

Algoritmi evolutivi pentru rezolvarea problemelor de optimizare multicriterială

- ❑ Specificul optimizării multicriteriale
- ❑ Metode de rezolvare a problemelor de optimizare multicriterială
- ❑ Optimizare în sens Pareto cu algoritmi evolutivi

Specificul optimizării multicriteriale

Optimizare multicriterială = optimizarea simultană a mai multor criterii

Exemple:

1. Determinarea parametrilor unui produs industrial care asigură maximizarea fiabilității și minimizarea costurilor
2. Rezolvarea unei probleme de rutare într-o rețea de telefonie astfel încât să fie minimizeate atât costurile cât și congestia rețelei
3. Gruparea datelor în clustere astfel încât să fie maximizată similaritatea datelor aparținând aceluiași cluster și minimizată similaritatea datelor din clustere diferite

Specificul optimizării multicriteriale

Formularea problemei: $f: \mathbb{R}^n \rightarrow \mathbb{R}^r$, $f(x) = (f_1(x), \dots, f_r(x))$

Se caută x^* care satisface:

- (i) Restricții de tip inegalitate: $g_i(x^*) \geq 0$, $i=1..p$
- (ii) Restricții de tip egalitate: $h_i(x^*) = 0$, $i=1..q$
- (iii) Optimizează (maximizează sau minimizează fiecare criteriu)

Obs:

1. criteriile pot fi contradictorii (ex: calitatea și prețul unui produs: cu cât produsul este mai bun calitativ cu atât va fi mai mare prețul)
2. multe probleme se caracterizează prin două criterii ($r=2$) însă există probleme și cu mai multe criterii

Specificul optimizării multicriteriale

Exemplu: $r=2$ (două criterii de optim)

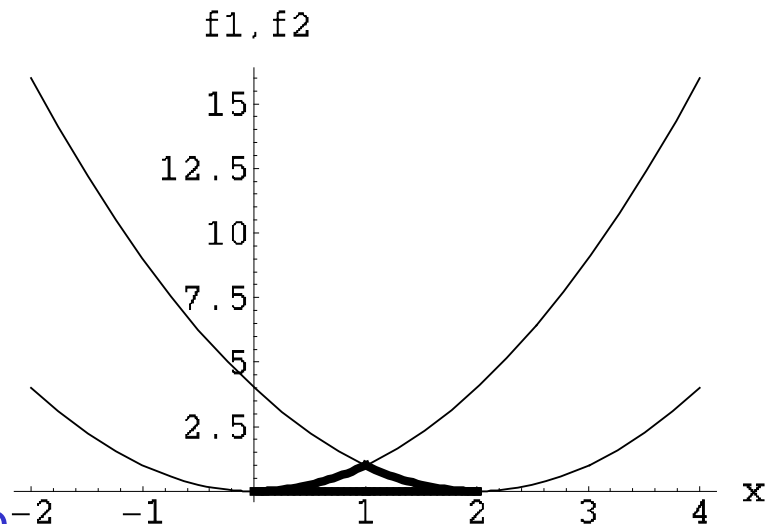
$f_1(x)=x^2$, $f_2(x)=(x-2)^2$ pentru x în $[-2,4]$

Se dorește minimizarea ambelor funcții. Nu există x^* care să minimizeze simultan cele două funcții

Se caută soluții de compromis:
suficient de bune din perspectiva
ambelor criterii:

x în $[0,2]$ – nu există x' cu
 $f_1(x') < f_1(x)$ și $f_2(x') < f_2(x)$

O astfel de soluție de compromis
este numită **soluție în sens Pareto**



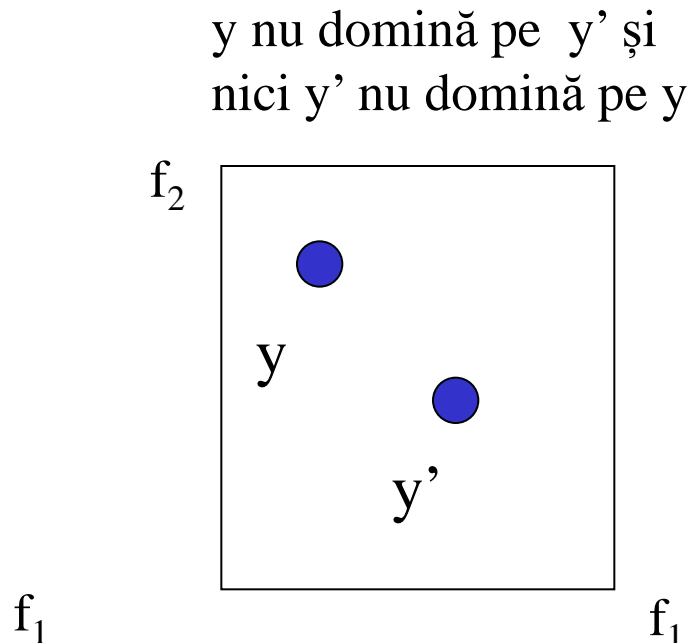
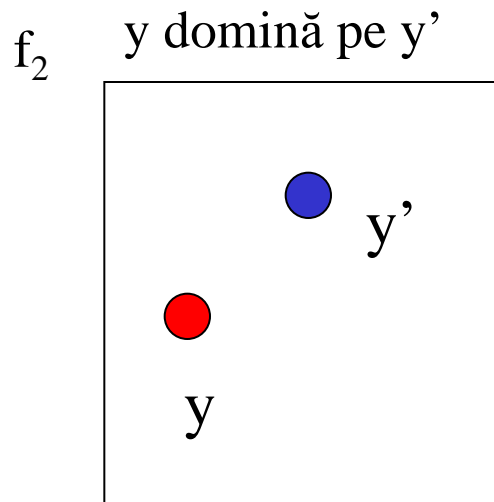
Specificul optimizării multicriteriale

Noțiuni de bază în optimizarea în sens Pareto:

1. Relația de dominare:

y domină pe y' (în cazul unei probleme de **minimizare**)

daca $y_i \leq y'_i$ pentru fiecare i și inegalitatea este strictă pentru cel puțin o componentă



Relația de dominare este o relație de ordine parțială (există elemente care sunt reciproc nedominate)

Specificul optimizării multicriteriale

Noțiuni de bază în optimizarea în sens Pareto:

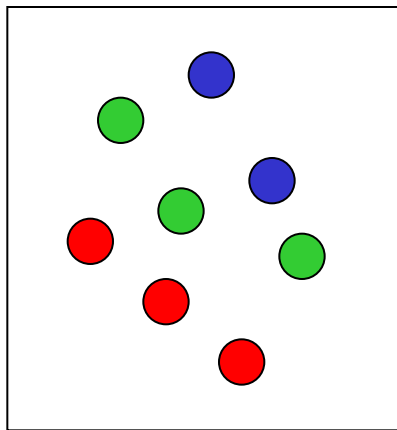
2. Element nedominat în raport cu o mulțime:

y este nedominat în raport cu V dacă nu există nici un element în V care să îl domine pe y

Elementele marcate cu roșu sunt nedominate în raport cu toate elementele

Elementele marcate cu verde sunt nedominate în raport cu celelalte marcate cu verde și cu cele marcate cu albastru

f_2



f_1

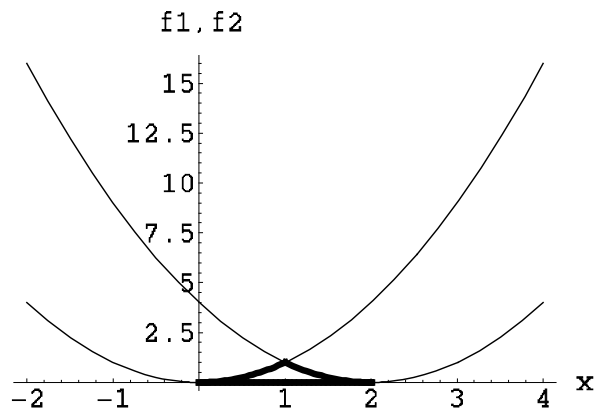
Specificul optimizării multicriteriale

Noțiuni de bază în optimizarea în sens Pareto:

3. Soluție optimală în sens Pareto

Un element x este soluție optimală în sens Pareto dacă nu există nici un element x' astfel încât $f(x')$ să îl domine pe $f(x)$

Mulțimea tuturor elementelor Pareto optimale ale unei probleme de optimizare multicriteriale se numește soluția optimală a problemei (în sens Pareto)



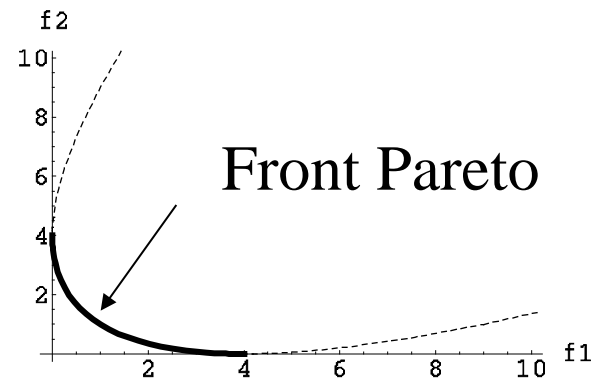
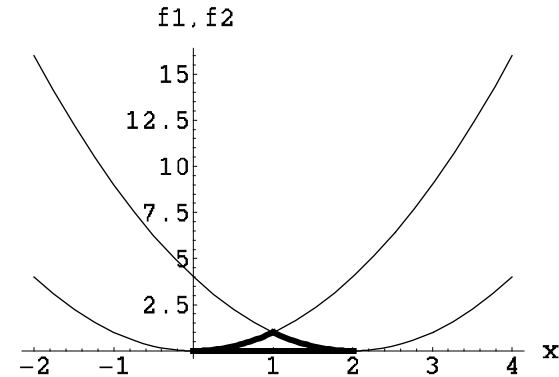
In cazul exemplului studiat mulțimea Pareto optimală este intervalul $[0,2]$

Specificul optimizării multicriteriale

Noțiuni de bază în optimizarea în sens Pareto:

4. Front Pareto

Mulțimea valorilor funcțiilor obiectiv (criteriile de optimizat) asociate elementelor unei mulțimi Pareto optimale se numește front Pareto



Metode de rezolvare

1. Transformarea într-o problemă de optimizare unicriterială: toate criteriile de optim se combină în unul singur

☐ Metoda agregării

$$\bar{f}(x) = \sum_{i=1}^r w_i f_i(x), \quad w_i \in (0,1), \quad \sum_{i=1}^r w_i = 1$$

Avantaje: se reduce la o problemă mai simplă de optimizare unicriterială

Dezavantaje:

- ☐ pentru un set de parametri w se obține o singură soluție; pentru estimarea frontului Pareto trebuie rezolvată pb pt mai multe seturi de valori ale lui w
- ☐ Trebuie specificați parametrii w
- ☐ nu permite estimarea fronturilor Pareto neconvexe

Metode de rezolvare

1. Transformarea într-o problema de optimizare unicriterială: toate criteriile de optim se combină în unul singur

□ Metoda deplasărilor față de valori țintă

$$\bar{f}(x) = \left(\sum_{i=1}^r |f_i(x) - y_i^*|^p \right)^{1/p}, \quad y_i^* - \text{valori tinta}$$

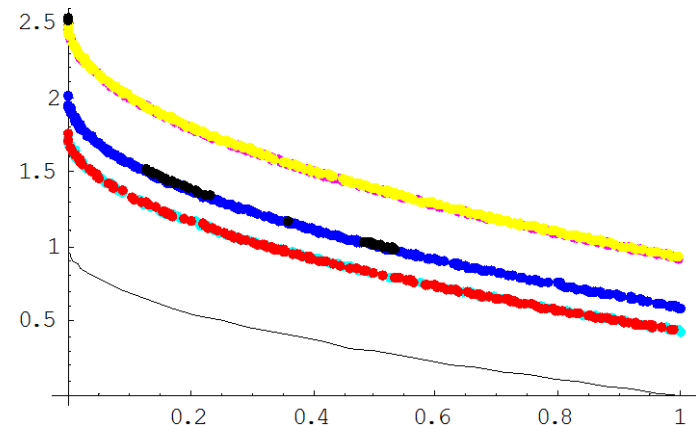
Avantaje: se reduce la o problemă de optimizare unicriterială

Dezavantaje:

- trebuie cunoscute valorile țintă
- problema la care se reduce este unicriterială dar poate fi multimodală (există mai multe valori pentru care se atinge optimul)

Metode de rezolvare

2. Aproximarea simultană (folosind o populație de soluții candidat) a mai multor elemente ale mulțimii optimale în sens Pareto
- ❑ Se folosește un algoritm evolutiv al cărui scop este să genereze într-o singură rulare o aproximare a mulțimii Pareto (și a frontului Pareto corespunzător)
 - ❑ Aproximarea frontului Pareto trebuie să satisfacă cel puțin două caracteristici:
 - ❑ Să fie cât mai apropiată de frontul real
 - ❑ Să fie suficient de diversă



Optimizare în sens Pareto cu algoritmi evolutivi

Pentru ca aproximarea frontului Pareto să aibă cele două proprietăți trebuie folosite tehnici specifice:

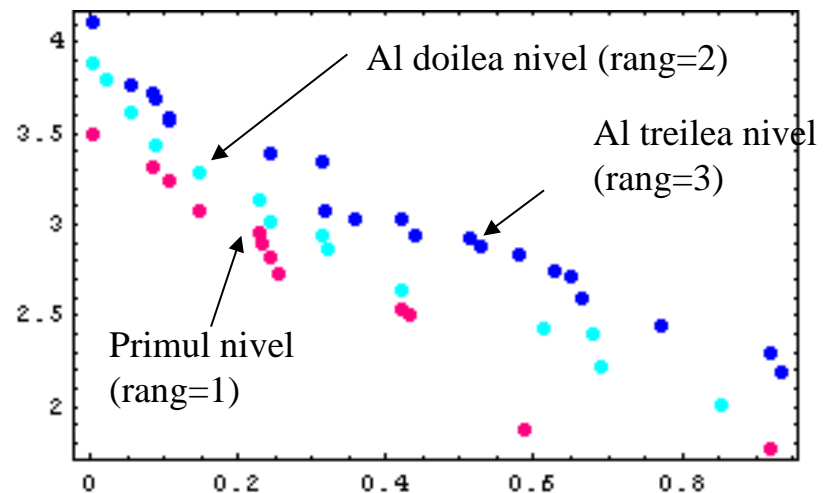
- ❑ Folosirea unui proces de selecție în care să se țină cont de relația de nedominare
- ❑ Utilizarea unui factor de aglomerare în evaluarea elementelor populației (“crowding factor”)
- ❑ Modificarea funcției de scor prin utilizarea unui mecanism de partajare (“sharing function”)
- ❑ Restricționarea încrucișării (“mating restriction”)
- ❑ Asigurarea elitismului prin utilizarea unei populații secundare (arhivă)

Optimizare în sens Pareto cu algoritmi evolutivi

Criterii specifice de selecție:

- ❑ Pe baza nivelului de nedominare (ex: NSGA – Nondominated Sorting GA)
 - ❑ Se organizează populația pe nivele de nedominare:
 - ❑ Primul nivel este constituit din elementele nedominate
 - ❑ Elementele nedominate din mulțimea obținută ignorând primul nivel formează al doilea nivel

- ❑ Un element este considerat mai bun dacă rangul de nedominare este mai mic
- ❑ La selecție se reunește populația părinților cu cea a urmașilor și se ordonează crescător după rang



Optimizare în sens Pareto cu algoritmi evolutivi

Criterii specifice de selecție:

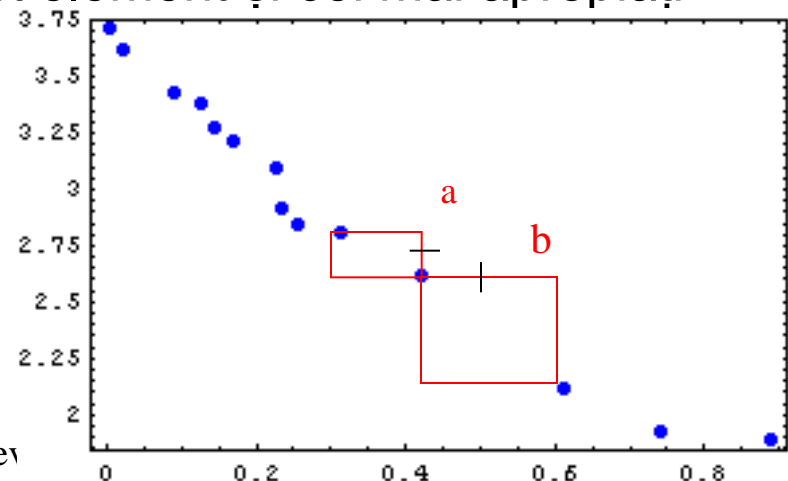
- ❑ **Gradul de adecvare** a unui element depinde de:
 - ❑ Numărul de elemente pe care el le domină (direct proporțional) – dominance count
 - ❑ Numărul de elemente de către care este dominat (invers proporțional)- dominance rank
 - ❑ Ex: SPEA – Strength Pareto EA
 - ❑ La compararea a două elemente se folosește unul dintre criteriile de optimizat. La fiecare etapă criteriul în baza căruia se compară elementele se alege:
 - ❑ pe baza unei reguli de parcurgere a setului de criterii
 - ❑ aleator, pe baza unor probabilități fixate sau adaptive
- Ex: VEGA

Optimizare în sens Pareto cu algoritmi evolutivi

Utilizarea unui factor de aglomerare (crowding factor)

- ❑ Scop: stimularea diversității aproximării frontului Pareto
- ❑ **Idee:** dintre două elemente care au aceeași calitate dpdv al criteriilor (de exemplu sunt reciproc nedominate, aparțin aceluiași nivel de nedominare sau au același grad de adecvare) este preferat cel care se află într-o regiune mai puțin aglomerată)
- ❑ Factorul de aglomerare asociat unui element se calculează în funcție de distanța dintre acest element și cei mai apropiați vecini

Valoarea factorului de aglomerare: $(a+b)/2$



Optimizare in sens Pareto cu algoritmi evolutivi

Mecanism de partajare (sharing mechanism):

- ❑ **Idee:** dacă un grup de indivizi partajează o resursă comună atunci șansa lor de supraviețuire este direct proporțională cu volumul resursei și invers proporțională cu dimensiunea grupului
- ❑ Gradul de adecvare (a_i) al unui element se ajustează prin împărțirea la o funcție de partajare care depinde de distanțele dintre elementele grupului

$$a_i^{(s)} = \frac{a_i}{\sum_{j=1}^m s(d(x_i, x_j))}, \quad s(d) = \begin{cases} 1 - (d / \sigma_s)^\alpha & d < \sigma_s \\ 0 & d \geq \sigma_s \end{cases}$$

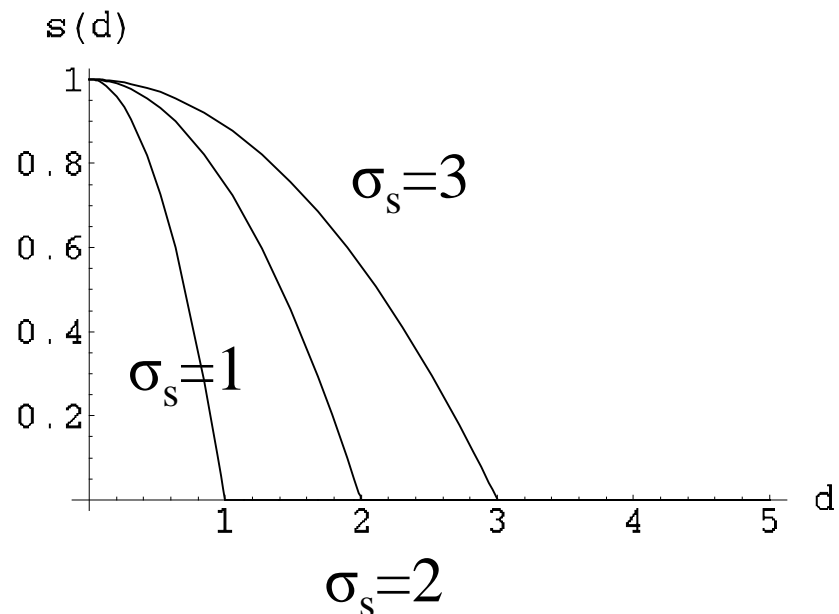
Optimizare in sens Pareto cu algoritmi evolutivi

Mecanism de partajare:

- ❑ Permite diferentierea între elemente care sunt reciproc nedominate
- ❑ Prezintă dezavantajul că necesită specificarea unei raze de acțiune a funcției de partajare (σ_s)

$$s(d) = \begin{cases} 1 - (d / \sigma_s)^\alpha & d < \sigma_s \\ 0 & d \geq \sigma_s \end{cases}$$

Are efect benefic și în cazul **optimizării multimodale** (când se urmărește aproximarea tuturor optimelor locale sau globale)



Optimizare in sens Pareto cu algoritmi evolutivi

Imperechere restrictionata (restricted mating):

- ❑ **Idee:** acceptarea încrucișării doar între elemente aparținând aceleiași specii (elemente suficient de similare)
- ❑ **Scop:** evitarea generarii de urmași de calitate slabă

Exemple:

1. Acceptarea ca părinți a unor elemente care sunt suficient de apropiate între ele
2. Acceptarea ca părinți doar a unor elemente nedominate

Optimizare in sens Pareto cu algoritmi evolutivi

Utilizarea unei arhive:

- ❑ **Scop:** asigurarea elitismului (conservarea elementelor nedominate obținute pe parcursul evoluției)
- ❑ Arhiva va reprezenta aproximarea mulțimii optime

- ❑ **Dezavantaj:** este necesară implementarea unui mecanism de gestiune a arhivei caracterizat prin:
 - ❑ Un nou candidat este acceptat în arhivă dacă nu este dominat de către nici un element al arhivei
 - ❑ Dacă noul element domină elemente existente în arhivă atunci acestea sunt eliminate
 - ❑ Pentru a se evita creșterea nelimitată a volumului arhivei, aceasta trebuie reorganizată periodic (de exemplu prin eliminarea elementelor aflate la o distanță mică față de alte elemente ale arhivei)

Exemple: VEGA (Vector Evaluating Genetic Algorithm)

- ❑ Primul algoritm evolutiv pentru optimizare multicriterială (1985) care nu folosește tehnica agregării
- ❑ Structura generală și operatorii de reproducere sunt similari algoritmilor specifici optimizării unicriteriale
- ❑ Singurul element care se modifică este **procesul de selecție**:
 - ❑ În cazul a r criterii se aplică selecția (de exemplu selecție proporțională) de r ori construindu-se k subpopulații (fiecare cu câte m/r elemente – m fiind dimensiunea populației globale)
- ❑ Avantaje: ușor de implementat
- ❑ Dezavantaje: similare cu cele ale tehnicilor bazate pe agregare

Exemple: Lexicographic Ordering

- ❑ Se asociază priorități criteriilor de optimizat
- ❑ Se rezolvă succesiv probleme de optimizare unicriteriale (pentru fiecare criteriu în parte) în ordinea descrescătoare a priorităților criteriilor
- ❑ Avantaj: simplu de implementat
- ❑ Dezavantaj: necesită stabilirea unor priorități
- ❑ Observație: se poate evita utilizarea priorităților prin selecție aleatoare a criteriilor

Exemple: Multiobjective GA (MOGA)

- ❑ Propus in 1993
- ❑ Fiecărui element i se asociază un rang proporțional cu numărul de elemente de către care este dominat (cu cât rangul este mai mic cu atât e mai bun elementul)
- ❑ Utilizează funcție de partajare și o regulă de selecție restricționată a părinților

Exemple: Nondominating Sorting GA (NSGA)

- ❑ Variante: NSGA-I (1993), NSGA-II (2000)

- ❑ Fiecărui element i se asociază un grad de nedominare bazat pe identificarea unor nivele de nedominare:
 - ❑ Elementele nedominate din populația curentă formează primul nivel de nedominare
 - ❑ Elementele nedominate din populația rămasă după eliminarea elementelor primului nivel, formează al doilea nivel s.a.m.d.

- ❑ Pentru **stimularea diversității**:
 - ❑ NSGA-I folosește funcție de partajare
 - ❑ NSGA-II folosește factor de aglomerare

Exemple: Niched Pareto GA (NPGA)

- ❑ Propus în: 1994

- ❑ Folosește selecție de tip turneu bazată pe nedominare în raport cu un eșantion din populație:
 - ❑ Selectează aleator două elemente din populație (x_1 și x_2)
 - ❑ Selectează aleator un eșantion de q elemente (din populația curentă având m elemente)
 - ❑ Verifică dacă x_1 și x_2 sunt dominate de către elemente din eșantion
 - ❑ Dacă x_1 e nedominat dar x_2 e dominat se selectează x_1
 - ❑ Dacă x_2 e nedominat dar x_1 e dominat se selectează x_2
 - ❑ Dacă ambele sunt dominate sau ambele nedominate atunci se folosește o funcție de partajare pentru discriminare

Example: Strength Pareto EA (SPEA)

- ❑ Propus în: 1999
- ❑ Folosește o arhivă de elemente nedominate actualizată pe parcursul evoluției
 - ❑ Pentru fiecare element din arhivă se întreține un indicator de dominare (strength factor) proporțional cu numărul de elemente pe care acesta le domină
 - ❑ Pentru fiecare element din populație scorul se calculează ca fiind invers proporțional cu suma indicatorilor elementelor din arhivă care îl domină
- ❑ SPEA 2 aduce câteva modificări:
 - ❑ La evaluarea unui element se iau în considerare atât elementele care îl domină cât și cele de către care este dominat
 - ❑ Utilizează o tehnică de estimare a densității frontului (bazată pe distanța la cel mai apropiat vecin)
 - ❑ Folosește o tehnică de trunchiere a arhivei

Exemple: tehnici bazate pe descompunere (MOEA/D)

- ❑ **Specific:** transformarea problemei de optimizare multicriterială în mai multe probleme de optimizare unicriterială
- ❑ MOEA/D a fost propus în 2007 și este la ora actuală printre cei mai eficienți algoritmi (în special în cazul a mai mult de două criterii)
- ❑ **Idee MOEA/D:**
 - ❑ Se consideră N vectori cu ponderi (w_1, w_2, \dots, w_N) ; fiecare vector conține r valori ($r = nr$ criterii)
 - ❑ Se cunosc valori țintă (referință) pentru fiecare dintre cele r criterii
 - ❑ Pentru fiecare vector w_i se rezolvă problema de optimizare unicriterială corespunzătoare:
 - ❑ Minimizează $\max_j \{w_{ij} |f_j(x) - z_j^*|\}$ (criteriu de tip Cebisev)

Exemple: tehnici bazate pe descompunere (MOEA/D)

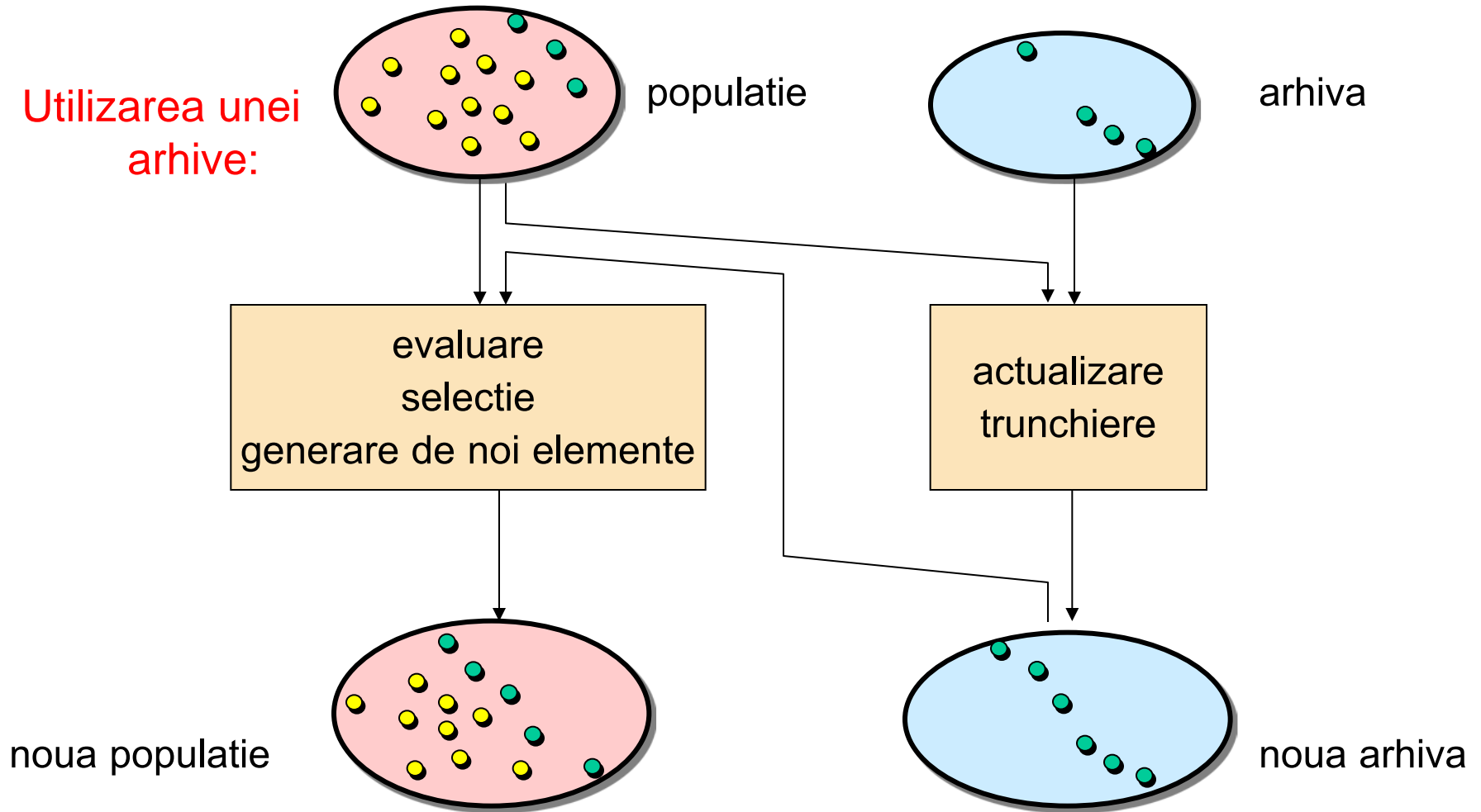
- ❑ Pentru implementarea unui alg de tip MOEA/D trebuie specificate:
 - ❑ N = număr subprobleme
 - ❑ N vectori cu ponderi (cât mai uniform distribuiți)
 - ❑ T = numărul de vectori din vecinătatea fiecărui vector pondere

Exemple: tehnici bazate pe descompunere (MOEA/D)

- Structura algoritm:
 - Inițializează N populatii; inițializează $z=(z_1, \dots, z_r)$ valori țintă; inițializează arhiva cu mulțimea vidă
 - La fiecare generație, pentru fiecare subproblemă i :
 - Selectează două elemente x_k și x_l din populațiile subproblemelor aflate în vecinătatea subproblemei i
 - Combină x_k cu x_l și construiește y
 - Actualizează z (dacă e cazul)
 - Inlocuiește vecinii cu y dacă acesta este mai bun
 - Elimină elementele dominate din arhivă și adaugă y la arhivă dacă este nedominat

Utilizarea unei arhive

Utilizarea unei
arhive:



Sumar

1984	first EMO approaches
1990	dominance-based population ranking
	dominance-based EMO algorithms with diversity preservation techniques
1995	attainment functions
	elitist EMO algorithms preference articulation convergence proofs
2000	test problem design quantitative performance assessment
	uncertainty and robustness running time analyses multiobjectivization
	quality measure design
2010	MCDM + EMO quality indicator based EMO algorithms
	many-objective optimization statistical performance assessment

D.Brockhoff - Tutorial MOEA – GECCO 2013

Resurse

<http://www.lania.mx/~ccoello/EMOO/EMOObib.html>

(2940 referinte – septembrie 2007)

(3519 referinte – noiembrie 2008)

(4388 referinte – octombrie 2009)

(4861 referinte – februarie 2010)

Din 2010 EMOObib nu a mai fost actualizată

(7806 referinte – martie 2013) – tutorial on MOEA – GECCO 2013