ where n is the number of potential facility locations, and m is the size of all subsets with positive benefit. The complexity of finding all preferred configurations when there are N such configurations is $O(mn \log n + N)$. These are substantial improvements in both applicability and complexity compared to the work of Witzgall and Saunders (1988) who used the out-of-kilter algorithm in their approach.

1.4. Overview

The paper is organized as follows. In Section 2 we illustrate the concept of preferred configurations demonstrating that not all points $(B, D(B))$ are preferred, and not all preferred configurations are nested. This section provides further evidence to the failure of the greedy strategy and shows how the generation of the efficient frontier provides an optimal plan. Section 3 discusses the relationship of the dynamic evolution problem to the maximum clique problem, and shows that the efficient frontier may provide almost no information about the size of the maximum clique in the graph if the densest subgraph is the entire graph and $n_1 = n$. Section 4 contains the main algorithmic result for finding all nested preferred configurations. It describes how to generate the entire efficient frontier with a parametric minimum cut procedure and multiple optima for each cut. Guidelines for the use of the efficient frontier in evolving a facility set are provided in Section 5. Section 6 analyzes the complexity of the algorithm, and Section 7 describes the weighted densest subgraph problem showing that it is the smallest positive budget preferred configuration. We conclude in Section 8 with closing remarks and strategic suggestions for optimal dynamic evolution of a set of facilities.

2. An illustrative example and properties of preferred configurations

We elaborate on a graph example of Witzgal and Saunders demonstrating that not all preferred configurations are nested and that not all maximum benefit configurations are preferred.

Consider the graph shown in Figure 3 with 6 potential facilities, each with a fixed cost of 1. The utilities of the 15 possible pairs are given in Figure 3 for two cases. In case (a) the utilities of edges adjacent to nodes $5, 6$ all have weight equal to $\frac{1}{4}$. Case (b) is identical except that those utility weights adjacent to $5, 6$ are $\frac{2}{7}$. The utilities (or benefits) of the missing pairs $[4, 5]$ and $[4, 6]$ are both of weight 0 in instances (a) and (b).

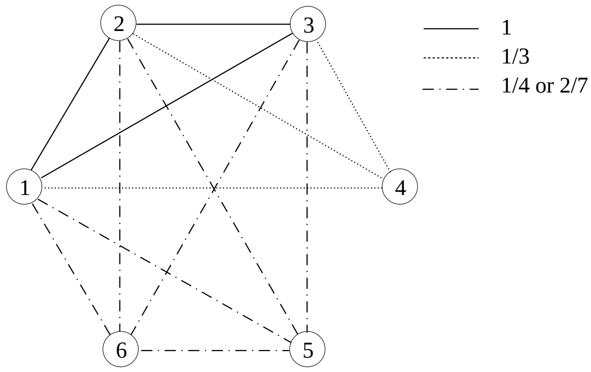

Figure 3 A 6-node graph

Figure 4 provides the density graphs for both cases and Table 1 lists the function values of $D(k)$.

Budget	Solution	(a)	(b)
2	{ 1,2}	1	1
3	{ 1,2,3}	3	3
4	{ 1,2,3,4}	4	4
5	{ 1,2,3,5,6}	$4\frac{3}{4}$	5
6	{ 1,2,3,4,5,6}	$5\frac{3}{4}$	6

Table 1 Optimal solutions for cases (a) and (b).

Case (a): For a budget or number of facilities equal to 2, 3 and 4 the maximum benefits are 1, 3 and 4 respectively. For 2 facilities the optimal set is any pair of nodes linked by an edge of weight 1. We choose the pair $\{1, 2\}$ but the nestedness property will hold whichever of the three possible pairs are selected. For 3 and 4 facilities the optimal sets are $\{1, 2, 3\}$ and $\{1, 2, 3, 4\}$ respectively. For

5 facilities the optimal system is $\{1,2,3,5,6\}$ of benefit of $4\frac{3}{4}$ which does *not* contain the optimal system of facilities for 4 facilities. For 6 facilities all facilities are selected at a total utility of $5\frac{3}{4}$. The trade-off curve which gives the maximum total benefits as a function of the total budget of the facilities is given in Figure 4. Note that the benefit value for 2 facilities and 5 facilities lie under the concave intersection of half planes that contain all the value points – the concave envelope. The density graph points $(B, D(B))$ lying on the envelope are the *preferred configurations*. The points on the piecewise linear envelope correspond to 1, 3, 4 and 6 facilities. In this example:

1. Not all maximum benefit configurations for given budgets are preferred nor do all reside on the concave envelope.

2. The breakpoints consist of the configuration for the budget of 4 and the trivial breakpoints corresponding to budgets of 0 and 6: $\{(0,0),(4,4\frac{3}{4}),(6,5\frac{3}{4})\}$.

3. In this example all the preferred configurations are nested, and all of them except the 3-configuration are breakpoints.

In terms of the evolution plan we determine from the efficient frontier that the configuration for $(4,4)$ should be expanded in the presence of incremental unit budget by adding facility 5 or 6.

Case (b), described next, demonstrates that preferred configurations are not always nested, unless they are also breakpoints.

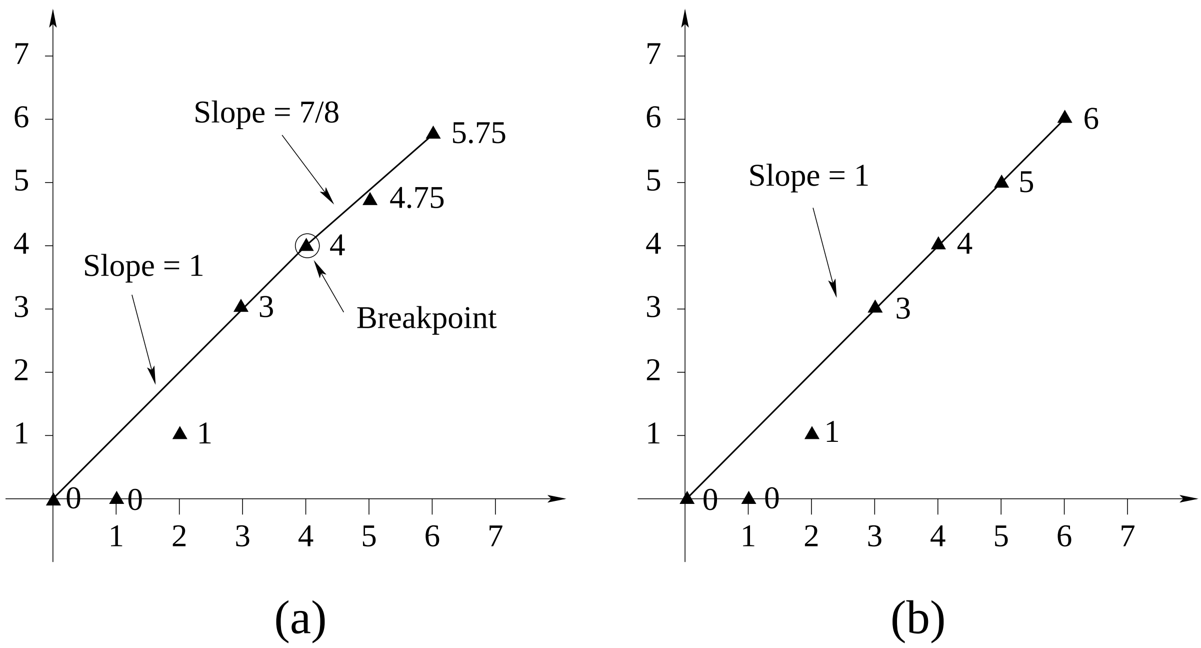

Figure 4 The concave envelopes for two benefit-cost graphs corresponding to cases (a) and (b).

Case (b): Here the weights of the edges adjacent to $5,6$ are $\frac{2}{7}$ each. The same configurations as in case (a) are optimal for each budget of k. The benefit values are however different. For 1, 2, 3, 4, 5 and 6 facilities, the optimal costs are 0, 1, 3, 4, 5 and 6 respectively. The optimal benefits for 3, 4, 5 and 6, all reside on one line segment of the concave envelope and the non-nestedness of the 4 facilities solution compared to the 5 facilities solution still holds. That means that the configuration for 4 facilities is not nested yet it is preferred. Still, all *breakpoints* – which are only the trivial ones – are nested configurations. The configurations for budgets of 3, 4, 5 and 6 are all preferred, but the 4-configuration is not contained in the 5-configuration.

This case illustrates that not all preferred configurations, or points on the concave envelope, are nested. Later we prove that the nestedness property is guaranteed only for preferred configurations that are also breakpoints.

The efficient frontier provides the preferred configurations for budgets of 3, 4 and 5. Because those are not nested, the list of preferred configurations allows for two alternative plans of expansion at budgets of 4 to 5: Either expand from $\{1,2,3,4\}$ to $\{1,2,3,4,5\}$, or else expand from $\{1,2,3,5\}$ to $\{1,2,3,4,5\}$. The benefit streams of these plans are 4 and $4\frac{6}{7}$ or $3\frac{6}{7}$ and $4\frac{6}{7}$ respectively. The first one is then the preferred alternative as the larger benefit payoff is available earlier.

In both cases (a) and (b) the maximum density subgraph corresponds to a budget of 3. This subgraph is the triangle consisting of the three nodes $1,2,3$ and three edges with density 1.

3. The efficient frontier and the maximum clique problem

In spite of the similarity of the maximum clique problem and the dynamic evolution problem for graphs, we show here that the entire efficient frontier may provide no information on the maximum clique in the graph. This explains why the generation of the efficient frontier in polynomial time does not contradict the hardness of the maximum clique problem. Even though the efficient frontier does not provide the information required to solve the maximum clique problem, it still provides insights to the structure of the graph. It is also demonstrated here that the most difficult graphs in terms of the maximum benefit problem (and maximum clique) are graphs where the densest subgraph is the entire graph itself.

The benefit maximization problem is NP-hard since the maximum clique problem is Turing-reducible to this problem. That is, several calls for benefit maximization on different values of budget can be used to solve the maximum clique problem as follows.

Consider the maximum clique problem on a graph $G=(V,E')$. Construct an instance of the budget maximization problem has with unit facility cost and with unit per pair/edge $e\in E'$. Missing edges $e\notin E'$ are assigned the weight 0. If the graph contains a clique of size $\bar{k}$ then $D(k)=\binom{k}{2}$ for all $k\le\bar{k}$. Therefore, if $k'=\min\{k|D(k)<\binom{k}{2}\}$ is the smallest value of k such that the maximum benefit configuration has fewer than $\binom{k'}{2}$ edges then there is a maximum clique in the graph of size $k'-1$.

The set of preferred configurations does not constitute the full description of the density graph as we saw in the examples in Section 2. Specifically, the maximum clique configuration may not lie on the concave envelope of the benefit cost curve of the density graph as shown in the following example.

We construct a graph with a maximum clique of size k on nodes $\{1,\dots,k\}$. The density of the maximum clique is $\binom{k}{2}/k=\frac{k}{2}-\frac{1}{2}$. Let node $k+1$ be adjacent to all but one node in the clique, nodes $2,\dots,k$; node $k+2$ be adjacent to nodes $3,\dots,k$ and to node $k+1$. The general pattern is to set node $k+p$ to be adjacent to nodes $p+1,\dots k+p-1$. The density of the $(k+1)$-subgraph on nodes $\{1,\dots,k+1\}$ is then $(\binom{k}{2}+k-1)/(k+1)=\frac{k}{2}-\frac{1}{k+1}>\frac{k}{2}-\frac{1}{2}$. Therefore the k-clique configuration in this graph is not preferred.

The remaining nodes $\{k+1,\dots,n\}$ each contribute $k-1$ to the total benefit. For $p=1,\dots,k-1$ the benefit increment from configuration of size p to size $p+1$ is p, and thus the slope is monotone increasing. Beyond the value k the slope is equal to $k-1$. For $p>k$ the slope is equal $k-1$. So the curve described by the density graph is convex and the derivatives or slopes of the tangents, are nondecreasing. Consequently the only preferred configurations that are on the concave envelope of the density graph are the trivial ones of size 0 and n. This does not provide any information about other points in the density graph, as illustrated in Figure 5. In this graph the maximum density subgraph is the entire graph, which is an indication that this is a hard instance for both the maximum clique and the maximum benefit problems.

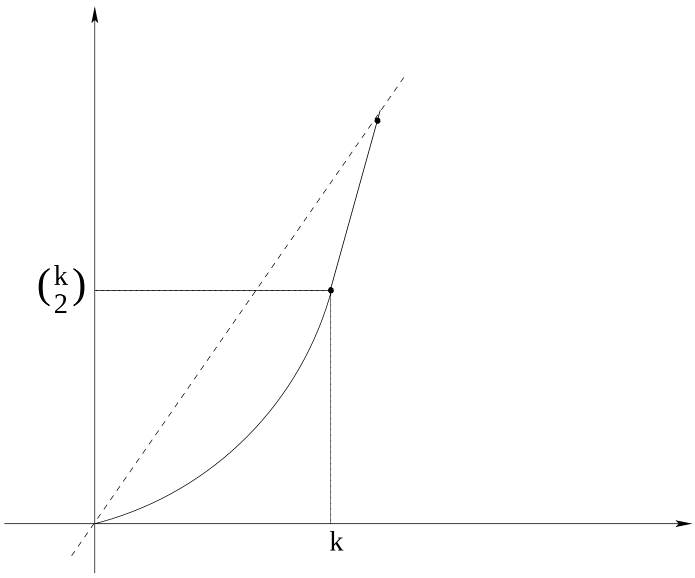

Figure 5 A density graph for a graph containing clique of size k.

If a graph containing a maximum k-clique is relatively sparse then the maximum clique is likely to be a preferred configuration. Sparsity means that the vertices not in a k-clique are adjacent to at most $k-2$ vertices in the clique. In this case, the maximum clique is necessarily the first breakpoint which is the maximum density subgraph.

The concave envelope, even if the maximum clique does not reside on it, provides an upper bound to the value of the maximum benefit configuration for each budget value. It is thus of value to have the concave envelope evaluated as a precursor to a maximum clique or maximum benefit optimization. Indeed, we show in a related paper Hochbaum (2006) that the concave envelope is useful in devising an approximation algorithm for maximum dispersion problem.

In summary, even though the problem of identifying *all* maximum benefit configurations is NP-hard, it is possible to find all the preferred configurations in polynomial time.

4. Finding all preferred configurations

Recall that a preferred configuration is a point lying on the concave envelope of the density graph, which has marginal benefit to extra cost ratio greater or equal to that of configurations of equal or larger budgets.

To find the concave envelope we identify all the lines tangent to that envelope. Let a line of a positive slope value λ be passing through a feasible configuration point corresponding to a subgraph H, (x,y), where $x = \sum_{j\in V_H} f_j$ is the sum of the fixed costs and $y = \sum_{ij\in E_H} w_{ij}$ is the sum of pairwise benefits, as in Figure 6. Let the line intersection with the x-axis be Δ. If the line is not tangent to the concave envelope then it is possible to find another configuration point where the value of Δ is smaller by pushing the line upwards. To find a tangent line of slope λ we seek among all the points in the graph the one for which the value of Δ is smallest (or the most negative) in order to define a half plane containing all feasible points in the density graph. For a given λ and a configuration H the value of Δ is,

$$\Delta = -\frac{1}{\lambda}\{\sum_{e\in E_H} w_e - \lambda \sum_{j\in V_H} f_j\}$$

Thus for a tangent line of slope λ, Δ is smallest for a configuration H maximizing $\sum_{e\in E_H} w_e - \lambda \sum_{j\in V_H} f_j$. Consequently, all tangents of any slope value λ can be found by maximizing this expression.

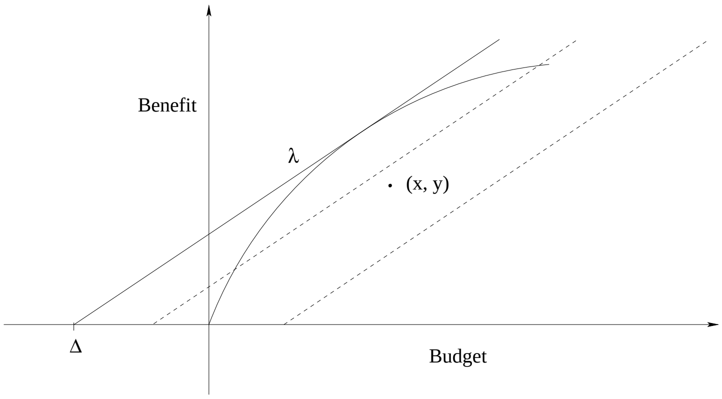

Figure 6 Identifying the concave envelope by minimizing Δ.

In order to find the entire concave envelope we solve a sequence of maximization problems, we call λ-problems, parameterized by the slopeλ,

$$\max_{H\subseteq G} \sum_{e\in E_H} w_e - \lambda \sum_{j\in V_H} f_j.$$

This discussion proves the following lemma.

Lemma 4.1*All configurations lying on the concave envelope of the density graph with marginal benefit-to-cost of λ are the subgraphs H that solve optimally the λ-problem.*

The λ-problem for a fixed value of λ is an instance of the *selection problem* introduced by Balinski (1970) and by Rhys (1970). The selection problem is defined for a set of items and a collection of m subsets each with a benefit associated with selecting all the items in the subset. For items $j \in \{1,\ldots,n\}$ each having an associated cost, f_j, and sets of items $S_i \subseteq \{1,\ldots,n\}$, $i = 1,\ldots,m$, each with benefit b_i, the objective is to maximize net benefit defined as total benefit minus total cost of items selected. A *feasible selection* is the pair $(J,\mathcal{S})$ with a collection of sets, $S_j, j \in J$, and the selected items $\mathcal{S} = \cup_{j\in J} S_j$. The objective of the selection problem is to find $(J,\mathcal{S})$ that maximizes:

$$\sum_{i\in J} b_i - \sum_{j\ \in \mathcal{S}} f_j.$$

Let $G = (V \cup \{s,t\}, A)$ be a directed graph containing source and sink nodes s and t and capacity upper bound u_{ij} for each arc $(i,j) \in A$. An s-t cut is a partition of $V \cup \{s,t\}$ to two sets S and T so that $s \in S$ and $t \in T$. S is called the *source set* of the cut and T is called the *sink set*. The set of arcs in the cut is $(S,T) = \{(i,j)|i \in S, j \in T\}$ and the capacity of the cut is $C(S,T) = \sum_{i\in S, j\in T} u_{ij}$. A cut is said to be finite if $C(S,T) < \infty$.

The selection problem is solved by finding a minimum capacity s-t cut in an appropriate bipartite network $G = (\{s\} \cup V_1 \cup V_2 \cup \{t\}, A)$ constructed as follows: One side of the bipartition, V_1, is a set of m nodes each representing a set S_i, and the other side of the bipartition, V_2, is a set of n nodes each representing an item. There is an arc (i,j) of infinite capacity if item $j \in S_i$. A source node s and a sink node t are added with arcs from s to $i \in V_1$ of capacity b_i, and from $j \in V_2$ to t of capacity f_j.

The source set of any finite cut corresponds to a feasible selection. Otherwise, if an element of a set is not included in the source set then an arc of infinite capacity is present in the cut. So the source set of the cut from which node s is removed forms a feasible selection consisting of a collection of sets and the union of their elements. For the sake of completeness we include here the proof that minimizing the capacity of an (s,t)-cut is equivalent to maximizing the net benefit.

For an s-t cut (S,T) of finite capacity $C(S,T)$, let $V_1 \cap S$ be the set nodes in the corresponding selection and $V_2 \cap S$ the item nodes in the selection. The net benefit of this selection is $NB(S) = \sum_{i\in S\cap V_1} b_i - \sum_{j\in S\cap V_2} f_j$.

$$\begin{aligned} C(S,T) &= \sum_{i\in T\cap V_1} b_i + \sum_{j\in S\cap V_2} f_j \\ &= \sum_{i\in V_1} b_i - \sum_{i\in S\cap V_1} b_i + \sum_{j\in S\cap V_2} f_j \\ &= B - [\sum_{i\in S\cap V_1} b_i - \sum_{j\in S\cap V_2} f_j] \\ &= B - NB(S). \end{aligned}$$

Since $B = \sum_{i=1}^{m} b_i$ is a constant, it follows that minimizing the cut capacity $C(S,T)$ is equivalent to maximizing the net benefits among all selections S.

For our λ-problem each set is an edge in the hypergraph e of benefit u_e and each item p is a facility of cost $c_p = \lambda f_p$. Figure 7 depicts a bipartite network parameterized by λ in which the solution to the minimum (s,t)-cut problem provides the answer to the λ-problem. We call this bipartite network the λ-network.

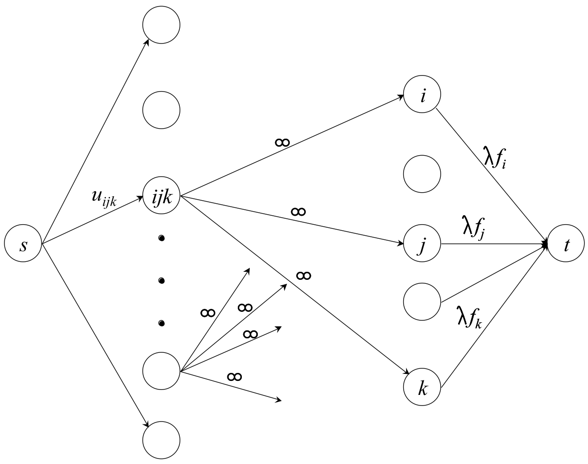

Figure 7 The bipartite λ-network for a prescribed value of λ set to solve the λ-problem.

When the value of λ in the λ-network is large enough, the minimum cut is $(\{s\}, V_1 \cup V_2 \cup \{t\})$ with all arcs adjacent to source in the cut. As the value of λ diminishes the source set of the corresponding minimum cut in the λ-network, S_λ, expands and contains all previous source sets. To make this statement more precise we let S_λ be the *minimal* source set of a minimum cut – that is, if there are multiple optimal solutions to the minimum cut problem, we select the one that is minimal in the sense that there is no other optimal source set contained in it. Formally, the statement of this fact is referred to as the *nestedness* lemma.

Lemma 4.2 *If* $\lambda_1 > \lambda_2$ *then* $S_{\lambda_1} \subseteq S_{\lambda_2}$.

This lemma is proved via the properties of specific algorithms for solving the minimum cut problem, and has also been known for a long time as part of the research "folklore". It was shown by Gallo et al. (1989), based on the properties of the push-relabel algorithm, and by Hochbaum (1998), based on the properties of the pseudoflow algorithm.

Because of the simple structure of the λ-network, the proof of the nestedness lemma for our case is relatively simple.

Proof: For a minimum cut (S, T) there is a corresponding maximum flow where the source set S includes all nodes reachable from s along arcs with positive residual capacity (capacity minus the value of the flow). The sink set T includes all nodes that have a residual path reaching t.

As the value of λ goes down only fewer nodes can reach the sink, whereas every node previously reachable from the source s is still reachable. Q.E.D.

We call the values of λ where the source set expands by at least one node, the *breakpoints* of the parametric cut. Let the breakpoints be $\lambda_1 > \lambda_2 > \ldots > \lambda_\ell$, with corresponding minimal source sets, $S_1 \subset S_2 \subset \ldots \subset S_\ell$. As a result of the nestedness lemma $\ell \leq n$ for a graph on n nodes since there can be no more than n different nested source sets. The capacity value of the minimum cut is increasing as a function of increasing values of λ along a piecewise linear concave curve which is the efficient frontier.

Theorem 4.1*All breakpoints of the density graph can be found by solving a parametric minimum cut problem where the sink adjacent capacities of arcs (i, t) are linear functions of the parameter, λf_i.*

Gallo et al. (1989) showed in how to find all the breakpoints and the corresponding minimum cuts in the same complexity as that required to solve a single minimum (s, t)-cut problem with the push-relabel algorithm. We tweak the analysis of complexity to achieve better results for the maximum benefit problem as described in Section 6. The pseudoflow algorithm for maximum flow and minimum cut (see Hochbaum (1998)) also finds the parametric breakpoints in the complexity of a single minimum (s, t)-cut. The pseudoflow algorithm has the further advantage of finding multiple minimum cut solutions – a feature that is needed here for generating *all* preferred configurations.

We will refer henceforth to the "source set" of a cut meaning the set of nodes in the source set excluding the source node s.

Example: We show how to generate, for each slope of the concave envelope for case (a) in Section 2, the sets of preferred configurations with benefit-cost ratio equal to this slope. Solving the minimum cut problem in the λ-network illustrated in Figure 8, for $\lambda = 1$, yields several optimal cuts with source sets, $\emptyset$, $\{1, 2, 3\}$ and $\{1, 2, 3, 4\}$. The minimum cut value $\sum_{ij \in E_H} w_{ij} - \lambda \sum_{j \in V_H} f_j$ for all these sets is equal to $5\frac{3}{4}$. Solving next for the value of $\lambda = \frac{7}{8}$ there are two optimal solutions (source sets): $\{1, 2, 3, 4\}$ and $\{1, 2, 3, 4, 5, 6\} = V$. Notice that in both these cases, the minimal source sets, $\emptyset$ for $\lambda = 1$ and $\{1, 2, 3, 4\}$ for $\lambda = \frac{7}{8}$, are breakpoints of the density graph. Solving the minimum cut problem in the λ network for $\lambda = \frac{15}{16}$ yields the unique solution $\{1, 2, 3, 4\}$ which is the only point on the concave envelope that this line is tangent to.

Similarly, solving case (b) for $\lambda = 1$ yields the source sets $\emptyset$, $\{1, 2, 3\}$, $\{1, 2, 3, 4\}$, $\{1, 2, 3, 5, 6\}$ and $\{1, 2, 3, 4, 5, 6\}$. Here it is evident that not all optimal source sets are nested as $\{1, 2, 3, 4\}$ is not contained in $\{1, 2, 3, 5, 6\}$.

The breakpoints are thus derived as the values where the minimal source sets change. Note that the parametric algorithm does not solve for one value at a time, as described in this example, but rather generates all the slopes and the breakpoints using a single efficient procedure. The other (non-breakpoint) optima are not necessarily nested. To find those we employ the pseudoflow algorithm, Hochbaum (1998), discussed next.

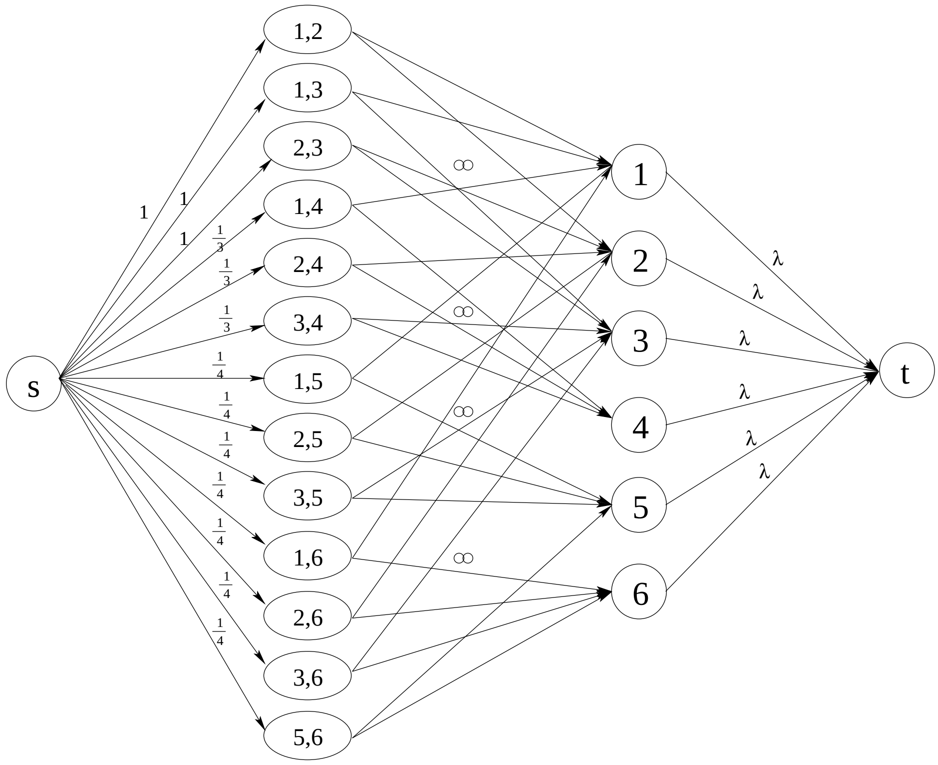

Figure 8 The λ-network for case (a).

4.1. Finding multiple optima for the minimum cut problem

We briefly sketch here how the pseudoflow algorithm identifies multiple optima. A pseudoflow is a flow that satisfies capacity constraints, but not the flow balance constraints specifying that for every node in V the total incoming flow is equal to the total outgoing flow. The pseudoflow algorithm works with a pseudoflow that saturates at all times all source adjacent arcs and all sink adjacent arcs. The algorithm maintains a construction called a "normalized tree" which is a collection of subsets of nodes, called branches. Some branches have positive excess (we call those the +sets); some have negative excess considered to be deficit (the −sets); and some are balanced, or have "0-deficit", (the 0sets). The +sets and any subcollection of the 0sets are called "strong" and the −sets along with the remaining 0sets are called weak. An iteration of the algorithm consists of a "merger" via a residual arc between a strong set and a weak set. The excess of a normalized tree is the sum of excesses of the +sets (or the strong nodes).

At termination, there are no residual arcs between the strong sets and the weak sets. If we choose for strong only + sets then there are no residual arcs between those and − or 0sets. The minimum cut then is (S,T) where S is the union of all +sets (and the source node s) and T is the complement set of nodes consisting of the −sets and 0sets (and the sink node t). The difference between the initial excess of the normalized tree and the excess at termination (also equal to the sum of excesses of S) is precisely equal to the value of the minimum cut and maximum flow (see Hochbaum (1998) for proof.)

The minimal source set of the minimum cut is formed of the +sets only. Alternative minimum cuts are formed when the strong sets consist of the +sets in the source set the −sets in the sink set and the 0sets partitioned between the source and the sink sets. Adding 0sets to the source set of +sets is maintains the minimum cut when there are no residual arcs between those 0sets and the remainder of the nodes in the sink set of the cut. Obviously all these partitions correspond to cuts of the same value – the minimum cut capacity, as the excess of the corresponding normalized tree is unchanged.

The process of generating the alternative optima is schematically described in Figure 9. Figure 9 (a) shows the status of the partition at the optimum where S is a minimal source set consisting only of +sets. The only residual arcs between the source and the sink are directed from sink to source sets.

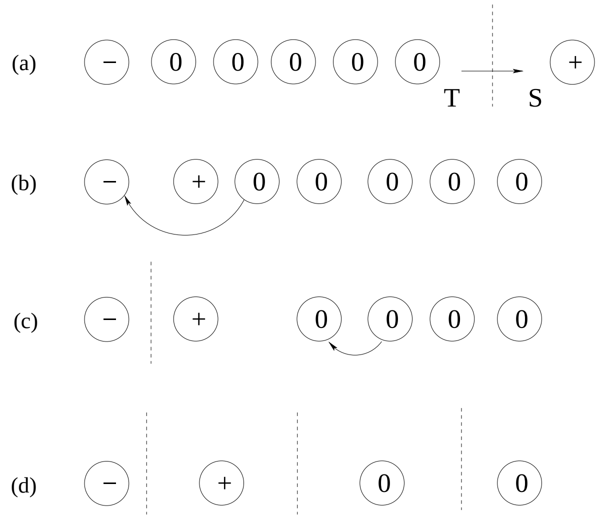

Figure 9 The stages of identifying all multiple optima.

To find the 0sets that can be added to the source set without changing the value of the cut, we first perform "0 to −" possible mergers between any 0sets to −sets if there is a residual arc going between them. This is shown in Figure 9 (b). Each merger of this type eliminates a 0set. All the remaining 0sets that do not have arcs to −sets are now candidates to be added to the source set. We next perform "0 to 0"mergers between 0sets and 0sets if there are residual arcs between them as in Figure 9 (c). This can cause the consolidation of some of those 0sets to larger 0sets. When this process terminates, there are some +sets, and some 0sets without outgoing residual arcs to any other 0 or −set. Any subcollection of those "isolated" 0sets can be added to the source set, while maintaining the validity of the cut and the resulting cut value remains the same. Any cut formed by the +sets and any subcollection of the 0sets in the source set is a minimum cut. Thus generating all possible multiple optima and all preferred configurations corresponding to the value of λ.

This extra work does not affect the complexity of the algorithm, $O(mn \log n)$, because the labels are maintained and the algorithm permits such mergers within the established complexity. The reader is referred to Hochbaum (1998) for additional details on generating multiple optima minimum cuts.

Suppose there are k 0sets that contain (facility, or V_2) nodes of cost values $B_1, \ldots, B_k$, then there are up to 2^k different solutions to the minimum cut problem. Let B^+ be the sum of costs of the nodes in the +sets. Finding a preferred configuration of specific budget or cost $\bar{B}$ is equivalent to solving the subset sum problem: Find $I \subseteq \{1, \ldots, k\}$ so that $\sum_{i \in I} B_i = \bar{B} - B^+$. The subset sum problem is weakly NP-hard (Karp (1972)), yet, enumerating all of the solutions takes time which is only proportional to their number.

5. Guidelines for using the efficient frontier to evolve a set of facilities

Once the efficient frontier has been fully evaluated, it is used as follows: A budget increment may be positioned at a breakpoint, in which one of the configurations optimal for that breakpoint can be selected. Suppose then that a budget increment is such that the total budget B falls between two breakpoints $B_j < B < B_{j+1}$. In that case the optimal facility configuration S_j corresponding B_j is selected, and then appended by a subset of $S_{j+q} \setminus S_j$. This added subset can be selected by a greedy algorithm, or, if there is another preferred configuration on the efficient frontier corresponding to budget not exceeding B (that is also contained in S_{j+1} then such configuration should be selected, and then possibly appended greedily by additional facilities if this preferred configuration requires budget strictly less than B.

Although this discussion ignores the present value discounting of the benefit stream, notice that it is impossible to generate higher benefits than any preferred configuration. The only flexibility in choosing configurations is in between breakpoints. That indeed can be done with a greedy approach that maximizes the present value of the benefits.

6. Analyzing and improving the complexity

The λ-network on which the parametric minimum cut is solved is an "unbalanced" bipartite graph. The term *unbalanced* refers to the relative sizes of the sets on each side of the bipartition. The set V_1 on one side represents the edges E in the original hypergraph, $G = (V, E)$, containing $|E| = m_1$ nodes. On the other side the set V_2 representing the set of nodes in G containing n nodes. For so-called "dense" graphs m_1 can be an order of magnitude larger than n. For a benefit maximization problem defined on a hypergraph $G = (V, E)$ the bipartite network set up for the selection problem has $O(|E| + |V|)$ nodes. The direct application of push-relabel algorithm has complexity of $O(m_1 n \log \frac{n^2}{m_1})$ and the complexity of pseudoflow is $O(m_1 n \log n)$. There are however improvements to this complexity exploiting the unbalanced property of the bipartite network.

The number of iterations required by the pseudoflow algorithm is bounded by a function of the length of the longest residual path in the graph – $O(m'n')$ – where m' is the number of arcs in the bipartite graph and n' is the maximum residual path length. In the λ-network this length n' is at most $2n + 2$ as each path alternates between the two sets in the partition. In our λ-network m' is equal to m, where m is the sum of sizes of the positive benefit edges in the original hypergraph, $\sum_{e \in E} |e|$. Each iteration of the pseudoflow algorithm takes $O(\log n)$ steps. Hence, the total complexity for finding all breakpoints is thus $O(mn \log n)$. For generating P additional preferred configurations, the total complexity is $O(mn \log n + P)$.

Ahuja et al. (1994) devised an improved push-relabel algorithms for unbalanced bipartite graphs. Among those, the most efficient for parametric minimum cut is an adaptation of the parametric push-relabel algorithm of Gallo, Grigoriadis and Tarjan with run time $O(m'n' \log(\frac{n'^2}{m'} + 2))$. This run time translates to $O(mn \log(\frac{n^2}{m} + 2))$ for the problem of generating all the breakpoints.

7. The maximum density subgraph problem

Finding a densest subgraph in a hypergraph $G = (V, E)$ differs from the densest k-subgraph problem in that the size of the subgraph is not specified. This seemingly minor distinction makes a significant difference to the complexity – whereas the densest k-subgraph is an NP-hard problem even for graphs, finding a densest subgraph is polynomially solvable.

Since for the first preferred configuration the density ratio is greater than for all configurations of larger budgets (or number of nodes) it follows that the maximum density subgraph is the smallest positive budget preferred configuration.

The maximum density subgraph problem can be solved by applying a parametric minimum cut procedure. For the weighted version of the maximum density subgraph problem defined on a hypergraph $G=(V,E)$ with node weights f_j and edge weights w_e, the problem is to find a subset of nodes $V_H \subseteq V$ and the induced subgraph $H=(V_H,E_H)$ with $E_H=E(H)$ that maximize,

$$\max_{H\subseteq G} \frac{\sum_{e\in E_H} w_e}{\sum_{j\in V_H} f_j}.$$

A general approach in maximizing any fractional (or as it is sometimes called, geometric) objective function is to reduce it to a sequence of calls to a procedure that finds the yes/no answer to the λ-*question*:
Is there a subset $V' \subseteq V$ such that $\sum_{[ij]\in E(V')} w_{ij} - \lambda \sum_{j\in D} f_j > 0$?

If the answer to the λ-question is *yes* then the optimal solution has value larger than λ. Otherwise, the optimal value is less than or equal to λ. A binary search procedure calls for the λ-question $O(\log(UF))$ times in order to solve the problem, where $U=\sum_{e\in E} w_e$, and $F=\sum_{j\in V} f_j$. The λ-question is identical to the λ-problem and can thus be solved more efficiently with a parametric minimum cut procedure.

The output of the parametric minimum cut procedure provides all the breakpoints, the first of which corresponds to the maximum density subgraph. So even though the generation of all the breakpoints is a more general problem than the maximum density subgraph, both can be solved with the same complexity.

8. Conclusions

We discuss here a problem of choosing sets of facilities with pairwise benefits subject to a budget restriction so as to maximize the total subsets benefits. The problem is studied in scenarios where budgets become available for facilities' set expansion over time. The optimal expansion strategy is to follow the nested preferred configurations on the efficient frontier. We show here that the entire efficient frontier is generated in polynomial time even though finding one specific optimal configuration for a given budget is an intractable problem. It is shown that in terms of complexity the problem is positioned between the NP-hard problem of maximum clique and the polynomially solvable problem of maximum density subgraph.

The generated information is also useful in providing upper bounds on the solution to the maximum clique and maximum dispersion problems. The efficient frontier provides an upper bound on the maximum benefit for a given budget and hence leads to an approximation algorithm that improves on the complexity of known approximation algorithms Hochbaum (2006). It also sheds light on the structure of the graph and the preferred configurations reflected in the sequence of breakpoints.

Acknowledgement

The author is grateful to Doug Shier for the reference of Witzgall and Saunders that motivated this research.

References

R. K. Ahuja, J. B. Orlin, C. Stein and R. E. Tarjan. Improved algorithms for bipartite network flow. *SIAM Journal of Computing* 1994, 23:5, 906–933.

M. L. Balinski. On a selection problem. *Management Science* 1970, 17, 230–S231.

F. Barahona. On the k-Cut Problem, *Operations Research Letters* 2000, 26:3, 99–105.

A. Feng, K. Venkatraman, J. Ward, B. Zhang. Managing product proliferation: a network flow approach. INFORMS Denver October 2004 presentation.

Gallo G., M. D. Grigoriadis, R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.* **18**(1), 1989, 30–55.

A. V. Goldberg, Finding a maximum density subgraph, UC Berkeley report UCB/CSD/84/171, 1984.

A. V. Goldberg and R. E. Tarjan. A New Approach to the Maximum Flow Problem. *Journal of the ACM*, 35, 1988, 921–940.

D. S. Hochbaum. The Pseudoflow algorithm: A new algorithm for the maximum flow problem. *Operations Research*, to appear. Extended abstract in: The pseudoflow algorithm and the pseudoflow-based simplex for the maximum flow problem. Procdeedings of IPCO98. *Lecture Notes in Computer Science*, Bixby, Boyd and Rios-Mercado (Eds.) 1412, Springer, 325-337, 1998.

D. S. Hochbaum. An efficient approximation algorithm for the k-densest subgraph problem. Manuscript, UC Berkeley, Feb 2006.

R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Communications*, 85–103. Plenum Press New York, 1972.

J. C. Picard, Maximal Closure of a Graph and Applications to Combinatorial Problems, *Management Science* 1976, 22,1268–1272.

R. Ravi and A. Sinha. Approximating k-cuts via network strength. *Proceedings of SODA* 2002, 621–622.

J. Rhys. Shared fixed costs and network flows. *Management Science* 1970, 17, 200–207.

D. D. Witzgall and R. E. Saunders. Electronic mail and the "Locator's" dilemma. In *Applications of Discrete Mathematics*, R.D. Ringeisen and F.S. Roberts eds. SIAM 65–84, 1988.

B. Zhang, J. Ward and Q. Feng. A Simultaneous Maximum Flow Algorithm for the Selection Model. HPL-2005-91.