11 Knowledge State Algorithms for the Server Problem

11.1 The Class $\mathcal{M}_{d,p}$

For any positive integers d, p such that $2d \leq p \leq 3d$, we define $\mathcal{M}_{d,p}$ to be the class of all metric spaces M such that:

1. If $x, y \in M$, then $d(x, y)$ is an integer, and is less than or equal to d.
2. If $x, y, z \in M$, then $d(x, y) + d(x, z) + d(y, z) \leq p$. (This is the perimeter of the triangle with vertices x, y, and z.)

For example, the class $\mathcal{M}_{1,3}$ is the class of all uniform spaces. It is well-known that the randomized competitiveness of the 2-server problem on $\mathcal{M}_{1,3}$ is $\frac{3}{2}$.

We will now concentrate on $\mathcal{M}_{2,4}$. Our result is as follows:

Lemma 11.1 *The randomized competitiveness of the 2-server problem in $\mathcal{M}_{2,4}$ is $\frac{19}{12}$.*

11.2 Cross-Polytope Spaces

Schlälfli proves that there are exactly three infinite families of regular hyperpolyhedra. These are the simplices, the cubes, and the cross-polytope spaces. Each family contains one regular hyperpolyhedron in each dimension.

We will measure distance between the vertices of a hyperpolyhedron by the number of edges of the shortest path of edges between the vertices. Thus, for example, the distance between the opposite corners of a 3-cube is 3, rather than the Euclidean distance of $\sqrt{3}$.

The *n-simplex* has $n + 1$ vertices. The distance between any two vertices is 1. A 2-simplex is an equilateral triangle, and a 3-simplex is a regular tetrahedron. Every finite uniform space is an n-simplex for some n.

The *n-cube* has 2^n vertices. We can represent any vertex by a bitstring of length n, and the distance between two vertices is the Hamming distance between the corresponding bitstrings. A 2-cube is a square, and a 3-cube is a regular hexahedron.

The *n-cross-polytope* has $2n$ vertices. Each vertex of a cross-polytope has a unique *antipode*. The distance between any vertex and its antipode is 2, while the distance between any two vertices that are not antipodes is 1. A 2-cross-polytope is a square, while a 3-cross-polytope is a regular octahedron. Every finite metric space in the class $\mathcal{M}_{2,4}$ embeds in a cross-polytope space.

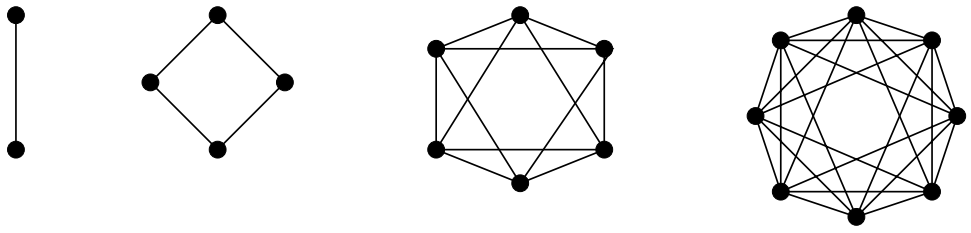


Figure 11.1: n-Cross-Polytopes for $n \leq 4$

11.3 Knowledge States

We allow the algorithm to use randomization to choose a distribution. This variation, called the *mixed model* of randomized algorithms is a generalization of both the behavioral model and the distributional model. A *knowledge state algorithm* is a mixed online algorithm that computes an *adjustment* and an *estimator* at each step, and uses the current estimator as its memory state. The estimator is a real-valued function on configurations that is updated at every step, and which estimates the cost of the optimal offline algorithm, while the adjustment is a real number that is computed at every step.

More formally, if $\mathcal{A}$ is a knowledge-state algorithm, then:

1. At any given step, the full state of $\mathcal{A}$ is a pair (ω, π), where π is a finite distribution on $\mathcal{X}$, and $\omega : \mathcal{X} \to \mathbb{R}$ is the current estimator. We call that pair the *current knowledge state.*

2. If $S = (\omega, \pi)$ is the knowledge state and the next request is r, then $\mathcal{A}$ computes an adjustment, a number which we call $adjust_{\mathcal{A}}(S, r)$, and uses randomization to pick a new knowledge state $S' = (\omega', \pi')$. More precisely, there are subsequent knowledge states $S_i = (\omega_i, \pi_i)$ and subsequent positive weights λ_i for $i = 1, \ldots m$, $\sum_{i=1}^{m} \lambda_i = 1$, such that

   (a) $(\omega \wedge r)(x) \geq adjust_{\mathcal{A}}(S, r) + \sum_{i=1}^{m} \lambda_i \omega_i(x)$ for each $x \in \mathcal{X}$, where we define function $\omega \wedge r$ as $(\omega \wedge r)(y) = \min_{x \in \mathcal{X}} \{\omega(x) + cost(x, r, y)\}$ and $cost(x, r, y)$ denotes the cost of serving r given cache configuration x while resulting in configuration y.

   (b) For each i, $\mathcal{A}$ chooses S' to be S_i with probability λ_i.

Our algorithm uses only eight classes of knowledge states, illustrated below.

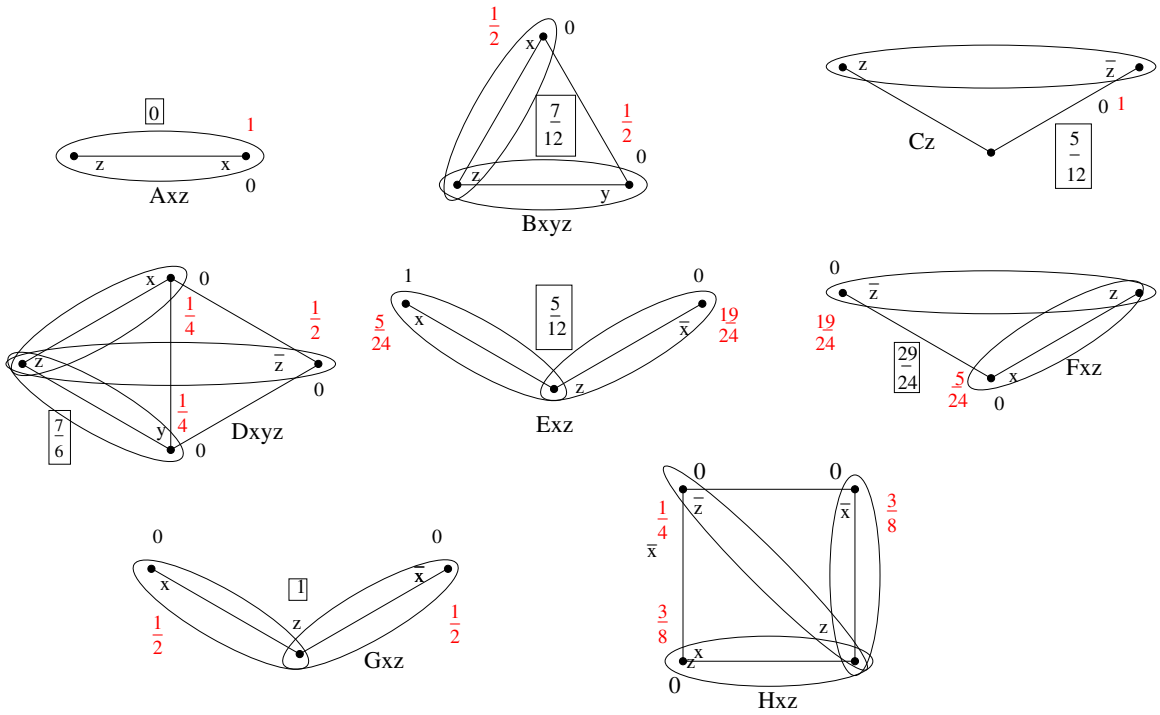


Figure 11.2: Knowledge States for the 2-Server problem on $\mathcal{M}_{2,4}$

11.4 Analyzing a Move

Given a knowledge state (ω, π) and a request point r, a knowledge state algorithm must choose a distribution on a set of knowledge states, which we call *subsequents*. That is, a knowledge state algorithm $\mathcal{A}$ chooses knowledge state $k_1 = (\omega_1, \pi_1), \ldots, k_1 = (\omega_m, \pi_m)$ and real numbers $\lambda_1, \ldots \lambda_m \geq 0$ $\sum_{i=1}^{m} \lambda_i = 1$, and such that, for each π_i there is a server at r.

Given that choice, the algorithm cost and the amortized optimal cost can be computed as follows.

$cost_{\mathcal{A}} = ||\pi, \sum_{i=1}^{m} \lambda_i \pi_i||$, the transportation cost between those two distributions.

$cost_{\text{OPT}} = \min_X \{(\omega \wedge r)(X) - \sum_{i=1}^{m} \lambda_i \omega_i(X)\}$ where the minimum is taken over all server configurations X. (In our case, X is a set of two points in the metric space. It is also sufficient to consider only those X which contain the request point, r.)

Example. We now let $\mathcal{A}$ be the $\frac{19}{12}$-competitive algorithm for the 2-server problem in $\mathcal{M}_{2,4}$ whose knowledge states are given in Figure 11.2. Suppose that $k = Dxyz$, and $r = \bar{x}$. The best choice for the $\mathcal{A}$ is as follows:

1. $k_1 = Gz\bar{x}$ and $\lambda_1 = \frac{1}{3}$.
2. $k_2 = Dyz\bar{x}$ and $\lambda_2 = \frac{1}{3}$.
3. $k_3 = Dy\bar{z}\bar{x}$ and $\lambda_3 = \frac{1}{3}$.

We then obtain

$cost_{\mathcal{A}} = \frac{13}{12}$
$cost_{\text{OPT}} = \frac{2}{3}$

Exercise 11.1 *Let $\mathcal{A}$ be the $\frac{19}{12}$-competitive algorithm for the 2-server problem in $\mathcal{M}_{2,4}$ whose knowledge states are given in Figure 11.2. If the current knowledge state is Exz and the request is x, what are the subsequents, and what is the distribution on those subsequents? What is the behavioral description of this move? Is this move critical?*

Exercise 11.2 *Let $\mathcal{A}$ be the $\frac{19}{12}$-competitive algorithm for the 2-server problem in $\mathcal{M}_{2,4}$ whose knowledge states are given in Figure 11.2. If the current knowledge state is $Dxyz$ and the request is a point $r \notin \{x, \bar{x}, y, \bar{y}, z, \bar{z}\}$ what are the subsequents, and what is the distribution on those subsequents? What is the behavioral description of this move? Is this move critical?*

11.5 Knowledge State Algorithms for Paging

When we design algorithms for the paging problem, we would like to minimize the number of pages we keep track of. We define a *bookmark* to be the name of a page that is not in our cache, and an *active page* to be a page that is either in our cache or bookmarked.

There is a knowledge state algorithm for the 2-paging problem that uses at most one bookmark. The knowledge states for that algorithm are as follows:

- $A_{a,b} = (ab||, \{a, b\})$, the cone over a and b, for any two pages a and b.

- $B_{a,b,c} = \big(a|bc|, \frac{1}{2}\{a,b\} + \frac{1}{2}\{a,c\}\big)$, for any three pages a, b, and c.

Exercise 11.3 *Define an H_2-competitive knowledge state algorithm for the 2-paging problem, using the knowledge states listed above.*

Consider the following list of knowledge states for the 3-paging problem.

- $A_{a,b,c} = (abc|||, \{a,b,c\})$, for any three pages a, b, and c.
- $B_{a,b,c,d} = \big(a|bcd||, \frac{1}{3}\{a,b,c\} + \frac{1}{3}\{a,b,d\} + \frac{1}{3}\{a,c,d\}\big)$ for any four pages a, b, c, and d.
- $C_{a,b,c,d,e} = \big(a|bcde||, \frac{1}{6}\{a,b,c\} + \frac{1}{6}\{a,b,d\} + \frac{1}{6}\{a,b,e\} + \frac{1}{6}\{a,c,d\} + \frac{1}{6}\{a,c,e\} + \frac{1}{6}\{a,d,e\}\big)$ for any five pages a, b, c, d, and e.
- $D_{a,b,c,d} = \big(ab||cd|, \frac{1}{2}\{a,b,c\} + \frac{1}{2}\{a,b,d\}\big)$ for any four pages a, b, c, and d.
- $E_{a,b,c,d,e} = \big(ab||cde|, \frac{1}{3}\{a,b,c\} + \frac{1}{3}\{a,b,d\} + \frac{1}{3}\{a,b,e\}\big)$ for any five pages a, b, c, d, and e.
- $F_{a,b,c,d,e} = \big(a|bc|de|, \frac{1}{3}\{a,b,c\} + \frac{1}{6}\{a,b,d\} + \frac{1}{6}\{a,b,e\} + \frac{1}{6}\{a,c,d\} + \frac{1}{6}\{a,c,e\}\big)$ for any five pages a, b, c, d, and e.

Exercise 11.4 *Define an H_3-competitive knowledge state algorithm for the 3-paging problem, using the knowledge states listed above.*