

# Soluție ”Load Balancing with Restrictions”

Ștefan Popescu

## Problemă

(15p) Se consideră un proiect format din  $n$  task-uri ce trebuie efectuate de către  $m$  mașini de calcul. Fiecare task poate fi procesat doar de către una din două mașini dintre cele  $m$ . Task-urile sunt caracterizate ca fiind un triplet de forma  $(T_i, x_i, y_i)$ , unde  $T_i$  este timpul necesar pentru a procesa task-ul  $i$ , iar  $x_i$  și  $y_i$  sunt indicii celor două mașini ce pot efectua task-ul  $i$  (celelalte  $m - 2$  mașini sunt incompatibile cu efectuarea taskului  $i$ ). Se dorește planificarea fiecărui task pe câte o mașină compatibilă cu task-ul astfel încât întregul proiect să se termine cât mai repede. (Altfel spus, se dorește minimizarea timpului de lucru a mașinii celei mai solicitate.)

### Cerințe

- Să se scrie problema anterioară sub forma unei **Probleme de Programare Liniară cu Numere Întregi** (en. *Integer Linear Programming Problem*). Apoi această problemă să fie *relaxată* și adusă sub forma unei **Probleme de Programare Liniară**. (10p)
- Folosindu-vă de Problemele de Programare Liniară descrise la punctul a), propuneți un algoritm 2-aproximativ pentru problema inițială. Justificați de ce algoritmul propus are factorul de aproximare 2. (5p)

### Notații și indicații:

$OPT$  - încărcătura mașinii celei mai solicitate în configurația optimă.

$ALG$  - încărcătura mașinii celei mai solicitate în urma algoritmului propus de voi.

$LP$ , respectiv  $ILP$  - expresiile ce trebuie maximizate sau minimizate pentru problemele voastre de programare liniară, respectiv programare liniară cu numere întregi.

Task-urile vor fi indexate cu variabile de forma ” $i, j, k$ ”. Mașinile vor fi indexate cu variabile de forma ” $q, p, r$ ”.

$Comp(q)$  - lista task-urilor compatibile cu mașina  $q$

Variabilele de tipul  $A_q^i$  vor indica dacă task-ul  $i$  este alocat mașinii  $q$  sau nu.

## Soluție

- minimizați  $ILP$  cu proprietatea că:

- \* pentru fiecare mașină  $q$  avem:  $\sum_{i \in Comp(q)} A_q^i * T_i \leq ILP$  (suma timpilor task-urilor alocate pe fiecare mașină nu poate depăși  $ILP$ )
- \*  $A_{x_i}^i + A_{y_i}^i = 1$  pentru orice  $i \in \{1, \dots, n\}$  (Un task trebuie programat în totalitate folosindu-ne de cele două mașini compatibile cu el)<sup>1</sup>

---

<sup>1</sup>Mulți ar tinde să scrie  $A_{x_i}^i + A_{y_i}^i \geq 1$ . Deși această restricție va furniza o soluție cu load optim, este posibil ca o activitate să fie programată de două ori, deoarece nu există o restricție.

- \* pentru fiecare  $(i, q) \in \{1, \dots, n\} \times \{1, \dots, m\}$   $A_q^i = 0$  dacă  $i \notin \text{Comp}(q)$  și  $A_q^i \in \{0, 1\}$  altfel.<sup>2</sup> (taskurile pot fi programate, adică  $A_q^i$  poate fi nenul, doar pe mașinile compatibile)

minimizăm  $LP$  cu proprietatea că:

- \* pentru fiecare mașină  $q$  avem:  $\sum_{i \in \text{Comp}(q)} A_q^i * T_i \leq ILP$  (suma timpilor task-urilor alocate pe fiecare mașină nu poate depăși  $ILP$ )
- \* pentru fiecare  $(i, q) \in \{1, \dots, n\} \times \{1, \dots, m\}$   $A_q^i = 0$  dacă  $i \notin \text{Comp}(q)$  și  $0 \leq A_q^i \leq 1$  altfel. (taskurile pot fi programate, adică  $A_q^i$  poate fi nenul, doar pe mașinile compatibile)
- \*  $A_{x_i}^i + A_{y_i}^i = 1$  pentru orice  $i \in \{1, \dots, n\}$  (Un task trebuie programat în totalitate folosindu-ne de cele două mașini compatibile cu el)

b) Știm că  $LP \leq ILP = OPT$ . Odată obținut  $LP$  putem face următoarea construcție: Pentru fiecare task  $i$ , alegem maximum dintre  $A_{x_i}^i$  și  $A_{y_i}^i$  și îl facem egal cu 1, respectiv pe celălalt îl facem egal cu 0. Valoarea maximă dintre cele două ( $A_{x_i}^i$  și  $A_{y_i}^i$ ) va fi măcar  $1/2$  deoarece suma lor este 1, asta conduce la faptul că încărcătura unei mașini în urma acestui artificiu va fi de cel mult de 2 ori mai mare decât inițial. Adică avem  $ALG \leq 2 * LP \leq 2 * ILP = 2 * OPT$ .

---

<sup>2</sup>La o primă vedere această constrângere ar fi acoperită de prima, anume că o activitate trebuie să fie realizată dintre una din cele două mașini. Dar, teoretic, fără această restricție pot fi planificate și duplicate ale unui task pe mașini incompatibile atâta timp cât nu se depășește load-ul maxim