

Teoría de la Complejidad y Heurísticas: El Problema de Asignación Cuadrática (QAP)

JCC 2016, Rosario

por

López Restrepo, Jheison
jalopez@ungs.edu.ar
Investigador/Docente UNGS
Maestría en Ingeniería - UdeA

&

Rojas Paredes, Andrés
arojas@ungs.edu.ar
Investigador/Docente de UNGS
Doctorando en FCEyN-UBA



Universidad
Nacional de
General
Sarmiento



Resumen

¿Qué sucede cuando afrontamos un problema computacional? Nada..., básicamente se formula un modelo matemático para este, se estudia su complejidad y se plantean posibles soluciones que deben ser eficientes, correctas y robustas (entre otros atributos).

Consideremos el clásico problema de optimización combinatoria llamado QAP (Problema de Asignación Cuadrática) que se refiere a la ubicación de instalaciones sujeta a costos.

QAP es NP-Hard; es decir, no admite soluciones exactas eficientes. En este contexto, vamos a introducir una nueva técnica para demostrar que un problema es *hard*, en particular, esta técnica es lo último en teoría de la complejidad ya que se puede aplicar a la programación orientada a objetos.

Por otro lado, los problemas complejos se pueden resolver con técnicas heurísticas; se sacrifica exactitud y se adoptan soluciones aproximadas. Explicaremos nuevas técnicas que son adecuadas para evitar la tremenda complejidad de los problemas NP-Hard.

Resumen

Qué sucede cuando afrontamos un problema computacional?



1. Hacer un modelo del problema

SOLUCIÓN
EXACTA

vs.

SOLUCIÓN
Heurística



3. Diseñar la mejor solución

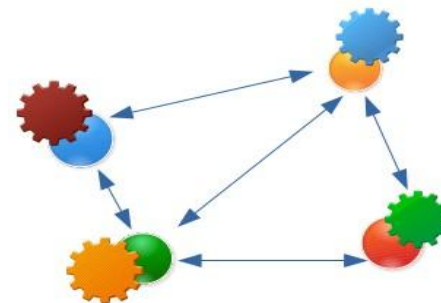


2. Estudiar su complejidad computacional

El Problema de Asignación Cuadrática

Formulación clásica

Asignar n instalaciones a n locaciones de manera que el costo sea mínimo (Koopmans-Beckmann, 1957).



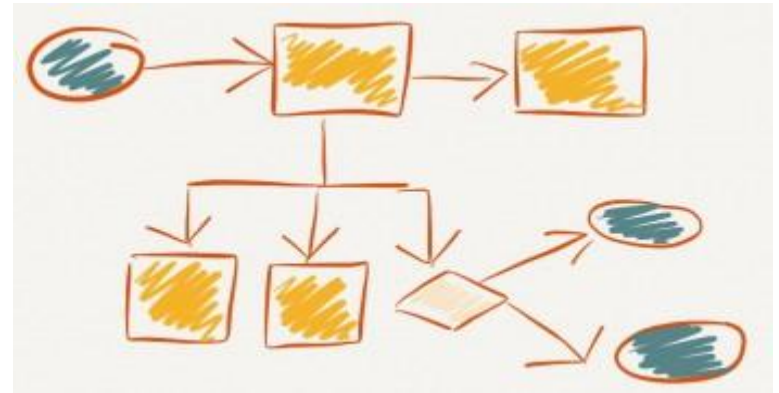
$$\min_{\phi \in \mathcal{S}_n} \sum_{i=1}^n \sum_{j=1}^n f_{ij} d_{\phi(i)\phi(j)} + \sum_{i=1}^n b_{i\phi(i)}$$

Aplicaciones (Interés práctico)

- Distribución de instalaciones en hospitales
- Distribución de componentes circuitales
- Disminución de la fatiga ocular
- Muchas aplicaciones más

El Problema de Asignación Cuadrática

Para ejecutar un algoritmo se deben llevar a cabo n pasos y cada paso tarda un tiempo x en ejecutarse



- Para QAP hay $n!$ formas diferentes de realizar la asignación (con $n=4 \rightarrow 24$)
- ¿Qué ocurre cuando n es muy grande?
(por ejemplo, con $n=10 \rightarrow 3.628.800$)
- Para cada asignación deben realizarse n multiplicaciones y $(n-1)$ sumas. Total: $[n! * n * (n - 1)]$

QAP es un problema NP-Hard (Interés teórico)

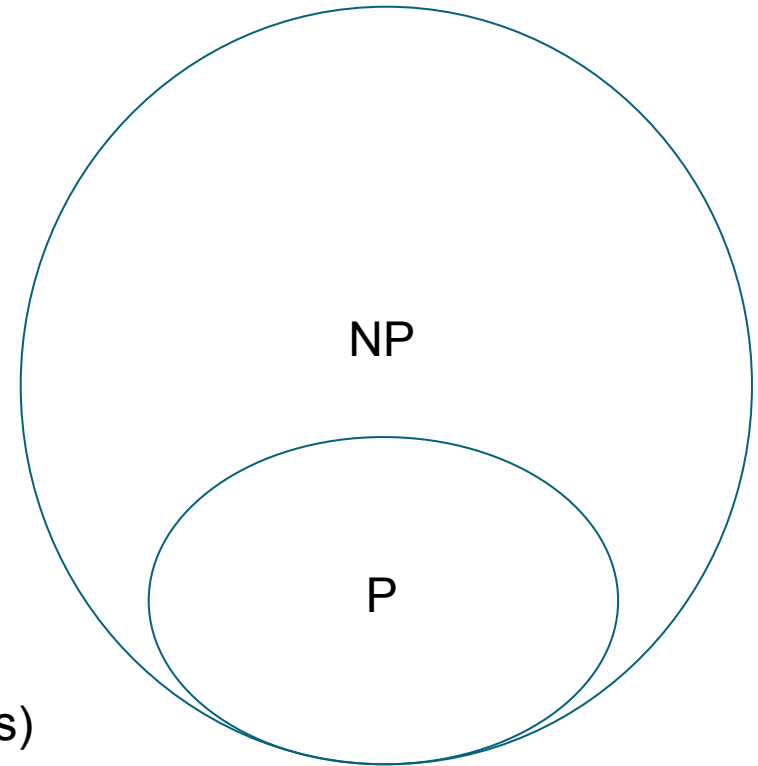
QAP es NP-Hard

Que significa que QAP es NP-Hard?

Clases de complejidad:

P = clase de los problemas que se pueden resolver en tiempo Polinomial

NP = clase de los problemas que se pueden verificar en tiempo Polinomial
(y no se conocen aún algoritmos polinomiales)

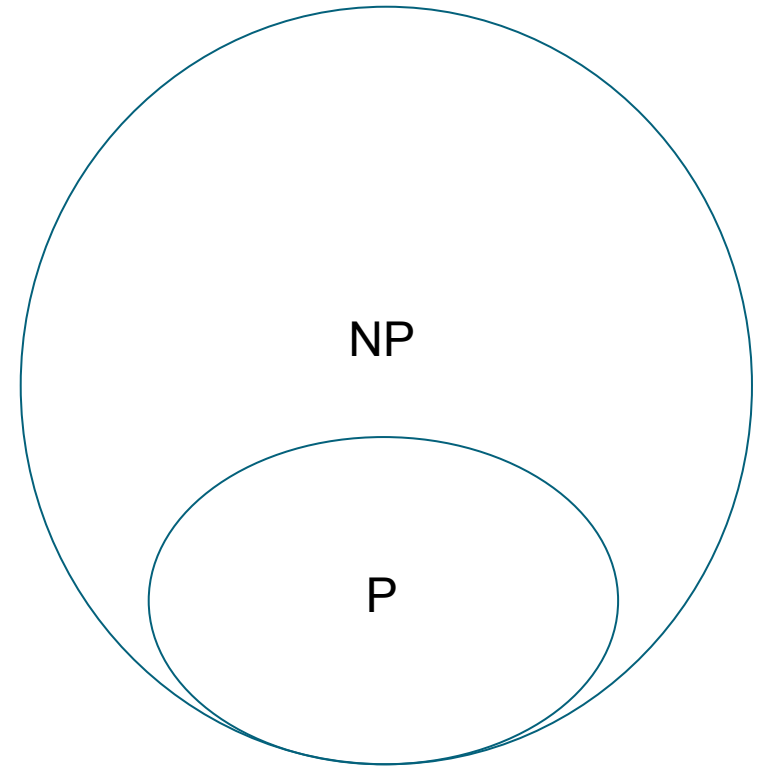


QAP es NP-Hard

$P = NP$ vs. $P \neq NP$ vs. $P ? NP$

P vs NP problema abierto desde
Cook-Levin 1971

Es el más codiciado
de los problemas
(matemáticos) del
milenio



Se dice que el nivel del pensamiento actual en Matemáticas, no permite demostrar la relación exacta entre estas dos clases

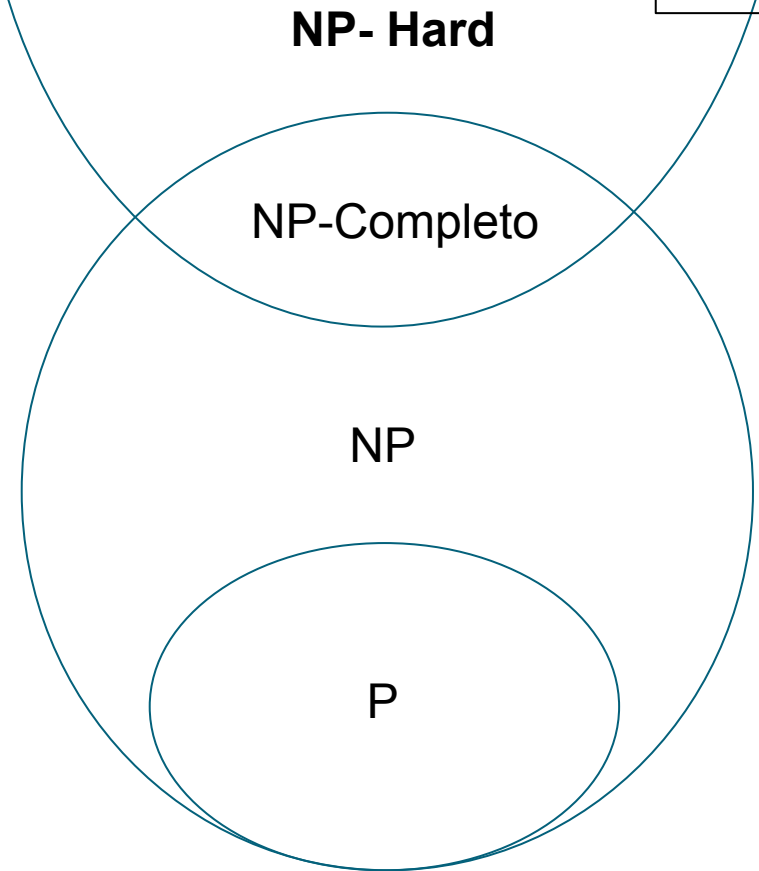
QAP es NP-Hard

Dentro de los problemas NP

Hay un subconjunto especialmente interesante

NP-Completo = clase de los problemas para los que cualquier otro problema NP se puede reducir polinomialmente a ellos.

NP - Hard = clase de problemas al menos tan difíciles como la clase NP-Completo



QAP es NP-Hard

NP- Hard

NP-Completo

NP

P

Copiar ese paso
es NP-Hard...

Copiar ese paso es muy difícil...



QAP es NP-Hard

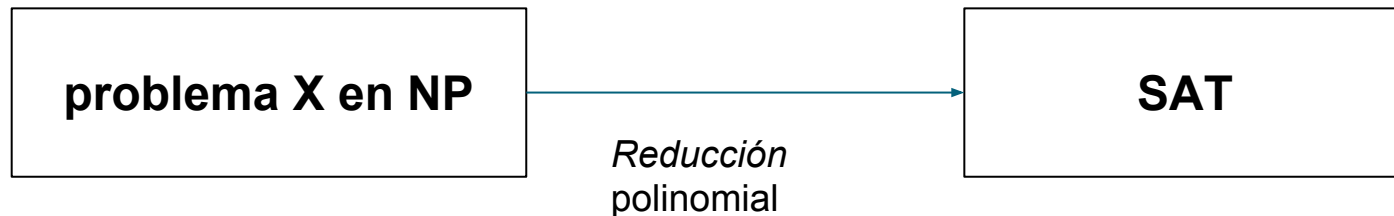


y... cómo hicieron para demostrar que QAP es NP-Hard?

Técnica básica: Reducción de Problemas

Una reducción es un algoritmo para transformar un problema en otro problema

SAT primer problema NP-Completo (Teorema de Cook-Levin 1971)
todo problema NP se reduce a SAT



Si existe un algoritmo polinomial llamado **Algoritmo** para SAT, entonces

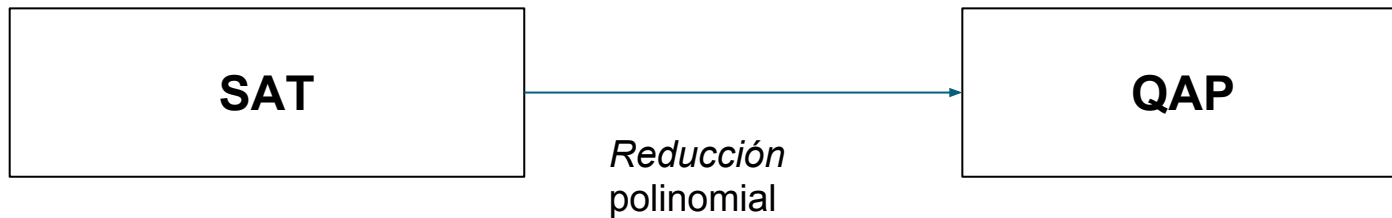
Algoritmo(Reducción(X))

es polinomial también para cualquier problema X en NP.

P vs NP Problem = existe este algoritmo?

Técnica básica: Reducción de Problemas

Para demostrar que QAP es NP-Hard podríamos hacer una reducción!



Entonces todo problema NP se puede reducir a QAP
(Definición de NP-Hard)

En otras palabras
QAP es al menos tan difícil como un problema NP-Completo

Técnica avanzada: Cotas inferiores

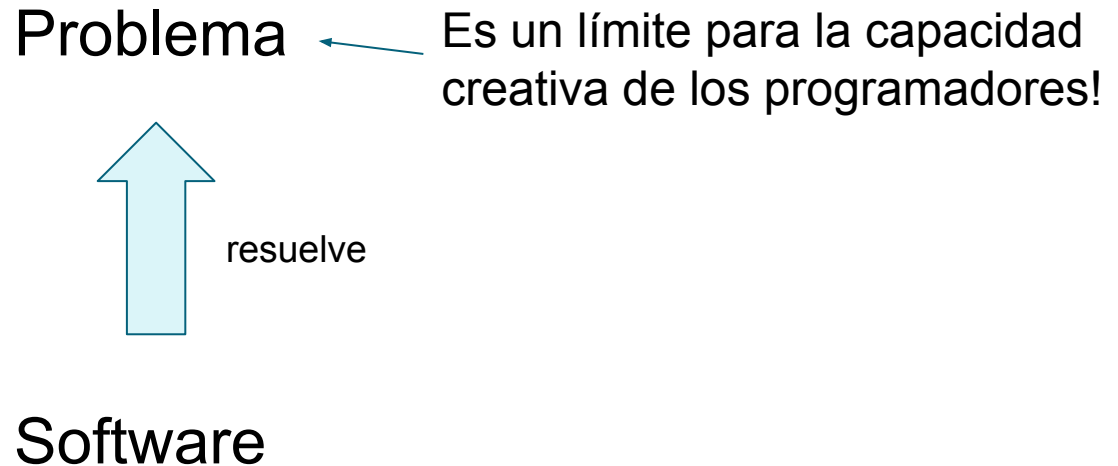


Así como el mar o el cielo, una cota inferior establece un límite

Cotas inferiores de complejidad

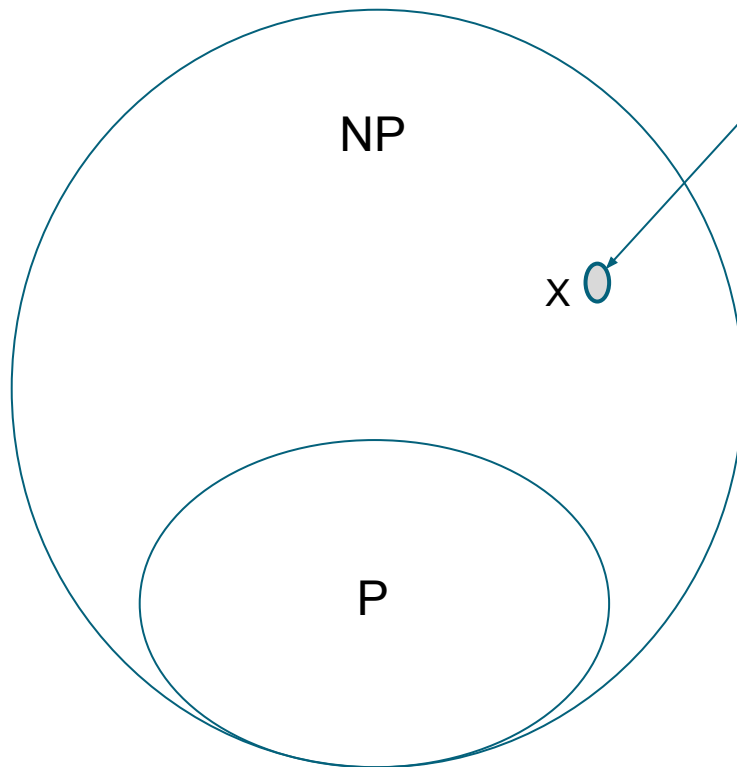
Que es una cota inferior?

Dado un problema, la cota inferior es un teorema que dice cuanto tarda *por lo menos* cualquier algoritmo que resuelve dicho problema



Cotas inferiores de complejidad

Importancia de las cotas inferiores



Suponer un problema X en NP

Suponer que existe una cota inferior exponencial $\Omega(2^n)$ con $n =$ tamaño del input para el problema X

Luego X no está en P

Entonces $P \neq NP$

Esta cota se buscó mucho en los 80's (desde Karp-Lipton Theorem)

Complexity Winter (1995 ... actualidad)

Así como los programadores [Lisp](#), los especialistas en complejidad se hicieron difíciles de encontrar

Por suerte encontré a mi director



Cotas Inferiores de Complejidad

antes

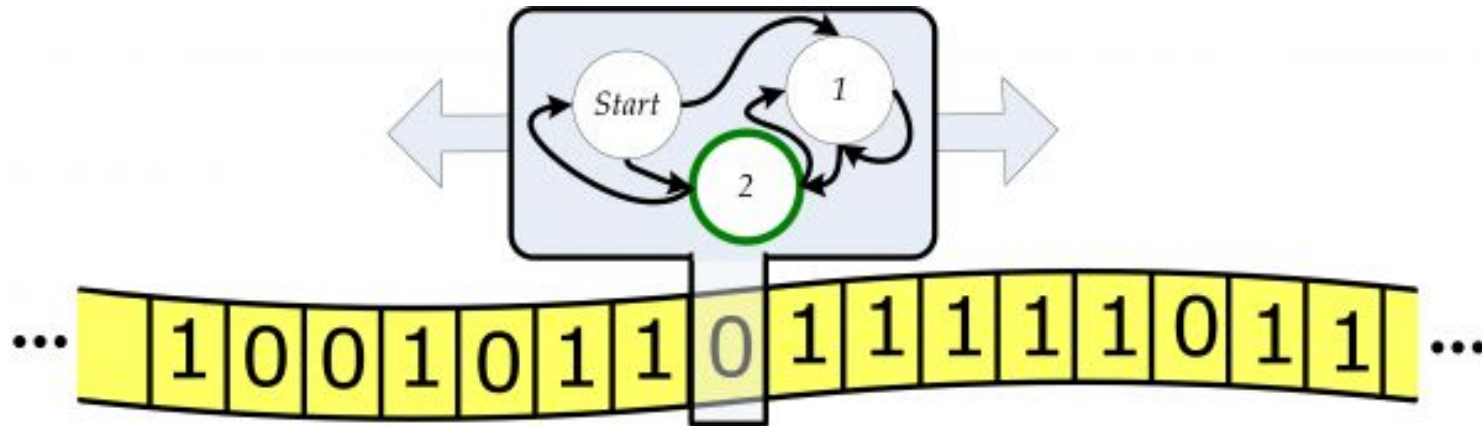
Usaban un modelo de cómputo dado por la máquina de Turing

Captura todos los algoritmos

Incluso los que tienen Bugs



Alan Turing



Cotas Inferiores de Complejidad

hoy

Usamos modelos de cómputo (mucho más) específicos

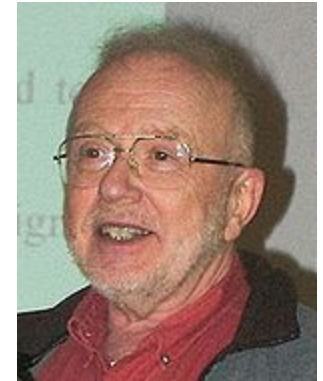
Captura algoritmos que implementan Information Hiding

No Bugs, correctos y robustos



CAR Hoare

En otras palabras..
modelo de cómputo basado en
Ingeniería de Software



D. Parnas

Cotas Inferiores de Complejidad

hoy

Cómo sería un modelo de cómputo que captura la noción de Information Hiding?



Cotas Inferiores de Complejidad

hoy

Quiz game (ó juego de preguntas y respuestas)



Class Polinomio
rep : _____
observadores() constructores()
eliminarVariables()

El programador (Moe) tiene todo el poder de cómputo que necesite pero está restringido a usar la interfaz que provee la clase (Fry)

Fry elige una instancia particular del problema (piensa en 3) y esconde la representación.

Moe tiene que encontrar el número sin tener acceso a la representación.

Volviendo al caso específico de QAP...



Si no es conveniente usar algoritmos exactos para QAP (porque es NP-Hard), que opción me queda?

¿Cómo hacerle frente a la complejidad?

Implementación (Tecnología)

- *Mas pequeño y mas rápido (reducir x)*



- Limite de frecuencia
- Calentamiento

- *Paralelismo (Simultaneidad)*



Reducción ideal por un factor α siendo α la cantidad de procesadores. No resulta práctico.

- Para $n=20$ solo fue posible una solución exacta a partir de los años 90.
- $n=25$ necesito de 64 procesadores trabajando continuamente durante 30 días.
- $n=30$ tomó 7 días con 650 procesadores en paralelo (aproximadamente equivalente a 7 años de procesamiento secuencial)

¿Cómo hacerle frente a la complejidad?

Algoritmia (Menor exactitud)

- *Algoritmos aproximativos*



– Aceptar un error cuantificable

- *Heurísticas*



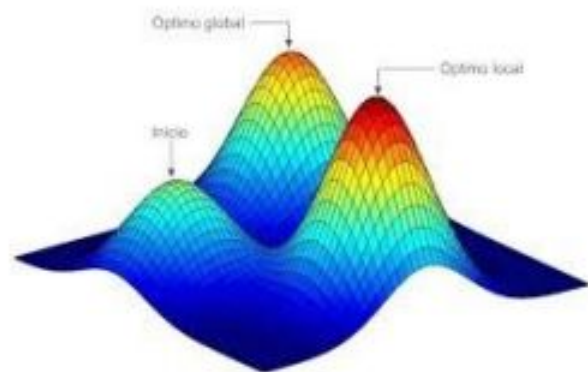
– Conformarse con la mejor solución encontrada en un tiempo dado.

La mayoría de las heurísticas encuentra, en un tiempo asumible, una solución que varía un 1% respecto a la mejor solución conocida de las instancias QAP.

Meta- Heurísticas

Heurísticas. Reglas que definen el “comportamiento” de un algoritmo al resolver un problema específico y que son derivadas del conocimiento de dicho problema y la experiencia previa.

Meta-heurísticas. Heurísticas generales que se pueden aplicar a grupos de problemas.



- Espacio de búsqueda
- Exploración vs Explotación

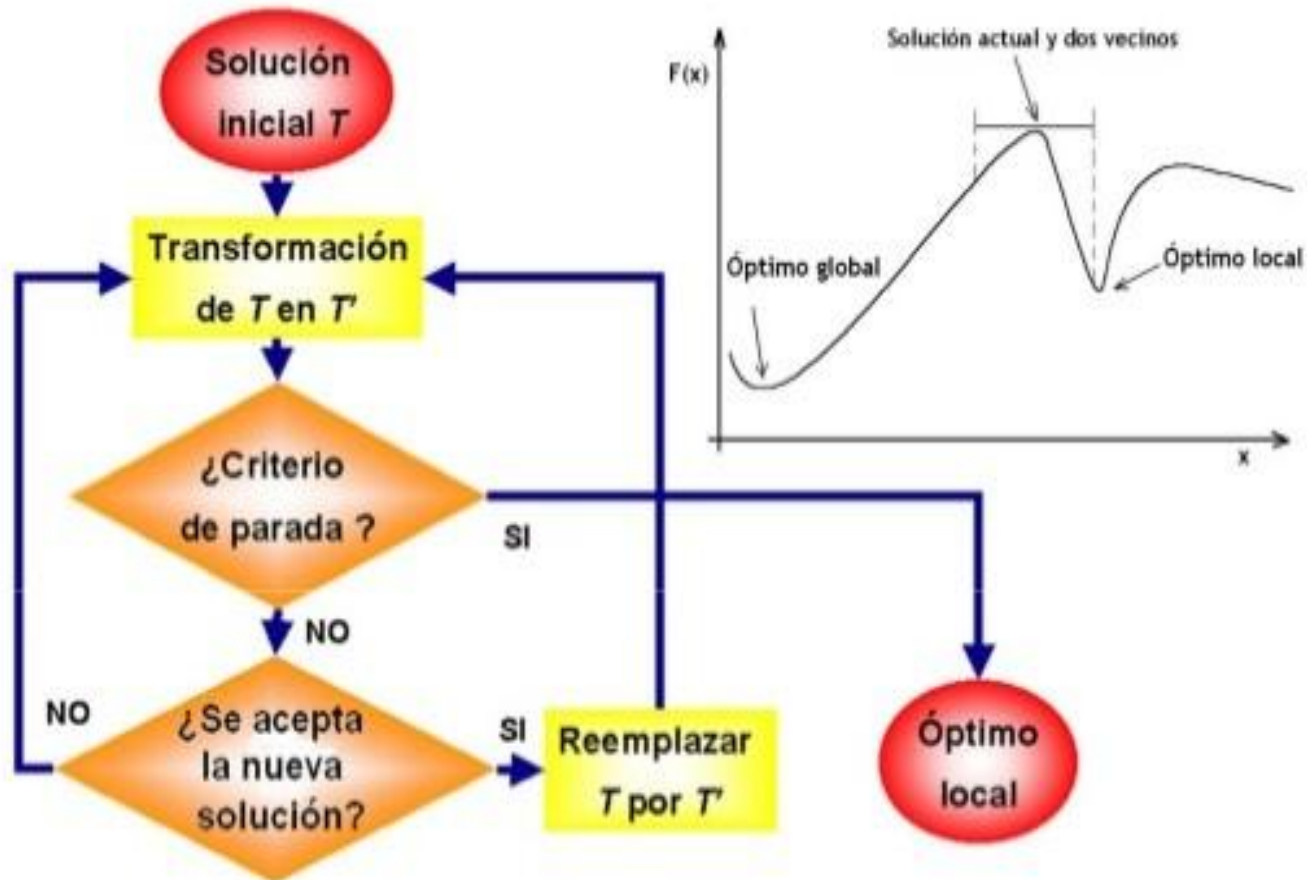
S-Metaheurísticas

- Recocido simulado
- Búsqueda local

P-Metaheurísticas

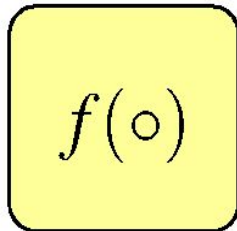
- Algoritmos genéticos
- Optimización de enjambres

Búsqueda local



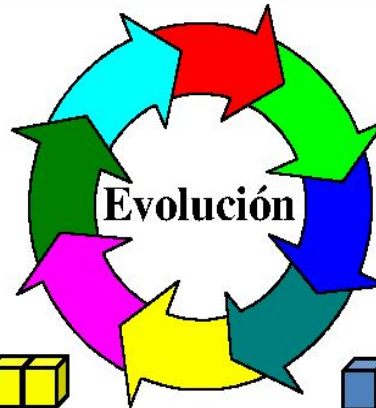
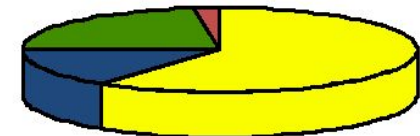
Algoritmos Genéticos

Evaluación de la descendencia

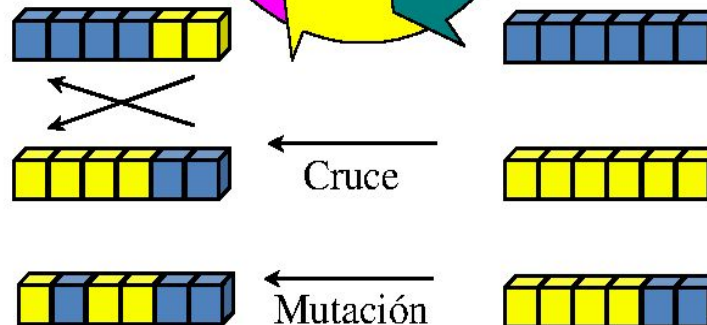


Individuo	genotipo	fitness
x ₁	100100	1296
x ₂	010010	324
x ₃	010110	484
x ₄	000001	1

Selección de Progenitores



Nueva población de descendientes



Reproducción

Hibridación

- NFL (No-Free Lunch Theorem) La efectividad de un algoritmo depende directamente de la cantidad y la calidad de la información del problema que tenga incorporada.
- Algoritmos meméticos (Evolución cultural)
- Trade-off: Intensificación y Diversificación.
 - Cooperativa vs Coercitiva
 - Síncrona vs Asíncrona
 - Cerradas vs Abiertas

Líneas de investigación en metaheurísticas

Paralelismo

- Descomposición en tareas o procesos para distribuirlos en los procesadores disponibles.
 - Nivel funcional
 - Nivel de datos
 - Nivel de la estructura del problema
- HPC (Computación de alto rendimiento)
 - Flynn (SISD, SIMD, MIMD)
 - Homogénea o heterogénea
 - Memoria compartida o distribuida
- Metaheurísticas robustas y fáciles de calibrar.

Hibridación de metaheurísticas y su implementación en un entorno de procesamiento paralelo aplicada a la resolución del Problema de Asignación Cuadrática QAP

- QAPLib como casos de prueba
 - 136 instancias prácticas y teóricas
 - De distintos tipos (!= Espacio de búsqueda)
- X10 como lenguaje de programación
 - Places, Activities, Clocks
- Paralelismo Cooperativo
 - Búsqueda “inteligente”
 - Cluster de 16 nodos de 4 × 16-core AMD Opteron 6376 (max 4 nodos y 32 núcleos/nodo)
 - InfiniBand FDR 4× (56 Gbps)
- Hibridación: búsqueda local y algoritmos genéticos

Resumen: ¿qué hemos aprendido?



- Estudiamos el Problema de Asignación Cuadrática (Caso específico de los problemas de optimización combinatoria)
- Una breve introducción al estudio de las metaheurísticas y sus tendencias de investigación.
- Y ya sabemos en qué andan algunos de los últimos sobrevivientes del Complexity Winter



Bibliografía

QAP

- C Cotta, E-G Talbi, and E Alba. Parallel Hybrid Metaheuristics. 2006.
- El-ghazali Talbi. SCI 434 - Hybrid Metaheuristics
- Umut Tosun. On the performance of parallel hybrid algorithms for the solution of the quadratic assignment problem. *Engineering Applications of Artificial Intelligence*, 39:267– 278, 2015.
- Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.
- Sartaj Sahni and Teofilo Gonzalez. P-Complete Approximation Problems. *Journal of the ACM*, 23(3):555–565, 1976.
- Ravi Kumar Bhati and Akhtar Rasool. Quadratic Assignment Problem and its Relevance to the Real World: A Survey. *International Journal of Computer Applications*, 96(9):42–47, 2014.

Complejidad

- B. Bank, J. Heintz, G. Matera, J. L. Montaña, L. M. Pardo, and A. Rojas Paredes, “Quiz games as a model for Information Hiding,” *Journal of Complexity*, vol. 34, pp. 1 – 29, 2016.
- J. Heintz, B. Kuijpers, and A. Rojas Paredes, “Software engineering and complexity in effective algebraic geometry,” *Journal of Complexity*, vol. 29, no. 1, pp. 92–138, 2013