ECAI 2010
H. Coelho et al. (Eds.)
IOS Press, 2010

doi:10.3233/978-1-60750-606-5-1013

A Knapsack-Based Approach to Bidding in Ad Auctions

Jordan Berg, Amy Greenwald, Victor Naroditskiy,[1] Eric Sodomka[2]

Abstract.

We model the problem of bidding in ad auctions as a penalized multiple choice knapsack problem (PMCKP), a combination of the multiple choice knapsack problem (MCKP) and the penalized knapsack problem (PKP) [1]. We present two versions of PMCKP, GlobalPMCKP and LocalPMCKP, together with a greedy algorithm that solves the linear relaxation of a GlobalPMCKP optimally. We also develop a greedy heuristic for solving LocalPMCKP. Although our heuristic is not optimal, we show that it performs well in TAC AA games.

1 INTRODUCTION

Most Internet advertising space is sold via ad auctions, which operate as follows: Well in advance, a publisher (e.g., a search engine or a newspaper's web site) solicits bids from advertisers on potential queries. Then, when a user submits one of those queries to a publisher, the user is shown sponsored search results (i.e., ads) alongside organic results. The order in which the ads are displayed is determined on-the-fly by the publisher, who runs an auction based on the previously submitted bids. Finally, if the user clicks on an ad, the corresponding advertiser pays the publisher some amount.

In this environment, both publishers and advertisers face difficult decision problems. The publishers must determine costs-per-click (CPCs), and in which slots to display the winning ads. The advertisers in turn must decide what queries to bid on and the maximum CPC they are willing to pay.

In 2009, the annual Trading Agent Competition (TAC) introduced a simulated market game that facilitates experimenting with autonomous bidding strategies for ad auctions. The TAC Ad Auctions (AA) game [2] captures many of the difficulties associated with bidding in real ad auctions, such as bidding on multiple queries and setting budgets in the presence of noisy, delayed, and only partial information. Our work makes use of the TAC AA framework to experimentally evaluate autonomous bidding strategies for ad auctions.

2 KNAPSACK PROBLEMS

MCKP Following [5], we view bidding in ad auctions as a kind of multiple choice knapsack problem (MCKP). MCKP is a generalization of the classic 0-1 knapsack problem (KP). In both problems, the objective is to fill a knapsack of finite capacity with weighted items in a way that maximizes the total value of the items taken. MCKP generalizes KP in that the given items are divided into sets, and at most one item can be taken from each set. To see how MCKP naturally models bidding in ad auctions, think of slots on a query's search results page as a set of items; then, taking one item from each set corresponds to choosing one slot per query. The sales target corresponds to the knapsack's capacity, and the expected number of sales associated with a slot corresponds to its weight.[3]

Let Q denote the set of all the given item sets (e.g., queries), and let R_q denote an enumerable set of items (e.g., slots) in the set $q \in Q$. Let $v_{qj} \in \mathbb{R}$ be the value associated with taking the jth item in the set q, let $w_{qj} \in \mathbb{R}^+$ be the weight of this item, and let x_{qj} be a binary decision variable that takes on value 1 iff this item is taken. Let $C \in \mathbb{R}$ denote the knapsack's capacity. Using this notation, we formulate MCKP as follows:

$$\max_{\vec{x}} \quad \sum_{q \in Q} v_{qj} x_{qj} \tag{1}$$

$$\text{subject to} \quad \sum_{q \in Q} \sum_{j \in R_q} w_{qj} x_{qj} \leq C \tag{2}$$

$$\sum_{j \in R_q} x_{qj} \leq 1 \quad \forall q \in Q \tag{3}$$

$$x_{qj} \in \{0, 1\}, \quad \forall q \in Q, j \in R_q \tag{4}$$

Much like KP, an approximate solution to MCKP can be obtained by running a greedy algorithm (GreedyMCKP) that solves the linear relaxation optimally [3]. GreedyMCKP actually converts an instance of MCKP into an instance of KP with *incremental* items (after removing dominated items; see [3]), and then solves the resulting KP in the usual greedy fashion: incremental items are taken in nonincreasing order of *efficiency*—defined as value divided by weight—until the knapsack has reached its capacity or there are no items left with positive efficiencies. The incremental items are so-called because taking the first k of them in the resulting KP is equivalent to taking the kth item in the original MCKP. In this way, a solution to this KP can be immediately interpreted as a solution to the original MCKP.

PKP The penalized KP [1] is another generalization of KP, where the hard constraint on capacity is replaced

[1] University of Southampton, UK; vnarodit@gmail.com
[2] Brown University, USA; {jberg,amy,sodomka}@cs.brown.edu

[3] Alternatively, the total advertising budget across queries could correspond to the knapsack's capacity, in which case a slot's expected cost-per-click times the expected number of clicks would correspond to its weight.

by a *global* penalty function p that depends on capacity used. PKP generalizes KP because a hard capacity constraint can be encoded in a penalty function that returns zero when the total weight of the items taken satisfies the constraint and an exceedingly high penalty otherwise.

We call the original PKP GlobalPKP because the penalty function is global. But in TAC AA the penalty function is not global; it is *local* in the sense that it differs across queries. We formulate LocalPKP as follows: given local penalty functions p_j for all $j = 1, \ldots, n$,

$$\max_{\vec{x}} \quad \sum_{j=1}^{n} (v_j - p_j(\kappa))\, x_j \tag{5}$$

$$\text{subject to} \quad x_j \in \{0,1\}, \quad \forall j = 1, \ldots, n \tag{6}$$

Here $\kappa = \sum_j w_j x_j$. (GlobalPKP is the special case of LocalPKP in which $p_j = p$, for all j.)

By redefining efficiency as marginal profit divided by weight, the usual greedy algorithm for solving the linear relaxation of the knapsack problem solves GlobalPKP optimally, assuming p is convex [1]. The *marginal profit* μ_j of an item j is the incremental value of that item less its incremental penalty. In the case of GlobalPKP, $\mu_j = v_j - p(\kappa') + p(\kappa)$. Here $\kappa' = \kappa + w_j$.

We extend this greedy algorithm (GreedyGlobalPKP) to one that heuristically solves LocalPKP (GreedyLocalPKP) by calculating marginal profit as follows:

$$\mu_j = v_j - \left(\sum_{\{i|x_i=1\}} p_i(\kappa') + p_j(\kappa') \right) + \left(\sum_{\{i|x_i=1\}} p_i(\kappa) \right)$$

Whereas GreedyGlobalPKP is $O(n \log n)$ (items are sorted by marginal profit once, in a preprocessing step), GreedyLocalPKP is $O(n^2)$ because marginal profit varies with capacity used differently for different items. This creates the need to first update marginal profits and then search through the items that have not yet been taken for one of maximal efficiency during each iteration. Still, GreedyLocalPKP does not solve LocalPKP optimally.

PMCKP Like MCKP, in PMCKP items are divided into sets, and at most one item can be taken from each set. Like PKP, penalty functions that depend on capacity used appear in the objective function. Formally, we define LocalPMCKP as follows: given values $v_{qj} \in \mathbb{R}$ and local penalty functions p_{qj}, for all q, j,

$$\max_{x_{qj}} \quad \sum_{q \in Q} \sum_{j \in R_q} (v_{qj} - p_{qj}(\kappa))\, x_{qj} \tag{7}$$

$$\text{subject to} \quad \sum_{j \in R_q} x_{qj} \leq 1 \quad \forall q \in Q \tag{8}$$

$$x_{qj} \in \{0,1\}, \quad \forall q \in Q, j \in R_q \tag{9}$$

(GlobalPMCKP is the special case of LocalPMCKP in which $p_{qj} = p$, for all q, j.)

Theorem If the penalty function p is convex, then GreedyMCKP with efficiency defined as marginal profit solves the linear relaxation of GlobalPMCKP optimally.

Proof Sketch It suffices to note that a global penalty function does not change the set of dominated items. □

We solve LocalPMCKP using yet another greedy algorithm, GreedyLocalPMCKP, which incorporates ideas from both GreedyMCKP and GreedyLocalPKP. First, we modify GreedyMCKP to eliminate dominated items based on their marginal profits instead of their values. But then we can no longer eliminate all dominated items in a preprocessing step (again, because marginal profit varies with capacity used differently for different items). Consequently, GreedyLocalPMCKP eliminates dominated items during each iteration. Second, following GreedyLocalPKP, GreedyLocalPMCKP, searches through the items that have not yet been taken for one of maximal efficiency during each iteration.

Eliminating dominated items requires sorting. Hence, GreedyLocalPMCKP is $O\left(\sum_{i=1}^{n} i \log i\right) = O(n^2 \log n)$, where n is the total number of items across item sets. In fact, in our experiments, we run HybridMCKP, an $O(n^2)$ algorithm that searches through the items that have not yet been taken during each iteration, but eliminates dominated items in a preprocessing step.

3 RESULTS AND CONCLUSION

The table below lists the mean scores and variances across 60 games played by the the best seven agents in the TAC AA agent repository and HybridMCKP. HybridMCKP placed second. We believe this result is of interest, and that it speaks to the strength of our approach to bidding in TAC AA, because unlike TacTex [4], HybridMCKP does not rely on any sophisticated modelling techniques (e.g., HMMs) tailored to the TAC AA game to estimate quantities like click probabilities and conversion probabilities, which are used to compute the values and penalty functions that define the TAC AA instance of PMCKP. Instead, HybridMCKP uses WEKA—off-the-shelf machine learning software. It remains to further study the performance of knapsack-based bidding heuristics using more sophisticated models.

Agent	Mean	Variance
TacTex	80.76	10.0
HybridMCKP	**77.83**	11.5
AstonTAC	76.30	9.2
Munsey	73.41	10.2
EPFLagent	72.43	9.8
QuakTAC	70.61	14.8
MetroClick	70.15	8.8
Mertacor	68.31	8.9

REFERENCES

[1] R. Freling, H.E. Romeijn, D.R. Morales, and A.P.M. Wagelmans, 'A branch-and-price algorithm for the multiperiod single-sourcing problem', *Operations Research*, **51**(6), 922–939, (2003).

[2] P.R. Jordan and M.P. Wellman, 'Designing an ad auctions game for the trading agent competition', in *Workshop on Trading Agent Design and Analysis*, (July 2009).

[3] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*, Springer, Berlin, Germany, 2004.

[4] D. Pardoe, D. Chakraborty, and P. Stone, 'TacTex09: A champion bidding agent for ad auctions', in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, (2010).

[5] Yunhong Zhou and Victor Naroditskiy, 'Algorithm for stochastic multiple-choice knapsack problem and application to keywords bidding', in *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 1175–1176, New York, NY, USA, (2008). ACM.