Programación lineal

Dr. Eduardo A. Rodríguez Tello

Cinvestav-Tamaulipas

5 de marzo del 2013

1. Programación lineal
   - Introducción
   - PL en espacios dimensionales mayores
   - PL en dos dimensiones
   - Construcción determinística incremental
   - Análisis de complejidad
   - Algoritmo aleatorizado
   - Análisis inverso para PL aleatorizada

Cinvestav

- El material de esta clase está basado en el capítulo 4 del libro: Mark de Berg, Otfried Cheong, Marc van Kreveld and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition (April 16, 2008), ISBN-10: 3540779736.

Cinvestav

1 Programación lineal
- Introducción
- PL en espacios dimensionales mayores
- PL en dos dimensiones
- Construcción determinística incremental
- Análisis de complejidad
- Algoritmo aleatorizado
- Análisis inverso para PL aleatorizada

Cinvestav

Introducción

- La última clase se estudió el problema del cálculo de la intersección de n semiplanos, y presentó un algoritmo óptimo $O(n \log n)$ para este problema
- En muchas aplicaciones no es importante conocer todo el polígono (o, en general todo el poliedro en dimensiones mayores), sino sólo encontrar un punto particular de interés
- Una aplicación especialmente importante es la de **Programación Lineal** (PL)

Cinvestav

Introducción

- En PL se nos da un conjunto de desigualdades lineales, o **restricciones**, que podemos ver como la definición de un poliedro (posiblemente vacío o no acotado) en el espacio, conocido como **región factible**
- También se da una **función objetivo** lineal, la cual debe ser minimizada o maximizada sujeta a las restricciones

Cinvestav

Introducción

- La descripción típica de un problema de programación lineal d-dimensional pueden ser:

$$\begin{array}{lll} \text{Maximizar:} & c_1 x_1 + \cdots + c_d x_d & \\ \text{Sujeto a:} & a_{1,1} x_1 + \cdots + a_{1,d} x_d & \leq \quad b_1 \\ & a_{2,1} x_1 + \cdots + a_{2,d} x_d & \leq \quad b_2 \\ & \vdots & \\ & a_{n,1} x_1 + \cdots + a_{n,d} x_d & \leq \quad b_n \end{array} \tag{1}$$

- donde $a_{i,j}$, c_i, y b_i son números reales

Introducción

- Esto también puede ser expresado en notación matricial de la siguiente forma:

$$\begin{aligned} &\text{Maximizar:} && c^T x, \\ &\text{Sujeto a:} && Ax \leq b \end{aligned} \tag{2}$$

- donde c y x son vectores d-dimensionales, b es un vector n-dimensional y A es una matriz de $n \times d$

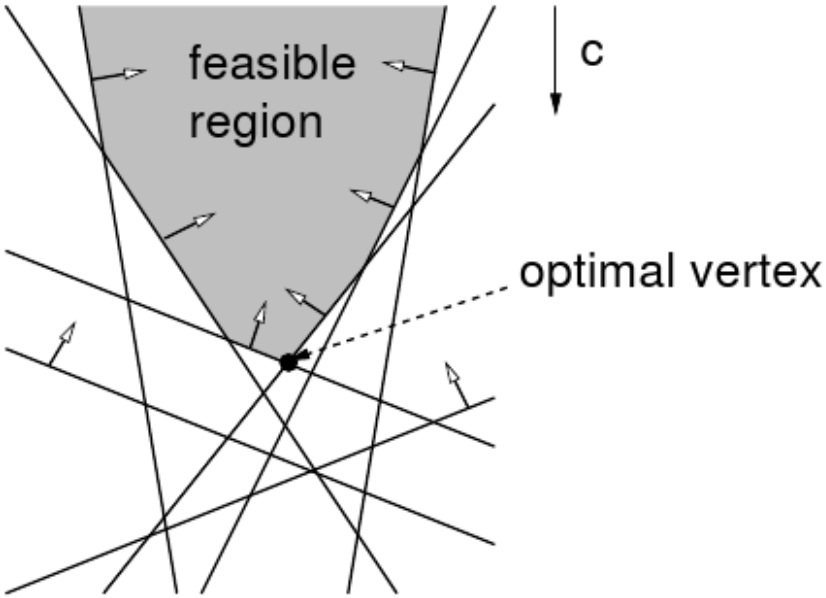
Introducción

- Desde una perspectiva geométrica, la región factible es la intersección de semiespacios, por lo tanto, es un poliedro convexo (posiblemente no acotado) en el d-espacio
- Podemos pensar en la función objetivo como un vector $\vec{c}$, y el problema es encontrar el punto de la región factible que es más lejano en la dirección $\vec{c}$, llamado **vértice óptimo**

Cinvestav

Introducción

- Programación lineal en 2 dimensiones



Introducción

- Tenga en cuenta que la magnitud de $\vec{c}$ es irrelevante, ya que el problema permanece sin cambios para cualquier escalar positivo múltiplo de $\vec{c}$
- En muchos de nuestros ejemplos, vamos a imaginar que $\vec{c}$ es el vector que apunta hacia abajo (es decir, $\vec{c} = (0, -1)$), y por lo tanto, el problema es sólo el de encontrar el punto más bajo (con mínima coordenada y) de la región factible
- Esto no implica pérdida de generalidad, ya que siempre es posible transformar el problema en uno en el que $\vec{c} = (0, -1)$ mediante una rotación adecuada

Cinvestav

Introducción

- Existen tres posibles resultados para un determinado problema de PL:
  - **Factible:** El punto extremo existe (y suponiendo la posición general es único) como un vértice de la región factible
  - **No factible:** La región factible es no acotada, y por lo tanto, no hay solución
  - **No acotado:** La región factible es no acotada en la dirección de la función objetivo, por lo que no existe solución óptima finita

Cinvestav

Introducción

- Como se mencionó anteriormente, sin la suposición de la posición general, es posible que un problema de PL factible tenga un número infinito de soluciones óptimas (finitas) de igual valor
- Esto sucede cuando una arista o cara de la región factible es ortogonal al vector de la función objetivo
- En estos casos es común romper los empates de alguna manera para producir una solución única
- Por ejemplo, podríamos exigir que la solución sea lexicográficamente máxima, es decir, entre todas las soluciones óptimas seleccionar una con el vector lexicográfico máximo

Cinvestav

1 Programación lineal
- Introducción
- PL en espacios dimensionales mayores
- PL en dos dimensiones
- Construcción determinística incremental
- Análisis de complejidad
- Algoritmo aleatorizado
- Análisis inverso para PL aleatorizada

Cinvestav

PL en espacios dimensionales mayores

- La PL es una técnica muy importante utilizada en la solución de problemas de optimización de gran tamaño
- Las instancias típicas pueden comprender cientos de miles de restricciones en un espacio altamente dimensional
- Es sin duda una de las formulaciones más importantes de problemas generales de optimización
- Los principales métodos utilizados para la solución de problemas de PL altamente dimensionales son el **algoritmo simplex** y diversos **métodos de punto interior**

Cinvestav

PL en espacios dimensionales mayores

- El algoritmo simplex trabaja encontrando un vértice en el poliedro de la región factible, y luego camina arista por arista hacia abajo hasta llegar a un mínimo local
- Por convexidad, el mínimo local es el mínimo global
- Es bien conocido que existen casos en los que el algoritmo simplex se ejecuta en tiempo exponencial
- La incógnita de si la PL ere resoluble en tiempo polinomial se mantuvo sin respuesta hasta que Khachiyan publicó su algoritmo del elipsoide (a finales de los 70) y Karmarkar un algoritmo más práctico de punto interior (a mediados de los 80)

Cinvestav

1 Programación lineal
- Introducción
- PL en espacios dimensionales mayores
- PL en dos dimensiones
- Construcción determinística incremental
- Análisis de complejidad
- Algoritmo aleatorizado
- Análisis inverso para PL aleatorizada

Cinvestav

PL en dos dimensiones

- Nos limitaremos a estudiar en esta clase instancias de PL de baja dimensionalidad
- Existen varios problemas interesantes de optimización que se pueden plantear como un problema de PL de baja dimensionalidad, o como problemas de optimización estrechamente relacionados
- Un ejemplo es el problema de encontrar el círculo de radio mínimo que encierra un conjunto de n puntos (capítulo 4 libro de Berg et al.)

Cinvestav

PL en dos dimensiones

- Consideremos por ahora el problema sólo en el plano
- Aquí sabemos que existe un algoritmo $O(n \log n)$ basado sólo en el cálculo del poliedro factible, y encontrando su vértice más abajo
- Sin embargo, ya que sólo estamos interesados en un punto del poliedro, parece que podemos esperar hacerlo de manera más eficiente
- Además, en un espacio d dimensional, un politopo delimitado por n semiespacios puede tener una complejidad máxima de $\Omega(n^{\lfloor d/2 \rfloor})$

Cinvestav

PL en dos dimensiones

- Mostraremos que los problemas de PL en 2 dimensiones pueden resolverse en tiempo $O(n)$
- Este algoritmo puede ser aplicado en cualquier dimensión d, pero el factor constante oculto por la notación asintótica crece como $d!$, por lo que no es práctico, excepto para dimensiones muy pequeñas

Cinvestav

PL en dos dimensiones

- Supongamos que se nos da un conjunto de n desigualdades lineales (semiplanos) $H = \{h_1, \ldots, h_n\}$ de la forma:

$$h_i : a_{i,x}x + a_{i,y}y \leq b_i \tag{3}$$

- y una función objetivo diferente de cero dada por el vector $\vec{c} = (c_x, c_y)$

- El problema es determinar si la instancia de PL es factible y acotada, y si lo es, calcular el punto factible $p = (p_x, p_y)$ que maximiza el producto punto $c_x p_x + c_y p_y$ (tratando p como vector)

Cinvestav

PL en dos dimensiones

- Por lo tanto, asumiremos que $\vec{c} = (0, -1)$
- Note que a medida que se decrementa la coordenada y de un punto, el valor de la función objetivo se incrementa
- De esta forma más grande no significa mejor

Cinvestav

PL en dos dimensiones

- Los dos algoritmos de PL que vamos a discutir son muy simples y se basan en un método llamado **construcción incremental**
- Un método es determinístico y el otro es aleatorizado (no determinístico)
- El método de *construcción incremental* es una de las técnicas algorítmicas más comunes en geometría computacional, y esta es otra razón importante por la cual estudiar el problema de PL
- En primer lugar, analizaremos el algoritmo determinístico incremental

Cinvestav

1 Programación lineal
- Introducción
- PL en espacios dimensionales mayores
- PL en dos dimensiones
- Construcción determinística incremental
- Análisis de complejidad
- Algoritmo aleatorizado
- Análisis inverso para PL aleatorizada

Cinvestav

Construcción determinística incremental

- Supongamos por ahora que la instancia del problema de PL es acotada, y además podemos encontrar dos semiplanos cuya intersección (un cono) es acotada con respecto a la función objetivo
- El libro de Berg et al. explica cómo tratar estos casos especiales en un tiempo $O(n)$ adicional
- Vamos a suponer que los semiplanos están numerados como h_1 y h_2, y que v_2 denota este primer vértice óptimo, donde las dos líneas asociadas a estos semiplanos se intersectan

Cinvestav

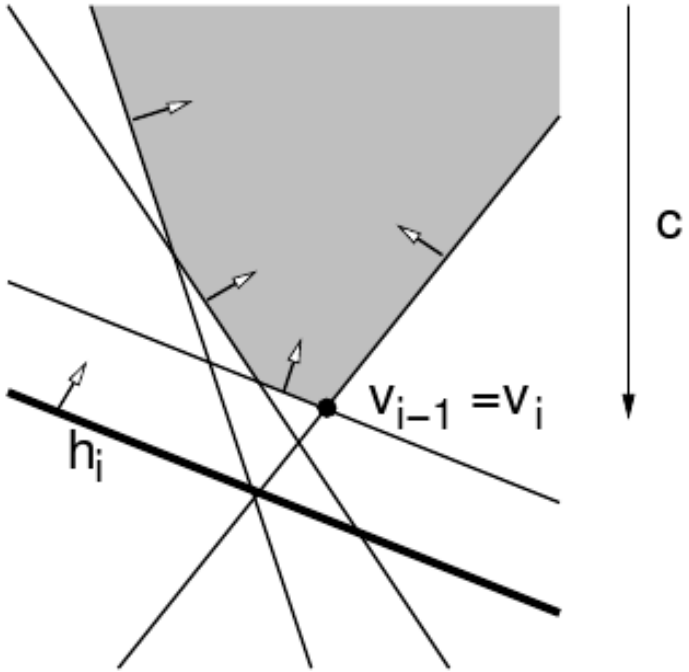
Construcción determinística incremental

- A continuación, añadiremos los semiplanos uno por uno, $h_3, h_4, \ldots,$ y con cada adición se actualiza el vértice óptimo
- Sea v_i el vértice óptimo factible después de la adición de $\{h_1, h_2, \ldots, h_i\}$
- El problema consiste en actualizar v_i con cada nueva adición
- Tenga en cuenta que con cada nueva restricción, la región factible generalmente se reduce, y por tanto el valor de la función objetivo en el vértice óptimo sólo puede decrecer

Cinvestav

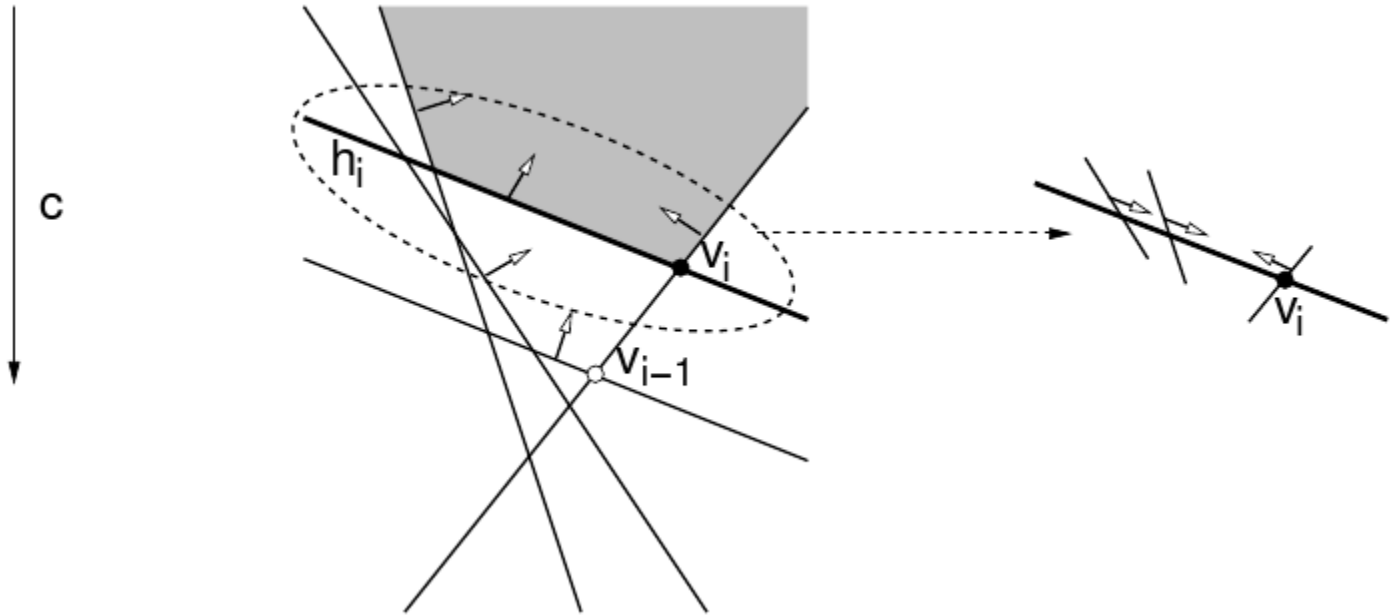
Construcción determinística incremental

- Hay dos casos que pueden surgir cuando se agrega h_i
- En el primer caso, v_{i-1} se encuentra dentro del semiplano h_i, por lo que cumple esta restricción
- Si es así, entonces es fácil ver que el mejor vértice no cambia, i.e. $v_i = v_{i-1}$



Construcción determinística incremental

- En el segundo caso, v_{i-1} viola la restricción h_i por lo que tenemos que encontrar un nuevo vértice óptimo



- Vamos a examinar este segundo caso con mayor detalle

Construcción determinística incremental

- Una observación importante es que (suponiendo que la región factible no es vacía) el nuevo vértice óptimo deberá situarse en la línea de que acota h_i
- Llamemos a esta línea ℓ_i
- Nuestro libro de texto demuestra este hecho formalmente (se recomienda estudiar la demostración), pero aquí presentaremos sólo un argumento intuitivo

Cinvestav

Construcción determinística incremental

- Supongamos que el nuevo vértice óptimo no cae en ℓ_i
- Dibujemos un segmento de línea de v_{i-1} al nuevo óptimo
- Observemos que:
  - Conforme caminamos a lo largo de este segmento el valor de la función objetivo decrece monótonamente (por linealidad)
  - Este segmento debe cruzar ℓ_i (ya que pasa de ser no factible con respecto a h_i a ser factible). Por lo tanto, es maximizado en el punto de cruce, que cae en ℓ_i
- Las propiedades de convexidad y linealidad son muy importantes para la demostración

Cinvestav

Construcción determinística incremental

- Por lo tanto, surge la pregunta de cómo encontrar el vértice óptimo situada en la línea ℓ_i
- Este resulta ser un problema de PL 1-dimensional
- Simplemente se intersectan cada uno de los semiplanos con esta línea
- Cada intersección adoptará la forma de un rayo que cae en la línea
- Podemos pensar que cada rayo representa un intervalo (no acotado a la izquierda ni a la derecha)

Cinvestav

Construcción determinística incremental

- Todo lo que necesitamos hacer es intersectar estos intervalos, y encontrar el punto que maximiza la función objetivo (es decir, el punto más bajo)
- Calcular la intersección de una colección de intervalos, es muy fácil y se puede resolver en tiempo lineal
- Sólo tenemos que encontrar la cota superior más pequeña y la cota inferior más grande
- Seleccionamos el punto que maximiza la función objetivo
- Si este intervalo es vacío, entonces la región factible es vacía, y terminamos reportando que no existe solución

Cinvestav

Construcción determinística incremental

- Observe que hemos resuelto un problema de PL 2-dimensional mediante una reducción a un problema de PL 1-dimensional (que es fácil de resolver)
- Este enfoque general se pueden aplicar a la solución de problemas PL en cualquier dimensión, mediante la reducción repetida a problemas de PL en la dimensión inmediatamente inferior
- Siempre que el vértice óptimo actual no sea factible

Cinvestav

Construcción determinística incremental

Algorithm 2DBOUNDEDLP($H, \vec{c}, m_1, m_2$)
Input. A linear program $(H \cup \{m_1, m_2\}, \vec{c})$, where H is a set of n half-planes, $\vec{c} \in \mathbb{R}^2$, and m_1, m_2 bound the solution.
Output. If $(H \cup \{m_1, m_2\}, \vec{c})$ is infeasible, then this fact is reported. Otherwise, the lexicographically smallest point p that maximizes $f_{\vec{c}}(p)$ is reported.

1. Let v_0 be the corner of C_0.
2. Let $h_1, \ldots, h_n$ be the half-planes of H.
3. **for** $i \leftarrow 1$ **to** n
4. **do if** $v_{i-1} \in h_i$
5. **then** $v_i \leftarrow v_{i-1}$
6. **else** $v_i \leftarrow$the point p on ℓ_i that maximizes $f_{\vec{c}}(p)$, subject to the constraints in H_{i-1}.
7. **if** p does not exist
8. **then** Report that the linear program is infeasible and quit.
9. **return** v_n

1 Programación lineal
- Introducción
- PL en espacios dimensionales mayores
- PL en dos dimensiones
- Construcción determinística incremental
- Análisis de complejidad
- Algoritmo aleatorizado
- Análisis inverso para PL aleatorizada

Cinvestav

Análisis de complejidad

- ¿Cuál es el tiempo de ejecución para el peor caso de este algoritmo?
- Hay aproximadamente n inserciones de semiespacios
- En el paso i, podemos encontrar, ya sea que el vértice óptimo actual es factible, en cuyo caso el tiempo de transformación es constante
- O que el vértice óptimo actual es no factible, entonces tenemos que resolver un programa lineal 1-dimensional con $i - 1$ restricciones

Cinvestav

Análisis de complejidad

- En el peor de los casos, este segundo paso se produce todo el tiempo, por lo que el tiempo de ejecución está determinado por la sumatoria:

$$\sum_{i=3}^{n}(i-1) \leq \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = O(n^2) \tag{4}$$

- Esto a partir de propiedades conocidas de las sumatorias

Cinvestav

Análisis de complejidad

- Sin embargo observe que este peor caso se basa más bien en la suposición pesimista de que el vértice actual es casi siempre no factible
- A continuación se estudiará si este supuesto es razonable, y la mejor manera de tratar esta situación

Cinvestav

1 Programación lineal
- Introducción
- PL en espacios dimensionales mayores
- PL en dos dimensiones
- Construcción determinística incremental
- Análisis de complejidad
- Algoritmo aleatorizado
- Análisis inverso para PL aleatorizada

Cinvestav

Algoritmo aleatorizado

- El tiempo de ejecución $O(n^2)$ del algoritmo determinístico incremental no es muy favorable
- Esto considerando que se podría calcular la totalidad de la región factible en tiempo $O(n \log n)$
- Pero se ha presentado ya que conduce a un muy elegante algoritmo aleatorizado cuyo tiempo esperado de ejecución es $O(n)$
- Tenga en cuenta que la expectativa es sobre la secuencia de números aleatorios, y no depende de ninguna suposición probabilística sobre la distribución de los datos de entrada

Cinvestav

Algoritmo aleatorizado

- El algoritmo funciona exactamente de la misma manera, pero insertamos los semiplanos en orden aleatorio, cada vez actualizando la respuesta en base a los semiplanos existentes
- Este es un ejemplo de una clase general de algoritmos llamados **algoritmos aleatorizados incrementales**
- Sólo hay una diferencia entre este tipo de algoritmos y los determinísticos incrementales que acabamos de estudiar
- La diferencia es que justo antes de ejecutar el algoritmo incremental, se llama un procedimiento que permuta aleatoriamente la lista de entrada inicial

Cinvestav

Algoritmo aleatorizado

Algorithm 2DRANDOMIZEDBOUNDEDLP$(H, \vec{c}, m_1, m_2)$

Input. A linear program $(H \cup \{m_1, m_2\}, \vec{c})$, where H is a set of n half-planes, $\vec{c} \in \mathbb{R}^2$, and m_1, m_2 bound the solution.

Output. If $(H \cup \{m_1, m_2\}, \vec{c})$ is infeasible, then this fact is reported. Otherwise, the lexicographically smallest point p that maximizes $f_{\vec{c}}(p)$ is reported.

1. Let v_0 be the corner of C_0.
2. Compute a *random* permutation $h_1, \ldots, h_n$ of the half-planes by calling RANDOMPERMUTATION$(H[1 \cdots n])$.
3. **for** $i \leftarrow 1$ **to** n
4. **do if** $v_{i-1} \in h_i$
5. **then** $v_i \leftarrow v_{i-1}$
6. **else** $v_i \leftarrow$the point p on ℓ_i that maximizes $f_{\vec{c}}(p)$, subject to the constraints in H_{i-1}.
7. **if** p does not exist
8. **then** Report that the linear program is infeasible and quit.
9. **return** v_n

1. Programación lineal
   - Introducción
   - PL en espacios dimensionales mayores
   - PL en dos dimensiones
   - Construcción determinística incremental
   - Análisis de complejidad
   - Algoritmo aleatorizado
   - Análisis inverso para PL aleatorizada

Cinvestav

Análisis inverso para PL aleatorizada

- Vamos a analizar el tiempo de ejecución del algoritmo aleatorizado incremental de PL mediante una técnica conocida como **análisis inverso** (*backwards analysis*)
- Para ello asumimos que el algoritmo ya terminó de ejecutarse y que devolvió el vértice v_n
- Sea p_i la probabilidad de que la inserción del i-ésimo semiplano en el orden aleatorio resulte en un cambio en el vértice óptimo

Cinvestav

Análisis inverso para PL aleatorizada

- Con probabilidad $(1 - p_i)$ no hay cambio (caso 1), y nos toma un tiempo $O(1)$ determinar ésto (tiempo $O(d)$ en general en d dimensiones)
- Con probabilidad p_i tenemos que invocar una instancia de PL 1-dimensional sobre un conjunto de $i - 1$ semiplanos en la dimensión 1, lo cual implica un tiempo de ejecución $O(i)$
- En general, si $T(n, d)$ es el tiempo esperado de ejecución para n semiespacios en d dimensiones, entonces el costo sería $T(i - 1, d - 1)$

Cinvestav

Análisis inverso para PL aleatorizada

- Combinando esto tenemos un tiempo esperado total de ejecución de:

$$T(n) = \sum_{i=1}^{n}((1-p_i)\cdot 1 + p_i \cdot i) \leq n + \sum_{i=1}^{n} p_i \cdot i \tag{5}$$

- Lo único que resta es determinar el valor de p_i, para ello vamos a aplicar la misma técnica
- Sea S_i un subconjunto arbitrario consistente en i de los semiplanos originales
- Entre todos las $i!$ permutaciones posibles de S_i, ¿en cuántas el vértice óptimo cambia con el i-ésimo paso?

Cinvestav

Análisis inverso para PL aleatorizada

- Sea v_i vértice óptimo para estos i semiplanos
- Note que v_i sólo depende del conjunto S_i y no en el orden de su inserción
- Asumiendo la posición general, hay dos semiplanos h' y h'' de S_i pasando a través de v_i
- Si ninguno de ellos es el último insertado, entonces $v_i = v_{i-1}$, y no hay ningún cambio

Cinvestav

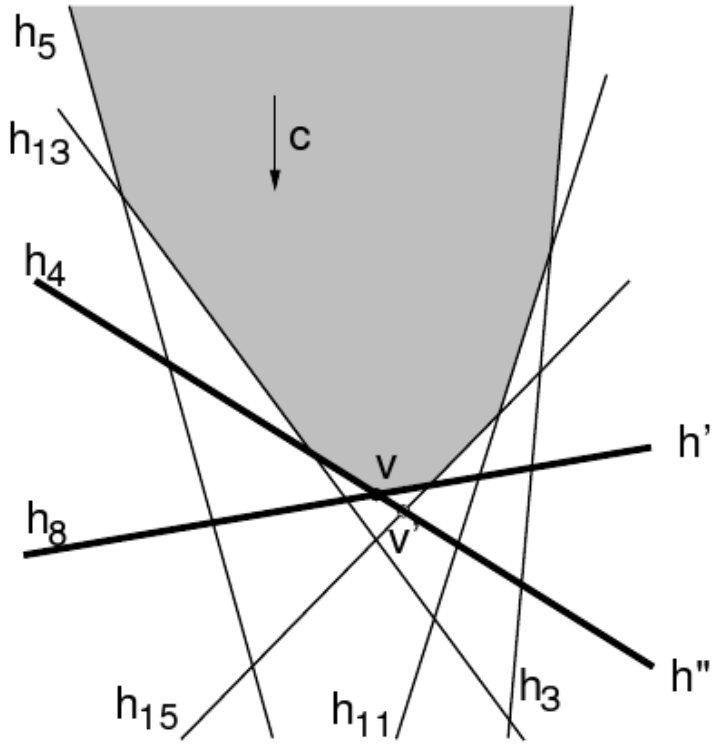
Análisis inverso para PL aleatorizada

- Si h' o h'' fue el último insertado, entonces v_i aún no existía, y por lo tanto el óptimo debió haber cambiado como resultado de esta inserción
- De esta manera, el óptimo cambia si y sólo h' o h'' fue el último semiplano insertado
- Dado que todos los i semiplanos tienen la misma probabilidad de ser los últimos en ser insertados, entonces esto ocurre con probabilidad $2/i$
- Por lo que $p_i = 2/i$

Cinvestav

Análisis inverso para PL aleatorizada

- Para ilustrar esto, consideremos el ejemplo que se muestra en la siguiente figura



- Tenemos $i = 7$ semiplanos que han sido insertados aleatoriamente hasta el momento, y $v_i = v$ es el vértice óptimo actual que está definido por h_4 y h_8

Análisis inverso para PL aleatorizada

- Asumamos que el último semiplano insertado fue h_5
- Consideraremos la situación con todos los semiplanos excepto h_5
- Antes de ésto v ya había sido considerado como el vértice óptimo, por lo tanto insertar h_5 tomó un tiempo $O(1)$
- Por otro lado, si h_8 fue el último en ser insertado, habríamos tenido un vértice óptimo diferente, llamado v' (el óptimo también cambia si el último insertado fuera h_4)

Cinvestav

Análisis inverso para PL aleatorizada

- En resumen, en 2 de 7 casos (h_4 y h_8) en la inserción i-ésima requerimos resolver una instancia de PL 1-dimensional en tiempo $O(i)$ y en 5 de 7 casos un tiempo $O(1)$ es necesario
- Volviendo a nuestro análisis, ya que $p_i = 2/i$ tenemos que:

$$T(n) \leq n + \sum_{i=1}^{n} p_i \cdot i = n + \sum_{i=1}^{n} \frac{2i}{i} = n + 2n = O(n) \tag{6}$$

- Por lo tanto, el tiempo de ejecución esperado es lineal en n

Cinvestav