Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Systems@ETH Zürich

Autumn Term 2018 Prof. T. Roscoe, Prof. R. Wattenhofer

Computer Systems

— Solution to Assignment 11 —

1 Game Theory

Quiz

1.1 Selling a Franc

We assume that there are two bidders, b_1 and b_2. If b_1 bids 5 rappen, his gain his 95 rappen. Now b_2 is inclined to bid 10 rappen and gain 90 rappen. This can continue until b_1 bid 95 rappen. Bidder b_2 now has the choice of losing 90 rappen (her las bid) or come out even. Since she is a rational player, she'll bid 1 franc. Bidder b_1 now faces a similar choice. Either he loses 95 rappen or he bids and has a chance of only losing 5 rappen. Since he is a rational player, he will bid 1.05 franc. This bidding war will continue indefinitely (or until one bidder runs out of money).

There are a few ways the bidders could have avoided this situation. Apart from the obvious, simply do not play, they could have also colluded. One bidder bids 5 rappen for the franc and the bidders will simply split the money they made. This requires that the bidders can trust each other. As you can guess, there are games that anticipate collaboration.
There exists however a strategy, which is profitable even for a non-colluding bidder. If the first bidder bids 95 rappen, she will win 5 rappen, because nobody else will also bid. Why will nobody else bid? If another bidder bids more, he will certainly not bid more than 1 franc, because this will yield a negative payoff. Instead, he could bid 1 franc. But then, the first bidder will bid 1.05 francs to minimize her loss. And then the game continues as outlined at the beginning and both bidders will incur a loss. Therefore, no rational player will bid 1 franc in this scenario.

Basic

1.2 Selfish Caching

To be sure that we find every Nash Equilibrium, we explicitly write down every best response.

i. The best response strategies are

u: cache only if nobody else does. (B1)

v: cache if neither u nor x cache. (B2)

w: cache unless u caches. (B3)

x: cache if neither u nor v cache. (B4)

Nash equilibrium. If we assume that u plays $Y_u = 1$ (u caches) the system can only be in a NE if $Y_v = Y_w = Y_x = 0$ due to (B1). Since for all v, w, and x it is the best response not to cache if u does, $x = (1000)$ is an Nash equilibrium. If $Y_u = 0$ then (B3) implies $Y_w = 1$. If furthermore, $Y_v = 1$ it must hold that $Y_x = 0$ due to (B2). This does not conflict with (B4), and (0110) constitutes another NE. Last, if $Y_v = 0$ then (B2) implies $Y_x = 1$, which is also okay with (B4). Hence (0011) is also a NE.

$$NE = \{(1000), (0110), (0011)\}$$

Price of anarchy. The social optimum is achieved in strategy profile (1000), namely $OPT = cost(1000) = 1 + \frac{1}{2} + \frac{3}{8} + \frac{3}{4} = \frac{21}{8}$. Since (1000) is also a Nash equilibrium we immediately get that $OPoA = 1$. The worst-case price of anarchy is

$$PoA = \frac{cost(0110)}{OPT} = \frac{\frac{1}{2} + 1 + 1 + \frac{7}{8}}{\frac{21}{8}} = \frac{9}{7} \approx 1.286.$$

ii. The best response strategies are

u: cache only if nobody else does. (B1)

v: cache unless u caches. (B2)

w: cache unless x caches. (B3)

x: cache if neither u nor w cache. (B4)

Nash equilibrium. If we assume that u plays $Y_u = 1$ (u caches) the system can only be in a NE if $Y_v = Y_w = Y_x = 0$ due to (B1). However, $Y_x = 0$ implies that $Y_w = 1$ due to (B3), and hence there can be no NE with $Y_u = 1$. In any NE it must hold that $Y_u = 0$. Consequently, it must hold that $Y_v = 1$ from (B2). Now if $Y_w = 1$ (B3) implies that x does not cache. This does not infringe rule (B4), and thus $x = (0110)$ is a Nash equilibrium. If $Y_w = 0$ then (B4) implies that x caches. As thus, rule (B3) is not violated $x = (0101)$ is also a Nash equilibrium.

Price of anarchy. The social optimum is achieved in strategy profile (0110), namely $OPT = cost(0110) = \frac{1}{3} \cdot 0.2 + 1 + 1 + \frac{1}{2} \cdot 0.2 = 2.1\overline{6}$. Since (0110) is also a Nash equilibrium we get that the optimistic price of anarchy is 1. The worst-case price of anarchy is

$$PoA = \frac{cost(0101)}{OPT} = \frac{1/3 \cdot 0.2 + 1 + 0.2 + 1}{2.1\overline{6}} = \frac{68}{65} \approx 1.046$$

1.3 Selfish Caching with variable caching cost

We define D_i to be the set of nodes that cover node i. A node j *covers* node i if and only if $d_i c_{i \leftarrow j} < \alpha_i$, i.e., node i prefers accessing the object at node j than caching it. Convince yourself that a strategy profile is a Nash Equilibrium if and only if for each node i it holds that

- if $Y_i = 1$ then $Y_j = 0$ for all $j \in D_i$, and
- if $Y_i = 0$ then $\exists j \in D_i$ with $Y_j = 1$.

i. $D_u = \emptyset$, $D_v = \{u, w\}$, $D_w = \{u\}$. D_u being empty implies $Y_u = 1$ (i.e. caches the file). Hence $Y_v = 0$, and $Y_w = 1$. $NE = \{(101)\}$. $PoA = 1$ since (101) is also the social optimum strategy.

ii. $D_u = \{v\}$, $D_v = \{u\}$, $D_w = \{u, v\}$. If $Y_u = 1$, then $Y_v = 0$ and $Y_w = 0$. If $Y_u = 0$, then $Y_v = 1$. Hence $Y_w = 0$. The equilibria are $NE = \{(100), (010)\}$.

$$PoA = \frac{cost(100)}{cost(010)} = \frac{3 + 1 + 8/3}{3/2 + 3/2 + 5/3} = \frac{40}{28} \approx 1.43$$

Dominant strategies. Every dominant strategy profile is also a Nash equilibrium. Hence we only have to check the computed NEs whether they consist of dominant strategies only.

Let us consider game i. Since every dominant strategy profile is also a Nash Equilibrium, it suffices to consider the NE. The game has no dominant strategy profile. Profile (101) is no dominant strategy profile in game i. since, although $Y_u = 1$ is the dominant strategy for u, $Y_v = 0$, and $Y_w = 1$ are not dominant strategies for v and w. If $Y_v = 1$, then it would be the best response of w to set $Y_w = 0$. Game ii: Since the decision of node u whether to cache depends on the decision of node v, this is not a dominant strategy. Therefore, this game has no dominant strategy profile.

Advanced

1.4 Matching Pennies

The bi-matrix of the game with Tobias as row player, and Stephan as column player looks as follows:

	H	T
H	1 , -1	-1 , 1
T	-1 , 1	1 , -1

This zero-sum game has no pure Nash equilibrium. For the mixed NEs, Tobias plays heads (H) with probability p, tails (T) with probability $1 - p$. Stephan plays H with probability q, and T with probability $1 - q$. We get the expected utility functions Γ:

$$\begin{aligned} \Gamma_T(p,q) &= p(q-(1-q)) + (1-p)(-q+(1-q)) = (4q-2)\cdot p + 1 - 2q \\ \Gamma_S(p,q) &= q(-p+(1-p)) + (1-q)(p-(1-p)) = (2-4p)\cdot q + 2p - 1 \end{aligned}$$

If Stephan plays $q = 1/2$ the term $4q - 2$ equals 0, and any choice of p will yield the same payoff for Tobias. If Tobias plays $p = 1/2$ then any choice of q is a best response for Stephan. Thus $(p, q) = (1/2, 1/2)$ is a mixed NE. Note that for any choice of $p > 1/2$, Stephan's best response is to choose $q = 0$. For a $p < 1/2$ Stephan would choose $q = 1$. However, Tobias' best response to $q > 1/2$ is $p = 1$, and $p = 0$ if $q < 1/2$. Hence $(p, q) = (1/2, 1/2)$ is the only pair of mutual best responses.

Mastery

1.5 PoA Classes

Let I^n be an instance of $\mathcal{A}^n_{[a,b]}$ that maximizes the price of anarchy, i.e. $PoA(\mathcal{A}^n_{[a,b]}) = PoA(I^n)$. Let $x, y \in X$ be two strategy profiles in I^n such that $PoA(I^n) = cost(y)/cost(x)$. We show the claim by constructing an instance $\hat{I}^n \in \mathcal{W}^n_{\left[\frac{1}{b},\frac{1}{a}\right]}$ out of I^n for which it holds that $PoA(\hat{I}^n) \geq \frac{a}{b} PoA(I^n) = \frac{a}{b} PoA(\mathcal{A}^n_{[a,b]})$. We construct $\hat{I}^n$ by setting $d_i = 1/\alpha_i$, $\hat{\alpha}_i = 1$ where α_i are the placement costs of player i in I^n. All edges remain as in I^n. This game has the same Nash equilibria as I^n since the cover sets D_i for each peer stay the same. A peer j is in D_i iff $c_{i \leftarrow j} < \alpha_i$, or $c_{i \leftarrow j}/\alpha_i < 1$ respectively. We get the bound by comparing the performance of the two strategies x, y that produce the PoA in I^n in $\hat{I}^n$. Note that x is not necessarily a social optimum, but y is

a Nash equilibrium also in $\hat{I}^n$.

$$PoA(\hat{I}^n) \geq \frac{\hat{cost}(y)}{\hat{cost}(x)} = \frac{\sum_{i=1}^{n}\left(y_i + (1-y_i)\frac{c_i(y)}{\alpha_i}\right)}{\sum_{i=1}^{n}\left(x_i + (1-x_i)\frac{c_i(x)}{\alpha_i}\right)} \quad (1)$$

$$= \frac{b \cdot a\sum_{i=1}^{n}\left(y_i + (1-y_i)\frac{c_i(y)}{\alpha_i}\right)}{b \cdot a\sum_{i=1}^{n}\left(x_i + (1-x_i)\frac{c_i(x)}{\alpha_i}\right)} \quad (2)$$

$$\geq \frac{a\sum_{i=1}^{n}\left(y_i\alpha_i + (1-y_i)c_i(y)\right)}{b\sum_{i=1}^{n}\left(x_i\alpha_i + (1-x_i)c_i(x)\right)} \quad (3)$$

$$= \frac{a \cdot cost(y)}{b \cdot cost(x)} = \frac{a}{b}PoA(I^n) \quad (4)$$

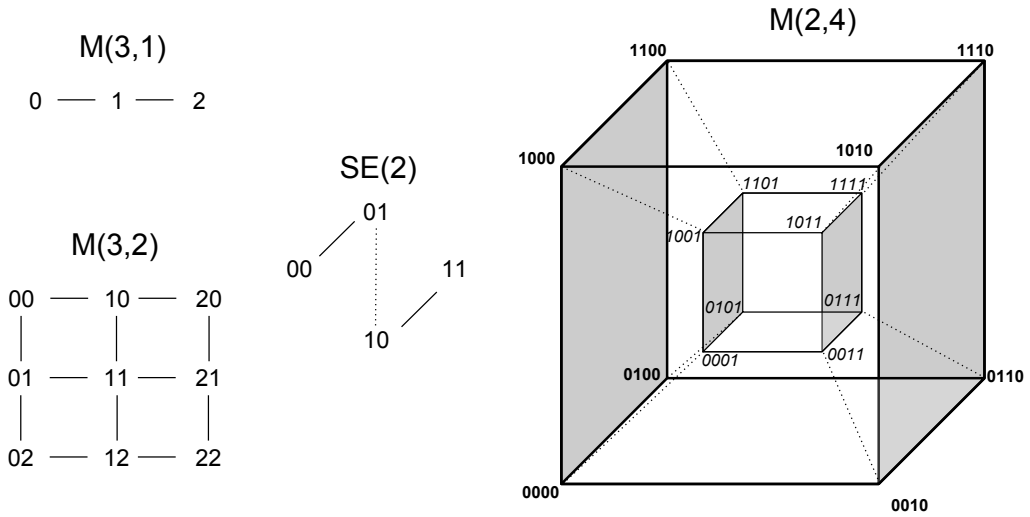
$\hat{cost}(x)$ denotes the cost function in $\hat{I}^n$. x_i, and y_i are either 1 or 0. x_i equals 1 if player i caches in strategy profile x, and 0 if she does not. With $c_i(y)$ we denote the cost of node i if it access the file remotely in strategy y. For step (3) we exploit the fact that $b \geq \alpha_i$ and $a \leq \alpha_i$ for all i.

2 Distributed Storage

Quiz

2.1 Hypercubic Networks

The hypergraphs are drawn as follows. The 4-dimensional hypercube ($M(2,4)$) can also be drawn as two cubes side-by-side with connected corners.



Basic

2.2 Iterative vs. Recursive Lookup

a) In the recursive lookup there is no difference between a request originating at the node and a request being forwarded. Only once the lookup is finished do we need to care about

forwarding the result to the previous node or returning it to the caller. This allows the same lookup logic to be reused for both. Furthermore, if the response is returned through the same path as the request was sent through, then the intermediate nodes can cache the result, potentially speeding up future lookups and distributing the load of a popular item.

b) Recursive lookups can easily be misused to mount Denial-of-Service attacks, since a single request message from an originating node is forwarded over multiple hops. Each hop multiplies the impact this message has on the network. Thus the attacker's bandwidth is potentially multiplied by the number of hops the request is routed through. Furthermore, if the result is returned over the same path as the request, then the attacker is hidden behind a number of hops and the victim only sees traffic originating from the last hop.

2.3 Building a set of Hash functions

The salted hashing function derivation allows random access to any of the derived hashing functions without having to recompute the intermediate functions, as is the case in the iterative hashing function derivation scheme.

Advanced

2.4 Multiple Skiplists

a) 1 shows the structure of a multi-skiplist with 8 nodes and 3 levels. Notice that each of the lists would wrap around at the ends.

b) Unlike in the single skiplist each node now has a constant degree of $2 \cdot (l + 1)$, i.e., on each level it has a right and a left neighbor, including the simple list at $l = 0$.

c) The number of hops is still $O(log(n))$, just like the simple skiplist.

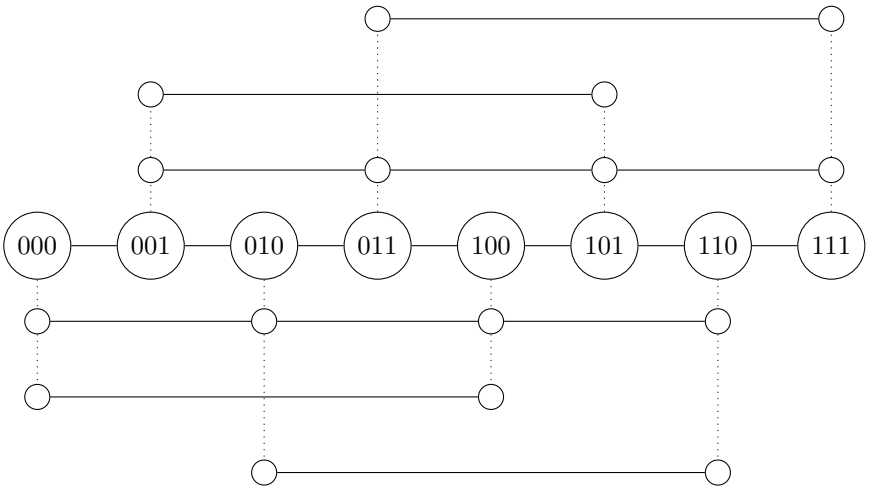


Figure 1: A multi-skiplist with 3 levels and 8 nodes