

Algoritmes Genètics i Problema del Venedor Viatger (TSP)

Optimització Computacional

Introducció a Algoritmes Genètics (AG)

- ▶ Els Algoritmes Genètics són tècniques d'optimització inspirades en la selecció natural.
- ▶ Són útils per trobar solucions aproximades a problemes complexos on el càlcul exhaustiu és impossible.
- ▶ Els AG imiten la **selecció natural** per "evolucionar" solucions a través de la **selecció, creuament i mutació**.

Explicació dels Algoritmes Genètics

- ▶ Imagineu-vos un concurs on hem de trobar la paraula correcta a partir de lletres aleatòries.
- ▶ Fem un munt d'endevinalles i anem millorant-les generació a generació.
- ▶ Cada generació es seleccionen les millors respostes i es "mesclen" per a formar-ne de noves.

Components d'un Algoritme Genètic

1. **Representació:** Cada possible solució s'anomena cromosoma.
2. **Funció de Fitness:** Mesura la qualitat de cada solució.
3. **Selecció:** Es trien les millors solucions per generar-ne de noves.
4. **Creuament:** Es combinen cromosomes per crear-ne de nous.
5. **Mutació:** S'introdueixen canvis aleatoris per evitar estancaments.

Algoritme Genètic en Forma Matemàtica

1. **Inicialització:** Genera una població inicial de N solucions candidates (cromosomes) $\{x_1, x_2, \dots, x_N\}$.
2. **Avaluació:** Calcula la **funció de fitness** $f(x)$ per cada cromosoma x_i a la població, on:

$$f(x) = \text{qualitat de la solució}$$

3. **Selecció:** Selecciona cromosomes per a reproducció proporcionalment a $f(x)$. Els cromosomes amb major fitness tenen més probabilitat de ser seleccionats.
4. **Creuament (Crossover):** Genera noves solucions (fills) combinant parts dels cromosomes seleccionats (pares). Si x_i i x_j són pares, els fills són:

$$\text{fill}_1 = \text{part}(x_i) + \text{part}(x_j), \quad \text{fill}_2 = \text{part}(x_j) + \text{part}(x_i)$$

Algoritme Genètic en Forma Matemàtica

1. **Mutació:** Aplica un canvi aleatori petit a alguns gens en els fills amb probabilitat $p_{\text{mutació}}$, per evitar convergència prematura.
2. **Actualització de la Població:** Reemplaça la població actual amb els nous cromosomes (fills), mantenint o eliminant els més antics segons l'estratègia.
3. **Repeteix:** Repetir els passos anteriors fins que es compleixi un criteri d'aturada, com ara el nombre màxim de generacions o una millora mínima en el fitness.

Exemple senzill d'Algoritme Genètic: Trobar la paraula ÇAT"

- ▶ Suposem que volem trobar la paraula correcta: **ÇAT"**.
- ▶ Comencem amb un conjunt de solucions aleatòries, com ara: "XYZ", "BATi ÇAT".
- ▶ **Avaluació (Fitness)**: Cada solució s'avalua en funció de la seva similitud amb la paraula objectiu ÇAT".
 - ▶ Per exemple, ÇAT" té una fitness del 100%, "BAT" té un 66% i "XYZ" té un percentatge molt baix.
- ▶ **Selecció**: Seleccionem les millors solucions, com ara ÇATi "BAT", per crear-ne de noves.
- ▶ **Creuament**: Combinar part dels millors resultats per generar nous intents.
- ▶ **Mutació**: Modificar una lletra de tant en tant per introduir noves possibilitats.
- ▶ **Repeteix**: Aquest procés es repeteix durant diverses generacions fins que trobem ÇAT".

Problema d'Optimització: Problema del Venedor Viatger (TSP)

- ▶ El Venedor Viatger ha de visitar diverses ciutats, començant per una, visitant-ne cada una exactament una vegada i tornant al punt inicial.
- ▶ L'objectiu és trobar la ruta amb la distància total mínima.
- ▶ Aquest problema és d'optimització perquè busquem minimitzar la distància del recorregut.

Resolució del TSP amb Algoritmes Genètics

1. **Representació de la Ruta:** Cada cromosoma representa una ruta possible.
2. **Funció de Fitness:** La distància total de la ruta. Quan més curta, millor el "fitness".
3. **Selecció:** Es trien les rutes més curtes per generar la següent generació.
4. **Creuament (Crossover):** Es combinen rutes per crear-ne de noves.
5. **Mutació:** S'intercanvien ciutats aleatòriament per mantenir la diversitat.

Exemple de TSP: Ciutats de Mostra

- ▶ Ciutats: $[0,0]$, $[1,5]$, $[5,3]$, $[6,6]$, $[8,3]$
- ▶ Cada ruta és una seqüència de ciutats, i la distància total és la suma de distàncies entre cada parell de ciutats.
- ▶ Es trien les millors rutes, es combinen, i es fan mutacions per trobar la ruta òptima.

Resultats: Algoritme Genètic per TSP

- ▶ L'algoritme evoluciona amb cada generació, millorant les rutes.
- ▶ Després de diverses generacions, s'obté una ruta que s'acosta a la mínima distància.
- ▶ Els resultats poden variar lleugerament segons els paràmetres: mida de la població, taxa de mutació, etc.

Per què Els Algoritmes Genètics Funcionen Bé per al TSP?

- ▶ Els AG són efectius per problemes combinatoris, on el nombre de rutes és massa gran per fer un càlcul exhaustiu.
- ▶ Els AG poden explorar una gran quantitat de rutes i trobar-ne una de "bona" sense revisar-les totes.
- ▶ Els AG ens permeten afegir diversitat a les solucions i evitar quedar atrapats en solucions subòptimes.

Aplicacio Industrial

Algunas de les empresas que estan fent servei de Algoritmes Genètics son els seguen

- ▶ Google DeepMind - AlphaFold Projecte.
- ▶ Tesla - vehiculus Autonomos
- ▶ Amazon - Operaciones logisticas
- ▶ Nvidia - OPTimització de GPU arquitectura
- ▶ Toyota, Siemens, etc...

Conclusió

- ▶ Els Algoritmes Genètics són eines potents per a problemes d'optimització complexos.
- ▶ El TSP és un exemple clàssic on els AG poden proporcionar una solució pràctica a problemes difícils.
- ▶ Els paràmetres de l'algoritme (mida de la població, taxa de mutació) són essencials per al rendiment.

Preguntes?