

Combinatorial Optimization Problems

$\langle S, f \rangle$: an instance

S : finite set of solutions

$f: S \longrightarrow \mathbb{R}$

Cost Function (to be minimized or maximized)

Find $s^* \in S$ s.t.

$f(s^*) \leq f(s), \forall s \in S$ (Min)

or

$f(s^*) \geq f(s), \forall s \in S$ (Max)

Συνδυαστική Βελτιστοποίηση

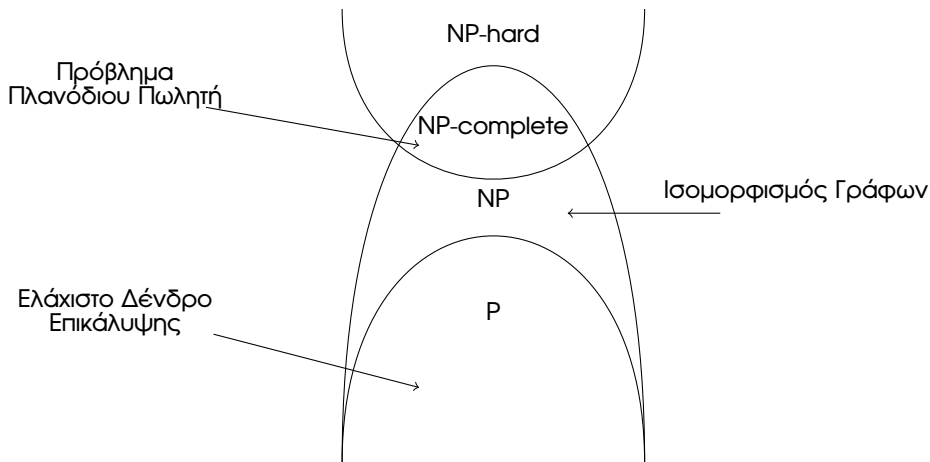
- Να βρεθεί η καλύτερη (μία;) λύση ανάμεσα σε ένα σύνολο λύσεων
- Πως να τυποποιηθεί;
 - φυσική γλώσσα
 - λογικές συνεπαγωγές
 - γράφοι
 - μαθηματικός προγραμματισμός
- Ακέραια λύση

Τα προβλήματα

- Περιήγηση Οχημάτων
- Εγκατάστασης
- Επένδυσης
- Ανάθεσης
- Χρονοπρογραμματισμός
- Κοπή-Τεμαχισμός

Προβλήματα γενικά NP-Hard

Κλάσεις Πολυπλοκότητας



Ο πλανόδιος πωλητής

(TSP)

$(n + 1$ πόλεις)

- Όλες οι μεταθέσεις
- Συντομότερος κύκλος Hamilton

$$\min \sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij}$$

$$\sum_{i=0}^n x_{ij} = 1, \quad j = 0, \dots, n$$

$$\sum_{j=0}^n x_{ij} = 1, \quad i = 1, \dots, n$$

$$x_{ij} = \begin{cases} 1 & \text{από } i \text{ στο } j, \\ 0 & \text{διαφορετικά.} \end{cases}$$

Σύνδεση τερματικών-Υπολογιστών

- Σύνολο m Τερματικών που θέλουν να συνδεθούν στο δίκτυο
- Σύνολο n Υπολογιστών του δικτύου
- Κάθε Τερματικό θέλει να συνδεθεί τουλάχιστον με a_i υπολογιστές
- Κάθε Υπολογιστής μπορεί να εξυπηρετήσει το πολύ b_j τερματικά
- Η σύνδεση τερματικού T_i με υπολογιστή Y_j έχει κόστος c_{ij}

- Μεταβλητές απόφασης

$$x_{ij} = \begin{cases} 1 & \text{Αν συνδεθεί το τερματικό } t_i \text{ με τον υπολογιστή } y_j \\ 0 & \text{Διαφορετικά} \end{cases}$$

- Περιορισμοί

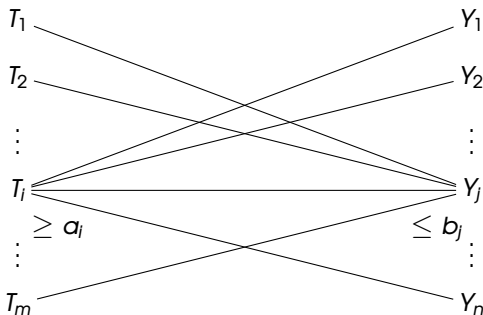
$$x_{i1} + x_{i2} + \dots + x_{in} \geq a_i, \forall i = 1, \dots, m$$

$$x_{1j} + x_{2j} + \dots + x_{mj} \leq b_j, \forall j = 1, \dots, n$$

- Αντικειμενική συνάρτηση

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

Διμερής γράφος ανάθεσης



Επίλυση με τεχνικές Θεωρίας Γράφων

$$\max q(x) = r^t x - \mu x^t A x$$

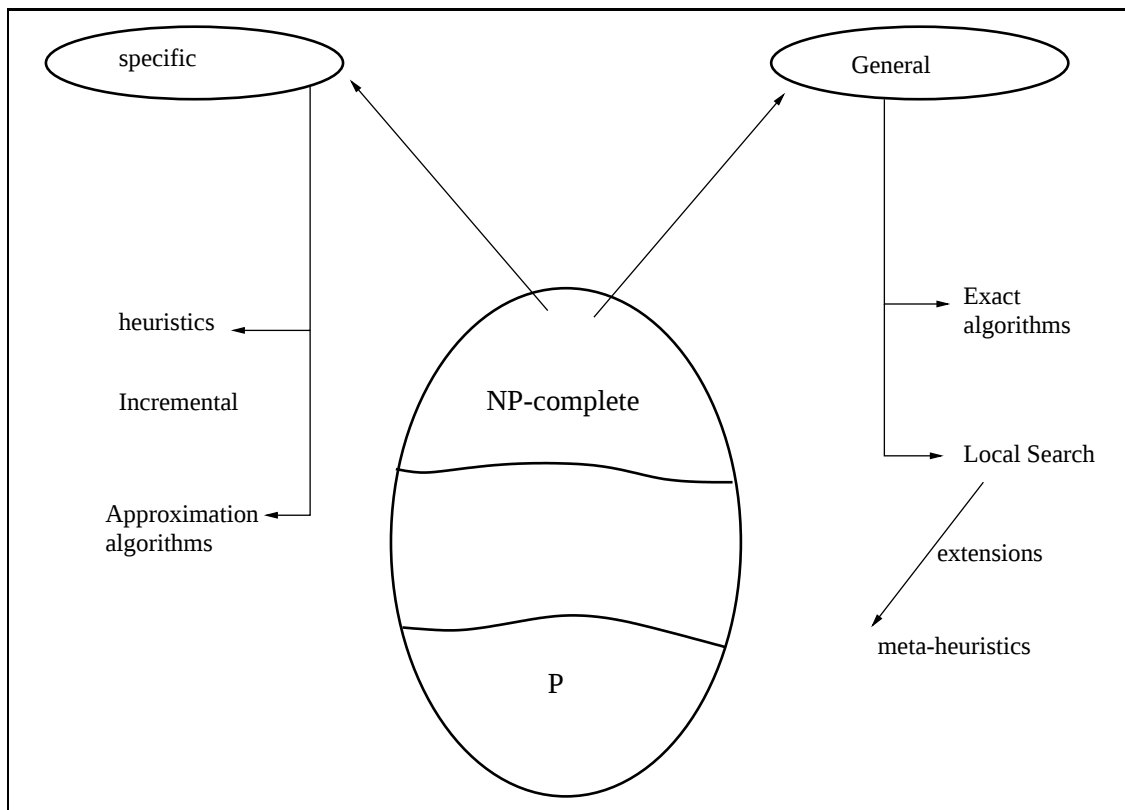
$$\text{s.t.} \quad \sum_{i=1}^n x_i = 1, i = 1, \dots, n$$
$$x_i \geq 0, i = 1, \dots, n$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \mu \text{ ένας συντελεστής βαρύτητας και } A \text{ ένας τετραγωνικός}$$

πίνακας $n \times n$, r διάνυσμα κολόνα $n \times 1$

(βέλτιστη διαχείριση χαρτοφυλακίων Markowitz, βραβείο Νόμπελ οικονομίας 1990).

The methods

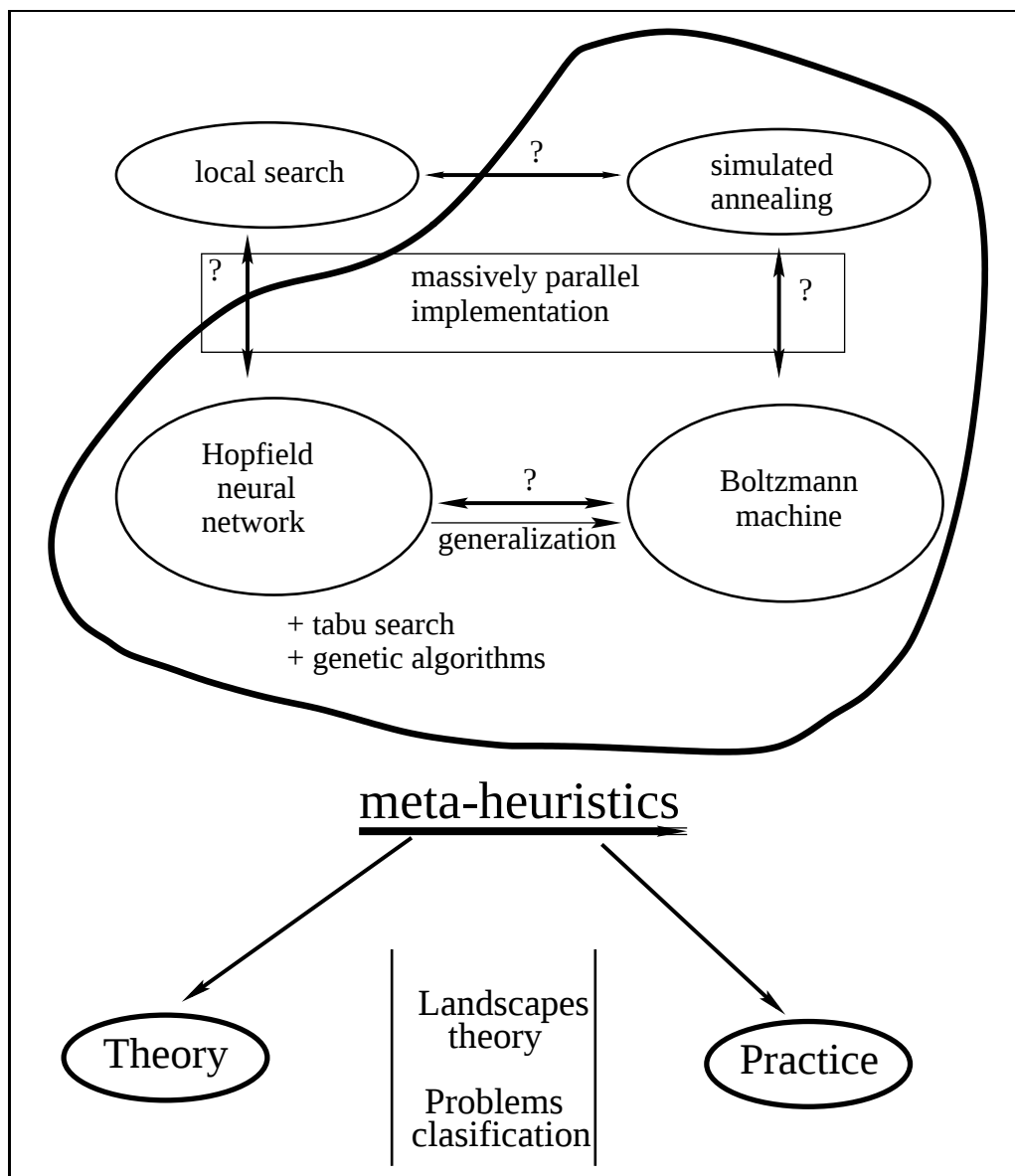


The class NP

The problems

- Graph Theory - Optimization
- Parallel architectures (task placement)
- Quadratic Assignment
- Interconnection Networks

The local search and its extensions



Exact Algorithms

- moderate size
- execution time?
- Parallelization?

Heuristics

- large sizes
- reasonable execution time
- approximation
- problem specific
- parallelization ?

Approximation Algorithms

- solutions quality guarantee
- full polynomial approximation schemes

Local Search Methods

- general methods
- approximation?
- local optimal!
- feasible initial solution?
- neighborhood structure?
- neighborhood's search?
- initial-solution-local optimum quality!

Simulated Annealing

- general technique
- escape from local optimal
- asymptotically optimal
- good solution quality
- execution time!
- approximation in finite time?

Neural Network Methods

- general methods
- escape from local optimal
- massively parallel methods
- flexible methods: add constraints code
- initial solution feasible not necessary
- good solutions
- approximation?

Ανάθεση Εργασιών

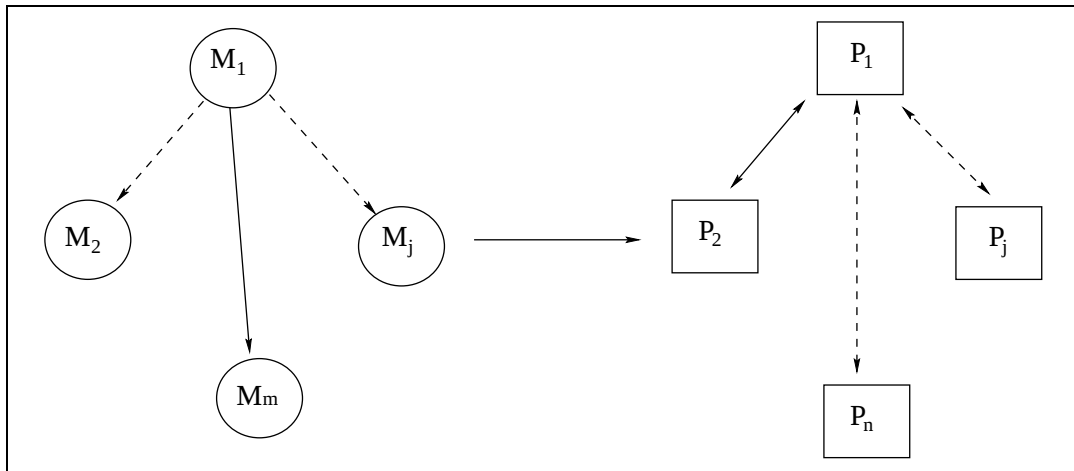
- **Minimize**

- κόστη επικοινωνίας
- κόστη 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

Οικογένειες Προβλημάτων

- $\min$ Κόστη επικοινωνιών + Κόστη ~~επικοινωνιών~~
- $\min$ Χρόνος τερματισμού

Μοντέλα

Μεταβλητές

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

Αντικειμενική Συνάρτηση

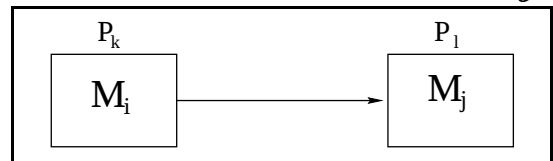
- κόστος εκτέλεσης + επικοινωνίες
- χρόνος αποπεράτωσης

Κόστος Εκτέλεσης:

$$\begin{aligned} q_{ik} &= \text{κόστος της } M_i \text{ στον } P_k \\ i &= 1, \dots, m \\ k &= 1, \dots, n \end{aligned}$$

Κόστος Επικοινωνίας:

C_{ijkl} = η εργασία i είναι στον P_k
η εργασία j είναι στον P_l
κόστος ανάμεσα i και j



- Αν P_k και P_l δεν είναι συνδεδεμένοι
 $\rightarrow C_{ijkl} = \infty$
- $C_{iikl} = 0 \quad \forall i$
 $C_{ijkk} = 0 \quad \forall k$

Κόστος για έναν επεξεργαστή K

- Κόστος εκτέλεσης

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

- Κόστος επικοινωνίας

$$cc_k(x) = \sum_{i=1}^m x_{ik} * \begin{array}{l} \text{Κόστος επικοινωνίας} \\ \text{ανάμεσα στην} \\ \text{εργασία } i \text{ και όλες} \\ \text{τις άλλες} \end{array}$$

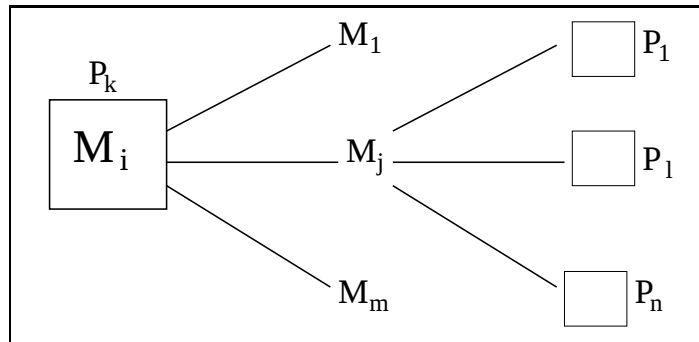
$$cc_k(x) = \sum_{i=1}^m x_{ik} * cc_{ki}(x)$$

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

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

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

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



Αντικειμενική συνάρτηση:

$$\min \sum_k (cP_k(x) + cc_k(x))$$

Περιορισμοί

- Πάνω στις εργασίες

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

- μνήμη, διαθέσιμος χρόνος

$$\left. \begin{array}{l} \bar{s}_k = \text{χώρος μνήμης} \\ \bar{t}_k = \text{συνολικός χρόνος} \end{array} \right\} = \text{διαθέσιμα στο } P_k$$

$$\left. \begin{array}{l} s_{ik} = \text{χώρος μνήμης} \\ t_{ik} = \text{χρόνος} \end{array} \right\} = \text{για } M_i \text{ στο } P_k$$

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

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

Άλλοι περιορισμοί

- Περιορισμοί εξαίρεσης

$$e_{ij} = \begin{cases} 1 & \text{αν } M_i \text{ και } M_j \text{ μπορούν,} \\ & \text{να είναι στον } P_k, \forall k \\ 0 & \text{διαφορετικά} \end{cases}$$
$$e_{ij} (x_{ik} + x_{jk}) \leq 1$$

- Περιορισμοί Απαγόρευσης

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

$\Rightarrow$ Διαγραφή μεταβλητών

Γενική Μορφή

$$\begin{aligned} \min \sum_k & \left[\sum_{i=1}^m x_{ik} q_{ik} + \sum_{i=1}^m \sum_{\substack{j=1 \\ j \neq i}}^m \sum_{l=1, l \neq k}^n c_{ijkl} x_{ik} x_{jl} \right] \\ \sum_{k=1}^n x_{ik} &= 1, \forall i = 1, \dots, m \\ \sum_{i=1}^m s_{ik} x_{ik} &\leq \bar{s}_k, \forall k = 1, \dots, n \\ \sum_{i=1}^m t_{ik} x_{ik} &\leq \bar{t}_k, \forall k = 1, \dots, n \\ e_{ij}(x_{ik} + x_{jk}) &\leq 1, \forall ijk \\ x_{ik} &= 0 \text{ ή } 1 \end{aligned}$$

Ανάθεση Εργασιών: οι μέθοδοι

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

Πολυπλοκότητα

- Εμπειρική:
 - γλώσσα, μεταγλωττιστής
 - μηχανή, προγραμματιστής
 - στιγμιότυπο (instance) προβλήματος
- Θεωρητική:
 - βέλτιστη περίπτωση
 - μέση
 - χειρίστη
 - φρόνηση
 - * simplex
 - * αλγόριθμος Dantzig(47):
 m^3 – κυκλικότητα
 - * κυκλικότητα (77)
 - * Khachiyan (79) πολυωνυμικά

Μικρή Υπενθύμιση !

- $O(\log n)$ $O(n^k)$ $O(2^n)$
όταν $n \rightarrow \infty$

$$\alpha v \quad f(n) = n^2 + 2^{28}n + 2^{30} \Rightarrow O(n^2)$$

- αν M_2 είναι 16 φορές γρηγορότερος από M_1

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	$O(\log n)$	2^{1024}	$(2^{1024})^{16}$

Λίγες Πράξεις !!

Πολ/τα Knapsack:

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_j x_j & \leq b \\ x_j & = 0 \text{ ή } 1 \end{aligned}$$

- ρητή απαρίθμηση = 2^n λύσεις
- μια πρόσθεση $250 \times 10^{-9}_s$ και 1 πολ.
 $\simeq 3$ προσθ.

$$n = 10 \rightarrow \dots\dots 10^{-2}_s$$

$$n = 20 \rightarrow \dots\dots 20_s$$

$$n = 50 \rightarrow \dots\dots 5 \times 10^{10}_s \quad !$$

$$n = 1000$$

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

$V(B)$: βέλτιστη τιμή

D: χώρος λύσεων

- Να βρεθεί x^* τ.ω. $V(B) = cx^* = \max\{cx | x \in D\}$

1. Αρχικοποίηση:

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

2. Ελάττωση: του χώρου D

- διαγραφή μεταβλητών
- διαγραφή περιορισμών
- πρόσθεση περιορισμών

3. Επίλυση του μειωμένου προβλήματος

- αλγεβρικές μέθοδοι
- δυναμικός προγραμματισμός
- μέθοδοι απαρίθμησης

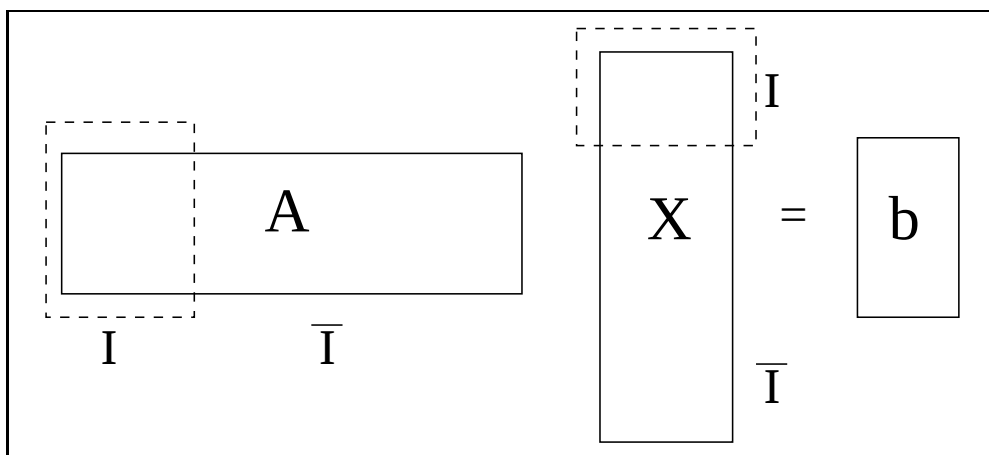
Μαθηματικός Προγραμματισμός

$$\begin{aligned} \max \quad & cx \\ & Ax \leq b \\ & x \geq 0 \\ & x \text{ ακέραιος} \rightarrow x = 0 \text{ ή } 1 \\ & A = (a_{ij})_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \end{aligned}$$

- αν $x \in \mathbb{R}$ και συναρτ.-περιορ. γραμμικά
 - Dantzig 1947
 - Kachiyan 1979
 - Karmakar 1984
- όχι γραμμικά, ακέραιοι, γραμμικό ή όχι

Γραμμικός Προγραμματισμός

Γραμμικό Σύστημα:



$$\begin{aligned}
 Ax &= b && \text{ή} \\
 A^I x_I + A^{\bar{I}} x_{\bar{I}} &= b && \text{ή} \\
 A^I x_I &= b - A^{\bar{I}} x_{\bar{I}} && \text{ή} \\
 x_I &= (A^I)^{-1} b - (A^I)^{-1} A^{\bar{I}} x_{\bar{I}}
 \end{aligned}$$

Τιμή συνάρτησης:

$$\begin{aligned}
 cx &= c^I x_I + c^{\bar{I}} x_{\bar{I}} = c^I (A^I)^{-1} b - c^I (A^I)^{-1} A^{\bar{I}} x_{\bar{I}} \\
 &= c^I (A^I)^{-1} b + [c^{\bar{I}} - c^I (A^I)^{-1} A^{\bar{I}}] x_{\bar{I}}
 \end{aligned}$$