

Técnicas avanzadas de optimización

Dr. Arno Formella

16 de febrero de 2006

Universidad de Vigo
Departamento de Informática
Área de Lenguajes y Sistemas Informáticos
E-32004 Ourense

<http://www.ei.uvigo.es/~formella>
formella@ei.uvigo.es

Índice

1. Curso	3
2. Tareas para una presentación	3
3. Motivación	3
4. Nociones básicas	3
5. No free lunch theorem	4
6. Ejemplo	5
7. Representación	5
8. Intento de clasificación	5
9. Optimización clásica: ejemplos	5
10. Optimización con métodos evolutivos	6
11. Paradigmas de los métodos evolutivos	7
12. Algoritmos genéticos	7
13. Programación evolutiva	7
14. Estrategias evolutivas	8
15. Programación genética	8
16. Evolución diferencial	8
17. Swarm intelligence	8
18. Colonias de hormigas	9

19. Particle swarm optimization	9
20. Particle swarm optimization	9
21. Particle swarm optimization	10
22. Particle swarm optimization	10
23. Evitar convergencia prematura	10
24. Optimización de funciones multi-objetivo	11
25. Técnicas de solución	11
26. Optimización de funciones multi-objetivo y métodos evolutivos	11
27. Híbridos	11

1. Curso

La página inicial del curso es:

<http://www.ei.uvigo.es/~formella/doc/tc05>

Estos apuntes se acompañan con ilustraciones en pizarra dónde se explica las notaciones y el funcionamiento de los algoritmos.

El texto es meramente una brevísima introducción (5 horas) a diferentes aspectos de la optimización donde se pincela ciertos aspectos más bien para motivar y despertar interés por este campo importante sobre todo desde el punto de vista de la informática aplicada.

2. Tareas para una presentación

El curso de doctorado ya tiene en su título *Desarrollo de Software*, por eso se ha pensado como tareas:

3. Motivación

Optimizar significa buscar alguna solución que se distingue de las demás posibles soluciones por ser (lo suficiente) extremo dentro de una ordenación (en un tiempo razonable).

Escalar el monte más alto.

Problemas que se quiere resolver:

- minimizar gastos
- maximizar ganancias
- maximizar ocupación
- minimizar energía
- minimizar recursos

4. Nociones básicas

espacio de búsqueda (puede ser discreta o continuo)

función objetivo, puede ser

- discreta o continua

- total o parcial
- simple o complejo, sobre todo respecto a su evaluación
- explícita, implícita, experimental
- derivable o no-derivable
- estática o dinámica

tiene que ser acotada.

- Minimización
- Maximización
- Obviamente cualquier problema de maximización se puede convertir en un problem de minimización.
- restricciones
- solución factible
- codificación de soluciones

Se distingue entre

optimización local: donde se parte de una solución inicial y se conforme con buscar algún mínimo local (cercano)

optimización global: donde se intenta encontrar la mejor solución entre todas las posibles soluciones

Problema principal de la optimización global: quedarse atrapado en un mínimo local (convergencia prematura).

Se puede intentar mejorar soluciones ya encontradas o construir nuevas soluciones.

Optimización con varios objetivos (Ejemplo).

5. No free lunch theorem

Básicamente dice:

El rendimiento de todos los algoritmos de optimización amortizado sobre todas las funciones objetivos siempre es igual (en espacios discretos).

Con la consecuencia: ningún algoritmo puede ser mejor (en general) que enumeración exhaustiva (o búsqueda aleatoria).

Menos mal que no estamos interesados en optimizar cualquier función, sino solamente aquellos de interés, o en otras palabras, un algoritmo de optimización solamente es útil en su ámbito (porque necesariamente existen ámbitos donde su rendimiento es muy mal).

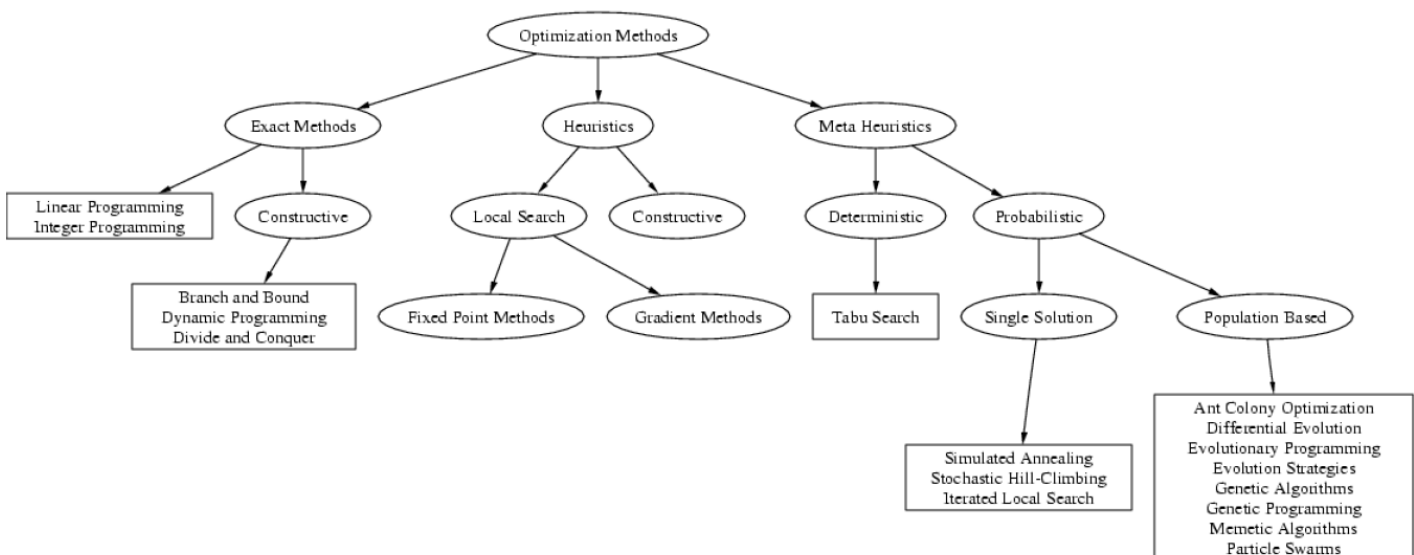
6. Ejemplo

Buscar el monte más alto en una sierra cuando haya niebla.

7. Representación

- conjunto de variables
- conjunto de restricciones
- función objetivo
- tratamiento de soluciones (intermedias) no factibles

8. Intento de clasificación



9. Optimización clásica: ejemplos

función objetivo discreta lineal, espacio continuo:

- programación lineal

función objetivo continua, espacio continuo:

- métodos basados en el gradiente
p.ej. Newton
- trajectory methods (enumeración de todos los puntos ‘extremos’)
- relajación (multi-resolución)
- métodos sin gradiente, p.ej.
Nelder–Mead (1965, and recent improvements 1994, 2002)
Rodríguez–Palomares (2002, and recent improvements 2005)
Dennis–Torczon (multidirectional search algorithm 1991)

función objetivo discreta lineal, espacio continuo:

- branch and bound
- divide and conquer
- bayesian search (cluster methods)
- monte carlos methods (stochastic search)

10. Optimización con métodos evolutivos

basadas en fenómenos naturales

- enfriamiento de materiales
- cristalización de materiales
- evolución (mutación, recombinación, selección)
- sistemas competitivos/colaborativos
- interacciones sociales

- búsqueda tabú (desde 1986)
- random search (desde 196X)
- simulated annealing (desde 196X)
- etc.

y

- genetic algorithms (desde 1975)
- programación genética
- (redes neuronales)

- ant colony optimization (desde 1992)
- particle swarm optimization (desde 1995)
- guided local search (desde 1997)
- iterated local search (desde 1999)
- variable neighborhood search (desde 1999)
- etc.

11. Paradigmas de los métodos evolutivos

- se trabaja con poblaciones de individuos
(tener solamente un individuo y una memoria...)
- hay procesos de modificación
(mutación, modificación, reproducción)
suele ser tema de grandes debates
- rendimiento de los individuos en el entorno basado en el *fitness* que suele ser la función objetivo
(pero no necesariamente exclusivo)
- la toma de decisiones tiene su aspecto probabilístico

12. Algoritmos genéticos

- se distingue genotipo (codificación) y fenotipo
- existe método de biyección entre genotipo y fenotipo
- modificación (mutación y cruce) se realiza sobre los genotipos
- el fitness se evalúa sobre los fenotipos
- mutación (¿tipos?), recombinación (¿tipos?), selección (¿tipos?)

13. Programación evolutiva

- existe solamente el fenotipo (con su codificación)
- modificación (mutación) se realiza sobre los fenotipos de copias
- el fitness se evalúa sobre los fenotipos
- mutación (¿tipos?), selección (¿tipos?)

14. Estrategias evolutivas

- una ampliación de la programación evolutiva
- se mantiene con cada individuo también parámetros que guían las mutaciones
- se modifica dichos parámetros al mismo tiempo que los propios fenotipos
- mutación (¿tipos?), selección (¿tipos?)

15. Programación genética

- la codificación del fenotipo es un programa
- se modifica los programas con operaciones adecuadas
- mutación (¿tipos?), selección (¿tipos?)

Ejemplo reciente.

16. Evolución diferencial

- la codificación del fenotipo es un vector de características
- se modifica el vector de un individuo con diferencias hacia otros vectores
- modificaciones (¿tipos?), selección (¿tipos?)

17. Swarm intelligence

- los individuos de la población interactúan de forma social
- las decisiones de cada individuo dependen del propio querer y la información disponible de (algunos de) los demás
- colonia de hormigas
- particle swarms

18. Colonias de hormigas

- los individuos dejan rastros (feromonas) en el espacio de búsqueda
- las decisiones se basan en una información individual y de las feromonas encontradas
- la información (feromonas) es volátil
- las feromonas o el compartamiento estadístico de los individuos define la solución

19. Particle swarm optimization

Se basa en la idea de simular el comportamiento social de una población de individuos donde cada individuo intenta de refinar/mejorar sus conocimientos con interacciones dentro de su entorno social (inventado alrededor del año 1995).

Las características principales son:

- fácil de describir
- fácil de implementar
- pocos parámetros a ajustar
- normalmente trabaja con poblaciones pequeñas
- el número de evaluaciones de la función objetivo suele ser pequeña
- suele ser rápida

convergencia prematura suele ocurrir si todos los individuos se concentran en una región pequeña del espacio de búsqueda

20. Particle swarm optimization

- cada individuo se comunica con una vecindad (las vecindades se solapan)
- y mantiene información local (mejor solución vista hasta ahora, dirección actual de búsqueda, etc.)
- la vecindad normalmente se mantiene fija
- se modifica la información local usando la información de los vecinos (o el mejor de ellos)
- se confina posibles cambios para evitar *explosiones*
- se puede resolver también problemas discretas

21. Particle swarm optimization

$$\begin{aligned}v_i &= \xi(v_i + U[0, \varphi_1](p_i - x_i) + U[0, \varphi_2](p_g - x_i)) \\x_i &= x_i + v_i\end{aligned}$$

con

- x_i vector de posición actual
- v_i vector de dirección actual
- p_i mejor vector de posición local
- p_g mejor vector de posición del grupo
- $\varphi_1 = 2,05$
- $\varphi_2 = 2,05$
- $\xi = 0,729$

22. Particle swarm optimization

versión binaria: Se *interpreta* las variables con un umbral para variables binarias segun una distribución

versión discreta: Se *interpreta* las variables (por ejemplo con redondeo) como variables discretas

versión dinámica: Si cambia el espacio de búsqueda se reinicializa las variables locales y se sigue el paso del algoritmo, la reinicialización puede ser: $p_i = x_i$ o reevaluar p_i y decidir entre p_i y x_i .

23. Evitar convergencia prematura

- los individuos en la población deben mantener cierta diversidad
- se necesita una función de similitud
- se adaptan dinámicamente los parámetros del algoritmo para aumentar la diversidad
- se usa justamente la diversidad como opción de parada
- se fuerza diversidad en la población

24. Optimización de funciones multi-objetivo

- se quiere optimizar varios objetivos a la vez
- Pareto optimal (global): todos los demás componentes de otra solución son peores o igual se llama también: puntos eficientes, no-dominante, o no-inferior
- Pareto optimal (local): todos los demás componentes de otra solución son peores o igual en una vecindad
- el conjunto de los puntos Pareto describe el *tradeoff* entre los objetivos (diagramas Pareto)

25. Técnicas de solución

- combinación convexa de los objetivos
(para obtener el conjunto pareto, hay que variar los pesos)
- técnicas de homotopía, es decir, calcular todo el conjunto pareto (funciona solamente para dos objetivos)
- programación con meta, es decir, optimizar un objetivo manteniendo los demás debajo de un umbral predeterminado
- normal-boundary intersection
- priorización (multi-level programming), es decir, se optimiza según una ordenación previa de los objetivos

26. Optimización de funciones multi-objetivo y métodos evolutivos

- métodos evolutivos pueden aproximar los límites Pareto en paralelo (por la diversidad en la población)
- se pueden usar particle swarms variando los pesos de la combinación convexa periódicamente con las iteraciones

27. Híbridos

Todo vale, teniendo en cuenta el teorema de no free lunch, y el ámbito de la optimización en mente.

Hay que comparar los resultados obtenidos no solamente con ejemplos, sino con un estudio estadístico riguroso.

Bibliografía

- Rui Mendes. *Population topologies and their influence in particle swarm performance*. PhD Thesis, Universidad de Minho, 2004. <http://www.di.uminho.pt/~rcm/>
- <http://www-fp.mcs.anl.gov/Otc/Guide/OptWeb/index.html> Online optimization project
- <http://www.coin-or.org/index.html> Operation research
- <http://www.cs.sandia.gov/opt/survey/ea.html> Evolutionary algorithms
- <http://iridia.ulb.ac.be/~mdorigo/ACO/> Ant colony optimization
- <http://www.mat.univie.ac.at/~neum/glopt.html> Global optimization
- <http://plato.asu.edu/gom.html> Continuous global optimization software: A brief review
- <http://www.swarmintelligence.org/index.php> Particle swarm optimization