2008-2 고급계산이론 기말고사

2008.12.12 김진일

1. Randomized Algorithm 설명

Las-Vegas Algorithm

- ➔ Given I to P, a Las-Vegas algorithm uses some (r) random numbers, but except for choosing random numbers it proceeds completely deterministically. LV's solution is always correct as in deterministic alg.
- ➔ We say that LV solves P in expected time T(n) if for every I such that |I|=n, LV solves I in expected time ≤ T(n). By expected time we mean the average of all solution times of I by LV for all possible choice sequences of r random numbers.

Monte-Carlo Algorithm

- ➔ A Monte-Carlo algorithm may produce an incorrect solution. Let e>0. We say that MC solves P with confidence greater than 1-e if for every I the probability that MC will produce an incorrect solution is ≤ e.

Difference) Las-Vegas 알고리즘은 항상 정답을 계산하는데 random 선택에 따라 수행 시간만이 달라지고 Monte-Carlo 알고리즘은 잘못된 답을 계산할 가능성이 있지만 정답을 계산할 확률이 일정 이상이 된다는 점이 다르다.

Example) Choosing a Large Number: n개의 숫자 중 중간(median) 이상으로 큰 숫자 고르기

Las-Vegas Algorithm: Pivot을 random하게 고르는 selection 알고리즘 (expected running time은 O(n))

Monte-Carlo Algorithm: k개의 숫자를 random하게 고른 다음 max 값을 리턴. (error 확률은 1/2^k 이하)

2. Competitive ratio

Competitive ratio of a deterministic online algorithm

정의: 어떤 (deterministic) online algorithm A가 모든 입력 string σ에 대하여 다음을 만족하면 c-competitive하다고 한다.

C_A(σ) ≤ c*C_OPT(σ)+b

A의 competitive ratio는 A가 c-competitive한 가장 작은 c를 의미한다. A가 가능한 가장 작은 competitive ratio를 가질 때 A를 strongly competitive하다고 한다.

Competitive ratio of a randomized online algorithm

정의: 어떤 randomized online algorithm B가 모든 입력 string σ에 대하여 다음을 만족하면 c-competitive하다고 한다. 여기서 B의 cost C_B(σ)는 random variable이다.

E[C_B(σ)] ≤ c*C_OPT(σ)+b

B의 competitive ratio는 B가 c-competitive한 가장 작은 c를 의미한다.

3. 2-server 문제

(1) 2-server 문에제 대한 Chrobak-Larmore 알고리즘

요청 위치 x에 대하여 line에서 active server(x와 자신 사이에 다른 서버가 없는 서버)들이 모두 x방향으로 같은 속도로 이동한다. 한 서버가 x에 도달하면 종료.

(2) 2-competitive 증명

논문과 같이 notation을 정의하고 potential function을

$$\Phi = 2\|M\| + \|s1, s2\|$$

로 정의하고 다음 두 가지를 보이면 2-competitive함이 증명된다.

(a) $\tilde{\Phi}^t - \Phi^{t-1} \le 2 \cdot A^t$

증명) adversary의 한 서버를 d만큼 움직이면 ‖M‖도 최대 d만큼 증가할 수 있다. S의 서버는 그대로이므로 Φ는 2d만큼 증가 (증명 끝)

(b) $\tilde{\Phi}^t - \Phi^t \le S^t$

증명) active 서버의 개수 q가 1인 경우와 2인 경우로 나누어 생각해보자.

q=1: $S^t = d$. ‖M‖는 d 감소하고 s1과 s2의 거리는 최대 d만큼 증가하므로 Φ는 최대 $-d = -S^t$만큼 증가한 값을 가질 수 있다.

q=2: $S^t = 2d$. 매치에서 최소 1개의 서버는 d만큼 가까워지므로 ‖M‖가 가장 커지는 경우는 남은 1개의 서버가 거리를 d만큼 증가시키는 경우이다. 그러므로 ‖M‖의 최대값은 이전 값과 같은 값이 된다. 그리고 s1과 s2의 거리는 2d만큼 줄어들기 때문에 Φ의 다음 값은 최대 $-2d = -S^t$만큼 감소한 값이 된다.

위 2가지 경우로부터 항상 $\tilde{\Phi}^t - \Phi^t \le S^t$ 임을 알 수 있다.

그러므로 Chrobak-Larmore 알고리즘은 2-competitive하다.

4. One-way trading

문제: 임의의 deterministic 알고리즘 A2에 대하여 모든 입력 σ 에 대해 E[C_A1(σ)]=C_A2(σ)가 되는 randomized 알고리즘 A2가 존재함을 보여라.

증명) A2가 i번째 단계에서 재산의 s_i 비율만큼을 trade할 때 A1이 s_i의 확률로 전체 재산을 trade하면 randomized 알고리즘 A2는 E[C_A1(σ)]=C_A2(σ)이 된다.

5. One-way trading에 대한 threat-based policy의 2가지 룰에 대하여 설명하라.

Let c be any competitive ratio that can be attained by some one-way trading algorithm. First, assume that c is known to the trader. The threat-based algorithm consists of two rules.

(1) Consider converting dollars to won only when the current rate is the highest seen so far.
(2) When converting dollars, convert just enough to ensure that a competitive ratio c would be obtained if an adversary dropped the exchange rate to the minimum possible rate and kept it.

6. 프로젝트 설명