Análisis y Diseño de Algoritmos

Búsqueda Local

Arturo Díaz Pérez
Sección de Computación
Departamento de Ingeniería Eléctrica
CINVESTAV-IPN
Av. Instituto Politécnico Nacional No. 2508
Col. San Pedro Zacatenco
México, D. F. CP 07300

Tel. (5)747 3800 Ext. 3755
e-mail: `adiaz@cs.cinvestav.mx`

Estrategias Generales

- Estrategias generales para resolver problemas de optimización sobre espacios de búsqueda exponenciales
  - Búsqueda Local
  - Recocido Simulado (Simulated Annealing)
  - Búsqueda Tabú
  - Algoritmos Genéticos
  - Redes Neuronales
  - Técnicas de Aleatorización

Búsqueda Local: Estrategia

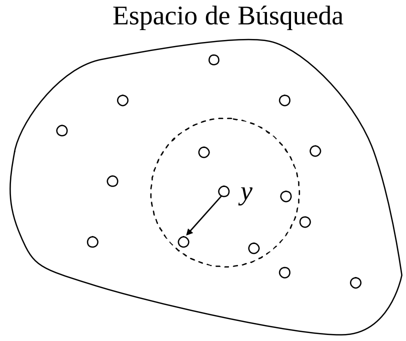
- La idea básica de los algoritmos de búsqueda local es:
  - Iniciar con una solución inicial
    - Generada aleatoriamente, o
    - Hallada con algún otro algoritmo
  - Aplicar a la solución actual una ***transformación*** de algún conjunto dado de transformaciones para mejorar la solución
    - La solución mejorada se convierte en la solución actual
  - Repetir lo anterior hasta que ninguna transformación del conjunto mejore a la solución actual

Análisis y Diseño de Algoritmos *LocalSearch*-3

Búsqueda Local: Estrategia

- Si *y* es una solución factible, entonces *z* es una solución factible si no difiere sustancialmente de *y*.
- Dada una solución factible *y*, un algoritmo de búsqueda local busca una solución vecina con un valor mejorado de la función objetivo.

Espacio de Búsqueda



Análisis y Diseño de Algoritmos *LocalSearch*-4

Búsqueda Local: Estrategia

- La solución resultante no necesariamente es óptima.
  - Dependerá de las transformaciones aplicadas.
  - Si el conjunto dado de transformaciones incluye todas las transformaciones que toman una solución *y* y la reemplazan por cualquier otra, *z*, entonces, la estrategia anterior no se detendrá hasta encontrar una solución óptima.
  - Sin embargo, en este caso el tiempo para aplicar generar todas las transformaciones sería el mismo que que se requiere para examinar todas las soluciones, entonces, el enfoque anterior no tiene sentido.
- Cuando no es posible obtener un mejoramiento en la solución el algoritmo se detiene en un óptimo local
  - Es tal que al alcanzar una solución *y* ninguno de sus vecinos es mejor.

Análisis y Diseño de Algoritmos — *LocalSearch*-5

Búsqueda Local: Estrategia

- El método tiene sentido cuando se puede restringir el conjunto de transformaciones a un conjunto pequeño, así se pueden considerar todas las transformaciones en un tiempo corto.
- Las transformaciones son conocidas como transformaciones locales.
- Es necesario definir una estructura de vecindad para una solución factible *y*
  - No es necesario encontrar todos los vecinos de *y*, es suficiente con determinar si existe uno que mejore a *y*

Análisis y Diseño de Algoritmos — *LocalSearch*-6

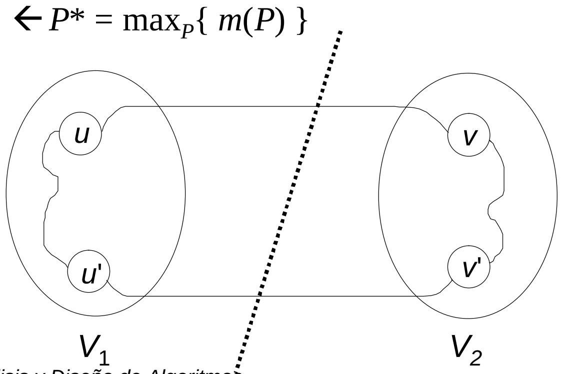
Vecindad y Tiempo Polinomial

- La estructura de vecindad nos permite movernos de una solución a una solución vecina mejor
  - Si esto se puede realizar en tiempo polinomial, y
  - El óptimo global se obtiene después de haber considerado un número polinomial de soluciones, entonces
  - El algoritmo de búsqueda local resuelve al problema de manera óptima en tiempo polinomial.
- Claramente, no esperamos que eso suceda para problemas NP con un espacio de búsqueda exponencial
- Lo factible es encontrar un algoritmo de búsqueda local que encuentre una solución aproximada correspondiente a
  - un óptimo local respecto
  - a una estructura de vecindad

Análisis y Diseño de Algoritmos — *LocalSearch*-7

Máximo Corte

- Sea $G=(V, E)$ una gráfica.
  - Sea $P=(V_1,V_2)$ una partición de V
    - $V = V_1 \cup V_2$ y $V_1 \cap V_2 = \emptyset$.
  - Sea $E_P = \{ (u, v) \in E \mid u \in V_1 \text{ y } v \in V_2 \}$
    - $m(E_P) = | E_P |$
- El problema es hallar P^* tal que
  - $P^* = \max_P\{ m(P) \}$



¿Cuántas particiones posibles existen en una gráfica con n vértices?

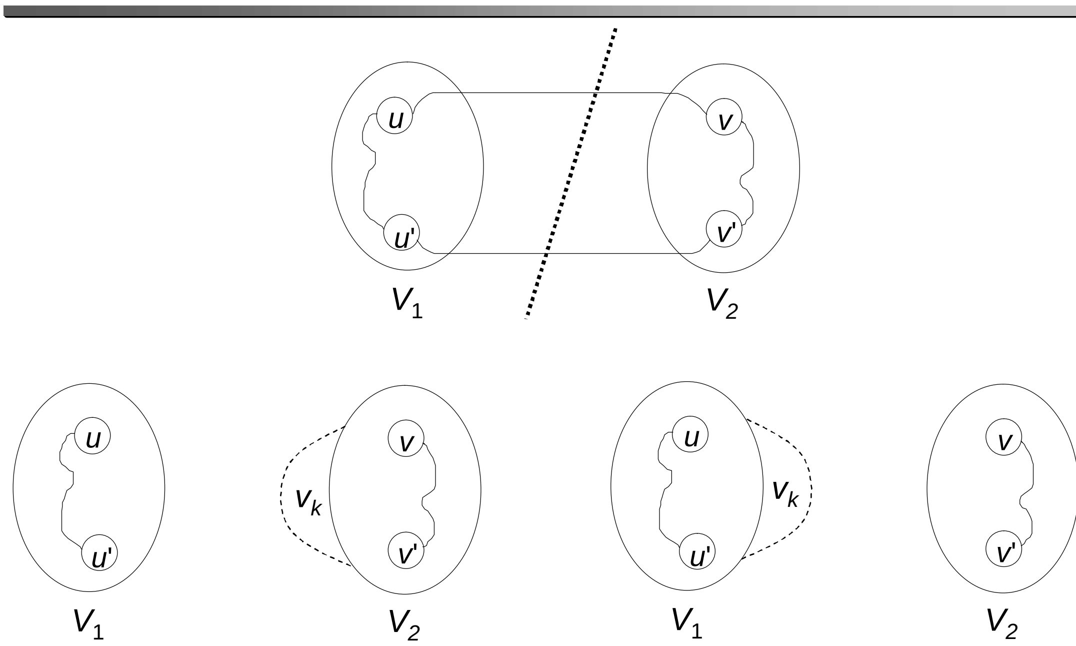
Análisis y Diseño de Algoritmos — *LocalSearch*-8

Búsqueda Local para MC

- Solución inicial
  - $V_1 = \emptyset$ y $V_2 = V$.
- Estructura de vecindad
  - Dada una solución $P=(V_1,V_2)$ consiste de todas las particiones $P_k=(V_{1k},V_{2k})$, para todos los $k = 1, ..., |V|$, tal que,
    - Si el vértice $v_k \in V_1$, entonces,
      - $V_{1k}= V_1 - \{k\}$ y $V_{2k}= V_2 \cup \{k\}$
    - Si el vértice $v_k \notin V_1$, entonces,
      - $V_{1k}= V_1 \cup \{k\}$ y $V_{2k}= V_2 - \{k\}$

Análisis y Diseño de Algoritmos *LocalSearch*-9

Estructura de Vecindad



Análisis y Diseño de Algoritmos *LocalSearch*-10

Exactitud de la Solución

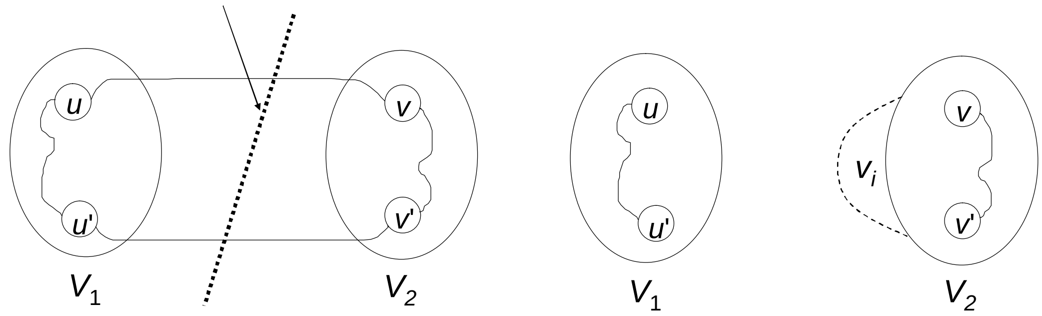
- Sean
  - x una instancia del problema de corte máximo,
  - $P=(V_1,V_2)$ un óptimo local con respecto a la vecindad N
  - $m_N(x)$ su medida ($m(E_P)$), y
  - $m^*(x)$ el valor de la solución óptima, entonces,
  - $m^*(x) / m_N(x) \leq 2$.
- Si m es el número de arcos de la gráfica es suficiente con probar que $m_N(x) \geq m/2$
- Si m_1 y m_2 son los arcos que unen a vértices dentro de V_1 y V_2, respectivamente, entonces
  - $m = m_1 + m_2 + m_N(x)$

Exactitud de la Solución

- Dado cualquier vértice v_i, se definen
  - $m_{1i}= \{v \mid v \in V_1 \text{ y } (v,v_i) \in E\}$
  - $m_{2i}= \{v \mid v \in V_2 \text{ y } (v,v_i) \in E\}$

Si se mueve v_i de V_1 a V_2, ¿cuántos arcos hay en el corte?

$m(E_P)$ vértices de V_1 a V_2

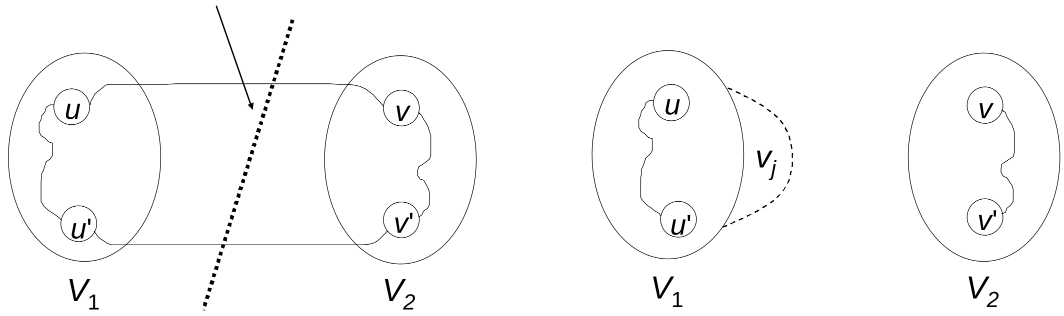


$$m(E_P) - |m_{2i}| + |m_{1i}| = m(E_P) + |m_{1i}| - |m_{2i}|$$

Estructura de Vecindad

Si se mueve v_j de V_2 a V_1, ¿cuántos arcos hay en el corte?

$m(E_P)$ vértices de V_1 a V_2



$$m(E_P) - |m_{1j}| + |m_{2j}| = m(E_P) + |m_{2j}| - |m_{1j}|$$

Exactitud de la Solución

- Si $P=(V_1,V_2)$ es una solución localmente óptima, entonces, para cualquier vértice v_k, la solución proporcionada por $P_k=(V_{1k},V_{2k})$, tiene un valor que no es mejor que $m_N(x)$.
  - Para cualquier $v_i \in V_1$, $|m_{1i}| - |m_{2i}| \leq 0$.
  - Para cualquier $v_j \in V_2$, $|m_{2j}| - |m_{1j}| \leq 0$.
- Sumando todos los vértices en V_1 y V_2 se obtiene

$$\sum_{v_i \in V_1} (|m_{1i}| - |m_{2i}|) = 2m_1 - m_N(x) \leq 0$$

$$\sum_{v_j \in V_2} (|m_{2j}| - |m_{1j}|) = 2m_2 - m_N(x) \leq 0$$

Exactitud de la Solución

$$\sum_{v_i \in V_1} (|m_{1i}| - |m_{2i}|) = 2m_1 - m_N(x) \le 0$$

$$\sum_{v_j \in V_2} (|m_{2j}| - |m_{1j}|) = 2m_2 - m_N(x) \le 0$$

- De aquí, $m_1 + m_2 - m_N(x) \le 0$
- Ya que $m = m_1 + m_2 + m_N(x)$,
- Se cumple que $m_N(x) \ge m/2$

Análisis y Diseño de Algoritmos *LocalSearch*-15

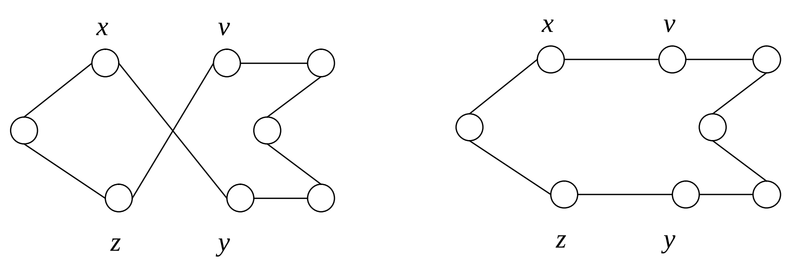
Comentarios

- La estructura de la vecindad determina
  - la calidad de la aproximación de la solución
  - el tiempo de ejecución del algoritmo
- Aspectos a considerar
  - la calidad de la aproximación deseada
  - el orden en el cual se consideran a los miembros de la vecindad
  - la complejidad para verificar que la vecindad no contiene una mejor solución
  - el número de soluciones generadas antes de que se encuentre un óptimo local

Análisis y Diseño de Algoritmos *LocalSearch*-16

Búsqueda Local para el TSP

- Solución inicial
  - Para una gráfica completamente conectada cualquier permutación es una solución factible.
    - $(c_1, c_2, ..., c_n)$
- Estructura de vecindad
  - Dado un tour I en una gráfica completamente conectada reemplazando dos arcos (x,y) y (v,z) en I por los arcos (x,v) y (y,z) produce un nuevo tour I'.



Análisis y Diseño de Algoritmos *LocalSearch*-17

Búsqueda Local para el TSP

- Estructura de vecindad
  - Dado un tour I en una gráfica completamente conectada reemplazando dos arcos (x,y) y (v,z) en I por los arcos (x,v) y (y,z) produce un nuevo tour I'.
  - La vecindad de I consiste de todas aquellas soluciones que se obtienen haciendo el reemplazo de pares arcos.
  - ¿Cuántos vecinos tiene I si consiste de n ciudades?
    - Respuesta $O(n^2)$.

Análisis y Diseño de Algoritmos *LocalSearch*-18

Caso de Estudio

- Ciudades colocadas aleatoriamiente en el plano euclideano
  - La calidad de la solución obtenida depende de la solución inicial
  - Al ejecutar varias veces el algoritmo de búsqueda local con soluciones iniciales diferentes, el algoritmo general puede ser muy efectivo
  - Estudios experimentales muestran que la solución obtenida está dentro del 5.5% de la solución óptima.
    - $I^*/I \geq .945$ (factor de aproximación)

Análisis y Diseño de Algoritmos *LocalSearch*-19