Μοντέλα

Συνδυαστική Βελτιστοποίηση

Βασίλης Ζησιμόπουλος

Θεωριτική Πληροφορική
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

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

Παραδείγματα προβλημάτων

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

Προβλήματα γενικά NP-Hard

Προβλήμα Βελτιστοποίησης

Ορισμός

$I =< S, f >$: Στιγμιότυπο προβλήματος
S: Πεπερασμένο σύνολο λύσεων
$f : S \to \mathbb{R}$: Συνάρτηση κόστους (πρέπει να μεγιστοποιηθεί ή να ελαχιστοποιηθεί)

Προβλήμα Βελτιστοποίησης

Ορισμός

$I =< S, f >$: Στιγμιότυπο προβλήματος
S: Πεπερασμένο σύνολο λύσεων
$f : S \rightarrow \mathbb{R}$: Συνάρτηση κόστους (πρέπει να μεγιστοποιηθεί ή να ελαχιστοποιηθεί)

Ζητούμενο

Λύση $s^* \in S$ τέτοια ώστε:

$$f(s^*) \leq f(s), \ \forall s \in S \quad \text{(ελαχιστοποίησης)}$$
$$f(s^*) \geq f(s), \ \forall s \in S \quad \text{(μεγιστοποίησης)}$$

Motivation

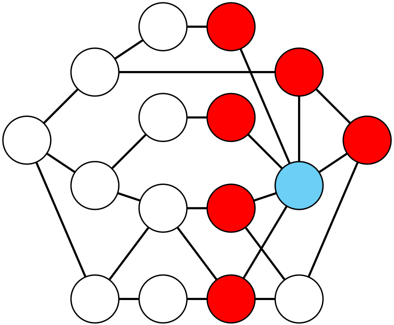
Dominating Sets' Applications

- Social network: maximizing influence (how a trend, a hit or a viral can be spread through the society), friendships

Motivation

Dominating Sets' Applications

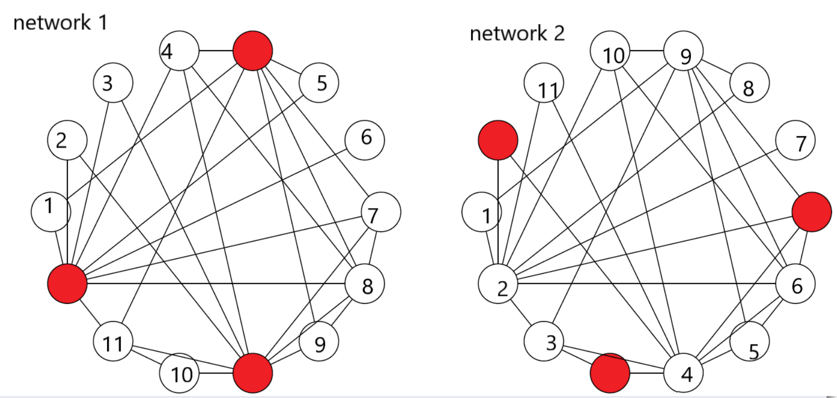
- Ad-hoc wireless (sensor) networks: coverage
- Electrical transmission lines: fault location



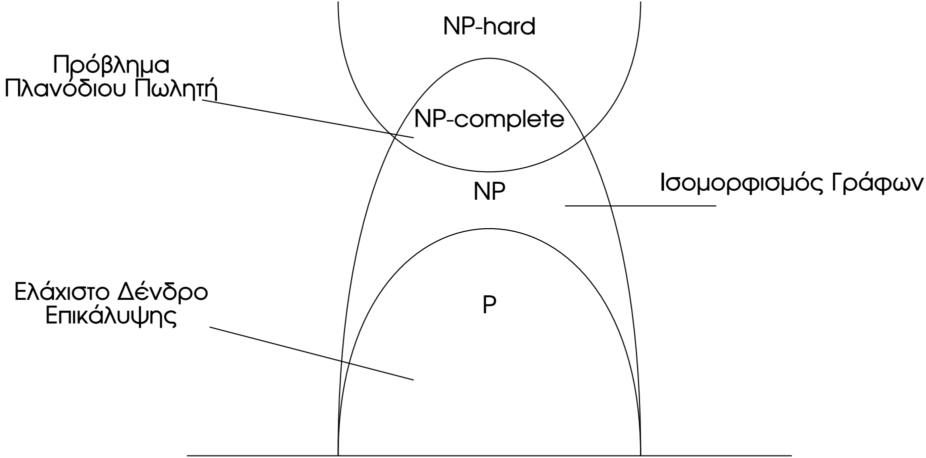
Conclusions

Related Problems: Fault Tolerant Dominating Sets

- Majority Illusion, Friendship Paradox
- help alleviate social problems: drinking, smoking, drugs



Κλάσεις Πολυπλοκότητας



Ο πλανώδιος πωλητής (TSP)

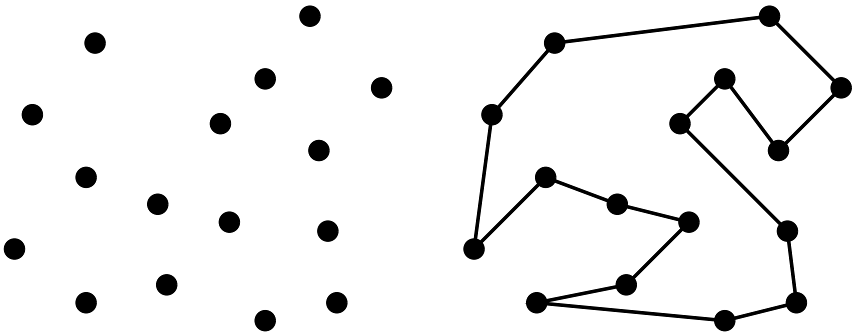
- $n + 1$ πόλεις
- Όλες οι μεταθέσεις $(n!)$
- Συντομότερος κύκλος Hamilton

Robot Tour Optimization

Suppose you have a robot arm equipped with a tool, say a soldering iron. To enable the robot arm to do a soldering job, we must construct an ordering of the contact points, so the robot visits (and solders) the points in order.

We seek the order which minimizes the testing time (i.e. travel distance) it takes to assemble the circuit board.

Find the Shortest Robot Tour



You are given the job to program the robot arm. Give me an algorithm to find the best tour!

Nearest Neighbor Tour

A popular solution starts at some point p_0 and then walks to its nearest neighbor p_1 first, then repeats from p_1, etc. until done.

Pick and visit an initial point p_0

$p = p_0$

$i = 0$

While there are still unvisited points

  $i = i + 1$

  Let p_i be the closest unvisited point to p_{i-1}

  Visit p_i

Return to p_0 from p_i

A Correct Algorithm: Exhaustive Search

We could try all possible orderings of the points, then select the one which minimizes the total length:

$d = \infty$

For each of the $n!$ permutations Π_i of the n points

        If $(cost(\Pi_i) \leq d)$ then

                $d = cost(\Pi_i)$ and $P_{min} = \Pi_i$

Return P_{min}

Since all possible orderings are considered, we are guaranteed to end up with the shortest possible tour.

Exhaustive Search is Slow!

Because it tries all $n!$ permutations, it is much too slow to use when there are more than 10-20 points.

No efficient, correct algorithm exists for the *traveling salesman problem*, as we will see later.

For Three Men
The Civil War
Wasn't Hell.
It Was
Practice!
CLINT EASTWOOD
"THE GOOD, THE BAD AND THE UGLY"
LEE VAN CLEEF
ALDO GIUFFRE | MARIO BREGA
ELI WALLACH
in the role of Tuco
SERGIO LEONE
TECHNISCOPE TECHNICOLOR

Ο πλανώδιος πωλητής (TSP)

Γραμμικό πρόγραμμα

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

Ο πλανώδιος πωλητής (TSP)

Γραμμικό πρόγραμμα

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

$$\sum_{i=0}^{n} x_{ij} = 1, \qquad\qquad j = 0, \ldots, n$$

Ο πλανώδιος πωλητής (TSP)

Γραμμικό πρόγραμμα

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

$$\begin{aligned} \sum_{i=0}^{n} x_{ij} &= 1, & j &= 0, \ldots, n \\ \sum_{j=0}^{n} d_{ij} &= 1, & i &= 0, \ldots, n \end{aligned}$$

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

Σύνδεση Τερματικών-Υπολογιστών

Το πρόβλημα

- Σύνολο m Τερματικών που θέλουν να συνδεθούν στο δίκτυο
- Σύνολο n Υπολογιστών του δικτύου
- Κάθε Τερματικό θέλει να συνδεθεί 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}

Σύνδεση Τερματικών-Υπολογιστών

- Μεταβλητές απόφασης

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

Σύνδεση Τερματικών-Υπολογιστών

- 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

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

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

Σύνδεση Τερματικών-Υπολογιστών

- 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

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

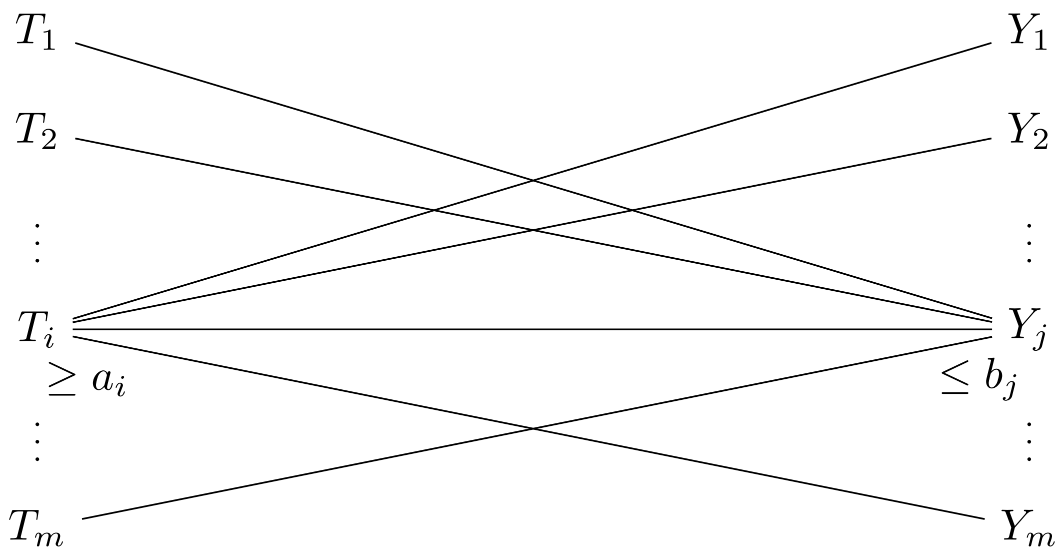
$$x_{1j} + x_{2j} + \cdots + x_{mj} \leq b_j, \forall j = 1, \ldots, n$$

- Αντικειμενική Συνάρτηση

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

Σύνδεση Τερματικών-Υπολογιστών

Διμερής γράφος ανάθεσης



Επίλυση με τεχνικές Θεωρίας Γράφων

Investment market (algos)

Τετραγωνικός Προγραμματισμός

$$\begin{aligned} \text{maximize} \quad & q(x) = r^t x - \mu x^t A x \\ \text{subject to} \quad & \sum_{i=1}^{n} x_i = 1,\ i = 1, \dots, n \\ & x_i \geq 0,\ i = 1, \dots, n \end{aligned}$$

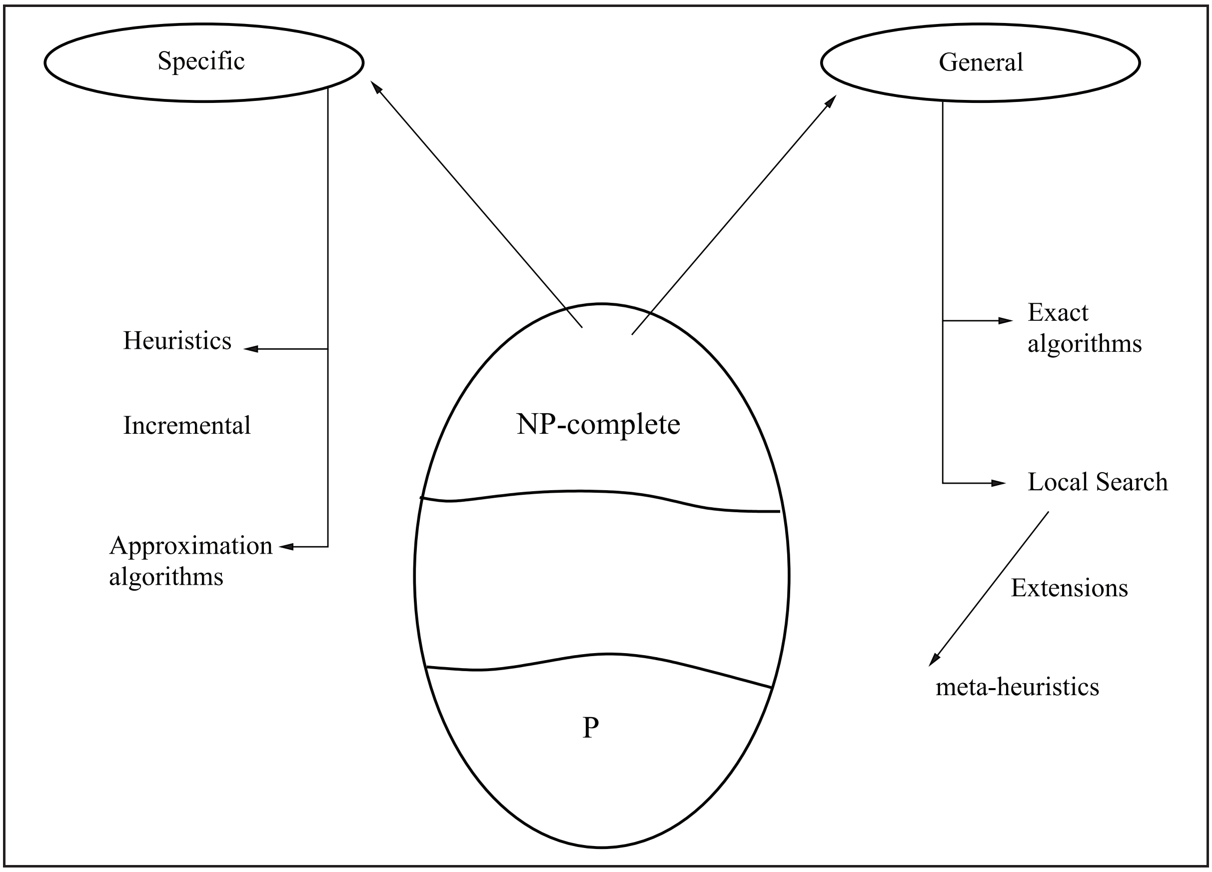
$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, μ ένας συντελεστής βαρύτητας και A ένας τετραγωνικός πίνακας $n \times n$, r διάνυσμα κολόνα $n \times 1$
(βέλτιστη διαχείριση χαρτοφυλακίων Markowitz, βραβείο Νόμπλελ οικονομίας 1990)

Η κλάση NP

Τα προβλήματα

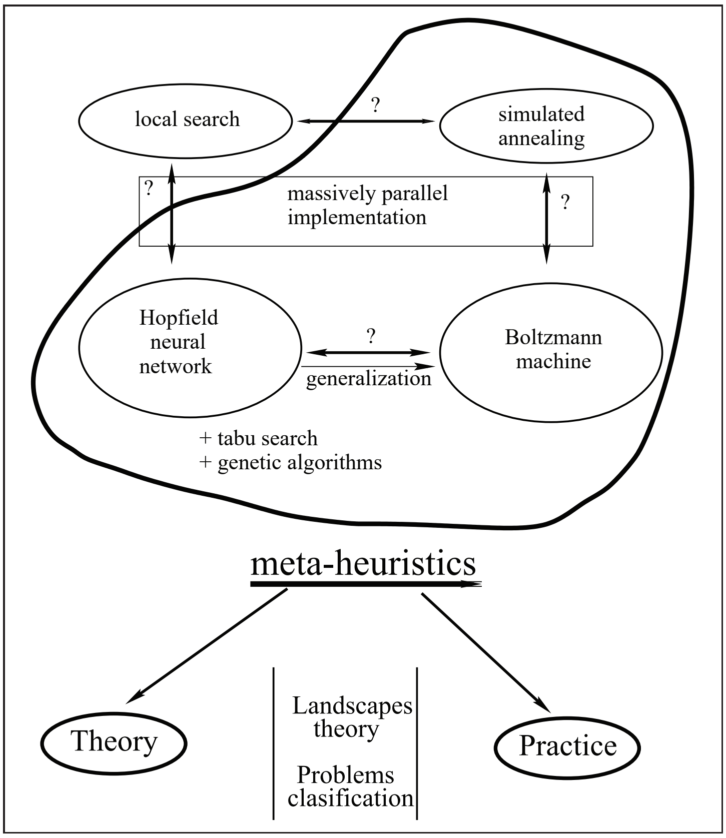
- Θεωρία Γράφων (βελτιστοποίηση)
- Παράλληλη Αρχιτεκτονική (task placement)
- Quadratic Assignment
- Interconnection Networks

Οι μέθοδοι



Specific
General
Heuristics
Incremental
Approximation algorithms
NP-complete
P
Exact algorithms
Local Search
Extensions
meta-heuristics

Η τοπική αναζήτηση και οι επεκτάσεις της



Οι μέθοδοι

- Exact Algorithms
  - moderate size
  - execution time?
  - parallelization?

Οι μέθοδοι

- Exact Algorithms
  - moderate size
  - execution time?
  - parallelization?
- Heuristics
  - large sizes
  - reasonable execution time
  - approximation
  - problem specific
  - parallelization?

Οι μέθοδοι

- Approximation Algorithms
  - solution quality quarantee
  - fully polynomial approximation schemes (FPTAS)

Οι μέθοδοι

- Approximation Algorithms
  - solution quality quarantee
  - fully polynomial approximation schemes (FPTAS)
- Local Search Methods
  - general methods
  - approximation?
  - local optimal!
  - feasible initial solution?
  - neighborhood structure?
  - neighborhood's search?
  - initial-solution-local optimum quality!

Οι μέθοδοι

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

Οι μέθοδοι

- Simulated Annealing
  - general technique
  - escape from local optimal
  - asymptoticaly 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 <u>not</u> necessary
  - good solutions
  - approximation?