Heurísticas

Leopoldo Taravilse[1]

[1]Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Algoritmos y Estructuras de Datos III

Menú del día

Menú del día

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

¿Ejemplos?

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

¿Ejemplos?

- El ejercicio de los ramales del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

¿Ejemplos?

- El ejercicio de los ramales del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- El ejercicio de las girl scouts del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

¿Ejemplos?

- El ejercicio de los ramales del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- El ejercicio de las girl scouts del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- Árbol Generador Mínimo/Máximo de un grafo (TP2)

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

¿Ejemplos?

- El ejercicio de los ramales del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- El ejercicio de las girl scouts del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- Árbol Generador Mínimo/Máximo de un grafo (TP2)
- Camino Mínimo/Máximo en un grafo (TP2)

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

¿Ejemplos?

- El ejercicio de los ramales del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- El ejercicio de las girl scouts del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- Árbol Generador Mínimo/Máximo de un grafo (TP2)
- Camino Mínimo/Máximo en un grafo (TP2)
- TSP

Problemas de Optimización Combinatoria

¿Qué es un Problema de Optimización Combinatoria?

Problema de Optimización (Combinatoria)

Dado un conjunto (finito) $\mathcal{S}$ de soluciones factibles y una función objetivo a optimizar $f : \mathcal{S} \to \mathbb{R}$, hallar $s^* \in \mathcal{S}$ tal que $f(s^*) \geq f(s)$ para todo $s \in \mathcal{S}$.

¿Ejemplos?

- El ejercicio de los ramales del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- El ejercicio de las girl scouts del TP 1 (¿qué es $\mathcal{S}$ y qué es $f(s)$?)
- Árbol Generador Mínimo/Máximo de un grafo (TP2)
- Camino Mínimo/Máximo en un grafo (TP2)
- TSP
- etc.

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas...

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas...
  (¡puede ser muy feo! podría haber muchísimas soluciones)

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas... (¡puede ser muy feo! podría haber muchísimas soluciones)
- Hacemos un algoritmo "eficiente" que no revise todas...

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas...
  (¡puede ser muy feo! podría haber muchísimas soluciones)
- Hacemos un algoritmo "eficiente" que no revise todas...
  (¡puede ser muy difícil!)

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas...
  (¡puede ser muy feo! podría haber muchísimas soluciones)
- Hacemos un algoritmo "eficiente" que no revise todas...
  (¡puede ser muy difícil!)
- Hacemos un algoritmo que sólo revise todas en el peor caso (e.g., un backtracking)...

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas... (¡puede ser muy feo! podría haber muchísimas soluciones)
- Hacemos un algoritmo "eficiente" que no revise todas... (¡puede ser muy difícil!)
- Hacemos un algoritmo que sólo revise todas en el peor caso (e.g., un backtracking)... (¡puede ser muy ineficiente en general!)

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas... (¡puede ser muy feo! podría haber muchísimas soluciones)
- Hacemos un algoritmo "eficiente" que no revise todas... (¡puede ser muy difícil!)
- Hacemos un algoritmo que sólo revise todas en el peor caso (e.g., un backtracking)... (¡puede ser muy ineficiente en general!)
- Implementamos una heurística que funcione rápido y encuentre una solución "cercana" al óptimo...

Problemas de Optimización Combinatoria

¿Cómo los resolvemos?

- Analizamos cada solución en $\mathcal{S}$ y nos quedamos con la mejor de ellas... (¡puede ser muy feo! podría haber muchísimas soluciones)
- Hacemos un algoritmo "eficiente" que no revise todas... (¡puede ser muy difícil!)
- Hacemos un algoritmo que sólo revise todas en el peor caso (e.g., un backtracking)... (¡puede ser muy ineficiente en general!)
- Implementamos una heurística que funcione rápido y encuentre una solución "cercana" al óptimo... (ok, vamos con eso...)

Menú del día

Heurísticas Golosas

Repasemos un poco: ¿Qué es una heurística golosa?

Heurísticas Golosas

Repasemos un poco: ¿Qué es una heurística golosa?

Heurística Golosa

Una heurística golosa es un algoritmo que va construyendo la solución haciendo en cada paso, la "mejor elección a nivel local" de un conjunto de elecciones candidato.

Heurísticas Golosas

Repasemos un poco: ¿Qué es una heurística golosa?

Heurística Golosa

Una heurística golosa es un algoritmo que va construyendo la solución haciendo en cada paso, la "mejor elección a nivel local" de un conjunto de elecciones candidato.

¿Qué es eso de la "mejor elección a nivel local"?

Heurísticas Golosas

Repasemos un poco: ¿Qué es una heurística golosa?

Heurística Golosa

Una heurística golosa es un algoritmo que va construyendo la solución haciendo en cada paso, la "mejor elección a nivel local" de un conjunto de elecciones candidato.

¿Qué es eso de la "mejor elección a nivel local"?

Veamos algunos ejemplos:

- AGM en un grafo
- Camino mínimo en un grafo
- Camino máximo en un grafo
- TSP

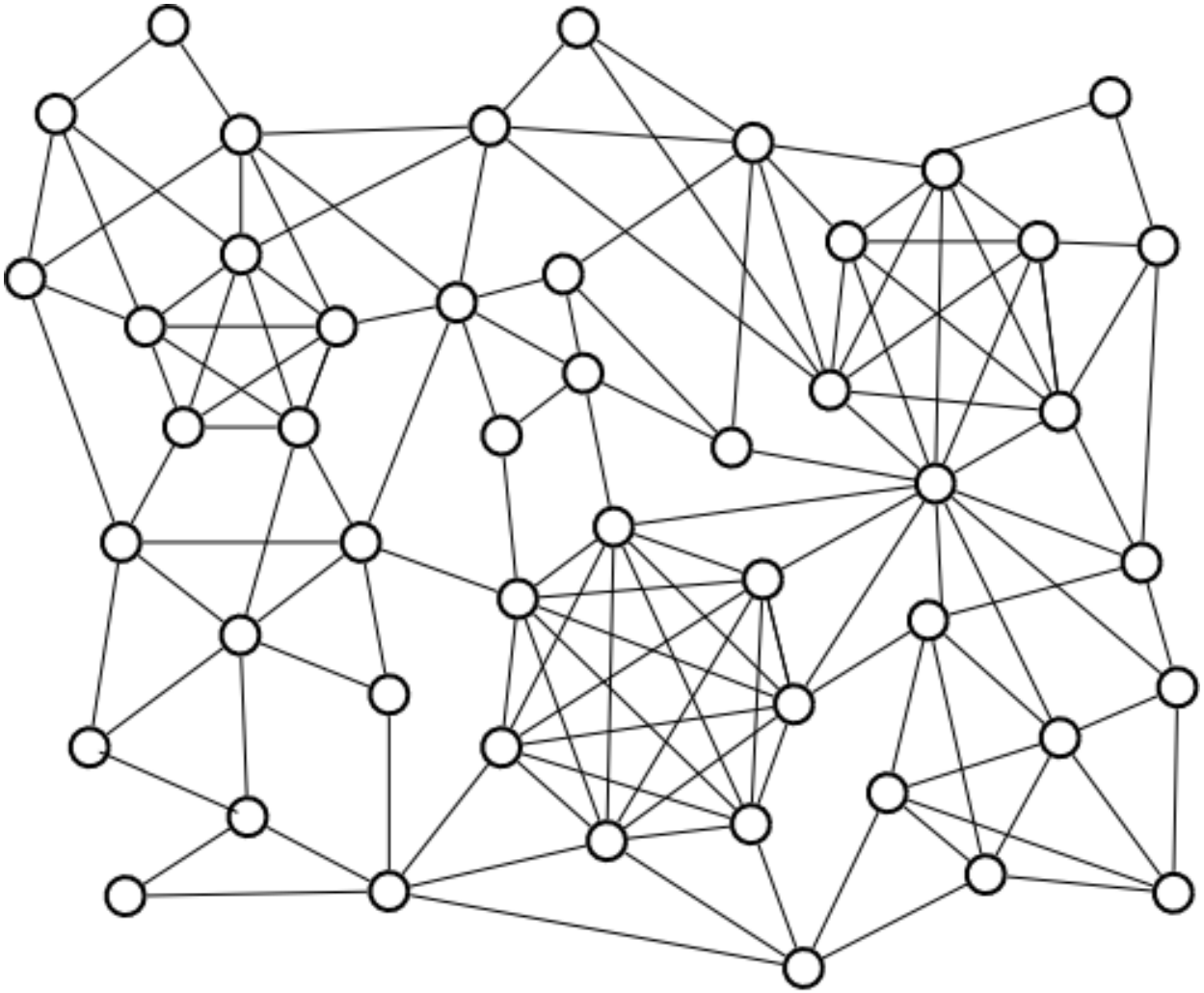
Heurísticas Golosas - Ejemplo: Clique Máxima

Tomemos como ejemplo el siguiente problema de OC:

Clique máxima en un grafo

Dado un grafo $G = (V, E)$, decimos que $K \subseteq V$ es una *clique* de G si el subgrafo de G inducido por K es completo. El problema de clique máxima consiste en hallar una clique de G de máxima cardinalidad

Heurísticas Golosas - Ejemplo: Clique Máxima



Heurísticas Golosas - Ejemplo: Clique Máxima

- ¿Se les ocurre alguna forma de resolverlo en forma eficiente?

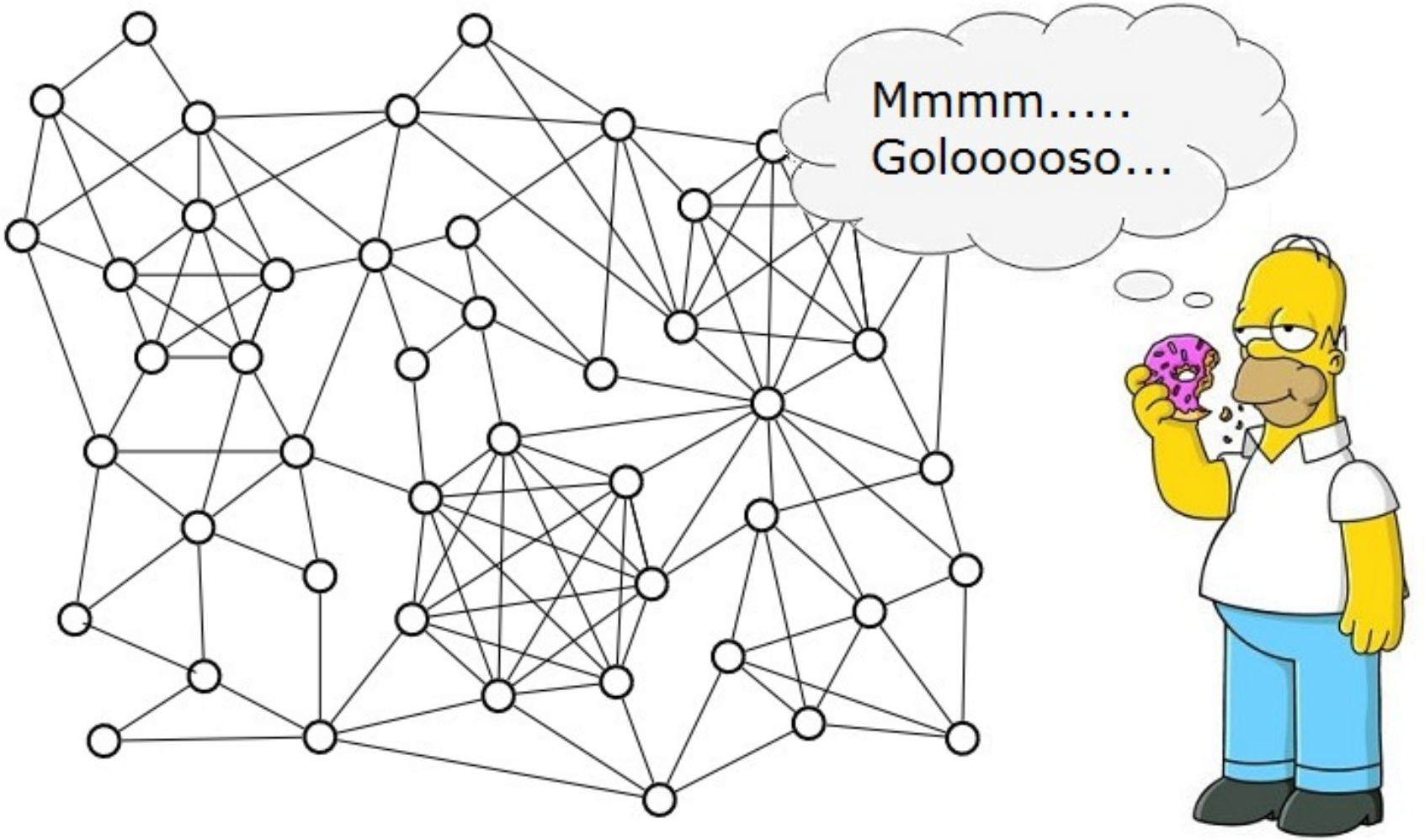
Heurísticas Golosas - Ejemplo: Clique Máxima

- ¿Se les ocurre alguna forma de resolverlo en forma eficiente?
- En realidad, no se conocen algoritmos eficientes para resolverlo... :-(

Heurísticas Golosas - Ejemplo: Clique Máxima

- ¿Se les ocurre alguna forma de resolverlo en forma eficiente?
- En realidad, no se conocen algoritmos eficientes para resolverlo... :-(
- Pensemos un rato, y propongamos heurísticas golosas para este problema...

Heurísticas Golosas - Ejemplo: Clique Máxima



Heurísticas Golosas

Analicemos un poco las propuestas...

- ¿Qué tan buenas son las heurísticas que propusimos?

Heurísticas Golosas

Analicemos un poco las propuestas...

- ¿Qué tan buenas son las heurísticas que propusimos?
- O mejor dicho, ¿qué tan malas pueden ser?

Heurísticas Golosas

Analicemos un poco las propuestas...

- ¿Qué tan buenas son las heurísticas que propusimos?
- O mejor dicho, ¿qué tan malas pueden ser?
- ¿Hay alguna forma de "medirlas"?

Heurísticas Golosas

Analicemos un poco las propuestas...

- ¿Qué tan buenas son las heurísticas que propusimos?
- O mejor dicho, ¿qué tan malas pueden ser?
- ¿Hay alguna forma de "medirlas"? (Sí, pero no nos vamos a meter con eso...)

Menú del día

Búsqueda Local

Sigamos repasando: ¿Qué es una heurística de búsqueda local?

Búsqueda Local

Sigamos repasando: ¿Qué es una heurística de búsqueda local?

Heurística de Búsqueda Local

Para cada solución factible $s \in \mathcal{S}$ se define $N(s)$ como el conjunto de "soluciones vecinas" de s. Un procedimiento de búsqueda local toma una solución inicial s e iterativamente la mejora reemplazándola por otra solución mejor del conjunto $N(s)$, hasta llegar a un óptimo local.

1. Sea $s \in \mathcal{S}$ una solución inicial
2. Mientras exista $s' \in N(s)$ con $f(s') > f(s)$
3. $\quad s \leftarrow s'$

Búsqueda Local

Sigamos repasando: ¿Qué es una heurística de búsqueda local?

Heurística de Búsqueda Local

Para cada solución factible $s \in \mathcal{S}$ se define $N(s)$ como el conjunto de "soluciones vecinas" de s. Un procedimiento de búsqueda local toma una solución inicial s e iterativamente la mejora reemplazándola por otra solución mejor del conjunto $N(s)$, hasta llegar a un óptimo local.

1. Sea $s \in \mathcal{S}$ una solución inicial
2. Mientras exista $s' \in N(s)$ con $f(s') > f(s)$
3. $\quad s \leftarrow s'$

Veamos en nuestros ejemplos: AGM, TSP...

Búsqueda Local

Sigamos repasando: ¿Qué es una heurística de búsqueda local?

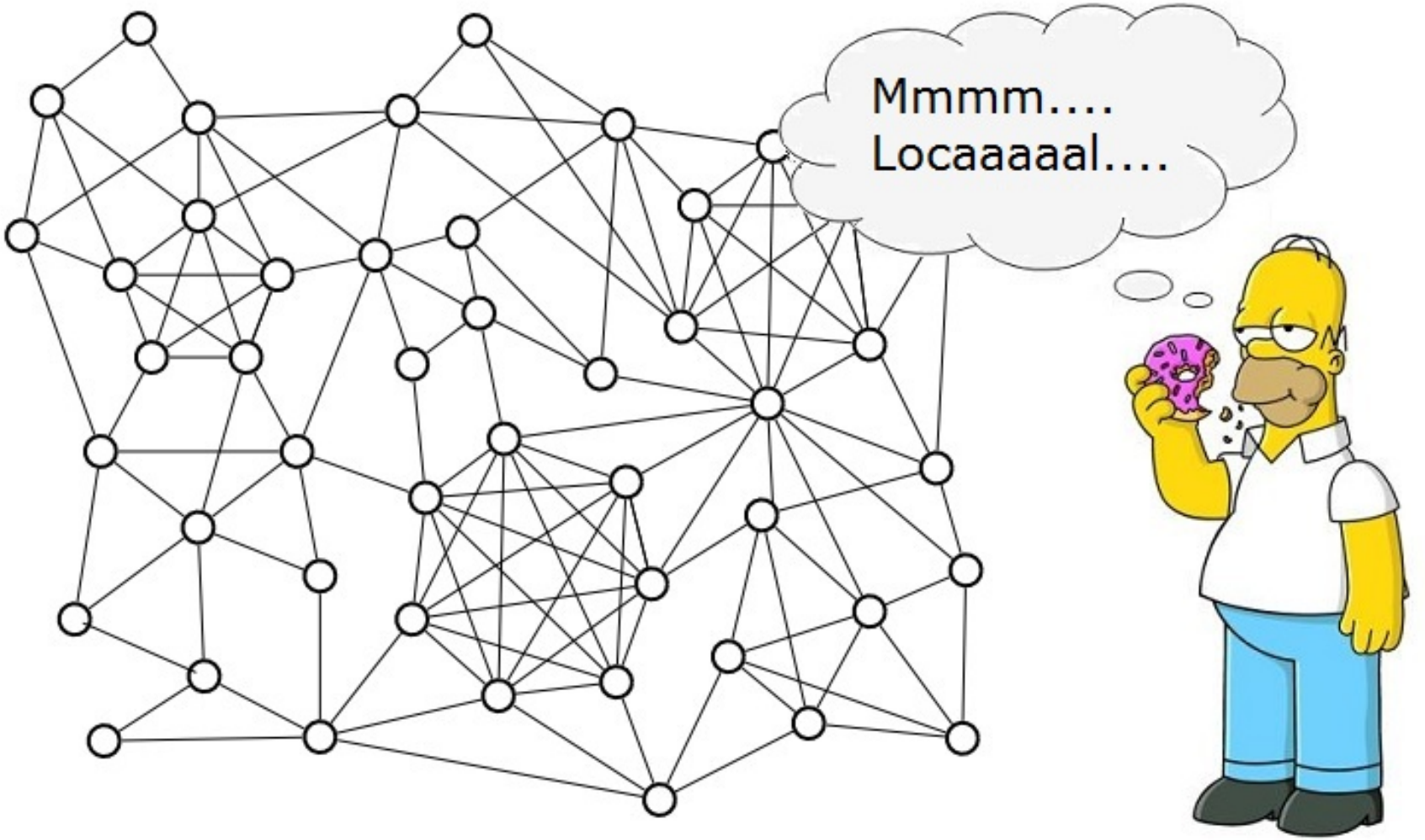
Heurística de Búsqueda Local

Para cada solución factible $s \in \mathcal{S}$ se define $N(s)$ como el conjunto de "soluciones vecinas" de s. Un procedimiento de búsqueda local toma una solución inicial s e iterativamente la mejora reemplazándola por otra solución mejor del conjunto $N(s)$, hasta llegar a un óptimo local.

1. Sea $s \in \mathcal{S}$ una solución inicial
2. Mientras exista $s' \in N(s)$ con $f(s') > f(s)$
3. $\quad s \leftarrow s'$

Veamos en nuestros ejemplos: AGM, TSP...
Pensemos un rato, y propongamos heurísticas de búsqueda local para el problema anterior...

Búsqueda Local - Ejemplo: Clique Máxima



Menú del día

GRASP

¿Qué es GRASP?

GRASP

¿Qué es GRASP?

Greedy Randomized Adaptative Search Procedure

GRASP es una combinación entre una heurística golosa "aleatorizada" y un procedimiento de búsqueda local.

1. Mientras no se alcance el criterio de terminación
2. Obtener $s \in \mathcal{S}$ mediante una heurística golosa aleatorizada.
3. Mejorar s mediante búsqueda local.
4. Recordar la mejor solución obtenida hasta el momento.

GRASP

¿Qué es GRASP?

Greedy Randomized Adaptative Search Procedure

GRASP es una combinación entre una heurística golosa "aleatorizada" y un procedimiento de búsqueda local.

1. Mientras no se alcance el criterio de terminación
2. Obtener $s \in \mathcal{S}$ mediante una heurística golosa aleatorizada.
3. Mejorar s mediante búsqueda local.
4. Recordar la mejor solución obtenida hasta el momento.

Ok, pero...

- ¿Cuál es el criterio de terminación?
- ¿Dónde está la aleatoriedad en la heurística golosa?

GRASP - Criterio de terminación

El criterio de terminación para el algoritmo de GRASP puede ser casi cualquier cosa "que tenga sentido". Por ejemplo:

GRASP - Criterio de terminación

El criterio de terminación para el algoritmo de GRASP puede ser casi cualquier cosa "que tenga sentido". Por ejemplo:

- No se encontró una mejora en las últimas k iteraciones.
- Se alcanzó un límite prefijado de k iteraciones.
- Se obtuvo una solución cercana a una cierta cota conocida.
- Se obtuvo muchas veces la misma solución.
- etc.

GRASP - Aleatoriedad en la heurística golosa

Como sabemos, una heurística golosa realiza en cada paso la mejor elección a nivel local.

GRASP - Aleatoriedad en la heurística golosa

Como sabemos, una heurística golosa realiza en cada paso la mejor elección a nivel local.
Para agregarle una componente aleatoria en GRASP se propone fabricar en cada paso una Lista Restringida de Candidatos (RCL) y elegir aleatoriamente un candidato de esta lista.

GRASP - Aleatoriedad en la heurística golosa

Como sabemos, una heurística golosa realiza en cada paso la mejor elección a nivel local.
Para agregarle una componente aleatoria en GRASP se propone fabricar en cada paso una Lista Restringida de Candidatos (RCL) y elegir aleatoriamente un candidato de esta lista.
Varias opciones:

- **Por valor**: la RCL contiene los candidatos cuyo valor sea no menor que un cierto porcentaje α del valor del mejor candidato.

GRASP - Aleatoriedad en la heurística golosa

Como sabemos, una heurística golosa realiza en cada paso la mejor elección a nivel local.
Para agregarle una componente aleatoria en GRASP se propone fabricar en cada paso una Lista Restringida de Candidatos (RCL) y elegir aleatoriamente un candidato de esta lista.
Varias opciones:

- **Por valor**: la RCL contiene los candidatos cuyo valor sea no menor que un cierto porcentaje α del valor del mejor candidato.
- **Por cantidad**: la RCL contiene los β mejores candidatos.

GRASP - Aleatoriedad en la heurística golosa

Como sabemos, una heurística golosa realiza en cada paso la mejor elección a nivel local.
Para agregarle una componente aleatoria en GRASP se propone fabricar en cada paso una Lista Restringida de Candidatos (RCL) y elegir aleatoriamente un candidato de esta lista.
Varias opciones:

- **Por valor**: la RCL contiene los candidatos cuyo valor sea no menor que un cierto porcentaje α del valor del mejor candidato.
- **Por cantidad**: la RCL contiene los β mejores candidatos.
- ... o una combinación de ambas.

GRASP - Aleatoriedad en la heurística golosa

Como sabemos, una heurística golosa realiza en cada paso la mejor elección a nivel local.
Para agregarle una componente aleatoria en GRASP se propone fabricar en cada paso una Lista Restringida de Candidatos (RCL) y elegir aleatoriamente un candidato de esta lista.
Varias opciones:

- **Por valor**: la RCL contiene los candidatos cuyo valor sea no menor que un cierto porcentaje α del valor del mejor candidato.
- **Por cantidad**: la RCL contiene los β mejores candidatos.
- ... o una combinación de ambas.

¿Cómo se aplicarían estas opciones a las heurísticas que planteamos antes?

Menú del día

Taboo Search

¿Qué es Taboo Search?

Taboo Search

¿Qué es Taboo Search?

- Taboo Search surge como una mejora de búsqueda local. El principal problema de la búsqueda local es que nos estancamos en óptimos locales.

Taboo Search

¿Qué es Taboo Search?

- Taboo Search surge como una mejora de búsqueda local. El principal problema de la búsqueda local es que nos estancamos en óptimos locales.
- En Taboo Search, a diferencia de la búsqueda local, nos movemos siempre al mejor vecino, aún si es peor que la solución donde estamos parados.

Taboo Search

¿Qué es Taboo Search?

- Taboo Search surge como una mejora de búsqueda local. El principal problema de la búsqueda local es que nos estancamos en óptimos locales.
- En Taboo Search, a diferencia de la búsqueda local, nos movemos siempre al mejor vecino, aún si es peor que la solución donde estamos parados.
- Para evitar ir y volver a un óptimo local generando un ciclo de 2 soluciones, guardamos una lista de soluciones que ya visitamos, la lista Taboo, para no volver a esas soluciones.

Taboo Search

¿Qué es Taboo Search?

- Taboo Search surge como una mejora de búsqueda local. El principal problema de la búsqueda local es que nos estancamos en óptimos locales.
- En Taboo Search, a diferencia de la búsqueda local, nos movemos siempre al mejor vecino, aún si es peor que la solución donde estamos parados.
- Para evitar ir y volver a un óptimo local generando un ciclo de 2 soluciones, guardamos una lista de soluciones que ya visitamos, la lista Taboo, para no volver a esas soluciones.
- Como la lista Taboo puede crecer demasiado, sólo guardamos las últimas T soluciones visitadas.

Diferencias entre búsqueda local y Taboo Search

Si ya tenemos nuestra solución con búsqueda local, es muy fácil adaptarla a una versión de Taboo Search.

Diferencias entre búsqueda local y Taboo Search

Si ya tenemos nuestra solución con búsqueda local, es muy fácil adaptarla a una versión de Taboo Search.
Lo único que hay que hacer es asegurarnos de movernos al mejor vecino también cuando estamos en un óptimo local y guardar la lista Taboo para no considerar soluciones en dicha lista.

Menú del día

Simulated Annealing

¿Qué es Simulated Annealing?

Simulated Annealing

¿Qué es Simulated Annealing?

Simulated Annealing

Simulated Annealing es una metaheurística que consiste en ir moviendonos de una solución a otra probabilísticamente según que tan buena es la nueva solución en comparación con la anterior

1. Empezar con $s \in \mathcal{S}$ elegida de alguna manera.
2. Mientras no se alcance el criterio de terminación
3. Obtener $s' \in \mathcal{S}$ vecino de s
4. Elegir un número $r \in [0, 1]$ al azar y calcular la función $P(s, s')$
5. Si $P(s, s') > r$ entonces $s \leftarrow s'$.
6. Recordar la mejor solución obtenida hasta el momento.

Simulated Annealing

¿Qué es Simulated Annealing?

Simulated Annealing

Simulated Annealing es una metaheurística que consiste en ir moviendonos de una solución a otra probabilísticamente según que tan buena es la nueva solución en comparación con la anterior

1. Empezar con $s \in \mathcal{S}$ elegida de alguna manera.
2. Mientras no se alcance el criterio de terminación
3. Obtener $s' \in \mathcal{S}$ vecino de s
4. Elegir un número $r \in [0, 1]$ al azar y calcular la función $P(s, s')$
5. Si $P(s, s') > r$ entonces $s \leftarrow s'$.
6. Recordar la mejor solución obtenida hasta el momento.

La función $P : \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$ es una función que cuanto mejor es la solución s' con respecto a s da un número más cercano a 1.