

# A Survey of Yao's Minimax Principle

Brian Shimanuki <sup>\*</sup>

Mayuri Sridhar <sup>†</sup>

Chelsea Voss <sup>‡</sup>

May 10, 2015

## Abstract

## 1 Introduction

## 2 Removable Online Knapsack Problem

In this section, we present the 2-competitive solution of Han et al. [1] to the removable online knapsack problem as well as use Yao's Principle show  $1 + 1/e$  is a lower bound on the competitiveness of any algorithm against an oblivious adversary.

The knapsack problem asks: Given a set of items  $e_i$  with weights  $w_i$  and values  $v_i$ , select a subset with the maximum sum of values such that the sum of the weights is at most the capacity of the knapsack. We call this the value of the knapsack. The offline version is a classic NP-hard optimization problem, though it has a simple dynamic programming pseudo-polynomial solution. Here, we consider the *online* version, meaning each item is given to the algorithm after the algorithm makes a decision on the previous item. *Removable* means the knapsack is a working set, from which we can remove items and add new items, but not add previously removed items. At all times, the knapsack is limited by its capacity. The objective is to maximize the value of the final knapsack. We say an algorithm is  $c$ -competitive if the expected value of the final knapsack is at least  $\frac{1}{c}$  times the expected value of the optimal knapsack.

We assume the knapsack has capacity 1. This is valid because only weights relative to the knapsack's capacity matters for the capacity constraint, so we can always scale down to a unit capacity.

---

<sup>\*</sup>bshimanuki@mit.edu

<sup>†</sup>mayuri@mit.edu

<sup>‡</sup>csvoss@mit.edu

## 2.1 Upper Bound

To prove an upper bound on the competitiveness of algorithms which yield a solution to the problem, we present a 2-competitive randomized algorithm.

Since the online nature of the problem is what makes it impossible to always achieve the optimal solution, it is insightful to first consider algorithms which behave similarly in both online and offline situations. We look at *MAX* and *GREEDY*.

*MAX* simply selects the item with the largest value whose weight does not exceed 1. Since in the offline case, we can process each item in an arbitrary order and keep track of the best item so far, in the online case, we can do exactly the same with the order given to us, storing the best item so far in our knapsack.

*GREEDY* selects the  $k$  items with the highest ratios of value to weight  $\frac{v_i}{w_i}$ , maximizing  $k$  while keeping the sum of the weights of the selected items at most 1. In the offline case, this can be done trivially by sorting the items by ratio and then selecting the highest  $k$  items. An online algorithm can perform at least as well by keeping a working set of the items with the highest ratio while processing each input.

---

**Algorithm 1:** online *GREEDY*

---

```

 $S \leftarrow \emptyset;$ 
foreach item  $e_i$ , in order of arrival do
  if  $w_i \leq 1$  then
     $S \leftarrow S \cup \{e_i\};$ 
    while  $\sum_{e_j \in S} w_j > 1$  do
      Remove the item from  $S$  with the smallest ratio

```

---

Since when the sum of the weights in  $S$  is greater than 1, an item with ratio less than that of the  $k$  items with the largest ratios must be in  $S$ , the  $k$  items will never be removed. Thus  $S$  is a superset of the items the offline *GREEDY* algorithm selects.

Neither of these algorithms alone are competitive. Consider an input sequence  $\{(w_i, v_i)\}$  of  $\{(\epsilon, 2\epsilon), (\epsilon, \epsilon), (\epsilon, \epsilon), \dots\}$  on the *MAX* algorithm and  $\{(\epsilon, 2\epsilon), (1, 1)\}$  on the *GREEDY* algorithm. In fact, no deterministic online algorithm is competitive with a constant factor [2]. However, using both with randomness can yield a competitive algorithm.

**Theorem 1.** [1] *Running MAX and GREEDY uniformly at random is a 2-competitive algorithm.*

*Proof.* For a given input sequence  $T$ , let  $GREEDY(T)$ ,  $MAX(T)$ , and  $OPT(T)$  be the values of the sets returned by *GREEDY*, *MAX*, and an optimal solution, respectively. Consider the fractional knapsack solution to the offline version, that is, when we are allowed to take fractional parts of items. Then a greedy solution is optimal: Sort by ratio and take the highest ratio items until the knapsack is full, potentially taking only part of the last item in the knapsack. In the fractional knapsack problem, we have relaxed the integrality constraint, so the fractional solution is not worse than the integral solution.

Now the offline *GREEDY* algorithm by definition selects all of these except the fractional item (if one exists). Thus the online *GREEDY* algorithm also selects all except the fractional item. Finally the *MAX* algorithm selects an item with value at least that of the

fractional item, so  $GREEDY(T) + MAX(T) \geq OPT(T)$ . Thus the competitive ratio is  $\frac{2 \cdot OPT(T)}{GREEDY(T) + MAX(T)} \leq 2$ .  $\square$

## 2.2 Lower Bound

We use Yao's Principle to prove a lower bound of  $1 + 1/e$  for the competitive ratio of any removable online knapsack algorithm.

To do so, we must find a distribution of inputs for which no deterministic algorithm performs "well", which means the expected value of the output from any deterministic algorithm is at most  $\frac{1}{1+1/e}$  times the expected optimum. As the power of an offline algorithm is in the additional information it has over an online algorithm, it is reasonable that in constructing our "bad" input, we would want to limit the information the algorithm has about future items. Thus it is not surprising that we will construct a family of inputs where each input is a prefix of the same base sequence.

Let the probability of  $k + 1$  items be  $p_k$ . For simplicity in analysis, we will let the first item  $e_0 = (w_0, v_0) = (1, 1)$  with future items having much smaller weights and values but a higher ratio so that the problem is essentially reduced to determining when to remove the first item from the knapsack.

This is reminiscent of the *ski rental problem*, one of the most well-understood online problems [3], where a client must choose whether and when to buy skis for a fixed cost versus paying a smaller cost each day. Our knapsack problem is the reverse of this: instead of deciding when to switch from renting skis to buying skis to minimize total cost, we are deciding when to switch from having a single large item in our knapsack to getting a stream of smaller items with a better ratio to maximize total value.

Finally, we want the expected value of removing  $e_0$  and selecting all following items to be close to 1 at any given point. A simple way to achieve this is to let the remaining items have a constant value and have the  $p_k$  form an exponential distribution.

This leads us towards the input sequence given by Han et al. [1]:

$$(1, 1), (1/n^2, 1/n), (1/n^2, 1/n), \dots (1/n^2, 1/n)$$

for a given value of  $n$  and where there are  $k + 1$  items with probability  $p_k = \frac{1 - e^{-1/n}}{1 - e^{-n}} \cdot e^{-(k-1)/n}$  for  $k = 1, 2, \dots, n^2$ .

**Theorem 2.** [1] *No online algorithm has competitive ratio less than  $1 + 1/e$  for the removable online knapsack problem.*

*Proof.* We apply Yao's Principle on the input distribution above. The optimal strategy with full information is to keep  $e_0$  for  $k \leq n$  and remove  $e_0$  immediately otherwise. This leads us to the expected optimal value

$$\sum_{i=1}^n 1 \cdot p_i + \sum_{i=n+1}^{n^2} \frac{i}{n} \cdot p_i = \frac{1 - e^{-1/n}}{1 - e^{-n}} \left( \sum_{i=1}^n e^{-(i-1)/n} + \frac{1}{n} \sum_{i=n+1}^{n^2} i \cdot e^{-(i-1)/n} \right)$$

which has value  $1 + 1/e$  as  $n \rightarrow \infty$ .

Now an online deterministic algorithm can only decide on which item  $e_l$  to remove  $e_0$ . Then the expected value of this algorithm is

$$\sum_{i=1}^l 1 \cdot p_i + \sum_{i=l+1}^{n^2} \frac{i-l}{n} \cdot p_i = \frac{1 - e^{-1/n}}{1 - e^{-n}} \left( \sum_{i=1}^l e^{-(i-1)/n} + \frac{1}{n} \sum_{i=l+1}^{n^2} (i-1) \cdot e^{-(i-1)/n} \right)$$

which has value 1 as  $n \rightarrow \infty$ .

Thus any online algorithm has competitive ratio at least  $1 + 1/e$ .  $\square$

## 2.3 Remarks

Yao's Principle is used here to give a lower bound on the competitiveness of algorithms for the removable online knapsack problem, which is an upper bound on their effectiveness. It can be used to show the tightness of a particular solution.

Also, as we saw here, there is a dual between using an algorithm to demonstrate a lower bound on the solvability of a problem (alternatively, an upper bound on the hardness of a problem), and using a series of inputs to demonstrate an upper bound on the solvability of a problem (alternatively, an lower bound on the hardness). As more work is done on a particular problem, the gap between the two shrink until a provably optimal algorithm is found.

## 3 Yao's Principle in Quantum Information

Another area where Yao's principle has found an application is in the study of quantum information, permitting proof of worst-case bounds for strategies in certain quantum information "games".

Consider setting up a "game" between two players and an adversary, where the two players receive random input from the adversary and must each return specific outputs based on those inputs without communicating with each other. Analyzing the best strategies for these games, under the assumptions of both classical physics and quantum physics, can be used to demonstrate that better strategies exist if the players are permitted access to quantum information.

An example of such a game is the CHSH (Clauser-Horne-Shimonyi-Holt) game:

- The players may collude to set up a strategy beforehand, but may not communicate afterwards.
- The adversary chooses a random  $\vec{x} = (x_1, x_2) \in X = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ . The random choice is uniform, with  $\frac{1}{4}$  probability each.
- The players 1 and 2 receive the values  $x_1$  and  $x_2$ , respectively.
- Using their strategy, the players 1 and 2 output the values  $a_1$  and  $a_2$ , respectively.
- The players win if  $a_1 \oplus a_2 = x_1 \wedge x_2$ .

Both under classical physics and under any ‘local hidden variable theory’ – an attempt to explain quantum phenomena as being due to extra local hidden variables – the best that the players can achieve is a win probability of 0.75. If the players are allowed to share entangled quantum bits, however, then 0.85... is the best win probability that the players can achieve. [[cite Clauser]] [[cite Ambainis]] The CHSH game was designed by its authors so as to be an actual realizable experiment, and experiments have since confirmed that the way the universe works is consistent with quantum entanglement, and not consistent with local hidden variable theories. [[cite Aspect]]

In *Worst Case Analysis of Non-local Games* by Ambainis et al., the authors generalize these results to games beyond the simple CHSH game, and analyze the worst-case performance of strategies in both classical and quantum physics.

To-do: the following citations

Ambainis, A., A. Bačkurs, K. Balodis, A. Škuškoviņš, J. Smotrovs, M. Virza: Worst Case Analysis of Non-local Games. P. van Emde Boas et al. (Eds.): SOFSEM 2013, LNCS 7741, pp. 121-132, 2013. <http://arxiv.org/abs/1112.2856>

Clauser, J., M. Horne, A. Shimony, R. Holt: Proposed experiment to test local hidden-variable theories. *Physical Review Letters* 23, 880 (1969). <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.23.880>

Aspect, A., P. Grangier, G. Roger: Experimental Tests of Realistic Local Theories via Bell’s Theorem. *Physical Review Letters* 47, 460 (1981). <http://journals.aps.org/prl/abstract/10.1103/PhysRevLett.47.460>

## 4 Conclusion

## References

- [1] Xin Han, Yasushi Kawase, and Kazuhisa Makino. Randomized algorithms for removable online knapsack problems. In *Frontiers in Algorithmics and Algorithmic Aspects in Information and Management, Third Joint International Conference, FAW-AAIM 2013, Dalian, China, June 26-28, 2013. Proceedings*, pages 60–71, 2013.
- [2] Kazuo Iwama and Guochuan Zhang. Optimal resource augmentations for online knapsack. In *Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX ’07/RANDOM ’07, pages 180–188, Berlin, Heidelberg, 2007. Springer-Verlag.
- [3] Anna R. Karlin, Claire Kenyon, and Dana Randall. Dynamic tcp acknowledgement and other stories about  $e/(e-1)$ . In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC ’01, pages 502–509, New York, NY, USA, 2001. ACM.
- [4] Andrew Chi-Chin Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, SFCS ’77, pages 222–227, Washington, DC, USA, 1977. IEEE Computer Society.