

Algoritmi Evolutivi: Codificarea binară, Permutări

I OBIECTIVE

- Proiectarea și implementarea algoritmilor evolutivi
- Discuție: reprezentarea soluției, funcția de fitness, strategia de selecție, operatorii de încrucișare și mutație
- Interpretarea rezultatelor obținute de un algoritm evolutiv

2 UN ALGORITM EVOLUTIV (AE) STANDARD

Notatii: t – iteratia (sau generația curentă), $P(t)$ – populația de la generația t

Parametri AE: N – numărul de indivizi din populație, M – numărul maxim de generații

$t=0$

Initializare $P(t)$ // populația inițială este formată din N indivizi, fiecare inițializat aleator
// conform reprezentării alese pentru problema

Evaluare $P(t)$ // se calculează calitatea fiecărui individ din $P(t)$ folosind funcția de fitness

while ($t < M$) do

$Parents(t)$ = Selecție părinți // se selectează perechile de părinți din $P(t)$ folosind
// un tip de selecție ex. selecție turnir, selecție proporțională, etc

$PX(t)$ = Incrucișare $Parents(t)$ // se aplică încrucișarea perechilor de părinți

$PM(t)$ = Mutație // se aplică mutație descendenților rezultati din încrucișare

$P(t+1)$ = Selecție supraviețuitori din $P(t)$, $PX(t)$ și $PM(t)$

$t = t + 1$

Evaluare $P(t)$

End-while

Observații:

- Un individ din populație corespunde unei potențiale soluții a problemei
- Populația $P(t)$ conține în fiecare generație N indivizi
- Rezultatul AE este cel mai bun individ din populația finală $P(t)$, unde $t=M$
- Pe lângă parametrii AE deja specificați (N , M), un algoritm evolutiv poate avea și alți parametri care provin din tipul de selecție, încrucișare, mutație folosite

Decizii AE:

- Decizii generale: reprezentare soluție, funcție de fitness
- Cum se selectează părinții? Selecția depinde de fitness. Pasul *Selecție părinți* poate folosi orice mecanism de selecție (vezi Curs 4-5) pentru primul și al doilea părinte sau o combinație (ex. un părinte selectat cu turnir și un părinte selectat aleator). Mărimea $Parents(t)$ poate fi egală cu N sau mai mică.

- *Ce încrucișare se folosește?* Depinde de reprezentare: există operatori specifici definiți pentru codificarea binară, reală și permutări (vezi Curs 6).
- *Ce mutație se folosește?* Depinde de reprezentare: există operatori specifici definiți pentru codificarea binară, reală și permutări (vezi Curs 6).
- *Cum se selectează supraviețuitorii?* Acest tip de selecție generează populația de la generația următoare. Se specifică un mecanism de selecție care alege indivizi din populația curentă și indivizii noi generați prin încrucișare / mutație. Pot fi folosite: selecția deterministă, selecție proporțională cu fitnessul, selecția turnir, selecția prin ordonare sau o combinație ex. 10% cei mai buni indivizi din $P(t)$ și 90% selectați cu turnir (vezi Curs 5).

3 REPREZENTARE

- Problema rucsacului: codificare binară
- TSP: codificare de tip permutare

4 SELECTIA

Selectia determinista:

- **Selectia ($\mu + \lambda$)**
 - λ descendenți sunt creați din μ părinți
 - $\mu + \lambda$ indivizi sunt evaluați
 - Selecția: Cei mai buni μ sunt păstrați
- **Selectia (μ, λ)**
 - λ descendenți sunt creați din μ părinți, $\lambda \geq \mu$
 - Selecția: Cei mai buni μ din cei λ indivizi sunt păstrați

Selectia stocastica:

- Selecția proporțională
- Selecția turnir

Curs 5 descrie toate tipurile de selecție.

5 INCRUCISAREA

Codificarea binară

- Incrucișarea cu un punct de tăietură
- Incrucișarea cu mai multe puncte de tăietură
- Incrucișarea uniformă
 - Parametru: p - probabilitatea ca gena unui descendent să provină din primul sau al doilea părinte
 - $X = X_1 X_2 \dots X_k \dots X_r$
 - $Y = Y_1 Y_2 \dots Y_k \dots Y_r$

- Pentru fiecare poziție i din x' se alege părintele care va da valoarea poziției respective cu probă p
- Pentru y' se ia valoarea poziției corespunzătoare din celălalt părinte

Codificarea prin permutări

- Order Crossover
 1. Alege aleator o parte $(i...j)$ din primul părinte
 2. Pentru primul descendent
 - 2.1 Copiază partea $(i...j)$
 - 2.2 Setează celelalte poziții astfel:
Începând de la poziția imediat următoare lui j
Folosind ordinea din al doilea părinte
Continuând circular până la poziția dinaintea lui i
 3. Al doilea descendent se creează similar cu primul dar cu rolurile părinților schimbate

Curs 6 descrie toate tipurile de încrucișare.

6 MUTATIA

Codificarea binară

- Mutatia tare
 - P1. Pentru fiecare cromozom al populației curente și pentru fiecare poziție a cromozomului se execută:
 - P1.1. Se generează un număr aleator q în intervalul $[0,1]$.
 - P1.2. Dacă $q < p_m$ atunci se execută mutația poziției respective, schimbând 0 în 1 și 1 în 0.
 - În caz contrar ($q \geq p_m$), poziția respectivă nu se schimbă
- Mutatia slabă
 - P1. Pentru fiecare cromozom al populației curente și pentru fiecare poziție a cromozomului se execută:
 - P1.1. Se generează un număr aleator q în intervalul $[0,1]$.
 - P'1.2. Dacă $q < p_m$ atunci se alege aleator una din valorile 0 sau 1. Se atribuie poziției curente valoarea astfel selectată.
Dacă $q \geq p_m$ atunci poziția curentă nu se schimbă.

Codificarea prin permutări

- Mutatia inserare
- Mutatia interschimbarea (2-swap)
- Mutatia inversiune (2-opt)
- Mutatia amestec

Curs 6 descrie toate tipurile de mutație.

7 VARIANTE AE

Diferite variante AE pot fi obținute prin schimbarea tipului de selecție, încrucișare, mutație, etc. Mai jos sunt enumerate câteva posibilități care pot fi considerate:

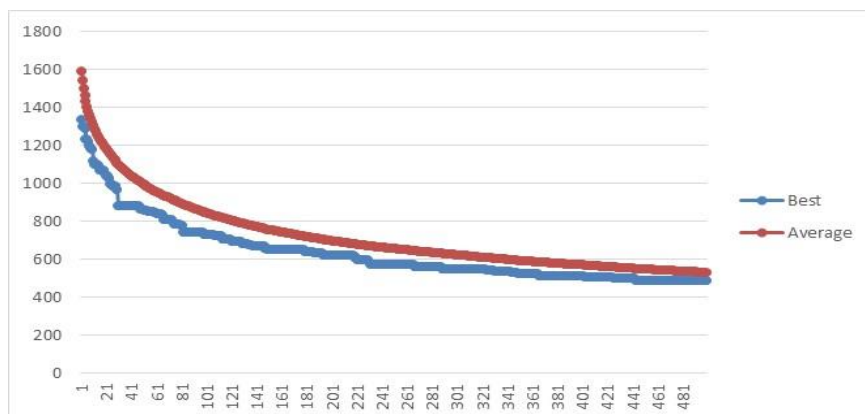
- **Initializarea populației (aleator / o parte din indivizi generați cu greedy / HC/ SA/ TS)**
- Operatorul de încrucișare (order crossover, partially mapped crossover, cycle crossover)
- Operatorul de mutație (mutația inserare, mutația interschimbare 2-swap, mutația inversiune 2-opt, mutația amestec)
- Strategia de selecție a părinților (selecție proporțională, selecție turnir, aleator)
- Strategia de selecție a supraviețuitorilor (selecție proporțională, selecție turnir, elitism, selecție deterministă)
- Hibridizare cu mecanisme de căutare locală
- Criteriul de terminare (număr maxim de generații, număr maxim de evaluări ale funcției de fitness, număr de generații în care fitnessul nu este îmbunătățit)

8 EXPERIMENTE

- Nici o concluzie nu poate fi trasă dintr-o singură rulare
 - 10 rulări independente pentru fiecare variantă AE
 - Best, average, timp de rulare
- Trebuie considerate mai multe valori ale principalilor parametri AE (mărime populație, număr maxim de generații)
- Comparatii:
 - Aceeași variantă AE, diferite valori ale parametrilor
 - Între diferite variante AE, cu alți algoritmi ex. SA/ TS
- *Ce putem măsura?*
 - *Rezultatul mediu obținut într-un anumit timp*
 - *Timpul mediu necesar obținerii unui anumit rezultat*
 - *Proportia de rulări în care s-a obținut o anumită soluție*
 - *Cea mai bună soluție din 10 rulări*

Exemplu tabel pentru comparații AE cu diferiți parametri

Population	Generations	Best	Average	Time (for 10 runs)
100	100	730	771	6 sec
100	1000	432	439	48 sec
100	2000	427	437	1 min 27 sec
500	100	731	752	22 sec
500	1000	427	432	3 min 42 sec
500	2000	427	431	6 min 37 sec
1000	100	706	736	43 sec
1000	500	463	478	4 min 21 sec
1000	1000	426	429	6 min 54 sec
1000	2000	426	427	14 min 14 sec

Exemplu grafic evoluție best*Population 1000, generations 500**Population 1000, generations 2000*