CC4102 - Control 3

Prof. Gonzalo Navarro

15 de Noviembre de 2012

P1 (3.0 pt)

Considere dos robots en una línea recta, y una secuencia de pedidos que implican que alguno de ellos se traslade a cierto punto de la recta. Se desea minimizar la distancia total recorrida por ambos robots. Si se conoce la secuencia de antemano, se puede construir un algoritmo óptimo (no importa cómo, para este control). Se busca un algoritmo competitivo que reciba cada pedido después de haber servido el anterior.

1. (1pt) Muestre que la estrategia de mover el robot más cercano al pedido no es competitiva.
2. (2pt) Muestre que modificando la estrategia para que, si el pedido cae entre los dos robots, se muevan *ambos* hacia el punto del pedido, hasta que uno de ellos llegue, es 2-competitiva. Para ello, llamemos s_1 y s_2 las posiciones de los robots en la estrategia óptima, y a_1 y a_2 en la aproximada. Considere la función potencial Φ tal que $\Delta\Phi = 2(|a_1 - s_1| + |a_2 - s_2|) + |a_1 - a_2|$. Muestre que (1) cuando el óptimo se mueve en x, $\Delta\Phi \leq 2x$, y (2) cuando el algoritmo online paga costo x, $\Delta\Phi \leq -x$. Muestre que eso implica que el algoritmo es 2-competitivo.

P2 (3.0 pt)

Recuerde el problema de recubrir los vértices de un grafo $G = (V, E)$ cuando hay pesos $c : V \rightarrow R^+$ asociados a cada nodo. Se desea encontrar un $V' \subseteq V$ tal que toda arista de E tenga al menos una de las dos puntas en V', y minimizar $C = \sum_{v \in V'} c(v)$.

En clase se vio una 2-aproximación sencilla para el caso sin pesos ($c(v) = 1$) y una basada en programación lineal para el caso con pesos. Considere ahora la siguiente variante *aleatorizada* del método más sencillo, pero que funciona con pesos:

Algoritmo: *Partir de un $V' = \emptyset$. Ir tomando las aristas $e \in E$ en algún orden. Para cada $e = \{u, v\}$, con probabilidad $\frac{c(v)}{c(u)+c(v)}$ incluir u en V', sino incluir v en V'. Luego eliminar todas las otras aristas incidentes en el nodo incluido, y continuar eligiendo otro e hasta vaciar E.*

(Observe que u se incluye con probabilidad $c(v)/\ldots$, no $c(u)/\ldots$).

Demuestre que este algoritmo aleatorizado obtiene, en el caso esperado, una 2-aproximación. Para ello, considere las aristas e que el algoritmo procesa, metiendo una de sus dos puntas en V'. Sume la contribución esperada a C de la *arista* e, en vez de pensar en los *nodos* de esa arista. Compare con lo que debería hacer un algoritmo óptimo con e, de forma similar a como se hizo en clase para el caso $c() = 1$.

Tiempo: 2 horas | Con una hoja de apuntes