Online and Offline Optimization Algorithms for High-Speed Ad Allocation and Performance Benchmarking

Maggie Zhang[1,4], Arav Chadha[1, 4], Lindsay Kossoff[2, 4], Isabella De La Garza[3, 4], Alina Ene[4]

Canyon Crest Academy, 5951 Village Center Loop Rd, San Diego, CA 92130[1], Holton-Arms School, 7303 River Rd, Bethesda, MD 20817[2], United High School, 2811 United Ave, Laredo, TX 78045[3], Boston University, Commonwealth Ave, Boston, MA 02215[4]

Introduction

- **Generalized assignment problem (GAP)** is a well-known problem in combinatorial optimization
  - GAP seeks to find the optimal allocation of tasks to agents given a bipartite graph
- Our research focuses within the context of **ad allocation**, an application of GAP
  - Ad impressions, or single instances of an ad being displayed to a user, must be allocated to budget-constrained advertisers

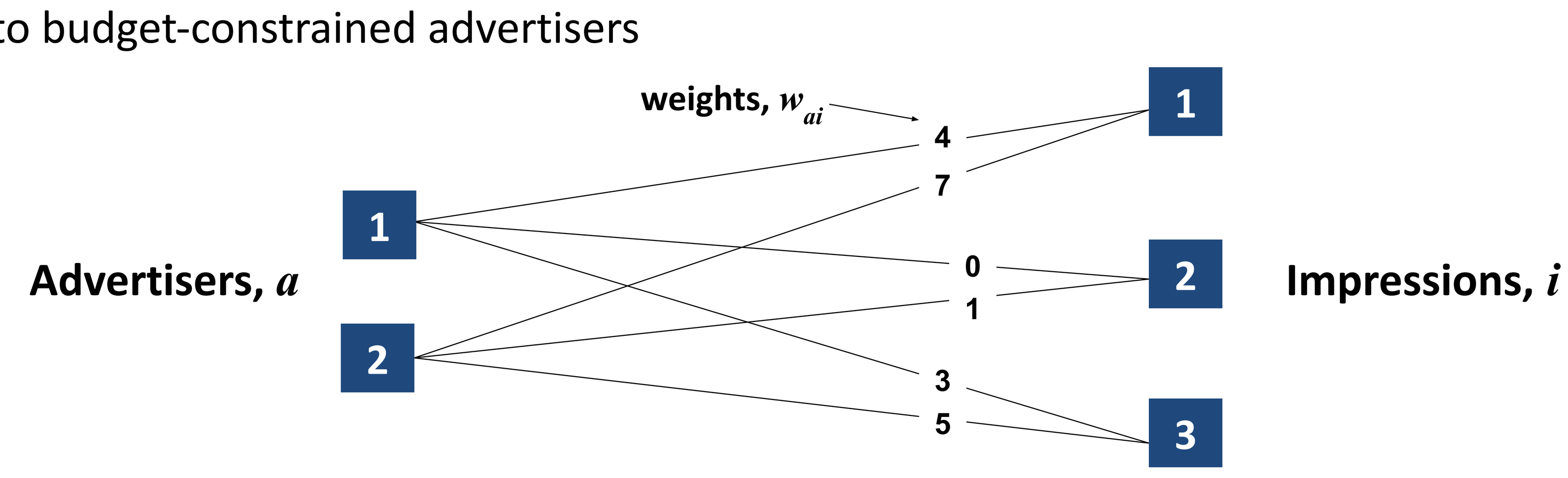

Fig 1. a bipartite graph representing advertisers and impressions

  - In online ad allocation, weights are calculated from user information (demographics, search queries, etc.) and arrive in real-time
  - In offline ad allocation, all weights are calculated and known beforehand
- We implement and compare an online algorithm integrating predictions and an offline algorithm with optimized parameters
  - These algorithms are analyzed for performance and efficiency on different data

Results

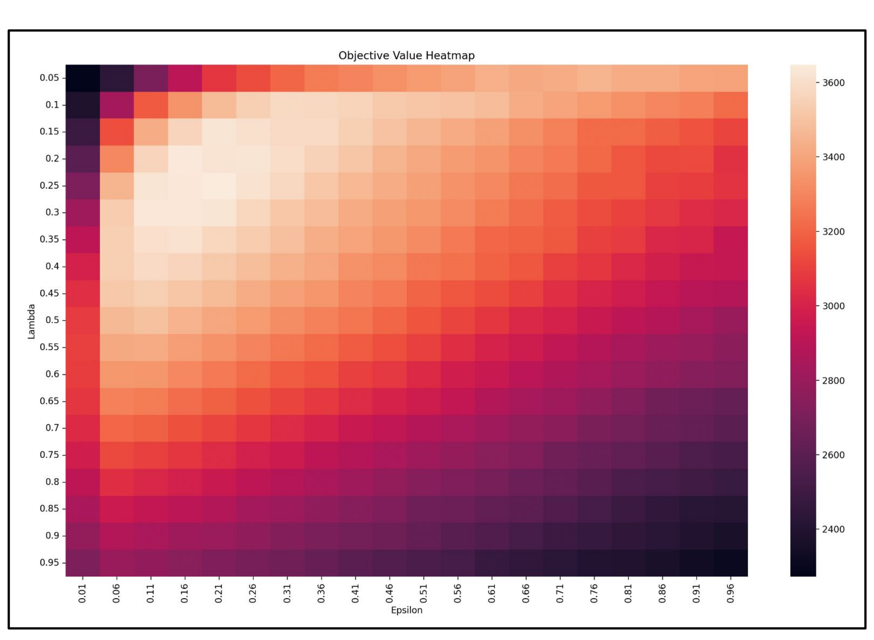

Fig 3. Heatmap, took ~90 minutes. Looped through epsilon (x-axis), starting at 0.01 and ending at 1.0 in increments of 0.05. Looped through lambda (y-axis), starting at 0.05 and ending at 1.0 in increments of 0.05. Set rounds = 50

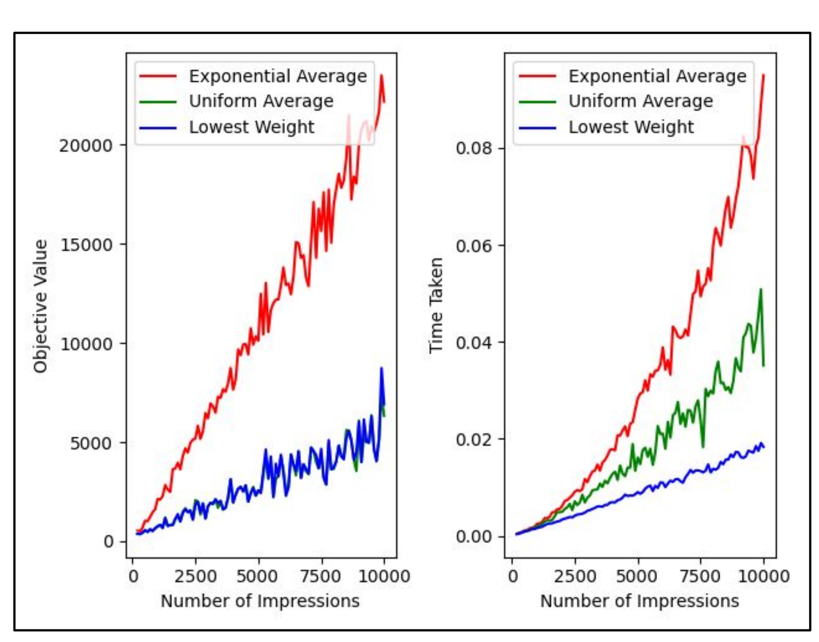

Fig 4. Graph comparing performance and efficiency for Alg. 1 using conservative, lowest weight, and uniform average update threshold steps for 100 advertisers and up to 10,000 impressions incrementing by 100

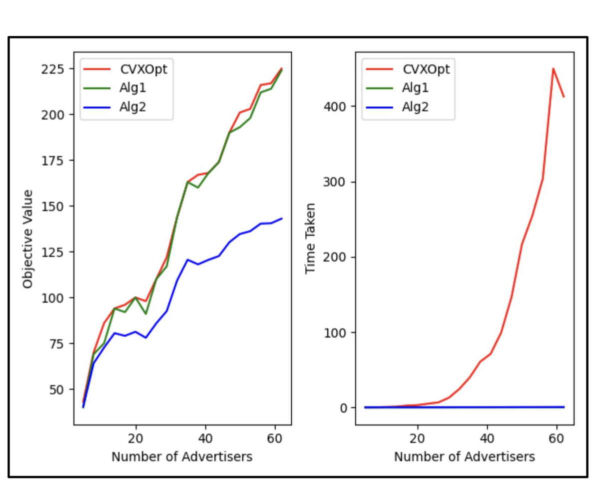

Fig 5. Graph comparing performance and efficiency for Alg. 1, Alg. 2, and CVXOPT on big data for up to 60 advertisers incrementing by 3 and a pre-determined number of impressions

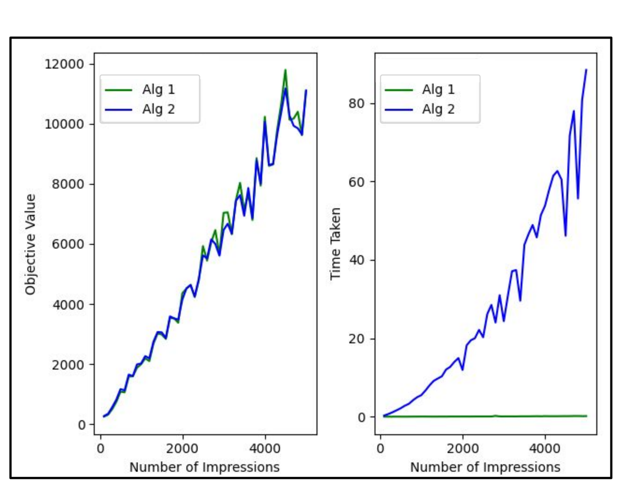

Fig 6. Graph comparing performance and efficiency for Alg. 1, Alg. 2, and CVXOPT on corrupted data for 50 advertisers and up to 5,000 impressions incrementing by 100

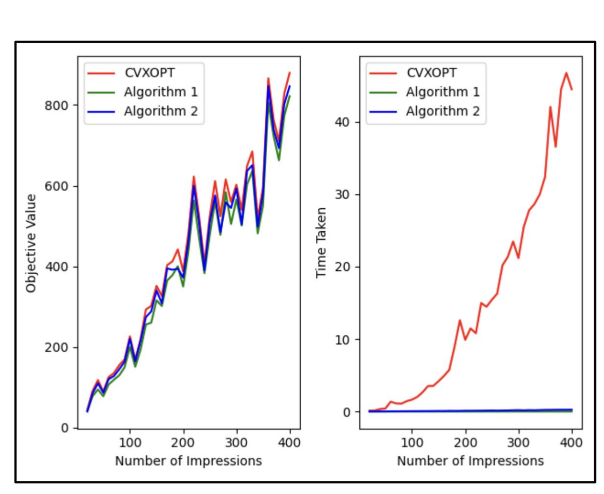

Fig 7. Graph comparing performance and efficiency for Alg. 1, Alg. 2, and CVXOPT on synthetic data for 20 advertisers and up to 400 impressions incrementing by 10

Discussion/Conclusion

Tuned Parameters: looking at Fig. 3, we identify that the Alg. 2 reaches optimal objective values when λ = 0.25 and ϵ = 0.21

Best Threshold: looking at Fig. 4, we determine that the conservative threshold update step outperforms the lowest weight and uniform average threshold update steps

Comparison on Performance and Efficiency: looking at Fig. 6, we see that Alg. 1 and Alg. 2 have similar performances. In Fig. 7, we see that Alg. 2 outperforms Alg. 1 on corrupted and synthetic data. Alg. 1 outperforms Alg. 2 on big data in Fig. 5; this could be attributed to Alg. 1's binary nature

Future Work:
- Incorporated mathematical predictions in Alg. 1, can shift to machine learning for "smarter" predictions in both algorithms
- Application of algorithms to bipartite graphs in other fields

Methods

Synthetic Data:
- Randomly assigned each impression a type using an exponential distribution, where each type is weighted differently by each advertiser
- Randomly assigned each advertiser a budget from an uniform distribution
- Created a weights array by using a nested for loop, first looping through each advertiser a, then each impression i, to get the weight of i for a
- Returned a list of advertisers, impressions, and weights

$$[w_{a_1 i_1}, w_{a_1 i_2}, \ldots\ w_{a_1 i_N},\ w_{a_2 i_1}, \ldots\ w_{a_N i_N}]$$

Fig 2. visualization of the weights array

Algorithm 1 (Online):
- Input: robustness-consistency trade-off parameter $\alpha \in [1, \infty)$, advertiser budgets $B_a \in \mathbb{N}$
- Define constants $B := \min_a B_{a'}$, $e_B := (1 + \frac{1}{B})^B$, and $\alpha_B := B\,(e_B^{\frac{\alpha}{B}} - 1)$
- Output: binary solution matrix

Algorithm 2 (Offline):
- Input: bipartite graph G of advertisers a, impressions i, and weights r, advertiser budgets C_a, parameters $\lambda, \epsilon \in (0, 1)$, and number of rounds R
- Initialize $\beta_a = (1 + \epsilon)^{-R}$, for all $a \in A$ and set $D_{i,a,\lambda} = e^{\frac{r_{i,a}}{\lambda}-1}$
- Output: fractional solution matrix

Algorithm 1:
For each a, initialize $\beta_a \leftarrow 0$
  For each arriving i
    Find expected a via $\text{argmax}_a(w_{ai} - \beta_a)$ and find predicted a using pred method
    Set a to $\text{argmax}_a(\alpha_B(w_{a(\text{PRD})i} - \beta_{a(\text{PRD})}), (w_{a(\text{EXP})i} - \beta_{a(\text{EXP})}))$
    Allocate i to a and remove least valuable i if B_a is exceeded
    Update $\beta_a \leftarrow \frac{e_{B_a}^{\alpha/B_a} - 1}{e_{B_a}^{\alpha} - 1} \sum_{i=1}^{B_a} w_i e_{B_a}^{\alpha(B_a - i)/B_a}$

Algorithm 2:
For rounds in R
  For each i, set allocation x

$$\mathbf{x}_{i,a} = \begin{cases} \beta_a D_{i,a,\lambda} & \text{if } \sum_{a' \in \mathbf{N}_i} \beta_{a'} D_{i,a',\lambda} \le 1 \\ \frac{\beta_a D_{i,a,\lambda}}{\sum_{a' \in \mathbf{N}_i} \beta_{a'} D_{i,a',\lambda}} & \text{otherwise} \end{cases}$$

For each a, update β_a
  $\text{alloc}_a \le \frac{\mathbf{C}_a}{(1+\epsilon)} \implies \beta_a := (1+\epsilon)\beta_a$
  $\text{alloc}_a \ge (1+\epsilon)\mathbf{C}_a \implies \beta_a := \frac{\beta_a}{(1+\epsilon)}$
For each a over budget
  Reduce $x_{i,a}$ until budget is met

CVXOPT:
- Python-based linear program solver used for convex optimization
- Returns the solution matrix

Testing:
- **Other data**: used corrupted and big data
  - Corrupted data: made by tampering with weights matrix (randomly removing edges, multiplying values by 100, etc.)
  - Big data: used and cleaned from Stanford Large Network Dataset Collection
- **Fine-tuning parameters**: generated heatmaps tuning λ and ϵ, where color represents varying objective values
- **Thresholds**: created lowest weight threshold update and uniform average threshold steps to compare against conservative paper method for Alg. 1
- **Comparison**: calculated objective values by dotting weights matrix with solution matrix for each algorithm and CVXOPT on same data instances

References

[1] Spaeh, F. and Ene, A. Online ad allocation with predictions. May 26, 2023. https://doi.org/10.48550/arXiv.2302.01827 (accessed August 6, 2024).

[2] Agrawal, S.; Mirrokni, V.; Zadimoghaddam, M. Proportional Allocation: Simple, Distributed, and Diverse Matching with High Entropy. 2018. https://proceedings.mlr.press/v80/agrawal18b/agrawal18b.pdf (accessed August 6, 2024).

[3] Andersen, M.; Dahl, J.; Vandenberghe, L. *CVXOPT*, version 1.3, 2023.

[4] Leskovec, J. and Krevl, A. Wikipedia adminship election data, 2008. Stanford Large Network Dataset Collection. https://snap.stanford.edu/data/wiki-Elec.html (accessed August 6, 2024).

Acknowledgements

I would like to express my deepest gratitude to Dr. Alina Ene for her unwavering support, patience, and mentorship throughout this entire research opportunity. I would also like to thank my lab partners for all their incredible support and collaboration.