

Clase de probleme de optimizare

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

Optimizare cu restricții

- Majoritatea problemelor din lumea reală au restricții
- Tipuri de restricții

- Constrângeri de domeniu

$$a_j \leq x_j \leq b_j \text{ pentru } j = 1, n$$

- Constrângeri inegalitate

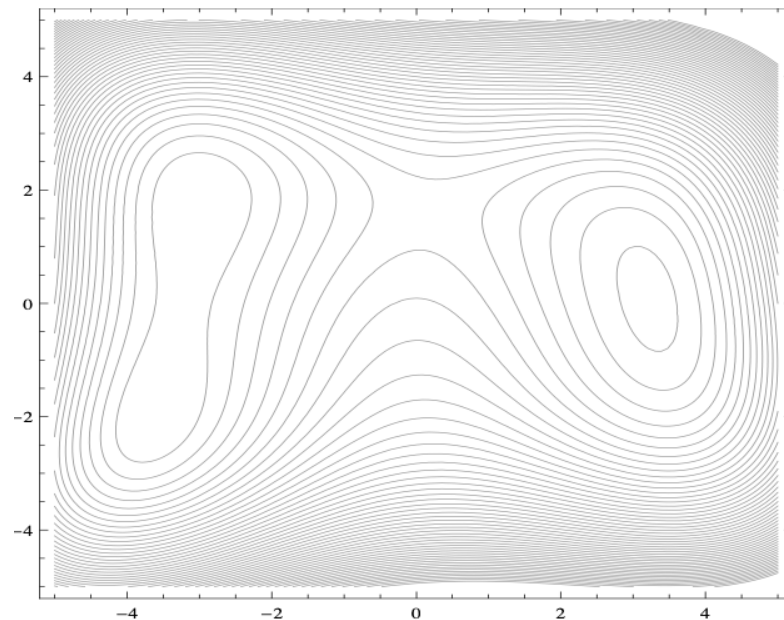
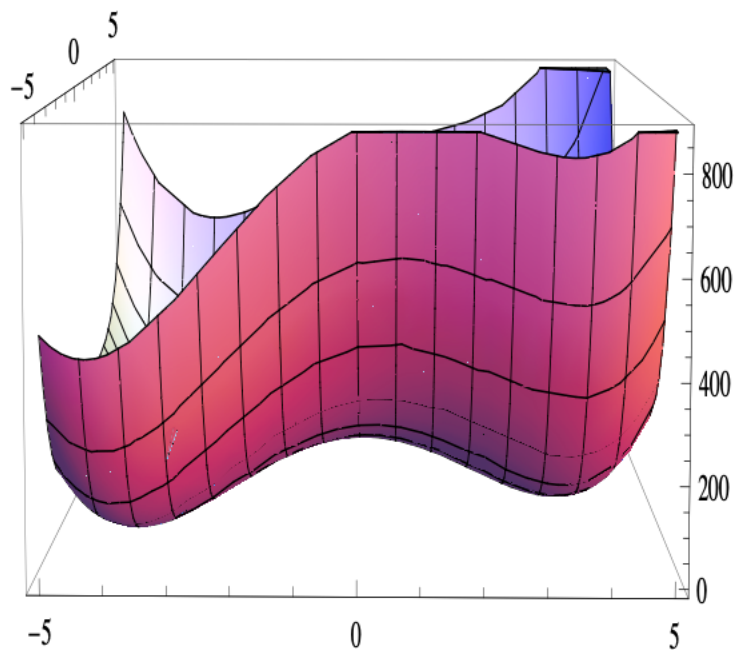
$$g_i(x) \leq 0 \text{ pentru } i = 1, n$$

- Constrângeri egalitate

$$h_i(x) = 0 \text{ pentru } i = 1, q \text{ (de obicei se transformă în } |h_i(x)| \leq \epsilon$$

Motivație

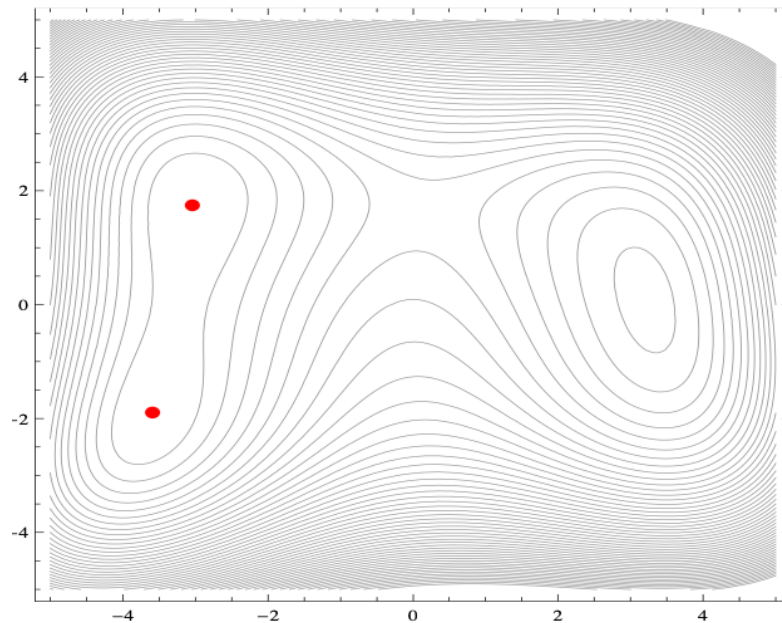
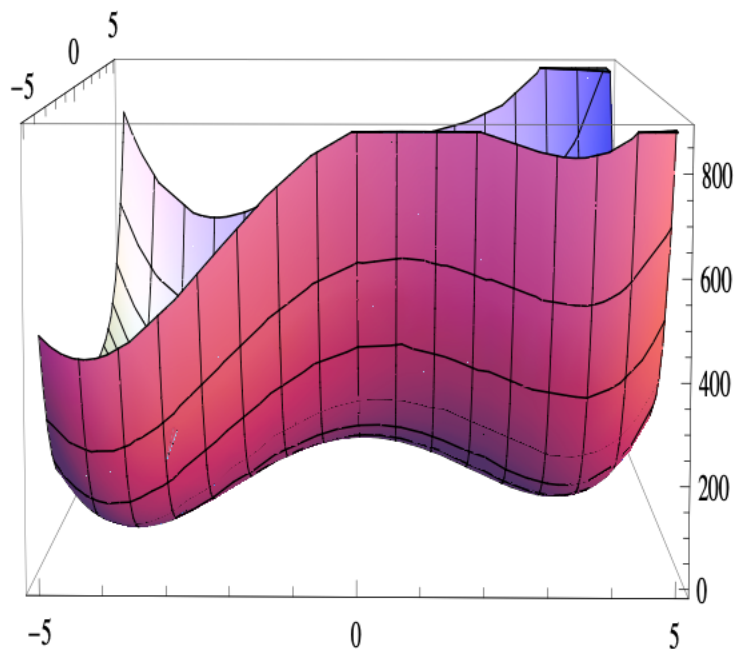
Exemplu: $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2)^2$, $x_1, x_2 \in [-5, 5]$



Unde se afla optimul?

Motivație

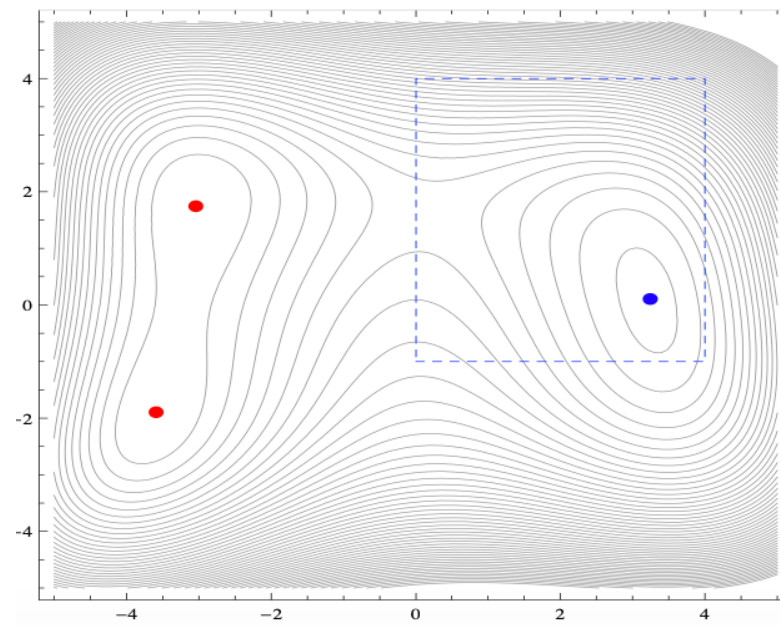
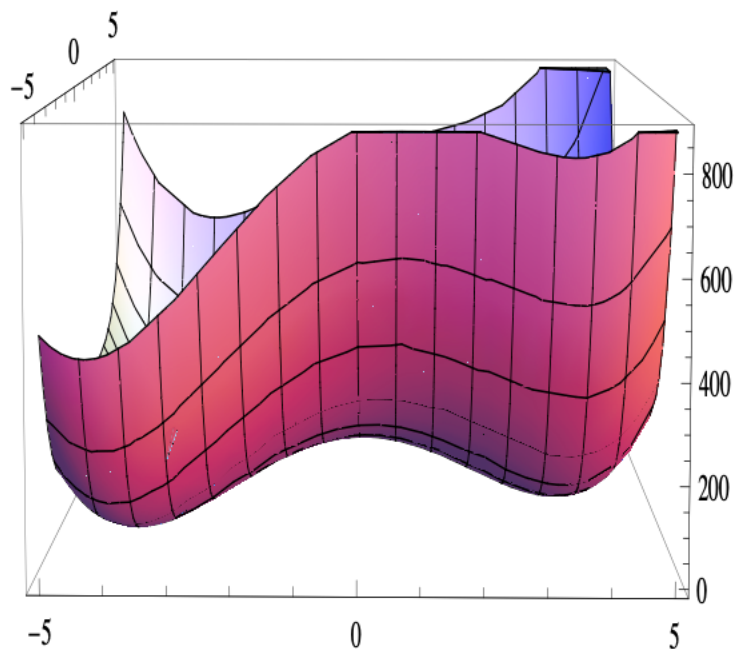
Exemplu: $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2)^2$, $x_1, x_2 \in [-5, 5]$



Dar dacă retrângem domeniul de fezabilitate la $[0, 4] \times [-5, -5]$

Motivație

Exemplu: $f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2)^2$, $x_1, x_2 \in [-5, 5]$



Punctul de optim este cel albastru și căutarea ar trebui redirecționată spre acesta.

Abordări

- Metode generale de tratare a metodelor de tratare a restricțiilor
 - Funcție de penalizare
 - Reguli de fezabilitate
 - Stochastic ranking
- Metode particulare de rezolvare a restricțiilor legate de domeniul de definire
 - Reșantionare (resampling)
 - Reinițializare aleatoare
 - Proiecție
 - Efecție

Abordări

- Principale abordări:
 - Căutarea numai în regiunile fezabilă (de exemplu, se începe cu un element fezabil și se păstrează constrângerile satisfăcute)
 - destul de ușor pentru constrângerile legate de domeniul de definire
 - pentru constrângerile generale, poate fi dificil chiar găsirea pozițiilor inițiale fezabile
 - Permiterea căutării în afara regiunii fezabile, dar favorizarea elementele fezabile sau aproape fezabile
 - Întrebare: Cum se poate decide că un element este aproape fezabil?
 - Răspuns: Prin estimarea cantității de constrângeri încălcate

Determinarea gradului de încălcare a restricțiilor

- Numărul de constrângeri încălcate
 - Nu respectă regiunile fezabile
- Gradul de încălcare

$$\phi(x) = \sum_{i=1}^p \max\{0, g_i(x)\} + \sum_{i=1}^q |h_i(x)|$$

- $\phi(x) = 0$ constrângerile sunt satisfăcute
- Valori mici pentru $\phi(x)$ corespund la elemente aflate ”aproape” de regiunile fezabile
- Poate fi interpretat ca un criteriu de optimizare care să influențeze selecția elementelor $\Rightarrow$ ghidând căutarea spre regiuni fezabile

Metoda penalizări funcție obiectiv

- Penalizarea soluțiilor nefiababile prin creșterea valorii funcției obiectiv bazată pe cantitatea de încălcare a constrângerii

$$F(x) = f(x) + \sum_{i=1}^p \alpha_i \max\{0, g_i(x)\} + \sum_{i=1}^q \beta_i |h_i(x)|$$

- Avantaje: ușor de implementat
- Dezavantaje: sensibilă la valorile α_i, β_i care depind de problemă

Reguli de fezabilitate.

Abordarea Deb

- Ideea
 - Folosirea unei funcții obiectiv separate (f) și a unei valori de penalitate (gradul de încălcare a restricțiilor ϕ când se compară două elemente

X este mai bun decât X'

- X și X' sunt ambele fezabile și $f(X) < f(X')$
- X este fezabil și X' nu este fezabil
- X și X' sunt ambele nefezabile și $\phi(X) < \phi(X')$

Reguli de fezabilitate.

Abordarea Deb

- Avantaje
 - ușor de implementat și de asociat cu diverși algoritmi de căutare
 - nu necesită parametri
- Dezavantaje
 - separarea constrângerilor și funcției obiective poate duce la pierderea diversității (deoarece abordarea favorizează puternic soluțiile fezabile)
 - Soluție: utilizați mecanisme de îmbunătățire a diversității (de exemplu, elemente aleatorii)
 - Combinarea încălcărilor constrângerii într-o singură funcție ($\phi(x)$) poate duce la pierderea particularității ale fiecăreia dintre constrângeri
 - Soluție: utilizați o abordare de clasificare Pareto peste valorile de încălcare a constrângerii calculate individual pe constrângere

Stochastic ranking

- Ideea
 - decide aleatoriu ce criteriu de selecție să folosească (funcția obiectivă sau funcția de penalizare)
 - în unele cazuri (decizie aleatorie) două soluții sunt comparate pe baza numai funcția obiectivă, chiar dacă nu sunt ambele fezabile

x este mai bun decât x'

$$\begin{cases} ((\phi(x) = \phi(x') = 0) \text{ or } (rand(0,1) < P_f)) \text{ and } (f(x) < f(x')) \\ \phi(x) < \phi(x') \end{cases}$$

Stochastic ranking

- Avantaje
 - limitează pierderea diversității (prin acceptarea candidaților promițători, dar imposibili)
- Dezavantaje
 - necesită specificarea unui parametru (P_f) - comportamentul algoritmului poate fi sensibil la valoarea lui P_f (o valoare utilizată în documente: $P_f = 0,45$ [T.Runarsson, X. Yao- Stochastic Ranking for Constrained Evolutionary Optimization, IEEE TEvC, 2000])

Constrângeri de domeniu

- Constrângeri de domeniu
: $a_j \leq x_j \leq b_j$ pentru $j = 1, n$
- Scop
 - repararea soluțiilor nefezabile
- Caracteristicile metodei de reparație care urmează să fie analizate:
 - Păstrează unele informații din elementul nefezabil?
 - Păstrează caracteristicile procesului de căutare (de exemplu, favorizând doar unele subregiuni ale regiunii fezabile)?
 - Variante: reeșantionare, reinițializarea aleatorie, proiecție, reflecție

Resampling

- Ideea principală
 - Ignorarea elementului nefezabil și generarea unui nou element selectând părinți noi sau alte valori ale unor parametri de control
 - Reesantionarea se poate face la nivelul componentelor sau la nivelul întregului vector
- Avantaje
 - Implementare ușoară (generarea repetată de elemente noi până la obținerea unuia posibil)
 - Păstrează caracteristicile strategiei de căutare (fără părtinire specifică)
- Dezavantaje
 - Generarea repetată de noi candidați ar putea fi costisitoare, în special în cazul în care întregul vector este reconstruit

Reinițializarea aleatoare

- Ideea principală
 - Componentele care încalcă constrângerile sunt reinitializate aleatoriu în domeniu
 - Pierde direcția anterioară de căutare (cel puțin pentru componente reinitializate)
- Avantaje
 - Implementare ușoară și costuri mici
 - Dacă se bazează pe o distribuție uniformă, atunci nu introduce o direcție specificată
 - Crește diversitatea populației (ajută la evitarea convergenței premature)
- Dezavantaje
 - Ar putea încetini convergența

Proiecția

- Ideea principală
 - Componentele care încalcă constrângerile sunt înlocuite cu cea mai apropiată margine
 - Păstrează direcția de căutare anterioară
- Avantaje
 - Implementare ușoară și costuri mici
 - Util atunci când optimul este la limită
- Dezavantaje
 - Introduce o direcție în căutare, concentrându-se pe graniță
 - Pentru unii operatori evolutivi, probabilitatea de încălcare rămâne mare, adică
 - regula de reparare joacă un rol important în procesul de căutare
 - S-ar putea reduce diversitatea populației

Reflexia

- Pentru fiecare componentă care încalcă limitele se iterează:

- $$x_j = \begin{cases} b_j - (x_j - b_j), & \text{daca } x_j > b_j \\ a_j + (x_j - a_j), & \text{daca } x_j < a_j \end{cases}$$

- Avantaje

- Poate crește diversitatea

- Dezavantaje

- Pentru unii operatori evolutivi, probabilitatea de încălcare rămâne mare, adică regula de reparație joacă un rol important în procesul de căutare
 - Pierde informațiile din direcția de căutare