Resource Allocation Algorithms
Lecture 08

Lecture by Dr. Tami Tamir
Typeset by Steven Karas

2018-05-15
Last edited 18:15:10 2018-05-15

Disclaimer These lecture notes are based on the lecture for the course Resource Allocation Algorithms, taught by Dr. Tami Tamir at IDC Herzliyah in the spring semester of 2017/2018. Sections may be based on the lecture slides written by Dr. Tami Tamir.

Agenda

- Recap of packing problems
- Game Theory

1 Packing problems

A family of problems characterized by a set of items and bins with various constraints.

1.1 Knapsack problem

A subset of packing problems with 1 bin with a limited size. Items can have varying weights. The general problem is $\mathcal{NP}$-hard, but we can build a dynamic programming exact solution that is polynomial in n and the volume of the knapsack. There is another variant of the DP solution that is polynomial in n and the sum of the gaps between weights. We showed a simple 2-approximation as well.

1.1.1 FPTAS

We showed an approximation scheme by which we can get arbitrarily close to the correct answer (formally: $1+\varepsilon$). In this case, we round the weights and run the DP on the reduced problem. The time complexity of this is $O(n^3/\varepsilon)$.

1.2 Bin packing

Items with weights within $0 < a_i < 1$. Multiple bins, but each has size 1. Objective function is to minimize the number of bins used.

$\mathcal{NP}$-hard by reduction from partition. From this, it also follows that the closest we can approximate is $\frac{3}{2}$ (with no constant factor).

We saw NEXT-FIT is a 2-approximation. We saw that FIRST-FIT is a 1.7-approximation. We then presented FFD (sort from largest to smallest, then run FIRST-FIT) which is a $3 + 1.22$-approximation.

1.3 Unit fractions bin packing

All items are in the form $\frac{1}{W}$ where $W \in \mathbb{N}$.

ANY-FIT-DECREASING is a $1 + \text{OPT}$ approximation.

1.4 Online bin packing

Bin packing, but with the twist that we do not know the number of items head of time, and a_i must be packed before we know $a_{i+1}, \ldots, a_n$.

1.4.1 Harmonic-k

HARMONIC-k is an $2 + 1.75$-approximation algorithm that classifies items into k intervals:

$(\frac{1}{2}, 1]$ has 1 item per bin, $(\frac{1}{k}, \frac{1}{k-1}]$ has $k-1$ items per bin. The last interval of $(0, \frac{1}{k})$ uses NEXT-FIT. All but the last bin of each interval is at least $\frac{k}{k+1}$ full. While running, there are $k-1$ bins open at any moment.

A full analysis of HARMONIC-3 can be found on slide 32-33.

For any k, HARMONIC-k is at most 1.691-competitive[1].

2 Game theory

A field of mathematics that studies the construction and analysis of models for scenarios that require decision making. Originally, the field arose from economics, and considered adversarial models. Computer Science has brought refinements to the field, notably many new problems, and some new techniques for assessing problems and solutions.

2.1 Nash Equilibrium

A stable state in which no agent has an incentive to switch strategy. There may be multiple Nash equilibriums for a given scenario.

Social Optimum The social optimum minimizes the total cost for all agents. The social optimum may not be a Nash equilibrium.

Price of Anarchy The ratio of the worst Nash equilibrium to the social optimum. A typical proof sketch is to show that for the worst NE, if the ratio is greater than some value, it implies there is a better option. The opposite direction of such a proof will typically show something by convergence of error.

Price of Stability The ratio of the best Nash equilibrium to the social optimum.

2.2 Network Formation Games

Given a directed graph $G = (V, E)$ with edge costs $c_e \geq 0$, a source vertex s, and k agents located at terminal vertexes $t_1, \ldots, t_k$. Agent j must construct a path P_j from s to t_j. If x agents use edge e, they each pay c_e/x. Agents behave in a selfish manner.

The strategy space is the set of all viable solutions for a given agent.

2.2.1 Price of Anarchy

For network formation games with k agents, provably k. The proof follows from showing that if the PoA is greater than k, then there must exist some agent that is better off switching. The proof in the other direction is to show that there is an instance for which the PoA is $> k - \varepsilon$ (i.e. slide 19).

2.2.2 Price of Stability

For network formation games with k agents, denote $H(k) = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$. Provably $H(k)$, with details on slides 26-27.

2.2.3 Best Response Dynamics

In turn, ask each agent if they would like to change their strategy. We will prove that this terminates for all inputs at a Nash equilibrium.

The proof sketch is that we will define a potential function that monotonically decreases when an agent improves their strategy.

If we use $\Phi(s) = \sum_{j=1}^{k} \text{cost}(t_i)$ as our potential function, the function does not monotonically decrease.

Consider a set of paths $P_1, \ldots, P_k$. Denote x_e as the number of paths that use edge e. Denote the harmonic sum $H(k) = \sum_{i=1}^{k} \frac{1}{i}$. Take note that $H(0) = 0$. Let $\Phi(P_1, \ldots, P_k) = \sum_{e \in E} c_e \cdot H(x_e)$. Consider an agent j switching from path P_j to P_j'. Note that the agent wants to switch because:

$$\sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1} < \sum_{e \in P_j - P_j'} \frac{c_e}{x_e}$$

Note that the change in the potential function is exactly the change in the agent's cost. Therefore, the net change is strictly negative. Because the aggregate strategy space of all agents is finite, this implies BRD terminates with a NE.

Bounding cost of stability Details and proof on slide 26-27.

2.2.4 Min-cost stable profile

For network formation, it is $\mathcal{NP}$-hard to determine if there is an Nash equilibrium with cost at most C. This follows by reduction from 3-DIMENSIONAL-MATCHING.

Proof is found on slides 28-32.

References

[1] Chung-Chieh Lee and Der-Tsai Lee. A simple on-line bin-packing algorithm. *Journal of the ACM (JACM)*, 32(3):562–572, 1985.

[2] Michael L Pinedo. *Scheduling: theory, algorithms, and systems*. Springer, 2016.