Multi-Channel Marketing with Budget Complementarities

Haifeng Zhang
Vanderbilt University
haifeng.zhang@vanderbilt.edu

Yevgeniy Vorobeychik
Vanderbilt University
yevgeniy.vorobeychik@vanderbilt.edu

Ariel D. Procaccia
Carnegie Mellon University
arielpro@cs.cmu.edu

ABSTRACT

Utility maximization under a budget constraint is a classical problem in economics and management science. It is commonly assumed that the utility is a "nice" known analytic function, for example, continuous and concave. In many domains, such as marketing, increased availability of computational resources and data has enabled the development of sophisticated simulations to evaluate the impact of allocating a fixed budget among alternatives (e.g., marketing channels) on outcomes, such as demand. While simulations enable high resolution evaluation of alternative budget allocation strategies, they significantly complicate the associated budget optimization problem. In particular, simulation runs are time consuming, significantly limiting the space of options that can be explored. An important second challenge is the common presence of budget complementarities, where non-negligible budget increments are required for an appreciable marginal impact from a channel. This introduces a combinatorial structure on the decision space. We propose to address these challenges by first converting the problem into a multi-choice knapsack optimization problem with unknown weights. We show that if weights (corresponding to marginal impact thresholds for each channel) are well approximated, we can achieve a solution within a factor of 2 of optimal, and this bound is tight. We then develop several parsimonious query algorithms for achieving this approximation in an online fashion. Experimental evaluation demonstrates the effectiveness of our approach.

Keywords

Budget optimization; multi-channel marketing; multi-choice knapsack; approximation guarantee; query strategy

1. INTRODUCTION

The emergence of digital media, such as the world wide web, search engines, and online social networks, has opened up tremendous opportunities for today's marketers to look for prospects and engage existing customers. A mix of these innovative channels with traditional ones, such as TV, direct mailing, and door-to-door marketing, has been widely adopted by many companies to generate more sales, maintain stronger customer relationships, and achieve a higher customer retention rate [20]. Despite its benefits, this practice has also significantly increased operational complexity, making marketing one of the key managerial challenges [16, 17]. The demand for effective budget allocation solutions in multi-channel marketing campaigns has in turn given rise to major software products aimed towards this goal, including those developed by SAS and IBM, among others.

In order to determine the optimal budget allocation among the marketing channels, the marketer needs a way to evaluate the effectiveness of alternative budget splits. Advanced simulation models, and abundant data that can be used to calibrate them, allow doing just that. The use of simulations, as compared to analytic objective functions (such as concave and continuous utility being maximized), introduces an important technical challenge: simulations are often slow, and parsimony is therefore crucial in query-based black-box optimization methods. A second technical challenge arises from the fact that the response function for each channel (such as the number of individuals who buy the product) commonly exhibits *budget complementarities*, requiring a non-trivial added expense on a channel to make a significant impact on the response function. For example, in door-to-door marketing, a budget increment needs to be sufficient to hire another salesman, or increase their working hours by a discrete amount. Similarly, in keyword auctions, moving up a slot requires a discrete added investment, the amount of which depends on specific pricing strategies and competition among bidders.

To address these challenges, we present a novel and powerful discrete budget optimization framework to generate near-optimal budgeting strategies when the budget allocation response is a *step function* represented by a simulator. We first show that the budget optimization problem can be readily cast into a multi-choice knapsack problem (MCKP), which admits effective state-of-the-art algorithms. Since the step-wise response function is represented using a simulator, the thresholds which identify the discrete jumps (serving as weights in the MCKP) are unknown, and a finite number of simulator queries can at best isolate these to small *intervals*. Consequently, the MCKP can at best be solved approximately. We show that under mild conditions, for sufficiently small bounds on weights, solving the MCKP with weight upper bounds yields a 2-approximation, and this bound is tight. Surprisingly, this bound holds even when the thresholds are not fully explored. Next, we develop two efficient query algorithms that allow us to obtain tight intervals around MCKP weights (as well as associated response val-

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

ues). The first, Generalized Binary Query (GBQ) is a generalization of the classic binary search applied to the case of multiple thresholds, which we show to be more efficient than simple linear search. The second approach, namely, Heuristic Binary Query (HBQ), is designed to reduce the number of queries needed per iteration, with the help of the solution of an auxiliary optimization problem that corresponds to the best possible payoff in the next round.

Our framework is implemented in a simulated marketing environment that mimics a real-world multi-channel campaign in a targeted geographical area. We use this simulator to conduct extensive experiments to demonstrate the usability of the proposed framework and compare the performance of two query algorithms as well as a Simulated Annealing (SA) algorithm, which is a well-known stochastic local search method typically used for problems with highly non-linear objectives. Our results show that HBQ achieves payoffs only slightly lower than GBQ, but using significantly less time. Moreover, both HBQ and GBQ outperform SA in all experiments, reaching competitive payoff levels significantly faster.

In summary, we make the following contributions:

1. A novel discrete budget optimization problem with an application to multi-channel marketing, transformed into a multi-choice knapsack problem;
2. A theoretical analysis of the resulting problem in which weights (corresponding to steps in the response function) can only be bounded, showing that solving the approximate MCKP with upper bounds on weights yields a tight 2-approximation;
3. Two novel simulation query strategies for obtaining upper and lower bounds on MCKP weights: Generalized Binary Query and Heuristic Binary Query;
4. A simulation platform to evaluate the multi-channel marketing algorithms;
5. Extensive experiments across different marketing situations and a variety of budgets demonstrating the usability of the proposed framework and efficacy of proposed query methods, compared to a simulated annealing algorithm.

2. RELATED WORK

Budget optimization is a classical problem in economics, operations research, and management science. The problem is traditionally tackled by maximizing specific objectives, e.g., sales, profit, or customer equity, based on a set of pre-specified constraints [16]. In practice, the real-world markets involve considerable complexity, for example, as a consequence of social interactions and influence [13]. Consequently, simulation-based optimization methods, such as system dynamic models, are often used to aid decision making [11]. Interestingly, most previous work assumes a continuous objective and solves the problem by smooth techniques which rely on computing a gradient or Hessian of the objective. In contrast, we present a novel combinatorial optimization framework in a setting where channel payoff exhibits strong budget complementarities which we model by considering a stepwise response function. Note that the non-smooth optimization techniques [3] such as the *sub-gradient* methods fail in our setting since the sub-gradient is not informative in step functions.

Recently, extensive work focused on budget optimization for a single marketing channel. For example, Yang et al. propose a hierarchical budget allocation framework for online advertising that links decisions at different decision levels, such as system, campaign, and keyword [22]. Using individual data, several authors model user responses to marketing actions as a Markov Chain and solve the budget optimization problem using a constrained Markov Decision Process (MDP). For instance, Abe et al. develop an MDP framework with reinforcement learning for direct mailing campaigns [1]. Under the assumption of positive carryover effects, Archak et al. propose an optimal greedy algorithm for online adverting in an MDP framework [2]. Boutilier and Lu address the allocation of a budget among multiple MDPs representing different types of users or groups [8]. Zhang and Vorobeychik develop a route planner for door-to-door marketing based on submodular optimization [23]. These specialized budget optimizers based on empirical data are quite effective for targeted marketing, but are still specialized sub-problems of the overall problem of optimally allocating a budget among a collection of marketing channels, which we address.

The multi-choice knapsack problem (MCKP) is a variant of the simple knapsack problem in which a class can have multiple items but only one can be chosen, and has been extensively explored in the literature [21, 19, 9]. Our theoretical analysis of approximation bounds is related to *sensitivity analysis* in operations research [5], which examines the sensitivity of the optimal solution to changes in the coefficient matrix, cost, price, and budget. Hifi et al. provide sensitivity intervals for the 0-1 knapsack problem subject to changes of item weights [12]. In contrast, we address the question about the worst-case performance of the MCKP in which weights are tightly bounded, as a means to a broader end of multi-channel marketing budget allocation. Finally, our work is related to, but distinct from *robust optimization* [4, 18, 15, 10]. This line of work usually imposes a limit for the number of uncertain parameters (e.g., weights) to avoid overly conservative solutions, while we use the upper-bounds for all weights to secure a feasible solution. Goerigk et al. study query strategies for a robust knapsack problem, rather than a general MCKP as in our case [10]. They assume that a single query returns "true" weight; by contrast, we design a sequence of queries to efficiently approximate weight bounds, but cannot in general obtain true weights, as is commonly the case when marketing response is simulated.

3. PROBLEM STATEMENT

Suppose that a marketer is given a fixed *budget* B to advertise a new product over n marketing channels. Let $x = (x_1, \ldots, x_n)$ represent a budget split with x_i the amount of the budget allocated to channel i. Let $r(x)$ be the net reward to the marketer (e.g., in terms of overall product uptake) given a budget split x. Our goal is to solve the following *multi-channel marketing optimization* (MCMO) problem:

$$\max_x r(x) \tag{1a}$$

$$\text{s.t.}: \sum_{i=1}^{n} x_i \leq B, \tag{1b}$$

that is, we aim to optimally split the budget B across the n channels to maximize the total net payoff. If $r(x)$ are concave and known and the budget divisible, as is commonly assumed in numerous related formulations, Problem 1 is straightforward to solve with the standard convex optimization methods (indeed, this is just the standard budget-constrained utility maximization problem in consumer theory [14]). What has not received much attention, and is of interest to us, is this problem in which (a) $r(x)$ exhibits strong, but imperfect, complementarities, and (b) $r(x)$ is not a priori known, but specified by a time consuming simulation model. For example, $r(x)$ may capture a complex social influence diffusion process which cannot be analytically characterized and is evaluated in simulations, as is the case for many important social influence models in the literature [13, 24]. Moreover, making a non-negligible impact on a given agent's decision (e.g., in seeding them by providing this agent a product at a low cost) incurs a non-zero cost which may be a complex function of contextual factors also embedded in a simulation, and therefore unknown a priori. As another example, online auction-based advertising channels (such as keyword auctions) require a sufficiently high investment to move into a higher priority slot, which makes a discontinuous impact on the expected number of clicks and, thus, conversions, and the precise amount of this investment is a complex function of bidding behavior by a collection of agents which can be captured in a simulation environment, but could be difficult to characterize in closed form.

In order to model such complementarities, we begin by assuming that $r(x) = \sum_i r_i(x_i)$, with $r_i(x_i)$ increasing and $r_i(0) = 0$. For each channel i, we suppose that there is a collection of thresholds w_{ij} so that crossing a threshold results in a jump in $r_i(x_i)$. Formally, we assume that $r_i(x_i)$ is a step function of the following form:

$$r_i(x_i) = \begin{cases} 0, & x_i < w_{i1} \\ r_{i1}, & w_{i1} \leq x_i < w_{i2} \\ \dots, & \dots \\ r_{iJ_i}, & x_i \geq w_{iJ_i} \end{cases}$$

where there are $J_i(\geq 1)$ thresholds $\{w_{ij}\}_{j=1,\dots,J_i}$ and non-zero payoff levels $\{r_{ij}\}_{j=1,\dots,J_i}$.

As the first step, we transform Problem 1 with the structure just described into an equivalent multi-choice knapsack problem (MCKP). Since in our model any investment level not corresponding to a threshold is wasteful, the decision problem is to determine at which threshold level j we should allocate the budget for each channel i. We encode this decision as a binary variable y_{ij}, which is 1 whenever we allocate budget at threshold level j for channel i. The MCKP is then

$$\max_{y_{ij} \in \{0,1\}} \sum_{i=1}^{n} \sum_{j=1}^{J_i} r_{ij} y_{ij} \tag{2a}$$

$$\text{s.t.}: \sum_{i=1}^{n} \sum_{j=1}^{J_i} w_{ij} y_{ij} \leq B \tag{2b}$$

$$\forall i \in \{1, \dots, n\}, \sum_{j=1}^{J_i} y_{ij} \leq 1, \tag{2c}$$

where the first inequality is the budget constraint, and the second implies that at most one budget level can be picked for any channel. Note that Problem 1 with known thresholds is harder than the MCKP, as a polynomial oracle for it can solve an arbitrary instance of MCKP. Throughout, it will be useful to denote the above MCKP as $MCKP(J_i, w_{ij})$, with a specified set of J_i weights for each i; the corresponding r_{ij} will be clear from context.

Armed with the MCKP formulation, we can now identify the key technical challenges: (1) w_{ij} can only be approximately determined from a finite number of queries, since these lie on a continuous interval, and (2) the problem parameters w_{ij} and r_{ij} must be obtained using time consuming simulations. We address these challenges below.

4. APPROXIMATE MULTI-CHOICE KNAPSACK

We begin by addressing the first challenge above: the threshold values w_{ij} cannot be identified exactly. Surprisingly, despite considerable prior work on approximate and robust knapsack problems, this particular problem remains open, to the best of our knowledge. Our analysis of approximate MCKP may thus be of independent interest, but for us it is just an important piece of the puzzle. We subsequently take up the complementary piece: efficient query strategies for achieving good MCKP approximations.

Formally, suppose that w_{ij} are not known, but we have lower and upper bounds so that $w_{ij} \in [\underline{w}_{ij}, \overline{w}_{ij}]$, and let $\epsilon := \max_{ij}\{\overline{w}_{ij} - \underline{w}_{ij}\} > 0$, which implies that $\overline{w}_{ij} - w_{ij} \leq \epsilon$ for all i, j. Since w_{ij} are unknown, we propose to approximate the associated MCKP with $MCKP(J_i, \overline{w}_{ij})$. Next, we demonstrate that under a set of conditions which can be guaranteed with sufficiently many simulation queries, we can obtain a 2-approximation of $MCKP(J_i, w_{ij})$, and this approximation is tight unless the weights are known *exactly*.

First, notice that we have thus far implicitly assumed that *every interval contains exactly one threshold* w_{ij}. This is a significant challenge: even if we can guarantee that for a particular fixed ϵ all thresholds are bounded within intervals of length at most ϵ, we would still be unable to distinguish thresholds that all cluster within some such interval. Fortunately, even in such a case we do know that all such thresholds are in one of the intervals we have identified. This turns out to be sufficient to obtain the approximation guarantees.

Formally, suppose that there are $J'_i \leq J_i$ thresholds for channel i, and we solve $MCKP(J'_i, \overline{w}_{ij})$. For ease of exposition, let us denote the optimal value of Problem 2 as $OPT(J_i, w_{ij})$, while the optimal value of $MCKP(J'_i, \overline{w}_{ij})$ will be denoted by $OPT(J'_i, \overline{w}_{ij})$. Finally, let $OPT(J'_i, \underline{w}_{ij})$ be the optimal value of the problem $MCKP(J'_i, \underline{w}_{ij})$, which uses lower-bound weights $\{\underline{w}_{ij}\}$ but upper-bound payoffs $\{r_i(\overline{w}_{ij})\}$ of J'_i intervals. The following is our key result:

THEOREM 4.1. *Assume that* $\overline{w}_{ij} \leq B, \forall i \in 1, \dots, n, \forall j \in 1, \dots, J'_i$ *and denote* $\overline{w}_{min} = \min\{\overline{w}_{ij}\}_{i=1,\dots,n; j=1,\dots,J'_i}$. *If* $\epsilon \leq \overline{w}_{min}/n$, *then, 1)* $OPT(J'_i, \overline{w}_{ij}) \geq \frac{1}{2} OPT(J_i, w_{ij})$; *and 2) the bound is tight.*

We prove this theorem in a series of steps.

LEMMA 4.2. $OPT(J_i, w_{ij}) \leq OPT(J'_i, \underline{w}_{ij}), \forall J'_i \leq J_i$.

PROOF. Since payoff increases with respect to weight, for channel i, any unexplored threshold h must be in one of the intervals we already discovered. Suppose it is in the interval $[\underline{w}_{ij}, \overline{w}_{ij}]$, where, $r(\underline{w}_{ij}) < r(\overline{w}_{ij})$. Clearly, option h $(w_{ih}, r(w_{ih}))$ (corresponding to threshold h) is dominated by option $(\underline{w}_{ij}, r(\overline{w}_{ij}))$, as the latter has lower cost but higher payoff. As to multiple channels, this suggests that

$OPT(J_i, w_{ij})$ is always upper-bounded by $OPT(J'_i, \underline{w}_{ij})$, although the number of intervals we identified is at most the number of thresholds: $J'_i \leq J_i$. □

LEMMA 4.3. *Assume that $\overline{w}_{ij} \leq B, \forall i \in 1, \ldots, n$, $\forall j \in 1, \ldots, J'_i$ and denote $\overline{w}_{min} = \min\{\overline{w}_{ij}\}_{i=1,\ldots,n; j=1,\ldots,J'_i}$. If $\epsilon \leq \overline{w}_{min}/n$, then, $OPT(J'_i, \underline{w}_{ij}) \leq 2OPT(J'_i, \overline{w}_{ij}), \forall J'_i \leq J_i$.*

PROOF. Let $\dot{Y} = \{\dot{y}_{ij}\}$ and $\hat{Y} = \{\hat{y_{ij}}\}$ be the optimal solution that corresponds to $OPT(J'_i, \underline{w}_{ij})$ and $OPT(J'_i, \overline{w}_{ij})$ respectively. Note that $\epsilon \geq \overline{w}_{ij} - \underline{w}_{ij}$, thus, we have

$$\sum_{i=1}^{n} \sum_{j=1}^{J'_i} \overline{w}_{ij} \dot{y}_{ij} \leq \sum_{i=1}^{n} \sum_{j=1}^{J'_i} \underline{w}_{ij} \dot{y}_{ij} + \epsilon \sum_{i=1}^{n} \sum_{j=1}^{J'_i} \dot{y}_{ij} \leq B + n\epsilon \quad (3)$$

where the last inequality holds due to the fact that $\dot{Y} = \{\dot{y}_{ij}\}$ is the optimal solution for $OPT(J'_i, \underline{w}_{ij})$, which satisfies the budget constraint. By the definition of $\overline{w}_{min}$ and the assumption

$$\epsilon \leq \frac{\overline{w}_{min}}{n} \quad (4)$$

we have $n\epsilon \leq \overline{w}_{min} \leq \overline{w}_{ij}, \forall i \in 1, \ldots, n, \forall j \in 1, \ldots, J'_i$.

Consider dropping any non-zero item s in $\dot{Y}$, the resulting solution must be feasible for the problem $MCKP(J'_i, \overline{w}_{ij})$ according to inequality (3), and bounded by $OPT(J'_i, \overline{w}_{ij})$. Thus, we have $\sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \dot{y}_{ij} - r_s \leq \sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \hat{y_{ij}}$, and by rearranging we get that

$$\sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \dot{y}_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \hat{y_{ij}} \leq r_s \quad (5)$$

We have assumed that $\overline{w}_i \leq B$, $\forall i \in 1, \ldots, n$, $\forall j \in 1, \ldots, J'_i$, so $\overline{w}_s \leq B$, which implies $r_s \leq \sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \hat{y_{ij}}$. By inequality (5) we know that

$$\sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \dot{y}_{ij} - \sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \hat{y_{ij}} \leq \sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \hat{y_{ij}}$$

and thus $\sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \dot{y}_{ij} \leq 2 \sum_{i=1}^{n} \sum_{j=1}^{J'_i} r_{ij} \hat{y_{ij}}$. Hence, $OPT(J'_i, \overline{w}_{ij})$ is a 2-approximation of $OPT(J'_i, \underline{w}_{ij})$. □

PROOF OF THEOREM 4.1. Part 1 of the theorem follows directly from Lemmas 4.2 and 4.3. For Part 2, we show that the factor-2 bound is *tight*. Consider a simple example, in which we are given a budget of 1 to advertise in only two channels, such that, $w_1 = 1/2 - \delta$, $w_2 = 1/2 + \delta$, and, $r_1 = r_2 = 1$, and $0 < \delta < 1/2$. Recall that in our setting, both w_1 and w_2 are unknown, but instead we use their upper bounds $\overline{w}_1$ and $\overline{w}_2$. If the upper bounds are not identical to the actual weights, we can only choose one of the channels. Therefore, the approximate problem gives us at most a payoff of 1, whereas we can get a payoff of 2 when exact weights are known by choosing both channels. Moreover, the example also suggests that one will never be able to get a better than 2-approximation no matter how small ϵ is. □

Note that $MCKP(J'_i, \overline{w}_{ij})$ is also an NP-hard problem. Suppose that we use a c-approximation algorithm to solve $MCKP(J'_i, \overline{w}_{ij})$, then the statement below naturally follows from Theorem 4.1.

COROLLARY 4.4. *Assume that $\overline{w}_{ij} \leq B, \forall i \in 1, \ldots, n$, $\forall j \in 1, \ldots, J'_i$ and denote $\overline{w}_{min} = \min\{\overline{w}_{ij}\}_{i=1,\ldots,n; j=1,\ldots,J'_i}$. If $\epsilon \leq \overline{w}_{min}/n$, a c-approximation algorithm for $MCKP(J'_i, \overline{w}_{ij})$ achieves at least 1/2c of the optimal value of $MCKP(J_i, w_{ij})$.*

PROOF. Let G be the near-optimal solution value given by the c-approximation algorithm for $MCKP(J'_i, \overline{w}_{ij})$, then $G \geq \frac{1}{c} OPT(J'_i, \overline{w}_{ij})$. From Theorem 4.1, we know that $OPT(J'_i, \overline{w}_{ij}) \geq \frac{1}{2} OPT(J_i, w_{ij})$. Thus, it must be the case that $G \geq \frac{1}{2c} OPT(J_i, w_{ij})$. □

5. QUERY STRATEGIES FOR BUDGET ALLOCATION

Our analysis so far assumed that we have been given a set of intervals for MCKP weights w_{ij} (that is, thresholds at which the response function jumps in value) which are sufficiently tight, in the sense of Condition (4), to ensure a 2-approximation using just the interval upper bounds in an MCKP. The key next question, which we now address, is how to obtain such intervals efficiently using a sequence of simulation queries. First, observe that there is a straightforward query mechanism which can produce intervals of arbitrary width in linear time: finely discretize each channel in the interval $[0, B]$, and query each discrete value for each channel independently. However, this approach can be extremely wasteful: for example, one channel can yield a small response and require a minimal investment of B; in most cases, we can quickly discover this and ignore this channel altogether. We will propose a more intelligent query algorithm which *interleaves* MCKP computation with queries. This allows more efficient exploration of the allocation space, and early termination once a near-optimal allocation is found.

5.1 Iterative Budgeting Algorithm

Data: maximum iteration K, total budget B,
  parameter θ
Result: budgeting plan $P_b = \{b_i\}_{i=1,\ldots,n}$
$\mathcal{I}_i \leftarrow \emptyset, \forall i = 1, \ldots, n$;
$b_i \leftarrow 0, \forall i = 1, \ldots, n$;
$k \leftarrow 0$;
foreach *channel* $i \in 1, \ldots, n$ **do**
  $v_0 \leftarrow (0, 0)$;
  $v_B \leftarrow (B, r_i(B))$;
  $\mathcal{I}_i \leftarrow \mathcal{I}_i \cup (v_0, v_B)$;
end
while $k < K$ **do**
  $\{\overline{w}_{ij}\} \leftarrow U_{bs}(\mathcal{I}_i)$, $\forall i \in 1, \ldots, n$;
  $\{\underline{w}_{ij}\} \leftarrow L_{bs}(\mathcal{I}_i)$, $\forall i \in 1, \ldots, n$;
  $\underline{y} \leftarrow MCKP(J'_i, \overline{w}_{ij})$;
  $\bar{y} \leftarrow MCKP(J'_i, \underline{w}_{ij})$;
  $P_b \leftarrow \underline{y} \circ \bar{w}$;
  if $\bar{y} \cdot r - \underline{y} \cdot r \leq \theta$ **then**
    break;
  end
  foreach *channel* $i \in 1, \ldots, n$ **do**
    $updateIntervals(\mathcal{I}_i)$;
  end
  $k \leftarrow k + 1$;
end
return P_b

Algorithm 1: Iterative Budgeting Algorithm.

Now we present our primary algorithm, termed Iterative Budgeting (IB); it is given as Algorithm 1. In Algorithm 1, $\mathcal{I}_i$ stands for the set of intervals for channel i. Moreover, an interval is represented as a tuple (v_l, v_u), consisting of a lower bound $v_l = (w_l, r(w_l))$ and an upper bound $v_u = (w_u, r(w_u))$. $r_i(\cdot)$ is the step-wise payoff function for channel i evaluated by simulation. For any $\mathcal{I}_i$, $U_{bs}(\cdot)$ and $L_{bs}(\cdot)$ are simply functions that return a set of upper-bound and lower-bound weights respectively. $\underline{y}$ and $\bar{y}$ are the solutions for $MCKP(J_i', \overline{w}_{ij})$ and $MCKP(J_i', \underline{w}_{ij})$ respectively (e.g., using CPLEX). Each round, a new budget allocation plan $P_b = \underline{y} \circ \bar{w} = \{b_i = \sum_j y_{ij}\bar{w}_{ij}\}$ is computed. The method $updateIntervals(\cdot)$ updates the set of intervals by sending more queries to the specified channel simulator, which will be described in more details in the next section.

The algorithm starts with one interval per channel and uses the upper-bound weights and payoffs to compute an initial solution. Next, in each iteration, it sends more queries to each channel and updates $\mathcal{I}_i$ with returned payoffs, and computes a new solution. As it runs more iterations, and the set of intervals is further refined, generally, the solved payoff $(\underline{y} \cdot r)$ will approach its upper-bound of the optimum $(\bar{y} \cdot r)$, where the notation $a \cdot b$ is the dot product of two vectors. θ is a parameter that controls the solution quality. Specifically, Lemma 4.2 shows that $OPT(J_i', \underline{w}_{ij})$ is an online upper bound for $OPT(J_i, w_{ij})$. This bound is very useful, since $OPT(J_i, w_{ij})$ is unknown; we can compute $OPT(J_i', \underline{w}_{ij})$ (suppose this is computationally feasible) and use

$$\Delta OPT = OPT(J_i', \underline{w}_{ij}) - OPT(J_i', \overline{w}_{ij})$$

to assess the quality of the approximate solution. In particular, when $\Delta OPT \leq OPT(J_i', \overline{w}_{ij})$, we are guaranteed to have a 2-approximation. This is because: $OPT(J_i', \underline{w}_{ij}) - OPT(J_i', \overline{w}_{ij}) \leq OPT(J_i', \overline{w}_{ij})$ and thus $OPT(J_i', \underline{w}_{ij}) \leq 2OPT(J_i', \overline{w}_{ij})$. As $OPT(J_i, w_{ij}) \leq OPT(J_i', \underline{w}_{ij})$, we have $OPT(J_i, w_{ij}) \leq 2OPT(J_i', \overline{w}_{ij})$. Consequently, we are guaranteed to get a 2-approximation if we set $\theta = \bar{y} \cdot r = OPT(J_i', \overline{w}_{ij})$. In practice, we set a much smaller θ to obtain a better solution.

5.2 Query Strategies

Now we discuss how Algorithm 1 updates intervals by queries. We first notice that in the case of a single threshold per channel, binary search is more efficient than linear search. Given budget B, using binary search to obtain an interval width ϵ, one has to send at most $\lfloor \log(B/\epsilon) \rfloor + 1$ queries (1 initial query for $r_i(B)$ and $r_i(0) = 0$). In particular, if $\exists i \in \mathbb{Z}^+$, such that $\epsilon = B(\frac{1}{2})^i$, then we need exactly $i+1$ queries to guarantee ϵ, where $i = \log(B/\epsilon)$. In contrast, using linear search, we need B/ϵ queries to achieve an interval width of ϵ, which is clearly less efficient than the binary search. The task is to extend binary search to handle the case that a channel could have a finite number of thresholds.

5.2.1 Generalized Binary Query Algorithm

We propose a Generalized Binary Query (GBQ) Algorithm (see Algorithm 2) to implement $updateIntervals(\cdot)$, which extends the binary search method to the case of multiple thresholds.

The algorithm scans each interval and creates a new query using a weight that is halfway between the lower and upper bound weights. In other words, it takes one binary search action within each interval available at an iteration. If the queried payoff equals to the lower (upper) bound payoff, then it updates the lower (upper) bound weight accordingly. Otherwise, a new interval is added to the current set of intervals $\mathcal{I}_i$. Therefore, the query action has two effects: 1) narrowing existing intervals, and 2) identifying new intervals (thresholds).

```
Data: a set of intervals I_i
Result: updated I_i
I'_i ← ∅
foreach interval (v_l, v_u) in I_i do
    w' ← (w_l + w_u)/2;
    r' ← (w');
    if r' = r_l then
        v_l ← (w', r_l);
    else if r' = r_u then
        v_u ← (w', r_u);
    else
        v' ← (w', r');
        I'_i ← I'_i ∪ (v_l, v');
        I'_i ← I'_i ∪ (v', v_u);
        I_i ← I_i \ (v_l, v_u);
end
return I'_i ∪ I_i
```

Algorithm 2: Generalized Binary Query (GBQ) Algorithm: updateIntervals($\mathcal{I}_i$).

Note that, to obtain an interval width of ϵ, we could also send B/ϵ queries using a simple linear search. Suppose a channel has J thresholds, $\{w_i\}_{i=1,\cdots,J}$, such that, $w_1 < w_2 <, \cdots, < w_J$. Define $d_{min} = \min_j\{w_j - w_{j-1}\}$, which is the minimum distance between two adjacent thresholds. Theorem 5.1 states that GBQ is more efficient than linear search.

THEOREM 5.1. *Assume $\epsilon = B(\frac{1}{2})^i$, where $i \in Z^+$. If $\epsilon \geq d_{min}$, in the worst case, GBQ uses the same number of queries as linear search; however, if $\epsilon < d_{min}$, GBQ always uses fewer queries.*

PROOF SKETCH. [1] First, we notice that for a given ϵ that satisfies $\epsilon = B(\frac{1}{2})^i$, besides the linear search with B/ϵ queries, we can use a naive binary search as follows: first, we query $r(B)$; next, $r(B/2)$; then, $r(B/4)$ and $r(3B/4)$, and so on. In total, we need $i+1$ iterations to achieve ϵ. In terms of queries, we need: $1 + 1 + 2 + \ldots + 2^{(i-1)} = 2^i$, which is exactly B/ϵ.

When $\epsilon \geq d_{min}$, it is possible that each query will find a new interval, and thus at most 2^i queries are required to obtain intervals of width ϵ. However, when $\epsilon < d_{min}$, we are guaranteed that GBQ has captured all thresholds at some iteration $\hat{i} < i$. For any iteration after $\hat{i}$, it only sends a fixed number of queries (equal to the actual number of thresholds), as the generalized binary search will skip intervals that do not have thresholds. By contrast, the naive binary search (equivalent to linear search as we have shown) has to explore all intervals available for each iteration. Therefore, if $\epsilon < d_{min}$, GBQ always uses fewer queries. □

Notice that as the IB algorithm iterates using the GBQ strategy, in each iteration it maintains several intervals for each channel with uniform length (ϵ in Section 4). We would

[1] A complete proof is available at: `https://github.com/haffwin/mcmo/blob/master/mcmo/mcmo_append.pdf`

expect that in some iteration, it will satisfy Condition (4) and ensure the 2-approximation in Theorem 4.1. Formally, we show this to be the case in Corollary 5.2, where k is the number of iterations of IB.

COROLLARY 5.2. *When $k \geq \log \frac{Bn}{w_{min}}$, IB algorithm with GBQ strategy achieves a 2-approximation of $MCKP(J_i, w_{ij})$, where k is the number of iteration.*

PROOF. Based on GBQ, we know that ϵ, the width of any interval satisfies: $\epsilon = B/2^k$. As Condition 4 requires that: $\epsilon \leq w_{min}/n$, we have $B/2^k \leq w_{min}/n$, which is $k \geq \log \frac{Bn}{w_{min}}$. □

5.2.2 Heuristic Binary Query Algorithm

Notice that using the query Algorithm 2, Algorithm 1 needs to send an increasing number of queries for each channel in each iteration. Although the number of queries required for an iteration will eventually be bounded by the actual number of thresholds, most queries are wasteful, especially when the solution approaches its optimum. To improve efficiency of the query search we propose a Heuristic Binary Query Algorithm (HBQ), described in Algorithm 3. Notably, as distinct from GBQ, HBQ only sends *one* query per channel to the "most profitable" interval in each iteration, where this channel is determined by $\hat{y}$, computed as:

$$\hat{y} \leftarrow MCKP(J'_i, (\underline{w}_{ij} + \overline{w}_{ij})/2).$$

That is, $\hat{y}$ is the solution of MCKP which uses the *average* of lower-bound and upper-bound weights, but the upper-bound payoffs of all intervals. Intuitively, $\hat{y}$ gives the highest payoff we would obtain in the next iteration if we query all current available intervals in a binary manner. Notably, if IB is to use HBQ, it is modified by using $\hat{y}$ in place of $\bar{y}$.

Data: a set of intervals $\mathcal{I}_i$, solution of best payoff in next iteration $\hat{y}$
Result: updated $\mathcal{I}_i$
$\mathcal{I}'_i \leftarrow \emptyset$
foreach *interval* (v_l, v_u) *in* $\mathcal{I}_i$ **do**
  if *item l is not selected in $\hat{y}$* **then**
    continue;
  end
  $w' \leftarrow (w_l + w_u)/2$;
  $r' \leftarrow (w')$;
  if $r' = r_l$ **then**
    $v_l \leftarrow (w', r_l)$;
  else if $r' = r_u$ **then**
    $v_u \leftarrow (w', r_u)$;
  else
    $v' \leftarrow (w', r')$;
    $\mathcal{I}'_i \leftarrow \mathcal{I}'_i \cup (v_l, v')$;
    $\mathcal{I}'_i \leftarrow \mathcal{I}'_i \cup (v', v_u)$;
    $\mathcal{I}_i \leftarrow \mathcal{I}_i \backslash (v_l, v_u)$;
end
return $\mathcal{I}'_i \cup \mathcal{I}_i$

Algorithm 3: Heuristic Binary Query (HBQ) Algorithm: updateIntervals($\mathcal{I}_i$, $\hat{s}$).

6. EXPERIMENTS

In order to evaluate our approach we developed a multichannel marketing simulator. In this simulator, a fixed budget is allocated to advertise a technology over four channels: *door-to-door*, *keyword auction*, *direct mailing*, and *broadcast*, where the payoff for each channel is modeled by a step function of the allocated budget and corresponds to the total number of adopters of the technology. One of the sources of complexity is that response is a function not merely of those reached by marketing directly, but also by those indirectly affected through social influence. Below we describe the details of the simulator.

The target population was comprised of 536 households in San Diego county, CA which is shared among the marketing channels.[2] The full marketing campaign was restricted to 3 months. As our baseline we implemented the well-known simulated annealing (SA) algorithm which was tuned to our problem domain [6]. Our algorithm used CPLEX 12.6.1 through a Java API to solve MCKP, and experiments were run on an Ubuntu Linux 64-bit PC with 32 GB RAM and four 8-core Intel Xeon 2.1 GHz CPUs.[3]

6.1 Marketing Simulator

Door-to-Door Marketing.

In door-to-door marketing, a sales agent knocks on a customer's door and attempts to initiate a discussion that could eventually lead to a sale. To simulate relevant marketing decisions, we adopt the door-to-door marketing route planner developed by [23], which uses an independent cascades (IC) model of social influence [13] to model the spread of successful adoptions by individuals to their geographic neighbors. The key parameter in the IC model is the *transmission probability* p representing the likelihood of a newly adopted customer affecting its socially-connected neighbors. For a fixed budget allocated to this channel, we follow Zhang and Vorobeychik in approximately computing the optimal routes for sales people to maximize influence using a greedy algorithm [23]. The step-function nature of the channel response arises from the fixed cost needed to add a sales person with sufficient allocated time to cover at least one household which has not already been covered by others.

Keyword Auction.

The keyword auction simulator mimics bidding decisions in a search engine keywords auction, analogous to Google AdWords. In the simulation, a marketer is given a fixed budget b to bid on k predefined keywords (relevant to the promoted product). A higher bid may help secure a higher position for one's advertised content in the search results page, and thus a higher *click-through* rate. However, the bid needs to be increased sufficiently to jump to a higher slot—hence the step-function nature of the response.

A keyword i is represented by a tuple (c_i, f_i), where c_i is its *cost-per-click* and f_i the fraction of users in the targeted region who click on the ads each day. c_i and f_i are determined as follows: $c_i = i + e_i^1$ and $f_i = \alpha(i + e_i^2)$ for all i, where, α is a pre-defined coefficient, and e_i^1 and e_i^2 are random variables drawn from the uniform distribution on $[0, 1]$ i.i.d. for each i. In expectation, c_i is approximately proportional to f_i. Finally, let R_{oad} be the conversion rate, or fraction of clicks on an ad that result in a purchase.

[2]Population was restricted primarily to restrict the time for each simulation run to enable a sufficient number of total runs for meaningful comparisons.
[3]Implementation of the marketing simulator and algorithms is available at: `https://github.com/haffwin/mcmo.git`.

We assume that for a given budget b, the marketer splits it equally over the 3-month period, and solves the following *fractional knapsack problem* every day to maximize the total number of clicks (since conversion rate is constant):

$$\max_{x} \sum_{i=1}^{k} x_i f_i \tag{6a}$$

$$M \sum_{i=1}^{k} x_i c_i f_i \leq b/d \quad \forall\, i, \tag{6b}$$

where x is the vector corresponding to the budget split, M is the total population size, and b/d the daily budget.

Direct Mailing.

A direct mailing marketer typically starts with a dataset of customer demographic information and purchase records, builds statistical models to predict *response rate*, and then ranks customers in descending order of response rate to send solicitations [7]. Our direct mailing simulator uses this strategy after randomly assigning a response rate to each customer from a uniform distribution on $[0, 2\bar{r}]$, where $\bar{r}$ is the pre-defined average response rate. In the simulation, an advertiser runs a direct mailing campaign weekly using an identical budget (i.e., the total budget is split equally among the 13 weeks comprising 3 months). In addition, we use a channel-specific conversion rate R_{dml} (probability of response to the ad).

Broadcast Marketing Simulator.

A marketer can also choose to advertise a product over a broadcast channel, such as TV or radio. In the simulation, we assume the marketer is facing J_b broadcast advertising options. Each option is represented by a tuple (c_j, r_j), with cost $c_j = e_j \bar{B}$ and response rate $r_j = \frac{1-e^{-\beta c_j}}{1+e^{-\beta c_j}}$ for each advertising option j, where, e_j is a random number drawn from a uniform distribution from 0 to 1, and $\bar{B}$ is the maximum budget allowed in the system. The distribution of r_j follows a *tanh* function with respect to c_j (essentially, a rescaled *logistic* function to ensure output is within [0,1] for positive values of cost c_j), with an exogenously specified coefficient β. The final parameter, conversion rate, is denoted by R_{brc}.

Similar to the other simulators, an advertiser divides the budget b equally into $w \leq 13$ weeks. Each week, with a budget b/w, the marketer chooses the best option (in terms of r_j). The expected response each week is the response rate multiplied by the market size. Finally, the marketer decides on an optimal duration of the marketing campaign in weeks, w^*.

6.2 Results

This section presents experimental results for 7 simulation configurations reflecting variability of relative channel efficacy for two channels, door-to-door marketing and keyword auctions. The parameter values are the transmission probability p in the former and conversion rate R_{oad} for the latter. The values of these for each configuration are given in Table 1.[4] Throughout, the conversion rates of direct mailing and broadcast marketing are 0.05 and 0.1 respectively.

[4]Experimental results regarding other parameters are provided in a supplement at: `https://github.com/haffwin/mcmo/blob/master/mcmo/mcmo_append.pdf`.

configuration (ID)	door to door (p)	online ads (R_{oad})
0	0.1	0.05
1	0.08	0.05
2	0.12	0.05
3	0.12	0.03
4	0.12	0.07
5	0.12	0.01
6	0.14	0.05

Table 1: Parameter configurations.

Figure 1 shows the simulated payoffs as a function of budget for each channel for configuration 0, where thresholds correspond to the "steps", and each step-wise line is consists of the simulated payoffs for the corresponding budget. The step-wise output suggests the strong combinatorial nature of our optimized objective.

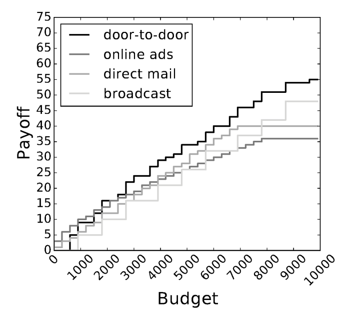

Figure 1: Simulated payoffs for Configuration 0 of four channels: door-to-door, online ads, direct mail and broadcast.

Figures 2 and 3 compare utility (overall rewards of the four channels) and running time of GBQ, HBQ and SA for different budgets, varying channel parameters of door-to-door marketing and keyword auction channels respectively. In all experiments, GBQ achieves the highest payoffs, but at a significant computational cost. Notably, HBQ is typically by far the most efficient in terms of computation time, and achieves utility that is always nearly optimal. In addition, GBQ is typically significantly more time efficient than SA, which does eventually achieve a near-optimal utility as well, but after considerable computing time.

7. CONCLUSION

We presented a novel discrete optimization framework to address the budget allocation challenge faced in multi-channel marketing, where channel-wise payoffs exhibit strong budget complementarities. We showed that the budget optimization problem in our setting can be transformed into a well-known multi-choice knapsack problem, which can be solved effectively using state-of-the-art MILP solvers. We then introduced effective approximation and query schemes (GBQ and HBQ) when the response function of multi-channel marketing is represented using a simulation model, where weights of knapsack items can only be bounded through queries. We showed that the transformed (multi-choice) knapsack problem using the upper-bound knapsack weights is a tight 2-approximation to the optimum with exact thresholds, when the weights satisfy mild conditions. We implemented our

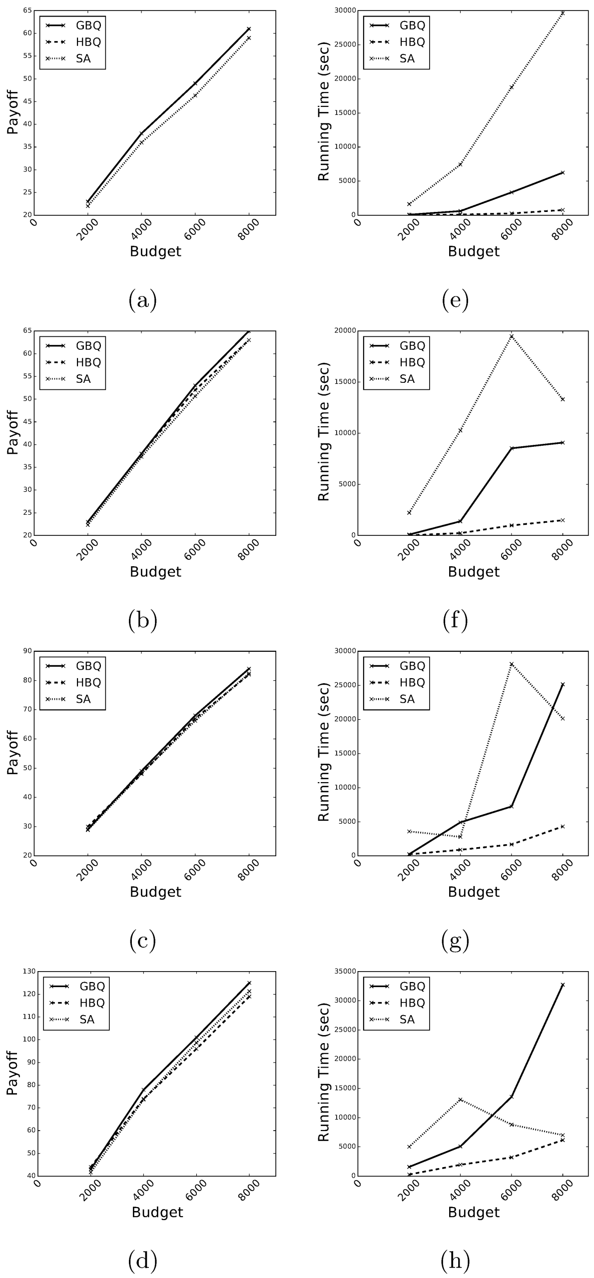

Figure 2: Payoff (a)-(d) & run time (e)-(h) comparison among algorithms over different budgets for different door-to-door marketing parameters. (a), (e) $p = 0.08$. (b), (f) $p = 0.1$. (c), (g) $p = 0.12$. (d), (h) $p = 0.14$.

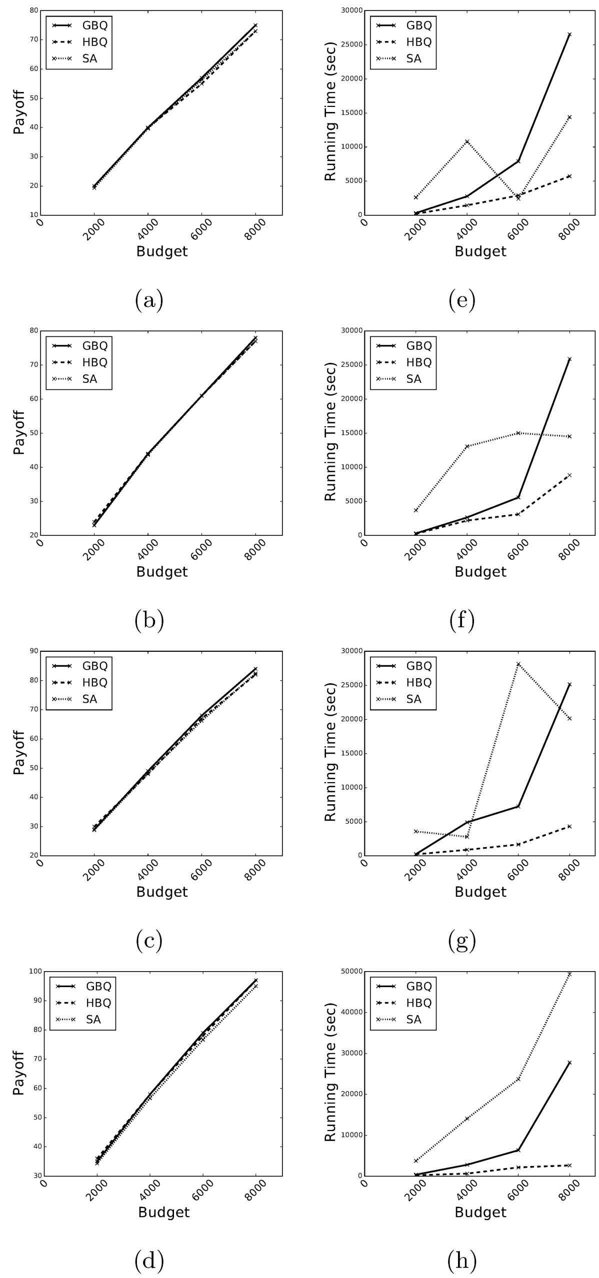

Figure 3: Payoff (a)-(d) & run time (e)-(h) comparison among algorithms over different budgets for different online ads marketing parameters. (a), (e) $R_{oad} = 0.01$. (b), (f) $R_{oad} = 0.03$. (c), (g) $R_{oad} = 0.05$. (d), (h) $R_{oad} = 0.07$.

framework in a simulated marketing platform motivated by real-world multi-channel marketing campaigns in a real geographical area. We conducted extensive experiments on different marketing configurations and showed that the proposed query algorithms significantly outperform a simulated annealing baseline.

8. 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 Office of Naval Research (N00014-15-1-2621), Army Research Office (W911NF-16-1-0069), the National Science Foundation (IIS-1526860, CNS-1640624) and the National Institutes of Health (UH2 CA203708-01). Procaccia was partially supported by the National Science Foundation (IIS-1350598, CCF-1525932), the Office of Naval Research, and a Sloan Research Fellowship.

REFERENCES

[1] Naoki Abe, Naval Verma, Chid Apte, and Robert Schroko. Cross channel optimized marketing by reinforcement learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 767–772. ACM, 2004.

[2] Nikolay Archak, Vahab Mirrokni, and S Muthukrishnan. Budget optimization for online advertising campaigns with carryover effects. In *Ad Auctions Workshop*, 2010.

[3] Adil Bagirov, Napsu Karmitsa, and Marko M Mäkelä. *Introduction to Nonsmooth Optimization: theory, practice and software*. Springer, 2014.

[4] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[5] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[6] Dimitris Bertsimas, John Tsitsiklis, et al. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.

[7] Indranil Bose and Xi Chen. Quantitative models for direct marketing: A review from systems perspective. *European Journal of Operational Research*, 195(1):1–16, 2009.

[8] Craig Boutilier and Tyler Lu. Budget allocation using weakly coupled, constrained Markov decision processes. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

[9] M.E. Dyer, W.O. Riha, and J. Walker. A hybrid dynamic programming/branch-and-bound algorithm for the multiple-choice knapsack problem. *Journal of Computational and Applied Mathematics*, 58(1):43–54, 1995.

[10] Marc Goerigk, Manoj Gupta, Jonas Ide, Anita Schöbel, and Sandeep Sen. The robust knapsack problem with queries. *Computers & Operations Research*, 55:12–22, 2015.

[11] Alan K Graham and Carlos A Ariza. Dynamic, hard and strategic questions: using optimization to answer a marketing resource allocation question. *System Dynamics Review*, 19(1):27–46, 2003.

[12] Mhand Hifi, Hedi Mhalla, and Slim Sadfi. Sensitivity of the optimum to perturbations of the profit or weight of an item in the binary knapsack problem. *Journal of Combinatorial Optimization*, 10(3):239–260, 2005.

[13] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[14] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[15] Michele Monaci and Ulrich Pferschy. On the robust knapsack problem. *SIAM Journal on Optimization*, 23(4):1956–1982, 2013.

[16] Scott A Neslin, Dhruv Grewal, Robert Leghorn, Venkatesh Shankar, Marije L Teerling, Jacquelyn S Thomas, and Peter C Verhoef. Challenges and opportunities in multichannel customer management. *Journal of Service Research*, 9(2):95–112, 2006.

[17] Scott A Neslin and Venkatesh Shankar. Key issues in multichannel customer management: current knowledge and future directions. *Journal of interactive marketing*, 23(1):70–81, 2009.

[18] Mustafa Ç Pínar. A note on robust 0-1 optimization with uncertain cost coefficients. *4OR*, 2(4):309–316, 2004.

[19] David Pisinger. A minimal algorithm for the multiple-choice knapsack problem. *European Journal of Operational Research*, 83(2):394–410, 1995.

[20] Arvind Rangaswamy and Gerrit H Van Bruggen. Opportunities and challenges in multichannel marketing: An introduction to the special issue. *Journal of Interactive Marketing*, 19(2):5–11, 2005.

[21] Prabhakant Sinha and Andris A Zoltners. The multiple-choice knapsack problem. *Operations Research*, 27(3):503–515, 1979.

[22] Yanwu Yang, Jie Zhang, Rui Qin, Juanjuan Li, Fei-Yue Wang, and Wei Qi. A budget optimization framework for search advertisements across markets. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 42(5):1141–1151, 2012.

[23] Haifeng Zhang and Yevgeniy Vorobeychik. Submodular optimization with routing constraints. In *AAAI Conference on Artificial Intelligence*, pages 819–825, 2016.

[24] Huiyuan Zhang, Thang N. Dinh, and My T. Thai. Maximizing the spread of positive influence in online social networks. In *International Conference on Distributed Computing Systems*, pages 317–326, 2013.