

Clase de probleme de optimizare

- Optimizare dinamică
- Optimizare multi-modală
- Optimizare cu restricții

Medii incerte si medii dinamice

- **Zgomote**
 - **Problema:** Functia obiectiv contine zgomote: erori de masurare, simulari incomplete
 - **Scop:** Eliminarea zgomotelor - outliers
- **Perturbatii**
 - **Problema:** Variabilele functiei obiectiv sunt afectate de zgomot
 - **Scop:** Gasirea unor solutii robuste
- **Aproximarea functiei obiectiv**
 - **Problema:** Functia obiectiv este aproximata deoarece o evaluare este prea complexa
 - **Scop:** Gasirea de solutii robuste, eliminarea zgomotelor
- **Functia obiectiv variaza în timp**
 - **Problema:** Functia obiectiv se schimba în timp
 - **Scop:** Urmarirea optimelor care variaza în timp

Probleme de optimizare

- **Optimizarea Statica**

- **Scop:** gasirea optimului global al functiei obiectiv
- **Problema:** convergenta prematura
- **Cauza:** populatia pierde din diversitate
- **Solutia:** stimularea diversitatii populatiei

- **Optimizarea Dinamica**

- **Scop:** urmarirea optimului care se schimba
- **Problema:** dificultatea urmaririi optimului
- **Cauza:** populatia a pierdut diversitatea
- **Solutia:** stimularea diversitatii populatiei

Optimizare dinamică

- Tradițional, metaheuristicile se focusează pe rezolvarea de probleme statice
 - scopul este de a găsi un optim repede și precis
- Dar multe probleme din lumea reală sunt dinamice
 - dynamic optimization problem (DOP) - în care modificări apar în timp
 - rețele de transport, timpul necesar parcurgerii distanței dintre două noduri poate să se modifice în timp
 - în logistică, cererile clienților se pot modifica

Ce sunt probleme de optimizare dinamică?

- Probleme de optimizare care se modifică în timp
 - mai sunt numite probleme dinamice sau probleme dependente de timp

$$F = f (X , y , t)$$

- unde X este o variabilă de decizie, y parametri, t timpul
- DOP
 - probleme dinamice care se rezolvă online de un algoritm pe parcursul treceri timpului

De ce se folosesc euristicile inspirate de natură pentru a rezolva DOPs?

- Multe probleme din lumea reală sunt DOPs
- Algoritmi inspirați din natură dacă sunt proiectați corespunzător sunt o alegere bună
 - inspirați din procese de evoluție naturală/biologică din medii dinamice
 - ar trebui să fie o soluție care să poată să rezolve DOPs

Benchmark și testarea DOPs

- Ideea de bază
 - modificarea problemelor statice entru a crea probleme dinamice
- În spațiul real
 - schibarea între diferite funcții
 - mutarea/remodelarea unor vârfuri
- În spațiul binar
 - adaugarea de măști binare
- În spațiul combinatorial
 - modificare variabilelor de decizie: in cazul problemei rucsacului
modificarea greutatei/profotul obiectelor
 - adaugarea/stergera de variabile de decizie: noi taskuri în cazul problemelor de planificare, adăugarea/ștergera de noduri în probleme de rutare

Tipuri de modificări

- Pași mici
- Pași mari
- Aleatorii
- Haotice
- Recurente
- Recurente cu zgomot
- ...

Caracteristici comune

- Majoritatea DOPs sunt probleme legate de timp
- Pentru majoritatea DOPs se presupune că modificările sunt detectabile
- În mare parte a cazurilor, funcția obiectiv se modifică
- Multe DOPs conțin modificări nepredictibile
- Majoritatea DOPs conțin modificări ciclice/recurente

Măsuri de performanță

- Collective mean fitness (mean best-of-generation)

$$\bar{F}_{BOG} = \frac{1}{G} \times \sum_{i=1}^{i=G} \left(\frac{1}{N} \times \sum_{j=1}^{j=N} F_{BOG_{ij}} \right)$$

G ,N: numarul de generatii, respectiva rulari
– $F_{BOG_{ij}}$: best-of-generation fitness
of generation i of run j

- Adaptation performance (Mori et al., 1997)

$$Ada = \frac{1}{T} \sum_{t=1..T} (f_{best}(t)/f_{opt}(t))$$

- Accuracy (Trojanowski and Michalewicz, 1999)

$$Acc = \frac{1}{K} \sum_{i=1..K} (f_{best}(i) - f_{opt}(i))$$

– $f_{best}(i)$: best fitness for environment i
(best before change)

Metode de adaptare a metaeuristicilor pentru DOPs

- Există multe abordări ale algoritmilor evolutivi pentru DOPs
- Abordări tipice:
 - Memorie: stochează și refolosește informații utile
 - Diversitate: abordează problema de convergența direct
 - Multi-populații: cooperarea subpopulațiilor
 - Adaptare: adaptează generatorii și parametrii
 - Predicție: prezic modificarea și acționează în avans
- Aplicate pentru diferiți algoritmi evolutivi de rezolvare DOPs

Metode de adaptare a metaeuristicilor pentru DOPs

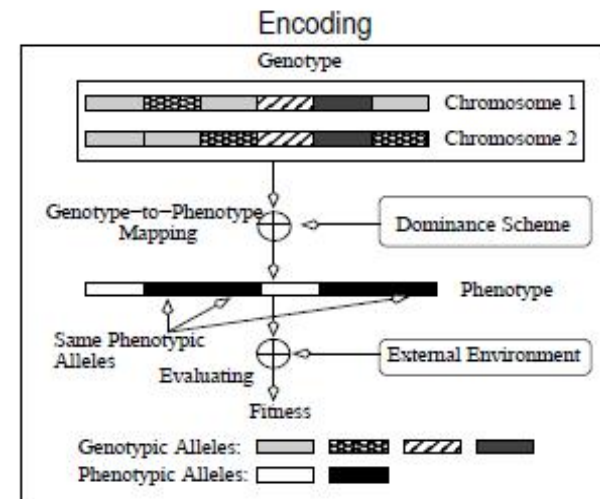
- Algoritmi traditionali evolutivi nu se descurcă bine în caul DOPs
 - de ce?
- Primă idee
 - restartarea algoritmului evolutiv după fiecare modificare
 - Nu e o alegere buna deoarece
 - nu este eficient, se pierde resurse computationale
 - pot conduce la soluții diferite înainte și după restartare. Pentru probleme reale se dorește ca soluțiile să fie similare
 - Au nevoie de mecanisme speciale pentru a putea să se descurce în cazul DOPs

Abordări bazate pe memorie

- DOPs ciclice: modificări ciclice ale unui set de stări
- Poate fi folosită memoria pentru stocarea și re folosirea de informații utile
- Două tipuri de memorie
 - memorie implicită - folosește o reprezentare redundantă
 - memorie explicită - folosește spațiu extra de stocare a informației

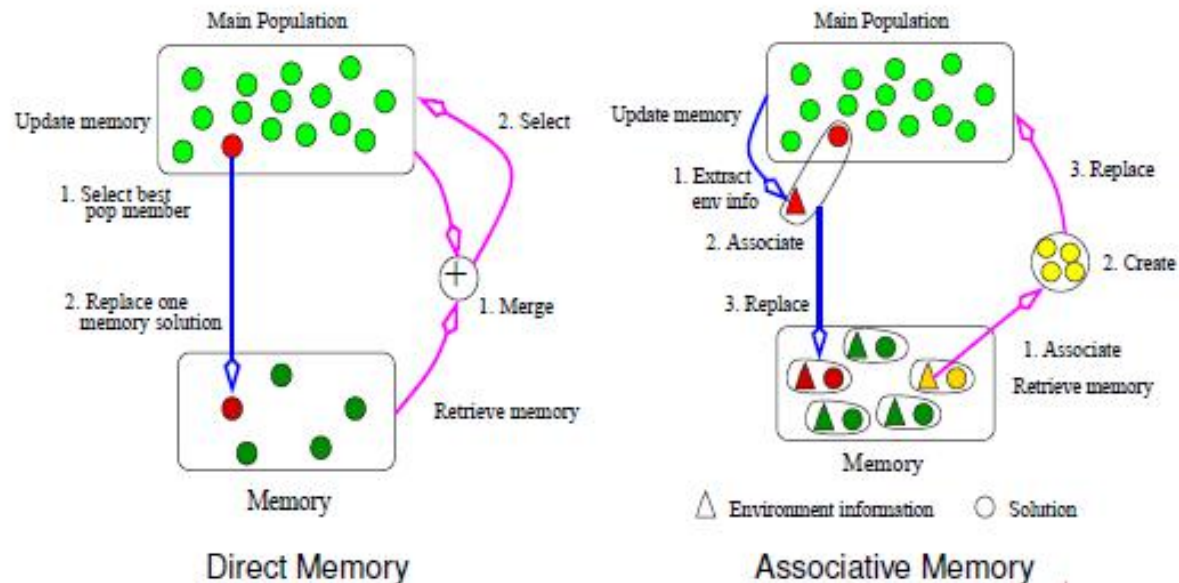
Abordări bazate pe memorie

- Memorie implicită: Diploid Genetic Algorithm
 - Fiecare individ conține o pereche de cromozomi
 - Schema de dominață mapează genotipurile to fenotipuri
 - Schema de dominanță se poate schimba sau fi adaptivă



Abordări bazate pe memorie

- Memorie explicită directă vs asociativă
Memory: Direct vs Associative
 - memoria directă: stochează soluții bune
 - memoria asociativă: stochează soluții bune + informații despre mediu

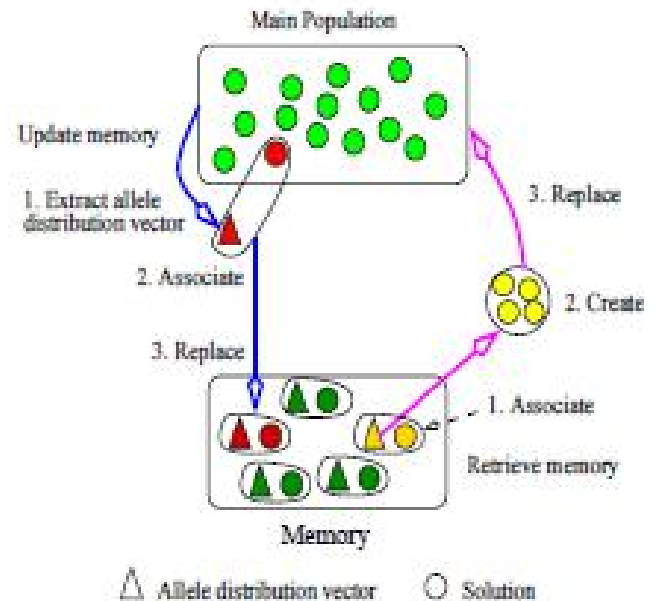


Abordări bazate pe memorie

- Abordări bazate pe memorie explicită
 - folosirea unei memorii extreme
 - în timp se stochează informații utile în populație
 - când are loc o modificare se folosește memoria pentru a identifica noul optim

Abordări bazate pe memorie

- Memorie asociativă
 - folosirea distribuției allelor pentru a reprezenta informațiile despre mediu
 - folosirea memoriei pentru a stoca informații despre soluție și mediu
 - modificarea memoriei pe baza similarității
 - re-evaluarea populației la fiecare generație. Dacă o modificare este detectată
 - extragerea celei mai bune perechi din memorie
 - crearea unei soluții prin exemplificare
 - înlocuirea în populație aleator

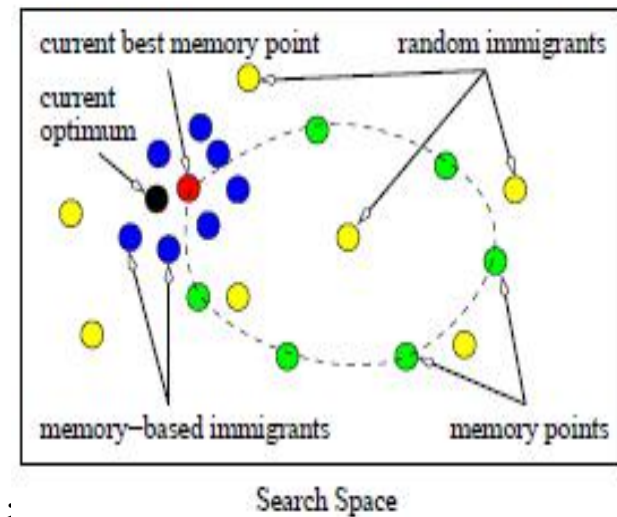


Abordări bazate pe densitate

- Convergența este o mare problemă pentru DOPs
- Imigranți aleatori
 - La fiecare generație se inserează câțiva indivizi aleatorii (numiți imigranți) în populație pentru a menține diversitatea
 - Când optimul se modifică, imigranția aleatori ghidează populația spre noul optim

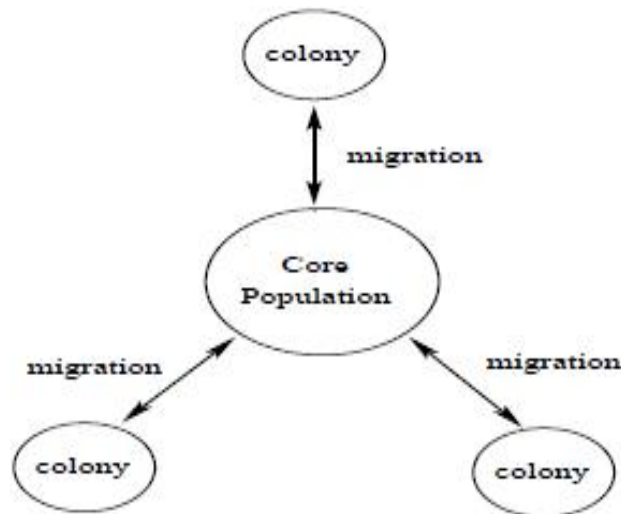
Abordări bazate pe densitate

- Imigranți aleatori mențin diversitatea în populație în timp de memoria adaptează mediul direct la noul mediu
- Imigranți bazați pe memorie
 - folosește memoria pentru a ghida imigranți în mediul curent
 - reevaluarea memoriei la fiecare generație
 - extrage cel mai bun element din memorie la momentul t - $BM(t)$
 - generează imigranți prin aplicarea de mutație asupra elementului $BM(t)$ cu o anumită probabilitate
 - înlocuiește cei mai "răi" membri din populație cu imigranți generați



Multi populații

- O metodă cu multipopulații care folosește colaborarea între subpopulații - Shifting Balance GA
 - populația de bază exploatează zona promișă
 - câteva coloni explorează spațiul de căutare



Multi populații

- Particle Swarm Optimisation (PSO):
 - motivat de?
- PSO este folosit la rezolvarea DOPs
- Clustering PSO (CPSO) pentru DOPs
 - Folosirea unei tehnici de grupare pentru construirea de sub-populații
 - Fiecare sub-populație va căuta un optim rapid
 - Verificare suprapunerii și convergență
 - Strategii pentru a răspunde la o modificare

Abordări bazate pe predicție

- Pentru unele probleme de optimizarea dinamică se pot identifica șablonuri predictibile
- Tehnici ca forecasting, Kalman filter, etc. pot fi folosite pentru prezicerea
 - locației următorului optim după o modificare
 - când va avea loc următoarea modificare și ce caracteristici va avea mediul

Abordări adaptive

- Scop: Adaptarea operatorilor/parametriilor, de obicei după o schimbare
 - Hipermutația - creșterea temporală a ratei de mutație când se observă o modificare în mediu
 - Hiperselectia - creșterea temporală a presiunii de selecție
 - Hyper-învățarea: creșterea temporară a ratei de învățare
 - Population-Based Incremental Learning (PBIL)
 - Combinate: hiperselectia și hiperînvățarea cu repornirea sau hipermutația

Caracteristici ale DOPs

- Detectarea modificărilor
 - Majoritatea studiilor presupun că modificările sunt ușor de detectat sau vizibile algoritmului când se întâmplă
 - de obicei modificările sunt greu de detectat pentru multe probleme DOPs
- Înțelegerea caracteristicilor DOPs
 - Ce caracteristici fac problema ușoară sau dificilă