Planificare cu minimizarea întârzierii maxime

Bibliografie

1. **Jon Kleinberg, Éva Tardos -** Algorithm Design**,** 2005 Addison-Wesley Professional

Se consideră o mulțime de n activități care trebuie planificate pentru a folosi o aceeași resursă. Această resursă poate fi folosită de o singură activitate la un moment dat. Pentru fiecare activitate i se cunosc durata l_i şi termenul limită până la care se poate executa t_i (raportat la ora de început 0). Dorim să planificăm aceste activități astfel încât întârzierea fiecărei activități să fie cât mai mică. Mai exact, pentru o planificare a acestor activități astfel încât activitatea *i* este programată în intervalul de timp [s_i, f_i), definim întârzierea activității *i* ca fiind durata cu care a depășit termenul limită: $\boldsymbol{p_i = \max\{0, f_i - t_i\}}$.

Întârzierea planificării se definește ca fiind **maximul întârzierilor activităților**:

Exemplu. Pentru n = 3 şi $l_1 = 1$, $t_1 = 3$ / $l_2 = 2$, $t_2 = 2$ / $l_3 = 3$, $t_3 = 3$

o soluţie optimă se obţine dacă planificăm activităţile în ordinea 2, 3, 1; astfel:

- activitatea 2 în intervalul [0, 2) – întârziere 0
- activitatea 3 în intervalul [2, 5) – întârziere 5 – t_3 = 5 - 3 = 2
- activitatea 1 în intervalul [5, 6) – întârziere 6 – t_1 = 6 - 3 = 3

Întârzierea planificării este max{0, 2, 3}=3 $\boldsymbol{P = \max\{p_1, p_2, ..., p_n\}}$

a) Să se determine o planificare a activităţilor date care să aibă întârzierea P minimă. Se vor afişa pentru fiecare activitate intervalul de desfăşurare şi întârzierea – **O(n log n)**

date.in	date.out
3 1 3 2 2 3 3	activitatea 2: intervalul 0 2 intarziere 0 activitatea 3: intervalul 2 5 intarziere 2 activitatea 1: intervalul 5 6 intarziere 3 Intarziere planificare 3
3 4 10 5 6 2 4	activitatea 3: intervalul 0 2 intarziere 0 activitatea 2: intervalul 2 7 intarziere 1 activitatea 1: intervalul 7 11 intarziere 1 Intarziere planificare 1

b) Este corect un algoritm Greedy bazat pe următoarea idee: planificăm activităţile în ordine crescătoare în raport l_i (adică în raport cu durata)? Justificaţi.

c) Este corect un algoritm Greedy bazat pe următoarea idee: planificăm activităţile în ordine crescătoare în raport cu diferenţa t_i-l_i (adică în raport cu timpul maxim la care trebuie să înceapă activitatea i pentru a respecta termenul limită)? Justificaţi.

Soluție.

O *soluţie* (stabilirea ordinii în care se vor planifica activitățile) înseamnă o permutare $\pi \in S_n$

Putem considera doar soluții în care **activitățile sunt planificate fără a lăsa momente de timp neutilizate** (libere) între activități, altfel activitățile programate după intervalele de timp libere ar putea fi programate mai devreme fără a lăsa astfel de intervale şi întârzierea noii planificării va fi cel puţin la fel de mică. Astfel, timpul de finalizare f_u pentru o activitate u va fi egal cu lungimea ei plus lungimea activităților programate înainte.

Pentru u=1,..,n notăm cu $p_u = \max\{0, f_u - t_u\}$ întârzierea unei activități u în planificarea corespunzătoare permutării π și cu $P(\pi) = \max_{1 \le i \le n} p_i$ întârzierea planificării π.

Dorim să determinăm acea permutare π pentru care P(π) este minim.

După ce criteriu alegem prima activitate pe care o programăm?

Posibile încercări:

- să aibă lungime cât mai mică pentru a se termina repede – **nu**, deoarece poate avea termen limită mare, deci poate fi programată mai târziu, de exemplu: dacă avem două activități cu l_1=2, t_1=10 și l_2=6, t_2=7, ordinea optimă este 2, 1, planificarea având întârzierea 0
- să aibă diferența t_i - l_i cât mai mică (cea care trebuie începută cât mai repede pentru a nu depăși termenul limită) - **nu**, deoarece poate avea lungime mare și activitățile de după întârzie mai mult; de exemplu: dacă avem două activități cu l_1=7, t_1=7 și l_2=2, t_2=3, ordinea optimă este 2, 1, planificarea având întârzierea 3

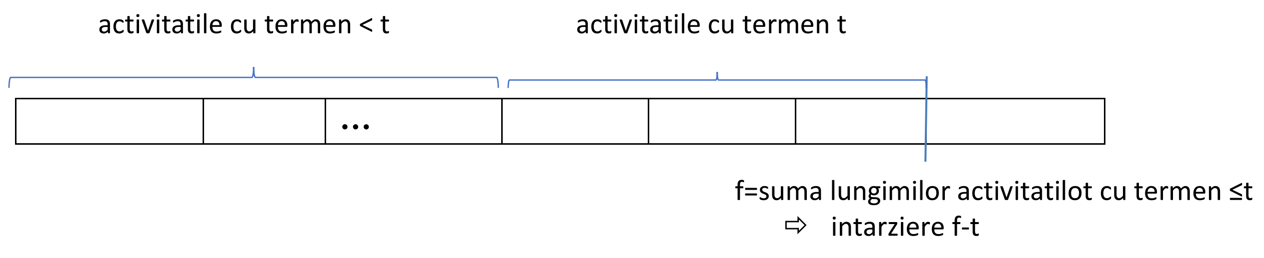
Strategia Greedy corectă – alegem la fiecare pas activitatea încă neplanificată care are termenul limită cel mai mic**.** Această strategie se reduce la a planifica activitățile în ordine crescătoare după termenul limită t_i**.**

Corectitudine

Renumerotăm activitățile astfel încât $t_1 \le t_2 \le \ldots \le t_n$. Astfel, soluția greedy corespunde permutării identice id. Intervalul de desfăşurare a activității k conform soluţiei greedy este $[l_1+\ldots+l_{k-1}, l_1+\ldots+l_k)$.

Presupunem prin absurd că soluţia greedy (permutarea id) nu este optimă.

Fie σ o permutare optimă. Dacă σ ar diferi de id doar prin ordinea în care alege activitățile care au același termen limită, atunci am avea T(σ) = T(id) deoarece dintre toate activitățile cu același termen limită t ultima programată are momentul de finalizare egal cu suma lungimilor activităților cu termen limită mai mic sau egal cu t, deci întârzierea ei este aceeași indiferent de ordinea în care executăm activitățile cu același termen limită.



Deoarece id nu este optimă, rezultă că σ conţine cel puţin o inversiune (i,j) cu $i < j$ şi $t_{\sigma(i)} > t_{\sigma(j)}$ (fără observația anterioară am fi putut considera ca inversiune o pereche $i < j$ şi şi $t_{\sigma(i)} \ge t_{\sigma(j)}$ și în caz de egalitate σ(i) > σ(j) și am fi obținut tot o contradicție).

Fie σ o permutare optimă cu număr minim de inversiuni. Există o inversiune în σ între elemente aflate pe poziții consecutive i și i+1, adică există i cu $t_{\sigma(i)} > t_{\sigma(i+1)}$ (dacă $t_{\sigma(1)}, \dots, t_{\sigma(n)}$ nu este ordonat crescător, există două elemente alăturate care nu sunt în această ordine).

Considerăm τ permutarea obţinută din σ interschimbând elementele de pe poziţiile i şi i+1 (permutarea τ va avea astfel mai puţine inversiuni, deci va fi mai asemănătoare cu soluţia greedy id decât σ).

$f'_u = f_v$

Planificare σ:	σ(1)	σ(2)	…σ(i-1)	u = σ(i)	**v = σ(i+1)**	σ(i+2)…
Planificare τ	**σ(1)**	**σ(2)**	**…σ(i-1)**	**v = σ(i+1)**	**u = σ(i)**	**σ(i+2)…**

Comparăm întârzierea corespunzătoare permutării τ cu cea a soluţiei optime σ (pentru a arăta că τ are întârzierea mai mică sau egală cu a lui σ, dar are mai puţine inversiuni decât σ, ceea ce contrazice alegerea lui σ).

Deoarece τ(k) = σ(k) ∀k≠i,i+1, singurele activități care pot avea întârzieri diferite în σ și τ sunt u=σ(i) și v=σ(i+1). Deoarece activitatea v este programată în τ mai devreme decât este în σ, este suficient să mai comparăm întârzierea activității $u = \sigma(i)$ în τ cu P(σ)
Momentul de finalizare f'_u al activității u în τ este egal cu cel a momentul de finalizare f_v al activității v în τ (= $l_{\sigma(1)} + \cdots + l_{\sigma(i-1)} + l_{\sigma(i)} + l_{\sigma(i+1)}$). În plus, activitatea v are termen limită mai mic decât cel al lui u, deoarece $t_{\sigma(i)} > t_{\sigma(i+1)}$ (sau, cu notatiile u si v, $t_u > t_v$), deci întârzierea activității u în τ este mai mică sau egală cu cea a lui v în σ (deoarece se termină la același moment, dar u are termenul limită mai mare):

$$f'_u - t_u = f_v - t_u \leq f_v - t_v \leq p_v$$

Deci și întârzierea p'_ua lui u în τ este mai mică sau egală cu P(σ)
Rezultă

$$\mathrm{P}(\pi) \leq \mathrm{P}(\sigma)$$

ceea ce contrazice fie optimalitatea lui σ (dacă P(τ) < P(σ)), fie faptul că σ este permutarea optimă cu număr minim de inversiuni.