

# Submodular Optimization with Routing Constraints

**Haifeng Zhang and Yevgeniy Vorobeychik**

Electrical Engineering and Computer Science

Vanderbilt University

Nashville, TN

{haifeng.zhang, yevgeniy.vorobeychik}@vanderbilt.edu

## Abstract

Submodular optimization, particularly under cardinality or cost constraints, has received considerable attention, stemming from its breadth of application, ranging from sensor placement to targeted marketing. However, the constraints faced in many real domains are more complex. We investigate an important and very general class of problems of maximizing a submodular function subject to general cost constraints, especially focusing on costs coming from route planning. Canonical problems that motivate our framework include mobile robotic sensing, and door-to-door marketing. We propose a generalized cost-benefit (GCB) greedy algorithm for our problem, and prove bi-criterion approximation guarantees under significantly weaker assumptions than those in related literature. Experimental evaluation on realistic mobile sensing and door-to-door marketing problems, as well as using simulated networks, show that our algorithm achieves significantly higher utility than state-of-the-art alternatives, and has either lower or competitive running time.

## Introduction

There has been much work on submodular maximization with cardinality constraints (Nemhauser, Wolsey, and Fisher 1978) and additive/modular constraints (Khuller, Moss, and Naor 1999; Sviridenko 2004; Krause and Guestrin 2005; Leskovec et al. 2007). In many applications, however, cost constraints are significantly more complex. For example, in mobile robotic sensing domains, the robot must not only choose where to take measurements, but to plan a route among measurement locations, where costs can reflect battery life. As another example, door-to-door marketing campaigns involve not only the decision about which households to target, but the best route among them, and the constraint reflects the total time the entire effort takes (coming from work schedule constraints). Unlike the typical additive cost constraints, such route planning constraints are themselves NP-Hard to evaluate, necessitating approximation in practice.

We tackle the problem of maximizing a submodular function subject to a general cost constraint,  $c(S) \leq B$ , where

$c(S)$  is the optimal cost of covering a set  $S$  (for example, by a walk through a graph that passes all nodes in  $S$ ). We propose a generalized cost-benefit greedy algorithm, which adds elements in order of marginal benefit per unit marginal cost. A key challenge is that computing (marginal) cost of adding an element (such as computing the increased cost of a walk when another node is added to a set) is often itself a hard problem. We therefore relax the algorithm to use a polynomial-time approximation algorithm for computing marginal cost. We then show that when the cost function is approximately submodular, we can achieve a bi-criterion approximation guarantee using this modified algorithm, which runs in polynomial time. To our knowledge, this offers the most generally applicable theoretical guarantee in our domain known to date.

Our experiments consider two applications: mobile robotic sensors and door-to-door marketing. In the former, we use sensor data on air quality in Beijing, China collected from 36 air quality monitoring stations, with a hypothetical tree-structured routing network among them. The objective in this case is to minimize conditional entropy of unobserved locations, given a Gaussian Process model of joint sensor measurements. In the door-to-door marketing domain, we use rooftop solar adoptions from San Diego county as an example, considering geographic proximity as a social influence network and the actual road network as the routing network. In both these domains, we show that the proposed algorithm significantly outperforms competition, both in terms of achieved utility, and, often, in terms of running time. Remarkably, this is true even in cases where the assumptions in our theoretical guarantees do not meaningfully hold.

In summary, this paper makes the following contributions:

1. a formulation of submodular maximization under general cost constraints (routing constraints are of particular interest);
2. a novel polynomial-time generalized cost-benefit algorithm with provable approximation guarantees;
3. an application of our algorithm to two motivating real-world optimization problems, mobile robotic sensing and door-to-door marketing, illustrating that our algorithm significantly outperforms state of the art.

## Related Work

Submodular optimization has received much attention due to its breadth of applicability, with applications including viral marketing, information gathering, image segmentation, and document summarization (Fujishige 2005; Krause and Golovin 2012). A number of efforts consider submodular optimization under cardinality or additive cost constraints (Nemhauser, Wolsey, and Fisher 1978; Khuller, Moss, and Naor 1999; Sviridenko 2004; Krause and Guestrin 2005; Leskovec et al. 2007), demonstrating the theoretical and practical effectiveness of simple greedy and cost-benefit algorithms in this context. The problem of minimizing travel cost to cover a set of nodes on a graph, which gives rise to our constraints, is a variant of the Traveling Salesman Problem (TSP), although in our variations we allow visiting the same nodes multiple times (this variation is sometimes called the Steiner TSP, or STPS) (Lam and Newman 2008). We adopt a well-known algorithm for approximating the shortest coverage route, referred to as a nearest-neighbor heuristic (Rosenkrantz, Stearns, and Lewis 1977). However, our results and approach are general, and admit alternative approximation algorithms, such as that proposed by Christofides (1976) which offers a  $3/2$  approximation factor. Moreover, it is known that TSP has submodular walk length on tree-structured graphs (Herer 1999), which motivates our relaxed submodularity assumption on the cost function due to (Alkalay-Houlihan and Vetta 2014).

A variation on the problem we study is the Orienteering Problem (OP), in which the goal is to maximize a total score collected from visiting vertices on a graph, subject to a travel time constraint (Golden, Levy, and Vohra 1987; Vansteenkoven, Souffriau, and Van Oudheusden 2011). Chekuri and Pal (2005) propose a quasi-polynomial time approximation algorithm that yields a logarithmic approximation guarantee for a more general *submodular* objective function. Singh et al. (2007) show that how this algorithm can be scaled up, and present results on planning informative paths for robotic sensors. However, our experimental results suggest that the running time of this algorithm is orders of magnitude slower than alternatives (including our proposed algorithm), and we therefore do not consider it in detail.

Perhaps the closest, and most practical, alternative to our algorithm is the framework proposed by Iyer and Bilmes (2013). Specifically, they consider submodular maximization under a submodular cost constraint, and propose several algorithms, including a greedy heuristic (GR) and iterative submodular knapsack (ISK) (their third proposed algorithm, involving ellipsoidal approximation of the submodular cost, scales poorly and we do not consider it). Our approach is a significant extension compared to Iyer and Bilmes (2013) and Iyer (2015): we present a new generalized cost-benefit algorithm, and demonstrate bi-criterion approximation guarantees which relax the submodularity assumption on the cost function made by Iyer. This generalization is crucial, as routing costs are in general not submodular (Herer 1999). Moreover, we demonstrate that our algorithm outperforms that of Iyer and Bilmes in experiments.

## Problem Statement

Let  $V$  be a collection of elements and  $f : 2^V \rightarrow R_{\geq 0}$  a function over subsets of  $V$ , and assume that  $f$  is monotone increasing. For any  $S \subseteq V$ , define  $f(j|S) = f(S \cup j) - f(S)$ , that is, the marginal improvement in  $f$  if element  $j \in V$  is added to a set  $S \subset V$ . Our discussion will concern *submodular functions*  $f$ , which we now define.

**Definition 1.** A function  $f : 2^V \rightarrow R_{\geq 0}$  is submodular if for all  $S \subseteq T \subseteq V$ ,  $f(j|S) \geq f(j|T)$ .

Our goal is to find a set  $S^* \subseteq V$  which solves the following problem:

$$f(S^*) = \max\{f(S) \mid c(S) \leq B\}, \quad (1)$$

where  $c : 2^V \rightarrow R_{\geq 0}$  is the cost function, which we assume is monotone increasing.

An important motivating setting for this problem is when the cost function represents a least-cost route through a set of vertices  $S$  on a graph. Specifically, suppose that  $G_R(V, E)$  is a graph in which  $V$  are nodes and  $E$  edges, and suppose that traversing an edge  $e \in E$  incurs a cost  $c_e$ , whereas visiting a vertex  $v \in V$  incurs a cost  $c_v$ . For a given set of nodes  $S \subseteq V$ , define a cost  $c_R(S)$  as the shortest walk in  $G_R$  that passes through all nodes in  $S$  at least once. The cost function for  $S$  then becomes

$$c(S) = c_R(S) + \sum_{s \in S} c_s,$$

that is, the total coverage cost (by a shortest walk through the graph), together with visit cost, for nodes in  $S$ .

## Generalized Cost-Benefit Algorithm

Maximizing submodular functions in general is NP-hard (Khuller, Moss, and Naor 1999). Moreover, even computing the cost function  $c(S)$  is NP-hard in many settings, such as when it involves computing a shortest walk through a subset of vertices on a graph (a variant of the traveling salesman problem). Our combination of two hard problems seems hopeless. We now present a general cost-benefit (GCB) algorithm (Algorithm 1) for computing approximate solutions to Problem 1. In the sections that follow we present theoretical guarantees for this algorithm under additional assumptions on the cost function, as well as empirical evidence for the effectiveness of GCB. At the core of the algorithm is the following simple heuristic: in each iteration  $i$ , add to a set  $G$  an element  $X_i$  such that

$$X_i = \arg \max_{X \in E \setminus G_{i-1}} \frac{f(X|G_{i-1})}{c(X|G_{i-1})}, \quad (2)$$

where  $G_0 = \emptyset$  and  $G_i = \{X_1, \dots, X_i\}$ . The simple heuristic alone has an unbounded approximation ratio, which is shown for a modular cost function by Khuller, Moss, and Naor (1999). The key modification is to return the better of this solution and the solution produced by a greedy heuristic that ignores the cost altogether. Next, we observed that  $c(\cdot)$  may not be computable in polynomial time. We therefore make use of an approximate cost function  $\hat{c}(\cdot)$  in its place which can be computed in polynomial time. The nature of

**Data:**  $B > 0$   
**Result:** Selection  $S \subseteq V$   
 $A := \arg \max\{f(X) | X \in V, \hat{c}(X) \leq B\};$   
 $G := \emptyset;$   
 $V' := V;$   
**while**  $V' \neq \emptyset$  **do**  
  **foreach**  $X \in V$  **do**  
     $\Delta_f^X := f(G \cup X) - f(G);$   
     $\Delta_c^X := \hat{c}(G \cup X) - \hat{c}(G);$   
  **end**  
   $X^* = \arg \max\{\Delta_f^X / \Delta_c^X | X \in V'\};$   
  **if**  $\hat{c}(G \cup X^*) \leq B$  **then**  
     $G := G \cup X^*;$   
  **end**  
   $V' := V' \setminus X^*$   
**end**  
**return**  $\arg \max_{S \in \{A, G\}} f(S)$   
**Algorithm 1:** Generalized Cost-benefit Algorithm.

our results will then depend on the quality of this approximation. Observe that, the greedy solution,  $A$ , in Algorithm 1 contains only a single element. However, a variant of this is to continue adding more elements until we violate the budget constraint. Similarly, when the budget constraint in GCB is saturated, we can continue adding elements that are not first-best, until no elements can be added. We show below that these variations yield the same approximation guarantees. Our implementations, however, use these enhancements.

## Theoretical Analysis

A major limitation of prior work on cost-constrained submodular maximization is that the cost function itself was assumed to be submodular. In most practical problems where the cost function is generated by routing problems (such as coverage of nodes on a graph, or TSP), the optimal cost is not submodular (Herer 1999). On the other hand, TSP has submodular special cases, such as when the graph is a tree (Herer 1999). Motivated by this observation, we make use of a natural relaxation of cost submodularity,  $\alpha$ -submodularity.

**Definition 2.** For a cost function  $c$ , we define it as  $\alpha$ -submodular for

$$\alpha = \min_x \min_{A, B: A \subset B} \frac{c(x|A)}{c(x|B)}.$$

In addition, just as in prior work, we need another notion for a function  $c$ , termed *curvature*, which essentially measures deviations from linearity (Conforti and Cornuéjols 1984; Iyer, Jegelka, and Bilmes 2013).

**Definition 3.** For a submodular function  $c$  over subsets of  $V$ , total curvature  $k_c$  is defined as

$$k_c = 1 - \min_{j \in V} \frac{c(j|V \setminus j)}{c(j)},$$

As mentioned earlier, since optimal cost is often infeasible to compute, our GCB algorithm makes use of approximate cost function,  $\hat{c}$ . We now make this notion of approximation formal: we assume that  $\hat{c}$  is a  $\psi(n)$ -approximation of the optimal cost, where  $n = |V|$ . In other words,  $c(X) \leq \hat{c}(X) \leq \psi(n)c(X)$ . For example, if an algorithm for TSP is a  $3/2$ -approximation (as is the algorithm by Christofides (1976)),  $\psi(n) = 3/2$  (independent of problem size). Below we use a much faster and simpler algorithm, nearest neighbor, which is a  $\log(n)$ -approximation.

Finally, we introduce another useful piece of notation, defining

$$K_c = \max\{|X| : c(X) \leq B\},$$

that is,  $K_c$  is the largest set  $X \subseteq V$  which is feasible for our problem.

Armed with these notions, we are now ready to prove a bi-criterion approximation guarantee on GCB, which presents a bound on the solution quality compared to the optimal solution with a slightly relaxed budget constraint.

Suppose the GCB algorithm starts with an empty set  $G_0 = \emptyset$ , and keeps adding nodes the set by the greedy rule (Equation 2). It generates a sequence of intermediate sets,  $G_1, \dots, G_l$ , until iteration  $l + 1$  when it violates the budget constraint and stops with set  $G_{l+1}$ .

**Lemma 1.** For  $i = 1, \dots, l + 1$ , it holds that

$$f(G_i) - f(G_{i-1}) \geq \frac{\hat{c}(G_i) - \hat{c}(G_{i-1})}{B} (f(\tilde{X}) - f(G_{i-1}))$$

where  $\hat{c}$  is an  $\alpha$  submodular  $\psi(n)$ -approximation of the  $\alpha$ -submodular function  $c$  and  $\tilde{X}$  is the optimal solution of  $\max\{f(X) | c(X) \leq \alpha \frac{B(1+\alpha(K_c-1)(1-k_c))}{\psi(n)K_c}\}$ .

The proof of this lemma is quite involved, making use of a series of auxiliary results, and is provided in the extended version of the paper.<sup>1</sup>

**Lemma 2.** For  $i = 1, \dots, l + 1$  it holds that

$$f(G_i) \geq \left[ 1 - \prod_{k=1}^i \left( 1 - \frac{\hat{c}(G_k) - \hat{c}(G_{k-1})}{B} \right) \right] f(\tilde{X})$$

where  $\hat{c}$  is an  $\alpha$ -submodular  $\psi(n)$ -approximation of an  $\alpha$ -submodular function  $c$ , and  $\tilde{X}$  is the optimal solution of  $\max\{f(X) | c(X) \leq \alpha \frac{B(1+\alpha(K_c-1)(1-k_c))}{\psi(n)K_c}\}$ .

*Proof.* For  $i = 1$ , we need to prove that

$$f(G_1) \geq \frac{\hat{c}(G_1) - \hat{c}(G_0)}{B} f(\tilde{X}).$$

Clearly, this follows from Lemma 1. Let  $i > 1$ , we have

$$\begin{aligned} f(G_i) &= f(G_{i-1}) + [f(G_i) - f(G_{i-1})] \geq \\ &f(G_{i-1}) + \frac{\hat{c}(G_i) - \hat{c}(G_{i-1})}{B} (f(\tilde{X}) - f(G_{i-1})) \\ &= \left( 1 - \frac{\hat{c}(G_i) - \hat{c}(G_{i-1})}{B} \right) f(G_{i-1}) + \end{aligned}$$

<sup>1</sup>See <http://vorobeychik.com/2016/gcb.pdf>.

$$\begin{aligned}
& \frac{\hat{c}(G_i) - \hat{c}(G_{i-1})}{B} f(\tilde{X}) \\
& \geq \left(1 - \frac{\hat{c}(G_i) - \hat{c}(G_{i-1})}{B}\right) \\
& \left[ \left(1 - \prod_{k=1}^{i-1} \left(1 - \frac{\hat{c}(G_k) - \hat{c}(G_{k-1})}{B}\right)\right) f(\tilde{X}) \right] + \\
& \frac{\hat{c}(G_i) - \hat{c}(G_{i-1})}{B} f(\tilde{X}) \\
& = \left(1 - \prod_{k=1}^i \left(1 - \frac{\hat{c}(G_k) - \hat{c}(G_{k-1})}{B}\right)\right) f(\tilde{X})
\end{aligned}$$

□

**Theorem 1.** *The GCB algorithm obtains a set  $X$  such that*

$$f(X) \geq \frac{1}{2}(1 - e^{-1})f(\tilde{X}),$$

where  $\tilde{X}$  is the optimal solution of  $\max\{f(X) | c(X) \leq \alpha \frac{B(1+\alpha(K_c-1)(1-k_c))}{\psi(n)K_c}\}$ ,  $\hat{c}$  is an  $\alpha$ -submodular  $\psi(n)$ -approximation of an  $\alpha$ -submodular function  $c$ .

*Proof.* Observe that for  $a_1, \dots, a_n \in R^+$ , such that  $\sum a_i = A$ ,  $1 - \prod_{i=1}^n (1 - \frac{a_i}{A})$  achieves its minimum at  $a_1 = \dots = a_n = \frac{A}{n}$ . It follows that

$$\begin{aligned}
f(G_{l+1}) & \geq \left[1 - \prod_{k=1}^{l+1} \left(1 - \frac{\hat{c}(G_k) - \hat{c}(G_{k-1})}{B}\right)\right] f(\tilde{X}) \\
& \geq \left[1 - \prod_{k=1}^{l+1} \left(1 - \frac{\hat{c}(G_k) - \hat{c}(G_{k-1})}{\hat{c}(G_{l+1})}\right)\right] f(\tilde{X}) \\
& \geq \left[1 - \left(1 - \frac{1}{l+1}\right)^{l+1}\right] f(\tilde{X}) \\
& \geq \left(1 - \frac{1}{e}\right) f(\tilde{X})
\end{aligned}$$

where the first inequality follows from Lemma 2 and the second inequality follows from the fact that  $\hat{c}(G_{l+1}) > B$ , since it violates the budget. Moreover, by submodularity, we note that  $f(G_{l+1}) - f(G_l) \leq f(X_{l+1}) \leq f(X^*)$ , where  $X^* = \arg \max_{X \in E} f(X)$ . Therefore,

$$f(G_l) + f(X^*) \geq f(G_{l+1}) \geq (1 - 1/e)f(\tilde{X})$$

and

$$\max\{f(G_l), f(X^*)\} \geq \frac{1}{2}(1 - 1/e)f(\tilde{X})$$

□

Having established a general approximation ratio for GCB, note that an exactly submodular cost function emerges as a special case with  $\alpha = 1$ , as does exact cost function when  $\psi(n) = 1$ , with the bound becoming tighter in both instances. Moreover, the bound becomes tighter as we approach modularity.

## Experiments

We implement the GCB algorithm to solve two realistic submodular maximization problems: mobile robotic sensing and door-to-door solar marketing, using both real and simulated networks. We show that the proposed GCB algorithm outperforms state-of-the-art alternatives, *particularly when routing problems do not yield a submodular optimal cost function*. We compare GCB to two state-of-the-art algorithms: simple greedy (GR) and iterative submodular knapsack (ISK).<sup>2</sup> All experiments were performed on an Ubuntu Linux 64-bit PC with 32 GB RAM and an 8-core Intel Xeon 2.1 GHz CPU. Each experiment used a single thread.

### Case Study 1: Mobile Robotic Sensing

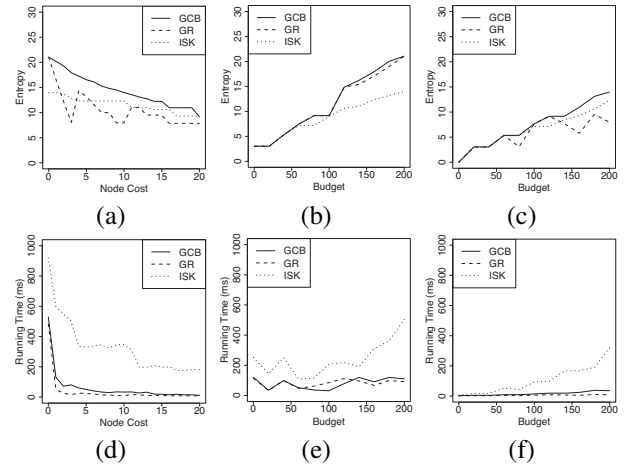


Figure 1: Entropy (a)-(c) & run time (d)-(f) comparison among algorithms for mobile robotic sensing scenario. (a), (d) As a function of visit (sensing) cost, fixing budget at 200. (b), (e) As a function of budget, fixing visit cost = 0. (c), (f) As a function of budget, fixing visit cost = 10.

Consider the following problem. A mobile robot equipped with sensors wants to optimally choose a subset of locations in a 2-D space to make measurements. A common criterion guiding such a choice is to minimize *conditional entropy* of unobserved locations of interest or, equivalently, maximize entropy of selected locations (Krause, Singh, and Guestrin 2008).<sup>3</sup> We suppose that the robot faces two kinds of costs (e.g., reflecting battery life or time): costs of moving between a pair of neighboring locations, and costs of making measurements at a particular location.

Our experiments use sensor data representing air quality measurements for 36 air quality monitoring stations in Beijing, China (Zheng, Liu, and Hsieh 2013), where we limit attention to temperature. We fit a multivariate Gaussian distribution to this data, and focus on a random subset of locations as the focus of prediction. We generated a hypothetical

<sup>2</sup>We have attempted to apply several other algorithms, but these two are the only candidates that scale to non-trivial problem instances.

<sup>3</sup>This objective is known to be submodular.

a tree-structure routing network (see the Supplement), and assume that the robot must return to the starting location.

The GCB algorithm starts with an empty set  $A_0 = \emptyset$  and iteratively adds to  $A_i$  a location  $x^*$  with highest ratio of conditional entropy to marginal cost,

$$x^* = \arg \max_x \frac{H(x|A_i)}{c(x|A_i)},$$

where

$$H(x|A_i) = \frac{1}{2} \log(2\pi e \sigma_{x|A_i}^2)$$

is the conditional entropy of  $x$  given selection of  $A_i$  and  $c(x|A_i)$  is the marginal cost of covering  $x$  if we already cover  $A_i$ . We use the Nearest Neighbor (NN) algorithm to compute approximately optimal routing cost (Rosenkrantz, Stearns, and Lewis 1977). The conditional variance,  $\sigma_{x|A_i}^2$ , can be obtained as follows:

$$\sigma_{x|A_i}^2 = K(x, x) - \Sigma_{xA_i} \Sigma_{A_i A_i}^{-1} \Sigma_{A_i x},$$

where  $K(x, x)$  is the variance of location  $x$ ,  $\Sigma_{xA_i}$  is a vector of covariances  $K(x, u), \forall u \in A_i$ ,  $\Sigma_{A_i A_i}$  the covariance submatrix corresponding to measurements  $A_i$ , and  $\Sigma_{A_i x}$  is the transpose of  $\Sigma_{xA_i}$ . Figure 1 shows the results of entropy (performance) and running time comparison between GCB and the two previous approaches (GR and ISK). GCB nearly always outperforms the other two, often by a large margin, in terms of entropy; GR is particularly weak in most comparisons. Moreover, GCB has competitive running time with GR, and scales far better than ISK.

## Case Study 2: Door-to-door Marketing of Rooftop Solar Photovoltaic Systems

Our next investigation focuses on door-to-door marketing problem, which we cast as social influence maximization (Kempe, Kleinberg, and Tardos 2003) under routing constraints. We formalize this problem by considering two interdependent networks: the social influence network, which captures the influence adopters (of a product) have on the likelihood that others adopt, and the routing network, which represents routes taken by a marketer to visit households of choice. The goal of the marketer is to select a subset of individuals on a network maximizing overall social influence, subject to cost (e.g., time) constraints. Below we consider both real and simulated networks. Whatever the choice of the network, we use the well-known *independent cascade* (IC) model to compute social influence of a subset of nodes (Kempe, Kleinberg, and Tardos 2003). In the IC model, given a subset of nodes initially seeded, each neighbor  $w$  of an adopter  $v$  is independently influenced to adopt with probability  $p_{vw}$ , and this adoption cascade process iteratively repeats until no new nodes adopt. The expected final number of adopters after this process terminates is denoted by  $\sigma(A)$ , where  $A$  is the set of initially seeded nodes. Kempe, Kleinberg, and Tardos (2003) showed that the function  $\sigma(A)$  is monotone submodular. In our experiments below, we used  $p_{vw} = 0.1$  for all network edges  $(v, w)$ . Since the IC model is stochastic in nature, we run it 1000 times to estimate  $\sigma(A)$ .

**Adoption of Visible Technology** Our first influence network was generated to represent adoption of a highly visible technology, such as rooftop solar (Zhang et al. 2015). Specifically, when a technology is visible, the primary social influence effect stems from geographic proximity; thus, in the case of rooftop solar adoption, adoption has been shown to have significant influence on geographic neighbors (Zhang et al. 2015). For this use case, we therefore take a household dataset for San Diego county, and induce a social influence network based on proximity as measured by a 165 foot radius, giving rise to the influence network shown in Figure 2 (left). Figure 2 (right) shows the corresponding



Figure 2: Top: social influence network arising from geographic proximity. Bottom: corresponding routing network.

routing network obtained from OpenStreetMap, where red dots are way points or intersections in the road networks. The costs of edges in the routing network correspond to physical distance.

Figure 3 shows the results of comparing our GCB algorithm to GR and ISK, both in terms of achieved average influence, and running time. In all cases, we can observe that GCB outperforms the others on both measures, often by a substantial margin. Particularly striking is the running time comparison with ISK, where the difference can be several orders of magnitude.

Our final experimental investigation involves random graph models for both social influence propagation and routing. In particular, we use the well-known Barabasi-Albert (BA) model (Albert and Barabási 2002) to generate a random social network (a natural choice, since BA model has been shown to exhibit a scale-free degree distribution, which is a commonly observed feature of real social networks), and the Erdos-Renyi (ER) model to generate the routing network (Erdos and Renyi 1959).

The BA model is an iterative generative model which starts with a small seeded network (e.g., a triangle), and adds a node one at a time, connecting it to  $m$  vertices, with each edge using the new node as a source generated with probability proportional to the target node’s degree. In our implementation, we generated a BA social network

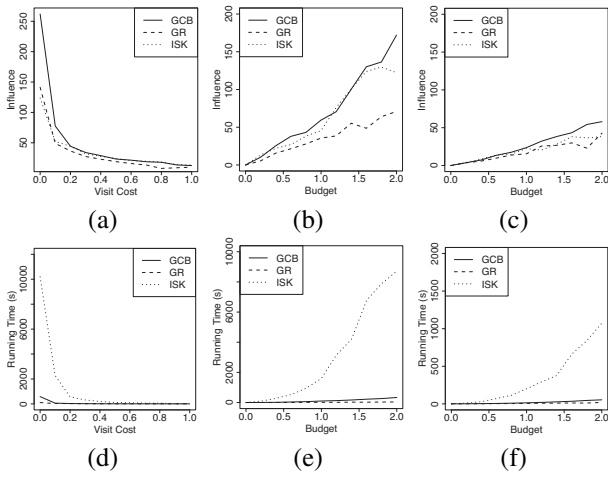


Figure 3: Influence  $\sigma$  (a)-(c) & run time (d)-(f) comparison among algorithms for the door-to-door marketing scenario with visible technology. (a), (d) As a function of visit (sensing) cost, fixing budget at 3. (b), (e) As a function of budget, fixing visit cost = 0. (c), (f) As a function of budget, fixing visit cost = 0.1.

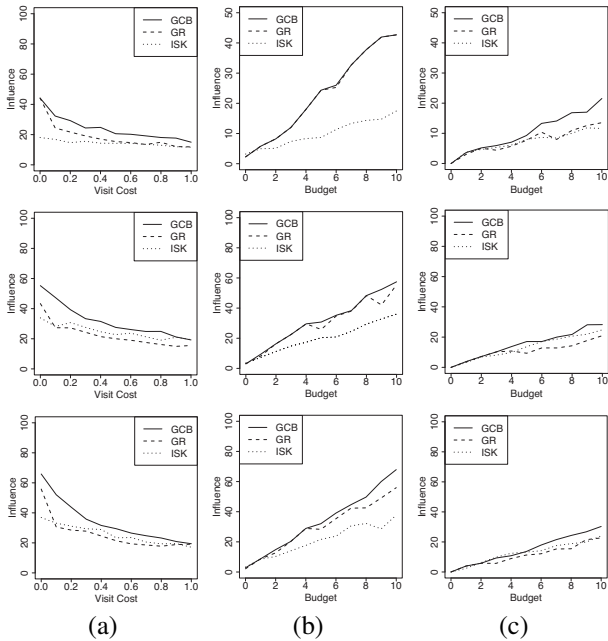


Figure 4: Influence  $\sigma$  comparison among algorithms for the door-to-door marketing scenario on random graphs. Top row:  $p = 0.01$  in the ER mode. Middle row:  $p = 0.02$ . Bottom row:  $p = 0.03$ . (a) As a function of visit (sensing) cost, fixing budget at 10. (b) As a function of budget, fixing visit cost = 0. (c) As a function of budget, fixing visit cost = 0.5.

over 200 nodes, adding  $m = 2$  edges in each iteration. The ER model is the simplest generative model of networks, where each edge is added to the network with a fixed probability  $p$ . In our experiments, we considered values of

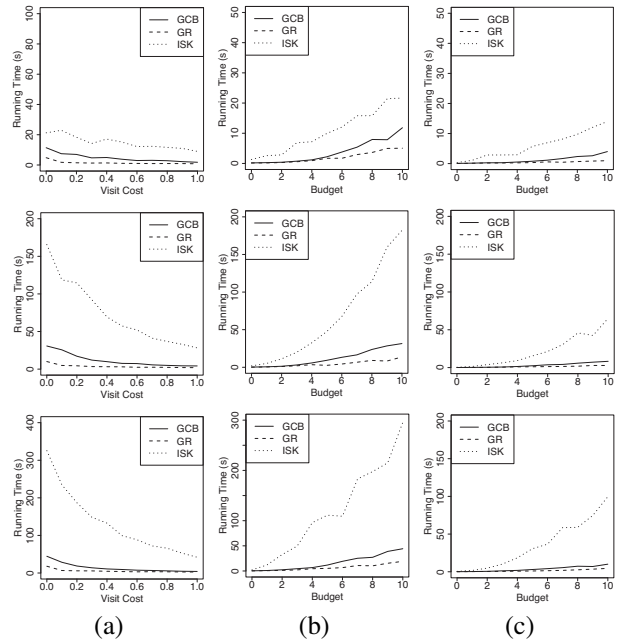


Figure 5: Run time comparison among algorithms for the door-to-door marketing scenario on random graphs. Top row:  $p = 0.01$  in the ER mode. Middle row:  $p = 0.02$ . Bottom row:  $p = 0.03$ . (a) As a function of visit (sensing) cost, fixing budget at 10. (b) As a function of budget, fixing visit cost = 0. (c) As a function of budget, fixing visit cost = 0.5.

$p \in \{0.01, 0.02, 0.03\}$ . To generate routing costs, we randomly assigned coordinates for the 200 nodes in 2-D space, and use the Euclidean distance between nodes as a proximity measure.

Figure 4 and Figure 5 show the results for the random graph experiments, which are consistent with the observations so far: GCB tends to outperform alternative algorithmic approaches both in terms of objective value (influence, in this case), and in terms of running time (it is comparable to GR, and much faster than ISK). Interestingly, in some cases GR or ISK have comparable objective value to GCB, but in almost all cases the other algorithm is far worse, and the relative performance of GR and ISK is not consistent: either one is sometimes observed to be better than the other. We can also note that as  $p$  increases (and the routing network becomes more dense), the running time of ISK increases rather dramatically, whereas both GCB and GR remain quite scalable.

## Conclusion

We considered a very general class of problems in which a monotone increasing submodular function is maximized subject to a general cost constraint. This problem is motivated by two very different applications: one is mobile robotic sensing, in which a robot moves through an environment with the goal of making select sensor measurements to make predictions about a location which is in-

feasible to measure, and another in door-to-door marketing. In both of these applications, the cost constraints arise from routing costs, as well as costs to visit nodes (e.g., to make sensor measurements or to make a marketing pitch). Our algorithmic contribution was a novel generalized cost-benefit algorithm, for which we showed bi-criterion approximation guarantees with a relaxed notion of cost submodularity as well as allowing optimal cost to be only approximately computed. Through an extensive experimental evaluation on both real and synthetic graphs we showed that our algorithm outperforms state-of-the-art alternatives both in terms of objective value achieved (often significantly) and running time. Moreover, we dramatically outperform an iterative submodular knapsack algorithm in terms of runtime, and have a competitive runtime with a greedy algorithm, which tends to perform poorly in terms of objective.

### Acknowledgments

This work was partially supported by the U.S. Department of Energy (DOE) office of Energy Efficiency and Renewable Energy, under the Solar Energy Evolution and Diffusion Studies (SEEDS) program, the National Science Foundation (IIS-1526860), and the Office of Naval Research (N00014-15-1-2621).

### References

Albert, R., and Barabási, A.-L. 2002. Statistical mechanics of complex networks. *Reviews of modern physics* 74(1):47.

Alkalay-Houlihan, C., and Vetta, A. 2014. False-name bidding and economic efficiency in combinatorial auctions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*, 538–544.

Chekuri, C., and Pal, M. 2005. A recursive greedy algorithm for walks in directed graphs. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, 245–253. IEEE.

Christofides, N. 1976. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, DTIC Document.

Conforti, M., and Cornuéjols, G. 1984. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete applied mathematics* 7(3):251–274.

Erdos, P., and Renyi, A. 1959. On random graphs i. *Publ. Math. Debrecen* 6:290–297.

Fujishige, S. 2005. *Submodular functions and optimization*, volume 58. Elsevier.

Golden, B. L.; Levy, L.; and Vohra, R. 1987. The orienteering problem. *Naval Research Logistics (NRL)* 34(3):307–318.

Herer, Y. T. 1999. Submodularity and the traveling salesman problem. *European journal of operational research* 114(3):489–508.

Iyer, R. K., and Bilmes, J. A. 2013. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, 2436–2444.

Iyer, R. K.; Jegelka, S.; and Bilmes, J. A. 2013. Curvature and optimal algorithms for learning and minimizing submodular functions. In *Advances in Neural Information Processing Systems*, 2742–2750.

Iyer, R. K. 2015. *Submodular Optimization and Machine Learning: Theoretical Results, Unifying and Scalable Algorithms, and Applications*. Ph.D. Dissertation.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 137–146. ACM.

Khuller, S.; Moss, A.; and Naor, J. S. 1999. The budgeted maximum coverage problem. *Information Processing Letters* 70(1):39–45.

Krause, A., and Golovin, D. 2012. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems* 3:19.

Krause, A., and Guestrin, C. 2005. A note on the budgeted maximization of submodular functions.

Krause, A.; Singh, A.; and Guestrin, C. 2008. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *The Journal of Machine Learning Research* 9:235–284.

Lam, F., and Newman, A. 2008. Traveling salesman path problems. *Mathematical Programming* 113(1):39–59.

Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; Van-Briesen, J.; and Glance, N. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 420–429. ACM.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14(1):265–294.

Rosenkrantz, D. J.; Stearns, R. E.; and Lewis, II, P. M. 1977. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing* 6(3):563–581.

Singh, A.; Krause, A.; Guestrin, C.; Kaiser, W. J.; and Batalin, M. A. 2007. Efficient planning of informative paths for multiple robots. In *IJCAI*, volume 7, 2204–2211.

Sviridenko, M. 2004. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters* 32(1):41–43.

Vansteenwegen, P.; Souffriau, W.; and Van Oudheusden, D. 2011. The orienteering problem: A survey. *European Journal of Operational Research* 209(1):1–10.

Zhang, H.; Vorobeychik, Y.; Letchford, J.; and Lakkaraju, K. 2015. Data-driven agent-based modeling, with application to rooftop solar adoption. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multi-agent Systems*, 513–521. International Foundation for Autonomous Agents and Multiagent Systems.

Zheng, Y.; Liu, F.; and Hsieh, H.-P. 2013. U-air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1436–1444. ACM.