

Técnicas avanzadas de optimización

Dr. Arno Formella

Departamento de Informática
Universidad de Vigo

06/02/09

Técnicas avanzadas de optimización I

- 1 Curso
- 2 Bibliografía y tareas para una presentación
- 3 Motivación
- 4 Nociones básicas
- 5 No free lunch theorem
- 6 Representación

Técnicas avanzadas de optimización II

- 7 Intento de clasificación
- 8 Paradigmas de los métodos evolutivos
- 9 Particle swarm optimization
- 10 Optimización de funciones multi-objetivo
- 11 Resumen

- La página inicial del curso es:
`http://www.ei.uvigo.es/~formella/doc/tc08`
- Estos apuntes se acompañan con ilustraciones en pizarra dónde se explican 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 pincelan ciertos aspectos más bien para motivar y despertar interés por este campo importante sobre todo desde el punto de vista de la informática aplicada.

(como disponibles en enero 2008)

- 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>
global optimization

- <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
- <http://www.swarmintelligence.org/index.php>
Particle swarm optimization

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

- escoger un tipo de optimización según las clasificaciones a continuación
- buscar librerías disponibles que implementan este método de optimización
- analizar la librería por lo menos según los siguientes criterios:
 - completitud, complejidad, entorno de uso, algoritmos concretos disponibles, filosofía de diseño, simplicidad de uso, aplicaciones donde se usó, documentación, recursos disponibles etc.

Optimizar significa

- buscar alguna solución
- que se distinga de las demás posibles soluciones
- por ser (lo suficiente) extremo
- dentro de una ordenación
- a lo mejor bajo ciertas restricciones
- (en un tiempo razonable).

Ejemplo: escalar dentro de una sierra de montañas (con niebla).

Problemas que se quieren resolver:

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

espacio de búsqueda y/o función objetivo pueden 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

La función objetiva tiene que ser acotada.

- Minimización
- Maximización
- Obviamente cualquier problema de maximización se puede convertir en un problem de minimización.

Nociones básicas

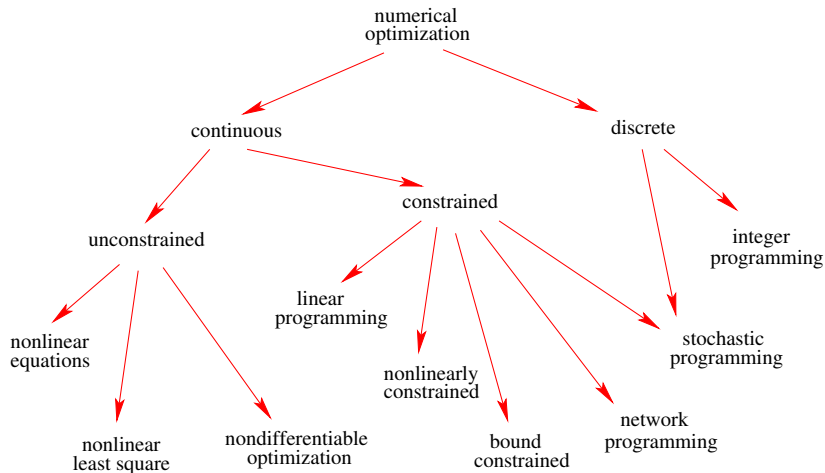
condiciones

- restricciones
- solución factible (feasibility problem)
- codificación de soluciones

Nociones básicas

clasificación

(según NEOS server (casi), Argonne National Laboratory)



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

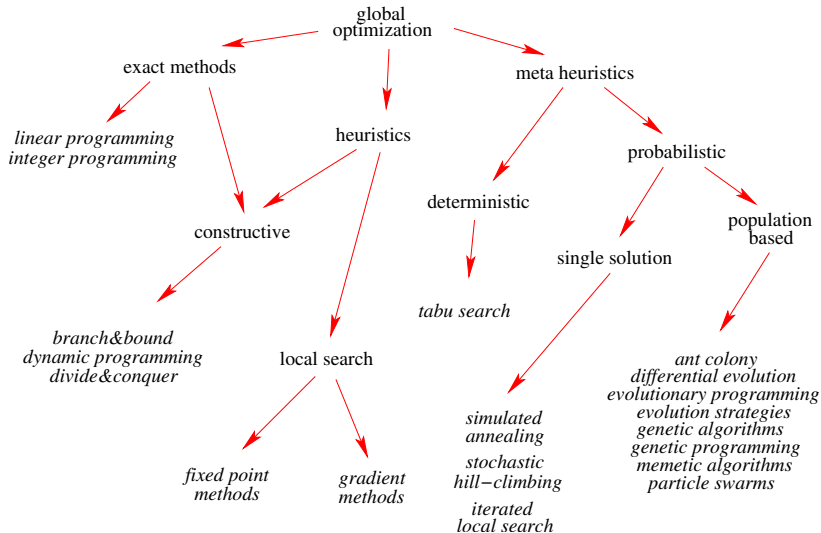
Optimización global

problema

- 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).

Nociones básicas

optimización global (un intento incompleto)



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).

No free lunch theorem

realidad

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).

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

Clasificación

según función objetiva y espacio de búsqueda I

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, ejemplos
Nelder–Mead (1965, improvements 1994, 2002)
Rodríguez–Palomares (2002, improvements 2005)
Dennis–Torczon (1991, multidirectional search algorithm)

función objetivo discreta lineal, espacio continuo:

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

Clasificación

basadas en fenómenos naturales

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

Clasificación

con meta heurísticas

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

- 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)

- 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

- 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?)

- 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?)

- 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?)

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

Ejemplo reciente (André Falcão, Residue fragment programs for enzyme classification, Proceedings BKDB2005, pp.24–28, 2005).

- 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?)

- 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

- 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

Particle swarm optimization

bases

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

Particle swarm optimization

algunos detalles

- 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

Particle swarm optimization

velocity actualización

$$v_i = \xi(v_i + U[0, \varphi_1](p_i - x_i) + U[0, \varphi_2](p_g - x_i))$$

$$x_i = x_i + v_i$$

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
- $\varphi_1 = 2,05$
- $\varphi_2 = 2,05$
- $\xi = 0,729$

Particle swarm optimization

versiones

- versión binaria:** Se *interpretan* las variables con un umbral para variables binarias segun una distribución
- versión discreta:** Se *interpretan* 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 .

Particle swarm optimization

convergencia

- 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

- 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)

- 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
- 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

Multi-objetivo

con 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

- “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.