Simultaneous Selection

Wojciech Olszewski* and Rakesh Vohra†

June 3, 2014

Abstract

We generalize Chade and Smith's (2006) simultaneous search problem to a class of discrete optimization problems. More precisely, we study the problem of maximizing a weighted sum of utilities of objects minus the sum of costs of acquiring these objects, given the constraint that the sum of weights cannot exceed the value of some submodular function.

We show that the problem has a simple solutions in the particular case in which the submodular function depends only on the number of objects. Namely, the optimal set of objects can be found by the steepest ascent algorithm. We provide some economic applications of this result. The particular case studied in the present paper, and the particular case studied by Chade and Smith complement one another, but they do not exhaust all instances of our general discrete optimization problem. We also show that in the general case the problem does not have a simple solution.

*Department of Economics, Northwestern University, 2001 Sheridan Rd. Evanston IL 60208-2600

†Department of Managerial Economics and Decision Science, Kellogg School of Management, Northwestern University, 2001 Sheridan Rd. Evanston IL 60208-2600

1 Introduction

Chade and Smith (2006) introduce an interesting discrete optimization problem involving a high-school student applying to colleges. The student must choose a subset of colleges. Each application costs a $c > 0$. Studying at different colleges gives different utilities, and the probability of being admitted also varies across colleges. From the set of colleges to which the student gets admitted, she chooses the one with highest utility. Chade and Smith point out that the threshold strategy of applying to colleges which give more than a certain utility may not be optimal, and they show that the optimum is attained by a simple *steepest ascent* algorithm.[1]

According to this algorithm, we first put on a list the college we would apply to, if we were allowed to apply to one college only. Then, we add to this list the college that we would apply to, if we were allowed to apply to one more college. We keep adding colleges to the list until the marginal benefit of adding any other college falls below the cost c.

In this paper we are concerned with a more general problem of maximizing a weighted sum of utilities of objects minus the sum of costs of acquiring these objects, given the constraint that the sum of weights assigned to any subset of objects S cannot exceed $f(S)$, where f is a non-decreasing, non-negative submodular function. Call this more general problem the *simultaneous selection problem*, or (SSP) in abbreviation.

This generalizes Chade and Smith's problem in two ways. It allows for different objects having different costs. In addition, in Chade and Smith's problem $f(S)$ has a very special form, which is the probability of being admitted by at least one college from S. Both ways in which Chade and Smith's problem is generalized, allow for new applications. In searching for a job instead of applying to colleges, the application process typically varies across employers. Other functions f also enlarge the set of applications. For example, imagine an inventory planning problem, in which you must order today some goods, which you will later be selling gradually, one per period. In this application, assuming geometric discounting, $f(S)$ is the present value of the stream of 1's over $|S|$ periods.

We show that the (SSP) can be solved by the steepest ascent algorithm in the case in which $f(S)$ depends on $|S|$ only. In particular, steepest ascent selects an optimal set in some versions of the applications described in the previous paragraph. Our result and that obtained by Chade and Smith complement one another. Neither subsumes the other, and they also apply to somewhat different settings.

[1] It should be noted that the optimality of steepest ascent for the college application problem is not the only result in Chade and Smith (see Section 2.1).

In the absence of additional assumptions on the function f, the (SSP) is NP-hard. An interesting open question is the design of good approximation algorithms for the (SSP). We show by an example that for any $\varepsilon > 0$, steepest ascent may select a set S that returns less than an ε share of the optimum.

2 Simultaneous selection problem

Let f be a nonnegative function defined on all subsets of a finite set N. The family of subsets of N is partially ordered by inclusion. This ordering is a lattice with the union of sets being their join, and the intersection being their meet. We will assume that f is non-decreasing, i.e.,

$$f(S) \leq f(T), \quad \forall_{S \subset T \subset N},$$

and submodular, i.e.,

$$f(S \cap T) + f(S \cup T) \leq f(S) + f(T), \quad \forall_{S,T \subset N}.$$

Given numbers (utilities) $w_i \geq 0\,, \forall i \in N$, define the function g on each $T \subseteq N$ by the following discrete optimization problem:

$$g(T) = \max \sum_{i \in T} w_i x_i$$

$$\sum_{i \in S} x_i \leq f(S), \quad \forall S \subseteq T$$

$$x_i \geq 0, \quad \forall i \in T$$

This function can be interpreted the maximum utility that one can obtain by assigning weights x_i to the elements of T, subject to the "resource" constraint that the aggregate weight assigned to any set S cannot exceed $f(S)$.

The submodularity of f guarantees that the optimization problem that defines function g has the following solution. Label the elements of T as $\{1, 2, \ldots, |T|\}$, and order them so that $w_1 \geq w_2 \geq \ldots w_{|T|}$. Then, it is optimal to assign the highest possible weight, $x_1 = f(1)$, to the element with the highest utility, that is, to element 1; and next, to assign the highest possible weight subject to the resource constraint, $x_2 = f(1,2) - f(1)$, to the element with the second-highest utility, that is, to element 2; etc., until assigning weight $x_{|T|} = f(1, 2, \ldots, |T|) - f(1, 2, \ldots, |T| - 1)$ to element $|T|$.[2]

In what follows, we denote by $x(T)$ the vector of optimal weights assigned to set $T \subseteq N$. Hence,

$$g(T) = \sum_{i \in T} w_i x_i(T).$$

[2] Dunstan and Wels (1973)

It is immediate that function g, defined on all subsets of N, is submodular.

Given numbers (costs) $c_i \geq 0, \forall i \in N$, we define a problem called the simultaneous selection problem (in short, SSP) as

$$\max_{T \subseteq N} \left\{ g(T) - \sum_{i \in T} c_i \right\}. \tag{SSP}$$

The number c_i can be interpreted as the cost of including element i to set T. For convenience, set

$$H(T) = g(T) - \sum_{i \in T} c_i.$$

In the following sections, we demonstrate that the (SSP) covers various, basic and seemingly unrelated applications.

2.1 Chade and Smith's college application problem

While we focus on high-school students deciding which colleges to apply to, one can imagine numerous other application processes. Let w_i be the utility of being admitted to college $i \in N$. Let α_i be the probability that one is admitted to college i, given one has applied to this college. Finally, let c_i be the cost of applying to college i. Suppose one has applied to the set $S \subset N$ of colleges. Some of these will admit you, and from these you select the one with largest utility. Denote by $g(S)$ the expected utility of applying to the colleges in S. Order the colleges in S by their decreasing utility w_i, i.e., $w_1 \geq w_2 \geq \ldots \geq w_{|S|}$. Then, the probability of j being the highest-utility college that admits you is

$$\alpha_j \Pi_{i=0}^{j-1}(1 - \alpha_j),$$

where $\alpha_0 = 0$, and the expected utility of applying to the colleges in S (disregarding the application costs) is

$$g(S) = \sum_{j=1}^{|S|} w_j \alpha_j \Pi_{i=0}^{j-1}(1 - \alpha_j).$$

Chade and Smith (2006) are concerned with the solution to the following problem:

$$\max_{S \subseteq N} \left[g(S) - \sum_{i \in S} c_i \right].$$

To show that this problem is an instance of the (SSP), let

$$f(T) = 1 - \Pi_{j \in T}(1 - \alpha_j).$$

It is easy to see that f is submodular; moreover, f satisfies, which is also easy to see, an additional condition called by Chade and Smith *downward recursive*: For $U, L \subset N$, write $U \succ L$ when

$$\min_{i \in U} w_i \geq \max_{i \in L} w_i.$$

Then, for any disjoint sets U and L such that $U \succ L$, we have

$$f(U \cup L) = f(U) + f(L)\Pi_{i \in U}(1 - \alpha_i).$$

It follows that

$$g(S) = \max \sum_{i \in S} w_i x_i$$

$$\sum_{i \in T} x_i \leq f(T) \ \forall T \subseteq S$$

The result in Chade and Smith is that this instance of the (SSP) can be solved by steepest ascent when all c_i's are the same.

Remark It should be noted that the optimality of steepest ascent for the problem described in this section is not the only result in Chade and Smith. They study a more general problem, and establish several important properties of the optimal set. In particular, Chade and Smith are motivated not only by solving the 'high-school student problem,' but also by comparing the sets S chosen in simultaneous and in sequential search.

2.2 Inventory planning

This application concerns inventory planning. Let c_i be the cost of ordering one unit of good i, and $\delta^t w_i$ be the discounted profit from holding that unit in inventory until period t and then selling it. Suppose the demand is a constant of one unit per period, and one is to order at period $t = 0$ a set of goods $S \subset N$, which will be sold one per period, in periods $t = 1, 2, ..., |S|$. Denote by $g(S)$ the utility of having the goods from set S.

Order the goods in S by decreasing profit, i.e., $w_1 \geq w_2 \geq \ldots \geq w_{|S|}$. Of course, you prefer to sell earlier goods that give you a higher profit. Thus,

$$g(S) = \sum_{t=1}^{|S|} \delta^t w_t,$$

or equivalently,

$$g(S) = \max \sum_{i \in S} w_i x_i$$

$$\sum_{i \in T} x_i \leq f(T), \ \ \forall T \subseteq S,$$

where

$$f(T) = \sum_{t=1}^{|T|} \delta^t.$$

It is easy to see that f is submodular.

Then, the problem of determining an optimal order is an instance of the (SSP):

$$\max_{S \subseteq N} \left[g(S) - \sum_{i \in S} c_i \right].$$

Another interpretation of this instance of the (SSP) is in terms of purchasing at period $t = 0$ a set of goods $S \subset N$, whose costs of purchasing are c_i and the utilities of consuming then at time t are $\delta^t w_i$. These goods are to be consumed one per period, in periods $t = 1, 2, ..., |S|$.

Our Theorem 1 implies that this instance of the (SSP) can also be solved by steepest ascent.

2.3 Project selection and sequencing

Here is another example of a problem that fits the Chade and Smith setting, and so is an instance of the (SSP). This problem was studied by Aspvall et al. (1995). Suppose we have a set of N projects. Project i has a duration t_i. If project i is initiated you earn V_i at the start of the project.[3] If project i is initiated after t_i time periods, the net present value (in short, NPV) of the project is $\delta_i V_i = (1 + r)^{-t_i} V_i$, where r is the discount rate. The cost of securing project i is c_i. You can only do one project at a time. Thus, your objective is to identify which subset of projects to secure and how to sequence them to maximize profits.

Suppose one has acquired the set $S \subset N$. As shown by Aspvall et al. (1995), one will sequence the projects by decreasing $V_i/(1 - \delta_i)$. If we order the elements of S by decreasing $V_i/(1 - \delta_i)$, the NPV of revenues from securing the set S is

$$g(S) = V_1 + \delta_1 V_2 + \delta_1 \delta_2 V_3 + \ldots + \Pi_{i<|S|} \delta_i V_{|S|}.$$

We are concerned with the solution to the following problem:

$$\max_{S \subseteq N} \left[g(S) - \sum_{i \in S} c_i \right].$$

[3] It would not make any difference if the profit was realized at the end of the project.

To show that this problem is an instance of the (SSP), let

$$f(T) = 1 - \Pi_{t \in T} \delta_t.$$

Then,

$$g(S) = \max \sum_{i \in S} V_i y_i$$

$$\sum_{i \in T} (1 - \delta_i) y_i \leq f(T) \; \forall T \subseteq S$$

By changing variables: $x_i = (1 - \delta_i) y_i$, we obtain that

$$g(S) = \max \sum_{i \in S} \frac{V_i}{1 - \delta_i} x_i$$

$$\sum_{i \in T} x_i \leq f(T) \; \forall T \subseteq S$$

Notice f is downward recursive for $w_i = V_i/(1 - \delta_i)$.

Here is another interpretation of the same problem. This version of the problem, for a given set of jobs, was studied by Stadje (1995). We have one machine and a set N of jobs. Each job has a processing time t_i and upon completion generates a reward V_i, discounted off course from the start. However, there is a possibility that when a job i is being processed, on the single machine, the machine may fail. When it fails, it cannot be repaired. The probability of the machine failing while processing job i is p_i. As shown by Stadje, given any set of jobs to be processed, it is optimal to process them by decreasing value of $w_i = p_i V_i / (1 - p_i e^{-r t_i})$.

The result in Chade and Smith implies that this instance of the (SSP) can also be solved by steepest ascent when all c_i's are the same.

2.4 Applying for jobs, and assigning time slots for tasks

The objective of this section is: (1) to provide other applications of our Theorem 1, and (2) to emphasize the fact that our result and that from Chade and Smith's complement one another.

Consider a version of the Chade and Smith problem from Section 2.1, in which an agent is searching for a job. Suppose the agent has a personal ranking of jobs, and the application cost varies across jobs. Suppose, in addition, that the agent assigns the same probability of any employer having an opening, for example, because this depends on factors unobserved by an outsider.

This setting is a version of Chade and Smith problem with w_i being the utility of being offered a job by employer $i \in N$, $\alpha_i = \alpha$ for all i's, being the probability of being offered a job by employer i, and c_i being the cost of applying to employer i.

Due to the assumption that $\alpha_i = \alpha$ for all i's, we have that

$$f(T) = 1 - (1-\alpha)^{|T|}$$

depends on $|T|$ only. Thus, this instance of the (SSP) can be solved by steepest ascent by our Theorem 1 for any configuration of c_i's.

Consider now a problem similar to that from Section 2.3, in which an agent has to schedule tasks. Assume that the time required for the completion of each task is uncertain, and perhaps because of the involvement of other parties, each task scheduled for completion has to receive a time slot of a given (fixed) length. This actually happens in several industries.

If the time slot for each task is of length t, task i once initiated is earning V_i, and the cost of securing task i is c_i, then this instance of the (SSP) coincides with that studied in Section 2.2, where

$$f(T) = \sum_{t=1}^{|T|} \delta^t, \ \delta = (1+r)^{-t}.$$

Again, our Theorem 1 implies that this instance of the (SSP) can be solved by steepest ascent for any configuration of c_i's. That is, our Theorem 1 assumes that all t_i's are the same, but allows for arbitrary c_i's.

3 Optimality of steepest ascent in special cases

We demonstrate in this section that a broad class of instances of the (SSP) can be solved by a polynomial-time algorithm; moreover, it can be solved by a particularly simple algorithm, called *steepest ascent*:

According to this algorithm, we first choose an element $i \in N$ that maximizes $g(i) - c_i$, call it i_1. This is the element that we would choose, if we could pick only one element of N. Then, we choose an element $i \in N \setminus \{i_1\}$ that maximizes $g(i_1, i) - g(i_1) - c_i$ and call it i_2. This is the element we would choose if we could add only one element to i_1, or the element with the highest value added. Next, we choose $i \in N \setminus \{i_1, i_2\}$ that maximizes $g(i_1, i_2, i) - g(i_1, i_2) - c_i$, and so on. We stop when the maximum possible value added is negative.

We will now show that:

Theorem *If $f(S)$ depends on $|S|$ only, then the (SSP) can be solved by steepest ascent.*

The formal proof is somewhat involved. However, one can build some basic intuition by means of a simple example. Consider the problem described in Section 2.2. Suppose steepest ascent selects just one element, call it 1. Let $S^* = \{1\}$. By the definition of steepest ascent, $H(S^*)$ is higher than $H(S)$ for any set S consisting either of a singleton or of two elements one of which being 1.

Consider any set S consisting of two elements none of which being 1. Call the two elements 2 and 3. Suppose that $w_2 \geq w_3$. We will show that $H(S) \leq H(S')$ for $S' = \{1, 3\}$, which will imply that $H(S) \leq H(S^*)$.

Case 1: $w_2 \leq w_1$

In this case, $w_3 \leq w_2 \leq w_1$, and so adding element 3 to each set $\{1\}$ or $\{2\}$ changes the objective function by $\delta^2 w_3 - c_3$. And since $H(\{1\}) \geq H(\{2\})$, we have that $H(S) \leq H(S')$.

Case 2: $w_2 > w_1$

If $w_3 \leq w_1$, then the same argument as in Case 1 applies. Suppose therefore that $w_1 < w_3 \leq w_2$. Then adding element 3 to each set $\{1\}$ changes the objective function by $\delta w_3 - c_3 - (\delta w_1 - \delta^2 w_1)$, while adding element 3 to each set $\{2\}$ changes the objective function again by $\delta^2 w_3 - c_3$. However, the latter expression is smaller than the former, because $w_1 < w_3$.

Proof. We will write $f(S)$ as $f(|S|)$ to emphasize the dependence on $|S|$ only. The optimal set of weights $x(T)$ assigned to set T is

$$x_i(T) = f(|\{j \in T : w_j > w_i\}| + 1) - f(|\{j \in T : w_j > w_i\}|). \tag{1}$$

For simplicity, we have assumed that $w_j \neq w_i$ for $i \neq j$. Order the elements of N by the order in which they are selected by steepest ascent. Suppose S is optimal for (SSP), and has $i^\#$ elements. Take $S^\# = \{1, 2, \ldots, i^\#\}$ comprising the first $i^\#$ elements picked by steepest ascent.

Hence, by the definition of steepest ascent, if $i^\# < i^*$, that is, $S^\#$ has fewer elements than the set picked by steepest ascent, we have that $H(S^*) \geq H(S^\#)$. We also have that $H(S^*) \geq H(S^\#)$ if $i^\# > i^*$ by the definition of steepest ascent, and the fact that function g is submodular. We will show that $H(S^\#) \geq H(S)$.

If $S \neq S^\#$, then there exists a j with $j \in S^\# \setminus S$. Choose the smallest j with this property. Next, choose $k > j$ such that $k \in S$ and w_k is the largest amongst all such numbers k. Let $S' = S \cup \{j\} \setminus \{k\}$. We will show that $H(S') \geq H(S)$. By iterating the same argument, we obtain that $H(S^\#) \geq H(S)$.

We can represent set S as $\{1, ..., j-1, k\} \cup S''$, and set S' as $\{1, ..., j\} \cup S''$, where $S'' \cap \{1, ..., j, k\} = \varnothing$. We will be progressively adding the elements of S'' to sets $\{1, ..., j-1, k\}$ and $\{1, ..., j\}$, and simultaneously showing that the value of H at the former set is no higher than that at the latter set.

By steepest ascent, we have that $H(\{1,...,j-1,k\}) \leq H(\{1,...,j\})$. Suppose $H(\{1,...,j-1,k\} \cup T) \leq H(\{1,...,j\} \cup T)$ for some proper subset $T \subset S''$, and take an $l \in S'' \setminus T$.

Case 1: $w_l \leq w_j$

In this case, $w_l \leq w_j, w_k$, and so adding element l to set $\{1,...,j-1\} \cup \{k\} \cup T$ and to set $\{1,...,j\} \cup T$ makes the same difference in the objective function, i.e.,

$$H(\{1,...,j-1,k\} \cup T \cup \{l\}) - H(\{1,...,j-1,k\} \cup T) =$$

$$= H(\{1,...,j\} \cup T \cup \{l\}) - H(\{1,...,j\} \cup T). \tag{2}$$

Indeed, let $i^1,...,i^q$ be all elements $i \in \{1,...,j-1,k\} \cup T$, and so all elements $i \in \{1,...,j\} \cup T$, such that $w_i < w_l$, ordered so that $w_{i^1} \geq ... \geq w_{i^q}$. Let p be the number of elements $i \in |\{1,...,j-1,k\} \cup T|$, and so the number of elements of $i \in |\{1,...,j\} \cup T|$, such that $w_i \geq w_l$. Then, by (1), both differences in (2) are equal to

$$-c_l + [f(p+1) - f(p)]w_l - \sum_{\lambda=1}^{q} [2f(p+\lambda) - f(p+\lambda-1) - f(p+\lambda+1)]w_{i^\lambda}.$$

Since

$$H(\{1,...,j-1,k\} \cup T) \leq H(\{1,...,j\} \cup T)$$

by inductive assumption, we have that

$$H(\{1,...,j-1,k\} \cup T \cup \{l\}) \leq H(\{1,...,j\} \cup T \cup \{l\}).$$

Case 2: $w_l > w_j$

In this case, $w_j < w_l \leq w_k$. Let $i^1,...,i^q$ be all elements $i \in \{1,...,j-1,k\} \cup T$, and so all elements $i \in \{1,...,j\} \cup T$, such that $w_j < w_i \leq w_l$, ordered so that $w_{i^1} \geq ... \geq w_{i^q}$, and let $i^{q+1},...,i^r$ be all elements $i \in \{1,...,j-1,k\} \cup T$, and so all elements $i \in \{1,...,j\} \cup T$, such that $w_i \leq w_j$, ordered so that $w_{i^{q+1}} \geq ... \geq w_{i^r}$. Recall that we assume that w_i's are different for different i's. Finally, let p be the number of elements i of $\{1,...,j-1,k\} \cup T$ such that $w_l < w_i$; it follows that $p-1$ is the number of elements i of $\{1,...,j\} \cup T$ such that $w_l < w_i$, because k is in the former set but not in the latter set.

Then

$$H(\{1,...,j-1,k\} \cup T \cup \{l\}) - H(\{1,...,j-1,k\} \cup T) = -c_l + [f(p+1) - f(p)]w_l$$

$$-\sum_{\lambda=1}^{q}[2f(p+\lambda)-f(p+\lambda-1)-f(p+\lambda+1)]w_{i^\lambda}$$

$$-\sum_{\lambda=q+1}^{r}[2f(p+\lambda)-f(p+\lambda-1)-f(p+\lambda+1)]w_{i^\lambda}.$$

In turn,

$$H(\{1,...,j\}\cup T\cup\{l\})-H(\{1,...,j\}\cup T)=-c_l+[f(p)-f(p-1)]w_l$$

$$-\sum_{\lambda=1}^{q}[2f(p+\lambda-1)-f(p+\lambda-2)-f(p+\lambda)]w_{i^\lambda}$$

$$-[2f(p+q)-f(p+q-1)-f(p+q+1)]w_j$$

$$-\sum_{\lambda=q+1}^{r}[2f(p+\lambda)-f(p+\lambda-1)-f(p+\lambda+1)]w_{i^\lambda}.$$

Thus, the difference between the two expressions is

$$+\sum_{\lambda=1}^{q}[2f(p+\lambda-1)-f(p+\lambda-2)-f(p+\lambda)]w_{i^\lambda}$$

$$+[2f(p+q)-f(p+q-1)-f(p+q+1)]w_j$$

$$-[2f(p)-f(p-1)-f(p+1)]w_l$$

$$-\sum_{\lambda=1}^{q}[2f(p+\lambda)-f(p+\lambda-1)-f(p+\lambda+1)]w_{i^\lambda}.$$

We will show that this difference is nonpositive, which will complete the proof since $H(\{1,...,j-1,k\}\cup T)\leq H(\{1,...,j\}\cup T)$ by inductive assumption.

Indeed,

$$[2f(p)-f(p-1)-f(p+1)]w_{i^1}\leq[2f(p)-f(p-1)-f(p+1)]w_l,$$

because $2f(p)-f(p-1)-f(p+1)\geq 0$ by submodularity, and $w_{i^1}\leq w_l$ by assumption. That is, the first component of the difference with '+' sign is no higher than the first component of the difference with '-' sign. Similarly,

$$[2f(p+1)-f(p)-f(p+2)]w_{i^2}\leq[2f(p+1)-f(p)-f(p+2)]w_{i^1},$$

that is, the second component of the difference with '+' sign is no higher than the second component of the difference with '-' sign. Continuing in this manner, we obtain that all subsequent components with '+' sign are no higher than the corresponding components with '-' sign, until

$$[2f(p+q) - f(p+q-1) - f(p+q+1)]w_j \leq [2f(p+q) - f(p+q-1) - f(p+q+1)]w_{i^q},$$

which completes the proof.

Remark (a) Our proof yields the result also in the case when the cost $c(T)$ of acquiring set T is a supermodular function that depends only on $|T|$, instead of being the sum of costs c_i across all elements of set T, which case has also been studied by Chade and Smith.

(b) The proof of our result and that from Chade and Smith have a number of arguments in common, but their proof cannot be easily modified to yield our result. For example, they mention (in Section 3D of their paper) the application we study in Section 2.2. However, they were able to solve this application only when $c(T)$ is a supermodular function that depends on $|T|$.[4]

(c) Our proof also applies to following cardinality constrained variant of SSP: $\max\{g(S)|S \subseteq N,\ |S| \leq K\}$.

4 NP-hardness of SSP

To show that SSP is NP-hard it suffices to show how to represent a known NP-hard problem as an instance of SSP. The known NP-hard problem we use is $\max_{S\subseteq N} f(S) - \sum_{i\in S} c_i$ where f is a monotone submodular function (see Feige and Vondrak (2010)). Choose $w_i = 1$ and $c_i = 0$ for all $i = 1, ..., n$ and set for each $T \subseteq N$,

$$g(T) = \max \sum_{i\in T} x_i$$
$$\sum_{i\in S} x_i \leq f(S)\ \forall S \subseteq T$$
$$x_i \geq 0\ \forall i \in T$$

Obviously, we see that $g(T) = f(T)$. Thus, the solution to the following instance of SSP

$$\max_{T\subseteq N} \left\{ g(T) - \sum_{i\in T} c_i \right\}.$$

is a solution to our original optimization problem.

[4]It is, however, important to emphasize that their general result allows f to depend on the composition of the set S, not just the number of objects in S. Thus, our result is more general than theirs only in this particular instance of the (SSP).

5 Suboptimality of steepest ascent in general case

In this section, we provide an example of the (SSP) in which steepest ascent returns an extremely poor solution. More precisely, for any $\varepsilon > 0$ steepest ascent selects a set S that gives less than an ε share of the optimum. This will be a special case of Chade and Smith's problem with costs varying across colleges. Chade and Smith have announced in the working paper version of their article (see Section 6B) that steepest ascent may be suboptimal for sufficiently disparate costs c_i.

Suppose the set $N = \{1, 2, ..., n\}$, $n \geq 2$, consists of:

- one college with $\alpha_1 = 0.1$, $w_1 = 10n$, and $c_1 = (n-1) + 0.9$;
- one college with $\alpha_2 = 0.5$, $w_2 = 2n$, and $c_2 = (n-1) + 0.89$;
- $(n-2)$ colleges with $\alpha_i = (1/n) \cdot 0.1$, $w_i = 2n$, and $c_i = 0.11$, $i = 3, ..., n$.

The steepest ascent algorithm selects college 2 first, which yields the payoff of 0.11, and other colleges yield the payoff of 0.1 and 0.09. Then, the algorithm stops, since the set of colleges $\{2, 1\}$ yields the payoff of

$$0.11 - 0.1 \cdot (n-1) < 0.11,$$

and the set of colleges $\{2, i\}$, $i > 2$, yields the payoff of 0.1.

Consider now the sets of colleges of the form $\{1\} \cup \{3, 4, ..., k+1\}$. Such a set yields the payoff of

$$0.1 + \sum_{i=1}^{k-1} \left[0.9 \cdot \left(1 - \frac{0.1}{n}\right)^{i-1} \cdot 0.2 - 0.11 \right]. \tag{3}$$

Suppose that k satisfies the condition

$$0.9 \cdot \left(1 - \frac{0.1}{n}\right)^{k-2} \cdot 0.2 - 0.11 \geq 0.05,$$

i.e.,

$$k \leq 2 + \frac{\ln \frac{0.16}{0.18}}{\ln \left(1 - \frac{0.1}{n}\right)}. \tag{4}$$

Then (3) exceeds $0.1 + 0.5 \cdot (k-1)$, which tends to ∞ for n tending to ∞ and k being the largest integer satisfying inequality (4).

5.1 Approximation bounds

The example above shows that steepest ascent cannot approximate the optimal objective function value of the (SSP) to within a constant factor. Steepest ascent, however, is known to deliver a constant factor approximation for certain cardinality constrained problems. For example, steepest ascent applied to maximizing a monotone, nonnegative valued submodular function subject to a cardinality constraint is guaranteed to deliver a solution with objective function value within $(1-1/e)$ of the optimal (see Cornuejols, Fisher and Nemhauser (1977)). For nonnegative valued submodular functions (not necessarily monotone) Buchbinder, Feldman, Naor and Schwartz (2012) exhibit a modification of the steepest ascent algorithm that yields a solution within a factor of half of the optimal.[5] This result does not apply to the (SSP) because the objective function, $H(S)$, while submodular can be negative. Finally, Cornuejols, Fisher and Nemhauser (1977) as well as Feige, Immorlica, Mirrokni, and Nazerzadeh (2009), under a different notion of approximation, analyze the accuracy of steepest ascent for a particular instance of the (SSP) called the Uncapacitated Facility Location problem.

6 References

1. Aspvall, B.; S.D. Flam h & K. P. Villanger, 1995. "Selecting among scheduled projects," **Operations Research Letters**, 17, 37-40.

2. Chade, H. & L. Smith, 2006. "Simultaneous Search," **Econometrica**, 74, 1293-1307.

3. Buchbinder, N.; M. Feldman, J. Naor & R. Schwartz, 2012. "A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization," **Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science**, 649-658.

4. Cornuejols, G.; M. L. Fisher & G. L. Nemhauser, 1977. "Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms," **Management Science**, 23, 789-810.

5. Feige U. and J. Vondrak, 2010. "The Submodular Welfare Problem with Demand Queries," **Theory of Computing** , 6, 247-290.

6. Dunstan, F. and D. J. A. Welsh, 1973. A greedy algorithm for solving a certain class of linear programmes, " **Mathematical Programming**, Volume 5, Number 1, Page 338-353.

[5]This result is in a sense best possible, see Feige, Mirrokni and Vondrak (2007).