

Exercițiul 2 (Cutting-Plane Gomory) + Exercițiul 3 (test: Colorare graf) — explicație de cod, variabile și pași

1. Context și idee

Implementarea urmează pseudo-codul cerut pentru Cutting-Plane:

rezolvă relaxarea LP \rightarrow dacă soluția e integrală, STOP \rightarrow altfel generează o tăietură Gomory \rightarrow ada

Pentru test, algoritmul este aplicat pe problema de minimizare din Ex. 3 (colorarea unui graf), pe care o transformăm într-o problemă de maximizare LP ca să folosim simplex-ul.

2. Notări și structuri de date

2.1. Structura LP în formă standard

Lucrăm cu o problemă LP în formă standard:

$$\max c^\top x \quad \text{s.t. } Ax = b, x \geq 0.$$

În cod, un astfel de LP este reținut în structura:

`StdLP(A, b, c, int_idx, names)`.

Semnificația câmpurilor:

- `A`: matricea coeficienților (listă de liste), dimensiune $m \times n$.
- `b`: vectorul termenilor liberi, dimensiune m .
- `c`: vectorul coeficienților funcției obiectiv, dimensiune n .
- `int_idx`: lista indicilor variabilelor care trebuie să fie întregi (în test: binare).
- `names`: nume simbolice ale variabilelor (util pentru interpretare/debug).

2.2. Variabilele numerice auxiliare

- `EPS`: toleranță numerică pentru comparații cu zero (erori de rotunjire).
- `fpart(x)`: partea fracționară $\{x\} = x - \lfloor x \rfloor$.
- `is01(x)`: test dacă x este aproximativ 0 sau 1.

3. Exercițiul 2: Implementarea Cutting-Plane (Gomory)

3.1. TwoPhaseMethod(P) din pseudo-cod

Pseudo-codul cere o metodă de două faze (Two-Phase) pentru rezolvarea LP-ului:

$$(x^0, f^0) = \text{TwoPhaseMethod}(P).$$

În cod, aceasta este implementată prin clasa `TwoPhase`, care construiește un **tableau** de simplex și rulează:

1. **Faza I** (fezabilitate): minimizează suma variabilelor artificiale (echivalent maximizează $-\sum a_i$).
2. **Faza II** (optimizare): maximizează funcția obiectiv originală $c^\top x$.

Tableau și bază:

- `tab`: tableau-ul simplex, cu dimensiune $(m + 1) \times (n + m + 1)$.
- `basic[i]`: indicele variabilei de bază (basic variable) pentru rândul i (constrângerea i).
- `art`: indicii variabilelor artificiale introduse în Faza I.

Metode-cheie:

- `_enter()`: alege coloana de intrare (cost redus negativ).
- `_leave(ec)`: testul raportului minim (rândul care iese din bază).
- `_pivot(r, c)`: pivotare Gauss–Jordan pe poziția (r, c) .
- `solve()`: execută Faza I + Faza II.
- `sol()`: extrage soluția x și valoarea obiectivului din tableau.

3.2. Testul de integritate $x^0 \in \mathbb{Z}^n$

Pseudo-cod:

dacă $x^0 \in \mathbb{Z}^n$ atunci returnează x^0 .

În cod:

```
all(is01(x[j]) for j in int_idx).
```

3.3. Alegerea unei variabile fracționare și a unui rând pentru tăietură

Tăietura Gomory se extrage dintr-un rând de tableau asociat unei variabile de bază integrale, dar fracționare. Funcția:

```
pick_frac_row(int_idx)
```

returnează un index de rând `rr` astfel încât termenul liber b are parte fracționară nenulă.

3.4. Construirea tăieturii Gomory fracționare

Dacă un rând de tableau (în formă canonică) arată:

$$x_B + \sum_j a_j x_j = b,$$

atunci tăietura Gomory fracționară este:

$$\sum_j \{a_j\} x_j \geq \{b\}.$$

3.5. Adăugarea tăieturii în problemă

O tăietură are forma $\alpha^\top x \geq \beta$. Pentru forma standard $Ax = b$, o scriem:

$$\alpha^\top x - s = \beta, \quad s \geq 0.$$

La fiecare iterație reconstruim LP-ul mărit, adăugând câte o variabilă slack pentru fiecare tăietură.

4. Exercițiul 3: Test (problemă de minimizare) — modelul de colorare

4.1. Datele (graful) și formatul DIMACS .col

Fișierul .col conține p edge n m și muchii e u v . Se citește cu:

$$\text{read_col}(\text{path}) \Rightarrow (n, \text{edges}).$$

4.2. Variabilele modelului de colorare

Luăm $k = n$ culori.

- $x_{u,j} \in \{0, 1\}$: nodul u primește culoarea j .
- $w_j \in \{0, 1\}$: culoarea j este utilizată.

4.3. Constrângeri

$$\sum_{j=1}^k x_{u,j} = 1, \quad \forall u, \quad x_{u,j} + x_{v,j} \leq w_j, \quad \forall (u, v) \in E, \quad \forall j.$$

Inegalitățile sunt transformate în egalități cu slack pentru forma standard.

4.4. Funcția obiectiv

$$\min \sum_{j=1}^k w_j \quad \Longleftrightarrow \quad \max \left(- \sum_{j=1}^k w_j \right).$$

La final:

$$\min \sum w_j = -(\text{obj_max}).$$

5. Mapare cod \leftrightarrow pseudo-cod

- construirea LP-ului: `build_coloring_std_lp`;
- loop-ul principal: `cutting_plane(lp0, imax)`;
- TwoPhaseMethod: `TwoPhase.solve()`;
- extragere soluție: `TwoPhase.sol()`;
- test integritate: `is01` pe `int_idx`;
- alegere rând fracționar: `pick_frac_row`;
- tăietură Gomory: `gomory_cut`;
- adăugare tăietură: `cuts.append((alpha,beta))`.

6. Compararea cu ILP (validare)

Pentru verificare, rezolvăm ILP cu PuLP+CBC și comparăm cu rezultatul Cutting-Plane. Diferențele evidențiază limita metodei Gomory pentru 0–1 ILP și motivează branch-and-cut în practică.

7. Ce trebuie să știi să spun oral

- forma standard $Ax = b, x \geq 0$;
- Two-Phase: Faza I (fezabilitate), Faza II (optimizare);
- tăietura Gomory $\sum \{a_j\}x_j \geq \{b\}$;
- transformarea minimizare \rightarrow maximizare;
- interpretarea variabilelor $x_{u,j}$ și w_j pentru colorare.