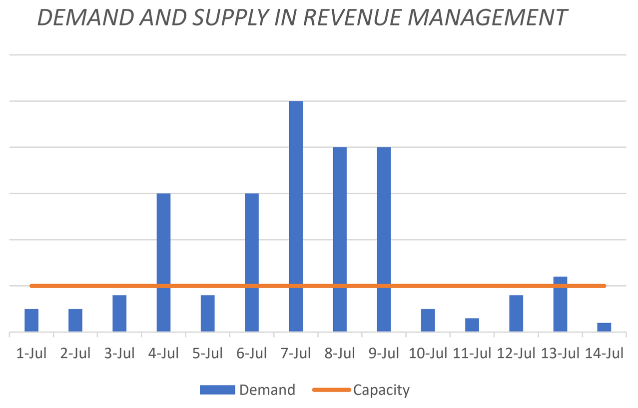
RESIDUAL ALGORITHMS VIA LINEAR PROGRAMMING NORMALIZATION

HAINING YU
AMAZON
*WORK DONE PRIOR TO JOINING AMAZON

RESEARCH MOTIVATION

- How to solve Approximate Dynamic Programming problems efficiently? [2] [3]
- How to improve residual algorithms? [1] [6]
- How to combine general-purpose and problem-specific approximation algorithms?

THE NETWORK INVENTORY CONTROL PROBLEM [8]

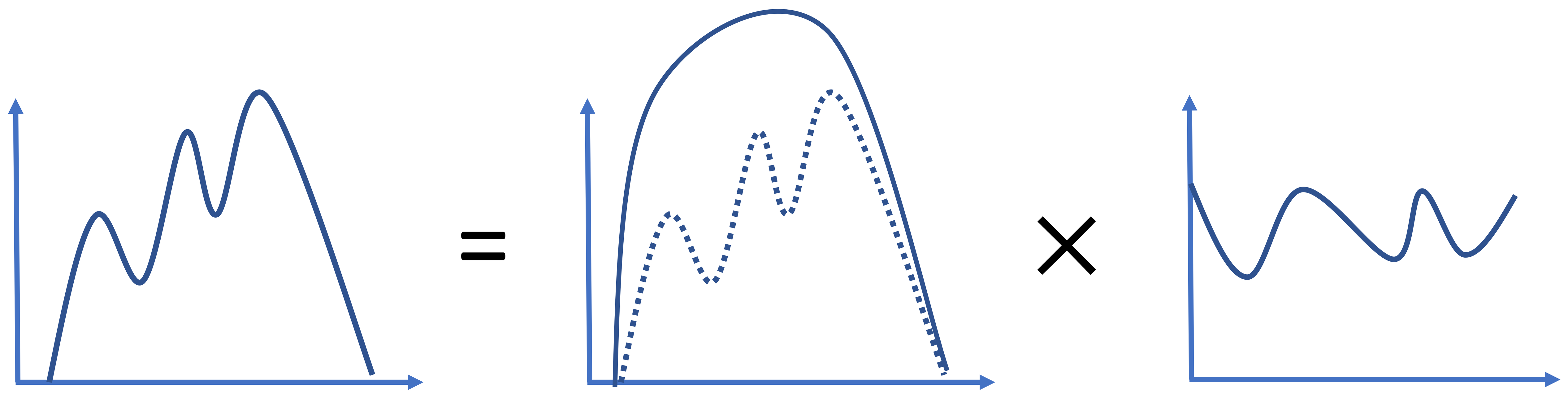


- Given inventory $\boldsymbol{x}(t)$ and a request for resource $\boldsymbol{a}$, should the algorithm accept or reject the request?

$$V_t(\mathbf{x}) = E\left\{\max_{u\in\Psi(\mathbf{x})}\left[r_n u(\mathbf{x},t,n) + V_{t+1}(\mathbf{x}-\mathbf{a}u)\right]\right\}$$

- High impact: 1 ppt improvement has significant financial benefit
- Well studied: current best practice is LP + DP decomposition [7]

MAIN IDEA



ORIGINAL PROBLEM

CUSTOM APPROXIMATION
FIRST ORDER EFFECT

DEEP NEURAL NETWORK
HIGHER-ORDER EFFECT

MATH

Approximate value function with $V_t^*(\mathbf{x}|\theta) = U_t(\mathbf{x})W_t(\mathbf{x}|\theta)$

where $U_t(\mathbf{x}) = \max \mathbf{r}^T\mathbf{y}$ s.t. $\mathbf{Ay} \leq \mathbf{x}$ and $0 \leq \mathbf{y} \leq \mathbf{p}^T t$.

Is the LP normalizer [4] [5] and *W* is the neural network

Minimize Bellman residual

$$U(\mathbf{x},t)W(\mathbf{x},t|\theta) - \sum_{0\leq n\leq N} p_n\left[\max_{u\in\Psi(\mathbf{x})}\left(r_n u + U_{t+1}(\mathbf{x}-\mathbf{a}_n u)W_{t+1}(\mathbf{x}-\mathbf{a}_n u|\theta)\right)\right]$$

IMPLEMENTATION DETAILS

- *Calculate U* on the fly using standard LP solver [4]
- Generate empirical average of Bellman residual via Monte Carlo simulation [5]
- Gradient descent via Adam
- Validation using discrete event simulation

PRELIMINARY RESULTS

CASE	M	N	T	$x(0)$	FIFO	LP BOUND	DECOMP	DNN
1	14	27	128,962	100	-10	-1.2	-	-0.3
2	14	39	47,811	500	-27.1	0.3	-	0.4
3	14	39	49,827	500	-32.8	-0.8	-	-0.3
4	14	39	18,684	200	-31.4	-1.5	-	0.3

- Optimality in same ballpark with current state-of-the-art approximation algorithm
- Savings on computation cost
- Scalability

NEXT STEPS

- Train general residual network across multiple problems
- Improve network architecture
- Test against pure deep neural networks
- Large scale industry experiment

REFERENCES

[1] Baird, L.~C., Residual Algorithms: Reinforcement Learning with Functional Approximation. ICML 1995
[2] Bertsekas, D.~P., Dynamic Programming and Optimal Control, Athena Scientific, 1995
[3] Bertsekas, D.~P., Tsitsiklis, J., Neuro-Dynamic Programming, Athena Scientific, 1996.
[4] Bertsimas, D., Popescu, I., Revenue Management in A Dynamic Network Environment, Transportation Sci. 37, 257–277
[5] de Farias, D.~P., Van Roy, B., The Linear Programming Approach to Approximate Dynamic Programming, Operations Research, 2003
[6] Geist, M., Piot, B., Pietquin, O. Is the Bellman Residual A Bad Proxy? NIPS 2017.
[7] Talluri, K., van Ryzin, G., An Analysis of Bid-price Controls for Network Revenue Management, Management Science. v44, 1998
[8] Talluri, K., van Ryzin, G., The Theory and Practice of Revenue Management. Springer, 2004