Ανάθεση Εργασιών

Συνδυαστική Βελτιστοποίηση

Βασίλης Ζησιμόπουλος

Θεωρητική Πληροφορική
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Ανάθεση Εργασιών

- Minimize
  - κόστοι επικοινωνίας
  - κόστοι υπολογισμού
  - χρόνος

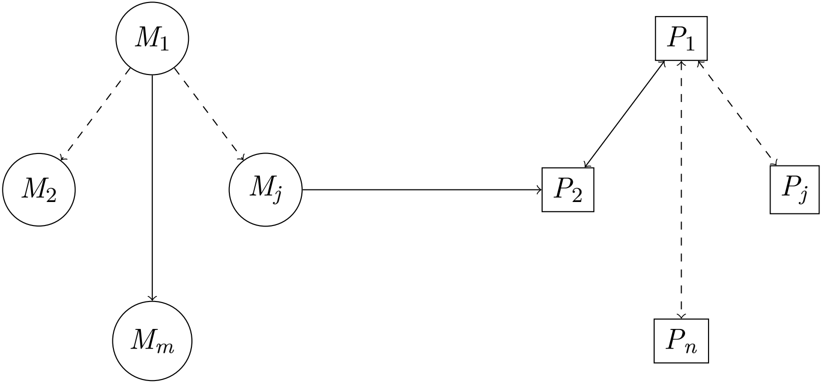
Ανάθεση Εργασιών

- Minimize
  - κόστοι επικοινωνίας
  - κόστοι υπολογισμού
  - χρόνος
- Περιορισμοί
  - στο χρόνο αποπεράτωσης
  - στην ποσότητα μνήμης
  - απαγόρευση
  - διπλασιασμός ή όχι εργασιών

Ανάθεση εργασιών σε επεξεργαστές

- συνδέσεις ανάμεσα στους επεξεργαστές
- σειρά ανάμεσα στις εργασίες
- διπλασιασμός εργασιών
- **min** κόστη επικοινωνιών + κόστη εκτέλεσης
- **min** χρόνος τερματισμού

Ανάθεση εργασιών σε επεξεργαστές



Μοντέλα

Μεταβλητές:

$$x_{ik} = \begin{cases} 1, & \text{αν } M_1 \text{ στο } P_k \\ 0, & \text{διαφορετικά} \end{cases}$$

Μοντέλα

Μεταβλητές:

$$x_{ik} = \begin{cases} 1, & \text{αν } M_1 \text{ στο } P_k \\ 0, & \text{διαφορετικά} \end{cases}$$

Αντικειμενική συνάρτηση:

- κόστος εκτέλεσης + κόστος επικοινωνίας
- χρόνος αποπεράτωσης

Μοντέλα

Κόστος εκτέλεσης:

$$
\begin{aligned}
q_{ik} &= \text{ κόστος της } M_1 \text{ στο } P_k \\
i &= 1, \ldots, m \\
k &= 1, \ldots, n
\end{aligned}
$$

Μοντέλα

Κόστος εκτέλεσης:

$$\begin{aligned} q_{ik} &= \text{ κόστος της } M_1 \text{ στο } P_k \\ i &= 1, \ldots, m \\ k &= 1, \ldots, n \end{aligned}$$

Κόστος επικοινωνίας:

$$\begin{aligned} C_{ijkl} = &\text{ η εργασία } i \text{ είναι στο } P_k \\ &\text{ η εργασία } j \text{ είναι στο } P_l \\ &\text{ κόστος ανάμεσα } i \text{ και } j \end{aligned}$$

$C_{ijkl} = \infty$ αν P_k και P_l δεν είναι συνδεμένοι
$C_{iikl} = 0, \ \forall i$
$C_{ijkk} = 0, \ \forall k$

Κόστος για έναν επεξεργαστή K

Κόστος εκτέλεσης:

$$cP_k(x) = \sum_{i=1}^{m} x_{ik} q_{ik}$$

Κόστος για έναν επεξεργαστή K

Κόστος εκτέλεσης:

$$cP_k(x) = \sum_{i=1}^{m} x_{ik} q_{ik}$$

Κόστος επικοινωνίας:

$$cc_k(x) = \sum_{i=1}^{m} x_{ik} * cc_{ki}(x)$$

όπου $cc_{ki}(x)$ το κόστος επικοινωνίας ανάμεσα στην εργασία i και όλες τις άλλες.

Κόστος για έναν επεξεργαστή K

$$
\begin{aligned}
cc_{ki}(x) &= \sum_{\substack{j=1 \\ j\neq i}}^{m} \text{κόστος για } k \text{ ανάμεσα σε } i \text{ και } j \\
&= \sum_{\substack{j=1 \\ j\neq i}}^{m} (\sum_{\substack{l=1 \\ l\neq k}}^{n} c_{ijkl} x_{jl})
\end{aligned}
$$

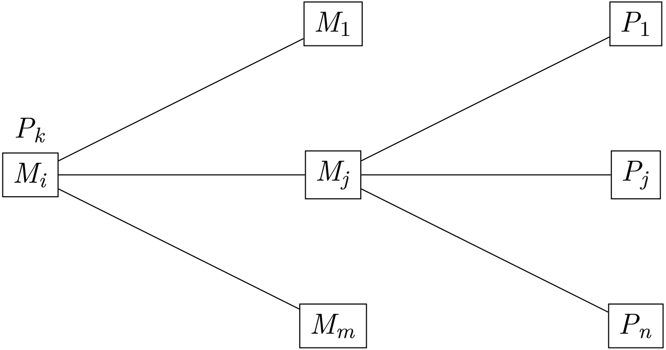
Κόστος για έναν επεξεργαστή K

$$cc_{ki}(x) = \sum_{\substack{j=1\\ j\neq i}}^{m} \text{κόστος για } k \text{ ανάμεσα σε } i \text{ και } j$$

$$= \sum_{\substack{j=1\\ j\neq i}}^{m} \Big(\sum_{\substack{l=1\\ l\neq k}}^{n} c_{ijkl} x_{jl} \Big)$$

$$cc_k(x) = \sum_{i=1}^{m} x_{ik} \sum_{\substack{j=1\\ j\neq i}}^{m} \sum_{\substack{l=1\\ l\neq k}}^{n} c_{ijkl} x_{jl}$$

$$= \sum_{i=1}^{m} \sum_{\substack{j=1\\ j\neq i}}^{m} \sum_{\substack{l=1\\ l\neq k}}^{n} c_{ijkl} x_{ik} x_{jl}$$


P_k
M_i
M_1
M_j
M_m
P_1
P_j
P_n

Γραμμικό πρόγραμμα - Αντικειμενική συνάρτηση

$$\min \sum_{k} \left(cP_k(x) + cc_k(x) \right)$$

Γραμμικό πρόγραμμα - Περιορισμοί

- Πάνω στις εργασίες

$$\sum_{k=1}^{n} x_{ik} = 1, \forall i = 1, \ldots, m$$

Γραμμικό πρόγραμμα - Περιορισμοί

- Πάνω στις εργασίες

$$\sum_{k=1}^{n} x_{ik} = 1, \forall i = 1, \ldots, m$$

- μνήμη, διαθέσιμος χρόνος

$$\sum_{i=1}^{m} s_{ik} x_{ik} \leq \bar{s}_k, \quad \forall k = 1, \ldots, n$$
$$\sum_{i=1}^{m} t_{ik} x_{ik} \leq \bar{t}_k, \quad \forall k = 1, \ldots, n$$

$\bar{s}_k$, $\bar{t}_k$ ο χώρος μνήμης και ο συνολικός χρόνος στο P_k αντίστοιχα
s_{ik}, t_{ik} ο χώρος μνήμης και ο χρόνος του M_i στο P_k αντίστοιχα

Άλλοι Περιορισμοί

- Περιορισμοί Εξαίρεσης

$$e_{ij} = \begin{cases} 1, & \text{αν } M_i \text{ και } M_j \text{ δεν μπορούν να είναι στον } P_k \ \forall k \\ 0, & \text{διαφορετικά} \end{cases}$$

$$e_{ij}(x_{ik} + x_{jk}) \leq 1$$

Άλλοι Περιορισμοί

- Περιορισμοί Εξαίρεσης

$$e_{ij} = \begin{cases} 1, & \text{αν } M_i \text{ και } M_j \text{ δεν μπορούν να είναι στον } P_k \ \forall k \\ 0, & \text{διαφορετικά} \end{cases}$$

$$e_{ij}(x_{ik} + x_{jk}) \leq 1$$

- Περιορισμοί Απαγόρευσης

$$b_{ik} = \begin{cases} 1, & \text{αν } M_i \text{ μπορεί να είναι στον } P_k \\ 0, & \text{διαφορετικά} \end{cases}$$

$$x_{ik} \leq b_{ik}$$

Γενική μορφή

$$\min \sum_{k} [\sum_{i=1}^{m} x_{ik} q_{ik} + \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} \sum_{\substack{l=1 \\ l \neq k}}^{n} c_{ijkl} x_{ik} x_{jl}]$$

$$\sum_{k=1}^{n} x_{ik} = 1, \forall i = 1, \ldots, m$$

$$\sum_{i=1}^{m} s_{ik} x_{ik} \leq \bar{s}_k, \forall k = 1, \ldots, n$$

$$\sum_{i=1}^{m} t_{ik} x_{ik} \leq \bar{t}_k, \forall k = 1, \ldots, n$$

$$e_{ij}(x_{ik} + x_{jk}) \leq 1, \forall ijk$$

$$x_{ik} = \{0, 1\}$$

Μέθοδοι

- Γράφοι
  - Ταίριασμα (matching)
  - Ροή (flow)
  - Επικάλυψη (covering), Διαμέριση (Partitioning)

Μέθοδοι

- Γράφοι
  - Ταίριασμα (matching)
  - Ροή (flow)
  - Επικάλυψη (covering), Διαμέριση (Partitioning)
- Ακέραιος Προγραμματισμός
  - Ροή (flow)
  - Επικάλυψη (covering), Διαμέριση (Partitioning)
  - Relaxation Decomposition

Μέθοδοι

- Γράφοι
  - Ταίριασμα (matching)
  - Ροή (flow)
  - Επικάλυψη (covering), Διαμέριση (Partitioning)
- Ακέραιος Προγραμματισμός
  - Ροή (flow)
  - Επικάλυψη (covering), Διαμέριση (Partitioning)
  - Relaxation Decomposition
- Διάδοση Περιορισμών

Πολυπλοκότητα

- Εμπειρική
  - γλώσσα, μεταγλωττιστής
  - μηχανή, προγραμματιστής
  - στιγμιότυπο (instance) προβλήματος

Πολυπλοκότητα

- Εμπειρική
  - γλώσσα, μεταγλωττιστής
  - μηχανή, προγραμματιστής
  - στιγμιότυπο (instance) προβλήματος
- Θεωριτική
  - βέλτιστη περίπτωση
  - μέση
  - χείριστη
  - φρόνηση
    - Simplex
    - αλγόριθμος Dantzig (47):
      m^3 - κυκλικότητα
    - κυκλικότητα(77)
    - Khachiyan (79) πολυωνυμικά

Μικρή Υπενθύμιση!

- $O(logn)$, $O(n^k)$, $O(2^n)$
- $f(n) = n^2 + 2^{28}n + 2^{30} \Rightarrow O(n^2)$

ΑΛΓΟ	Πολ/τα	M_1	M_2
1	$O(2^n)$	10	$10 + 4$
2	$O(n^2)$	32	$32 * 4$
3	$O(n)$	1024	$1024 * 16$
4	$O(logn)$	2^{1024}	$(2^{1024})^{16}$

Πίνακας: M_2 16 φορές γρηγορότερος από M_1

Λίγες πράξεις!

Knapsack:

$$\max \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j = \{0, 1\}$$

Λίγες πράξεις!

- ρητή απαρίθμηση $= 2^n$ λύσεις
- μια πρόσθεση $250 \times 10^{-9}s$ και 1 πολ/σμός $\simeq$ 3 προσθέσεις

n	10	20	50	1000
$time$	$10^{-2}s$	$20s$	$5 \times 10^{10}s!$	

Πίνακας: Χρόνοι ρητής απαρίθμησης

Ιστορία: Knapsack 1000 μεταβλητές

	Έτος	Χρόνος
Greenberry, Hegerich	70	4990
Fayard, Plateau	71	4502
Horowitz, Sahni	74	2770
Lauriere	76	860
Martello, Toth	78	203
Fayard, Plateau	79 - 82	83
Balas Zemel	80	105
Martello, Toth	81	100
Metairie, Polian, Fayard	88	36

Γενικός Αλγόριθμος

Πρόβλημα: B

$V(B)$: βέλτιση τιμή
D: χώρος λύσεων Να βρεθεί x^* τ.ω.

$$V(B) = cx^* = \max\{cx | x \in D\}$$

Γενικός Αλγόριθμος

1. Αρχικοποίηση

- εύρεση $\underline{V}(B) = c\underline{x}$ κάτω φράγμα της $V(B)$
- εύρεση $\bar{V}(B) = c\bar{x}$ άνω φράγμα της $V(B)$

Γενικός Αλγόριθμος

1. Αρχικοποίηση

- εύρεση $\underline{V}(B) = c\underline{x}$ κάτω φράγμα της $V(B)$
- εύρεση $\bar{V}(B) = c\bar{x}$ άνω φράγμα της $V(B)$

2. Ελλάτωση: του χώρου D

- διαγραφή μεταβλητών
- διαγραφή περιορισμών
- πρόσθεση περιορισμών

Γενικός Αλγόριθμος

3. Επίλυση του μειωμένου προβλήματος

- αλγεβρικές μέθοδοι
- δυναμικός προγραμματισμός
- μέθοδοι απαρίθμησης