DM826 – Spring 2014

Modeling and Solving Constrained Optimization Problems

Exercises 6

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

Matching under preferences

Matching agents one to another, subject to various criteria.

Examples:

- junior doctors to hospitals
- pupils to schools
- kidney patients to donors

subject to ordinal preferences over a subset of the others. That is, there is a ranking or list of preferences with first choice, second choice, etc. The list need not be strictly ordered.

Typically other constraints: such as capacity

Relevant and large applications:

- in Hungary in 2011, 140 953 students applied for admission at universities
- In US National Resident Matching Program in 2012, 38 777 aspiring residents, 26 772 available positions.
- Free-for-all markets: free negotiations: issues of unraveling, congestion, exploiting offers

Centralized Matching Schemes

Centralized clearinghouses

Third party computes (automatically) optimal matching. Eg. maximizing number of places filled at hospitals, giving the maximum number of school-leavers their first-choice university, or ensuring no junior doctor and hospital have an incentive to reject their assignees and become matched together.

Classification

- Bipartite matching problems with two-sided preferences
  - Stable Marriage problem (SM)
  - Hospitals Resident problem (HR) (many-one SM generalization)
    Workers Firm problem, Student-Project Allocation problem

  Optimality criteria: Stability: no two agents prefer another to one of their current assignees

- Bipartite matching problems with one-sided preferences
  - House Allocation problem (HA)
  - Capacited House Allocation Problem (CHA) (many-one HA generalization)

  Optimality criteria: Pareto optimality, popularity, profile-based optimality

- Non-bipartite matching problems with preferences
  - Stable Roommates problem (SR)
    chess players, kidney exchanges patient-donor, P2P network
  - Stable Fixtures, S. Multiple Activities, S. Allocation (many-many)
  - Coalition Formation Game (partnerships of size > 2)

  Optimality criteria: Stability

Variants and Extensions

- Indifference in agents' lists, ie ties
- Incomplete/bounded lists
- Exchange stability: no pair of residents who could exchange one another's assigned hospitals so as to improve their outcome
- tripartite matching problem with preferences
- find all stable matchings
- find stable matching with other properties

Literature

- Seminal paper by Gale and Shapley (1962). polytime algorithm for SM
- D.E. Knuth. Stable Marriage and its Relation to Other Combinatorial Problems. American Mathematical Society, 1976 (translated to English 1997)
- D. Gusfield and R. Irving. The Stable Marriage Problem: Structure and Algorithms. MIT Press, 1989
- K. Iwama and S. Miyazaki. A Survey of the Stable Marriage Problem and Its Variants. International Conference on Informatics Education and Research for Knowledge-Circulating Society (ICKS), 2008
- D.F. Manlove. Algorithmics of Matching Under Preferences. World Scientific, 2013
- http://optimalmatching.com/

Context

In economics (game-theory):

- matching theory as part of market design in microeconomics
- matching algorithm as a mechanism
- interest in strategy-proof or truthful mechanisms: make a dominant strategy for the agents to reveal their true preferences
- A.E. Roth and M. Sotomayor. Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis. Cambridge University Press, 1990

In CS

- Computational social choice theory (collective decisions) -> Algorithmic mechanism design (social welfare)
- Algorithmic Game Theory concerned with computational questions

Stable Marriages

- A set of men $U = \{m_1, \ldots, m_{n_1}\}$
- A set of women $W = \{w_1, \ldots, w_{n_2}\}$
- $n_1 = n_2 = n$ (if not add men or women with empty preference list)
- In SM $E = U \times W$. In SMI $E \subset U \times W$
- Each agent (man or woman) $a_k \in U \cup W$ has a preference list in which it ranks agents from the other set in strict order
- Given any man $m_i \in U$ and given any women $w_j, w_k \in W$, w_i is said to prefer w_j to w_k ($w_j \succ_{m_i} w_k$) if $(m_i, w_j) \in E$, $(m_i, w_k) \in E$ and w_j precedes w_k on m_i's preference list (same mutatis mutandis for any man)
- $rank(m_i, w_j)$ is 1 plus the number of women that m_i prefers to w_j. (similarly for $rank(w_j, m_i)$)
- An assignment / matching M is a subset of E. For each $a_k \in U \cup W$ the set of assignees of a_k is denoted by $M(a_k)$. $|M(a_k)| = 1$ (else unassigned)
- Underlying graph of an instance Π of SM is a bipartite graph $G = (U \cup W, E)$

Definition (Stable Matching)

A pair $(m_i, w_j) \in E \setminus M$ blocks a matching M, or is a blocking pair for M, if

- m_i is unassigned or prefers w_j to $M(m_i)$
- w_j is unassigned or prefers m_i to $M(w_j)$

A matching M is said to be stable if it admits no blocking pair.

Stability Checking Algorithm


```
for m := 1 to n do
    for each w such that m prefers w to M(m) do
        if w prefers m to M(w) then
            return unstable;
    return stable;
```

Gale Shapley algorithm (1962)


```
assign each person to be free
while some man m is free do
    w :=first woman on m's list to whom m has not yet proposed
    if w is free then
        assign m and w to be engaged (to each other)
    else
        if w prefers m to her fiance' m' then
            assign m and w to be engaged and m' to be free
        else
            w rejects m (and m remains free)
```

Linear Programming

Let $P(m, w)$ be the set of women whom m strictly prefers to w
Let $\hat{P}(w, m)$ be the set of men such that w strictly prefers m to each man in $\hat{P}(w, m)$

$$\sum_{w} x(m, w) = 1 \qquad \text{for each man } m \quad (1)$$

$$\sum_{m} x(m, w) = 1 \qquad \text{for each woman } w \quad (2)$$

$$x(m, w) \geq 0 \qquad \text{for each pair } (m, w) \quad (3)$$

$$\sum_{m' \in \hat{P}(w,m)} x(m', w) - \sum_{w' \in \hat{P}(m,w)} x(m, w') \leq 0 \qquad \text{for each pair } (m, w) \quad (4)$$

Constraint Programming

Length of preference lists $l(m_i)$ and $l(w_j)$, respectively.
$m = |E|$

Variables:

- $2n$ variables:
  x_i, $i \in U$, $dom(x_i) = \{1, 2, \ldots, l(m_i\} \cup \{n+1\}$
  y_j, $j \in W$, $dom(y_j) = \{1, 2, \ldots, l(w_j\} \cup \{n+1\}$
  ($x_i = p, 1 \leq p \leq l(m_i)$ then m_i marries the woman w_j such that $rank(m_i, w_j) = p$)

Constraints:

1. $x_i \geq p \implies y_j \leq p \quad 1 \leq i \leq n, 1 \leq p \leq l(m_i)$
2. $y_j \geq q \implies x_i \leq q \quad 1 \leq j \leq n, 1 \leq q \leq l(w_j)$
3. $y_j \neq q \implies x_i \neq p \quad 1 \leq j \leq n, 1 \leq q \leq l(w_j)$
4. $x_i \neq p \implies y_j \neq q \quad 1 \leq i \leq n, 1 \leq p \leq l(m_i)$

Extensions

- Incomplete lists: handled by a slight modification of Gale Shapley algorithm.
- Ties: three stability notions:
  - super-stability
    blocking pair is defined as a pair (m, w) such that $M(m) \neq w$, $w \succeq_m M(m)$, and $m \succeq_w M(w)$.
  - strong stability, (x, y) is a blocking pair if $M(x) \neq y$, $y \succ_x M(x)$, and $x \succeq_y M(y)$
  - weak stability. a blocking pair is defined as (m, w) such that $M(m) \neq w$, $w \succ_m M(m)$, and $m \succ_w M(w)$.

  super-stable matching $\Longrightarrow$ strongly stable, strongly stable $\Longrightarrow$ weakly stable.

  Weakly stable matching always exists and can be found in polynomial time. In contrast, there are instances that have no super-stable nor strongly stable matching. Nevertheless, there is a polynomial time algorithm that decides if a super-stable (strongly stable, resp.) matching exists and finds one if any, whose running time is $O(n^2)$ $O(n^3)$
- MAX SMTI (SM with ties + incomplete lists): finding a largest

Course Instructor Problem

Extension of SM: find a stable matching that matches as many agents as possible, given an instance of SM where the preference lists may involve ties and may be incomplete and constraints on the number of assignees that agents canmust obtain in a stable matching.

- A set of instructors $S = \{s_1, \ldots, s_{m_S}\}$
- A set of courses $C = \{c_1, \ldots, c_{m_C}\}$
- $m_C \leq m_S$
- Each course $c_j \in C$ requires a number of assistants n_j (posts or capacity)
- Each instructor $s_i \in S$ has a required minimum p_i and a required maximum q_i number of courses to receive
- There is a set $E \subseteq S \times C$ of acceptable course-instructor pairs
  Each assistant has an acceptable set of courses
  $A(s_i) = \{c_j \in C : (s_i, c_j) \in E\}$
  Each course has an acceptable set of instructors
  $A(c_i) = \{s_i \in S : (s_i, c_j) \in E\}$ (this handles the semester issue and other issues)
- Each agent (instructors and courses) $a_k \in S \cup C$ have a preference list in which it ranks $A(a_k)$ in partial order

- Given any instructor $s_i \in S$ and given any courses $c_j, c_k \in C$, s_i is said to prefer c_j to c_k ($c_j \succ_{s_i} c_k$) if $(s_i, c_j) \in E$, $(s_i, c_k) \in E$ and c_j precedes c_k on s_i's preference list (same mutatis mutandis for any course)

- $rank(s_i, c_j)$ is 1 plus the number of courses that s_i prefers to c_j. (similarly for $rank(c_j, s_i)$)

- An assignment M is a subset of E. For each $a_k \in S \cup C$ the set of assignees of a_k is denoted by $M(a_k)$. (unassigned, undersubscribed, full, oversubscribed)

- A matching is an assignment such that $p_j \leq |M(s_i)| \leq q_j$ for all $s_i \in S$ and $|M(c_j)| = n_j$ for each $c_j \in C$

Definition (Stable Matching)

Let Π be an instance of the problem and let M be a matching in Π. A pair $(s_i, c_j) \in E \setminus M$ blocks M, or is a blocking pair for M if:

1. s_i is undersubscribed or (strictly) prefers c_j to at least one member of $M(s_i)$
2. c_j is undersubscribed or (strictly) prefers s_i to at least one member of $M(c_j)$ (or both)

M is said to be stable if it admits no blocking pair.

A CP model

Variables

- $x_i, 1 \leq i \leq \sum_{l \in C} n_l$,

Outline

Edge Finding

$$\text{If } L_J - E_{J\cup\{i\}} < p_i + p_J, \text{ then } i \gg J \quad (a)$$
$$\text{If } L_{J\cup\{i\}} - E_J < p_i + p_J, \text{ then } i \ll J \quad (b)$$

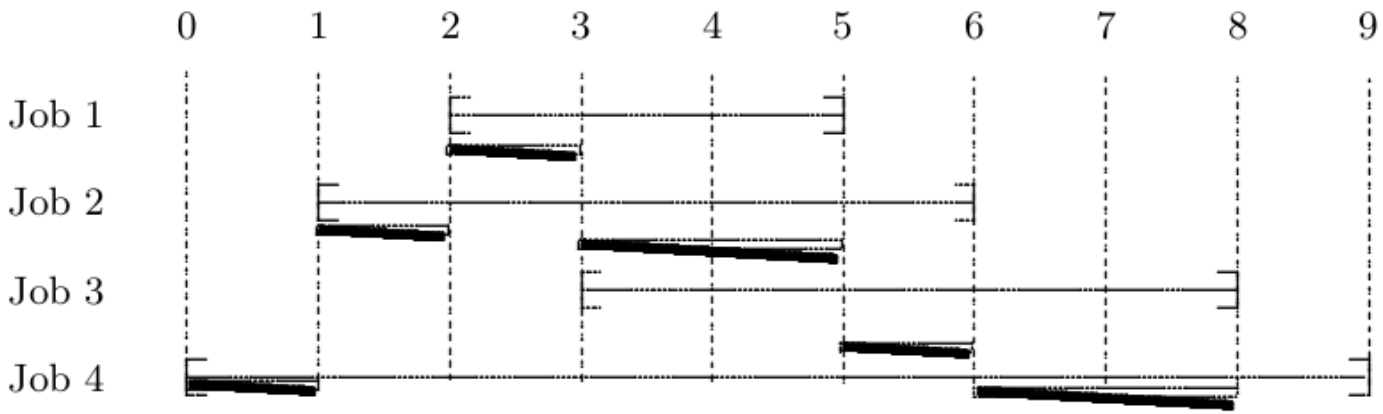
$$\text{If } i \gg J, \text{ then update } E_i \text{ to } \max\left\{E_i, \max_{J'\subset J}\{E_{J'} + p_{J'}\}\right\}.$$

$$\text{If } i \ll J, \text{ then update } L_i \text{ to } \min\left\{L_i, \min_{J'\subset J}\{L_{J'} - p_{J'}\}\right\}.$$

j	p_j	E_j	L_j
1	1	2	5
2	3	1	6
3	1	3	8
4	3	0	9

$O(n^2)$ algorithm

j	p_j	E_j	L_j
1	1	2	5
2	3	1	6
3	1	3	8
4	3	0	9



i	J_i	$\bar{p}$	k	J_{ik}	$L_k - E_i$	$p_i + \bar{p}_{J_{ik}}$
1	$\{1,2,3,4\}$	$(1,2,1,2)$	4	$\{2,3,4\}$	$9-2$	$1+5$
			3	$\{2,3\}$	$8-2$	$1+3$
			2	$\{2\}$	$6-2$	$1+3$
2	$\{1,2,3,4\}$	$(1,3,1,2)$	4	$\{1,3,4\}$	$9-1$	$3+4$
			3	$\{1,3\}$	$8-1$	$3+2$
			1	$\{3\}$	$5-1$	$3+1$
3	$\{2,3,4\}$	$(0,2,1,2)$	4	$\{2,4\}$	$9-3$	$1+4$
			2	$\{2\}$	$6-3$	$1+2$
4	$\{1,2,3,4\}$	$(1,3,1,3)$	3	$\{1,2,3\}$	$8-0$	$3+5$
			2	$\{1,2\}$	$6-0$	$3+4$
Conclude that $4 \gg \{1,2\}$ and update E_4 from 0 to 5						

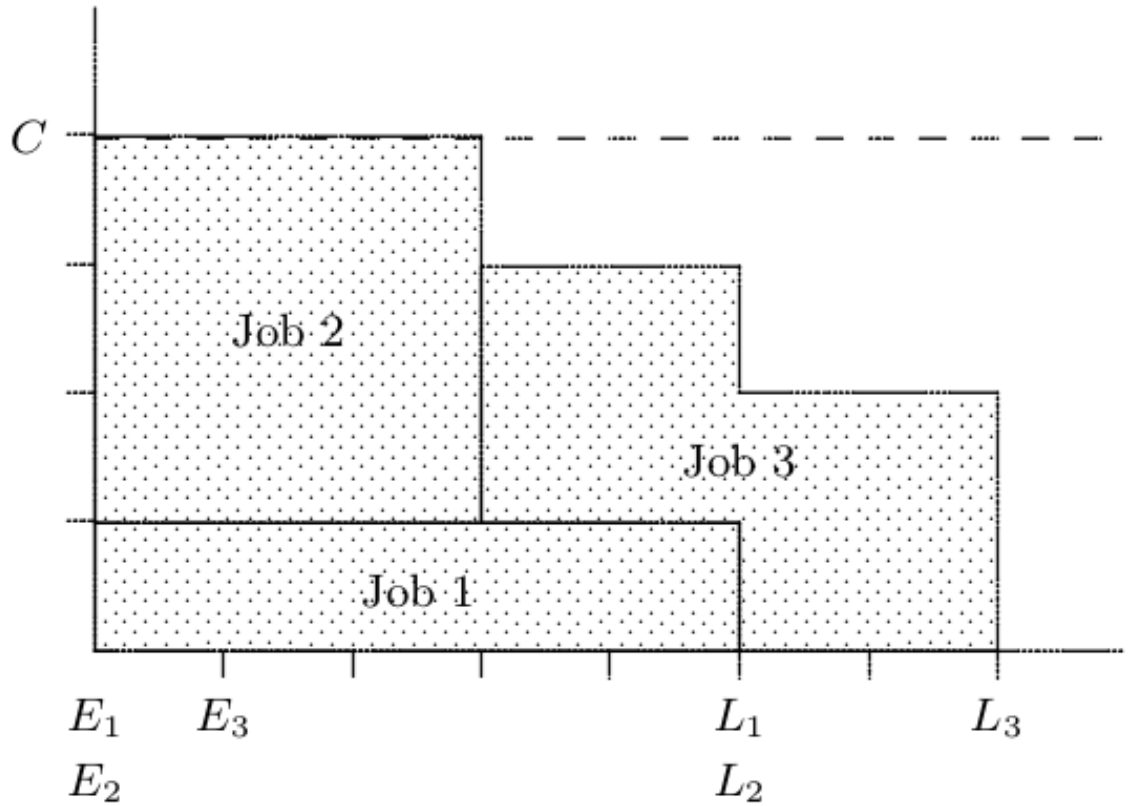
Not first, Not Last

$$\text{If } L_J - E_i < p_i + p_J, \text{ then } \neg(i \ll J). \quad (a)$$
$$\text{If } L_i - E_J < p_i + p_J, \text{ then } \neg(i \gg J). \quad (b)$$

$$\text{If } \neg(i \ll J), \text{ then update } E_i \text{ to } \max\left\{E_i, \min_{j \in J}\{E_j + p_j\}\right\} \quad (a)$$
$$\text{If } \neg(i \gg J), \text{ then update } L_i \text{ to } \min\left\{L_i, \max_{j \in J}\{L_j - p_j\}\right\} \quad (b)$$

Cumulative Scheduling

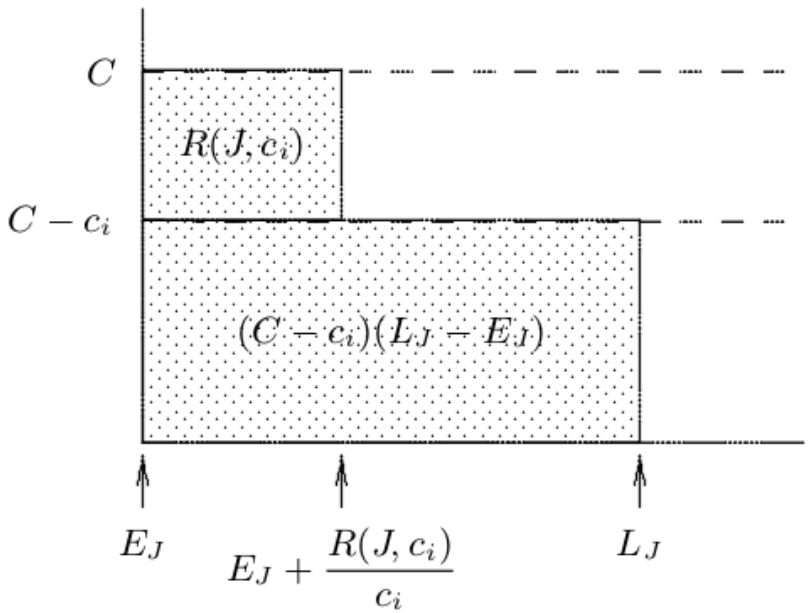
j	p_j	c_j	E_j	L_j
1	5	1	0	5
2	3	3	0	5
3	4	2	1	7



Edge Finding

If $e_i + e_J > C \cdot (L_J - E_{J\cup\{i\}})$, then $i > J$. (a)

If $e_i + e_J > C \cdot (L_{J\cup\{i\}} - E_J)$, then $i < J$. (b)



If $i > J$ and $R(J, c_i) > 0$, update E_i to $\max\left\{E_i, E_J + \dfrac{R(J, c_i)}{c_i}\right\}$.

If $i < J$ and $R(J, c_i) > 0$, update L_i to $\min\left\{L_i, L_J - \dfrac{R(J, c_i)}{c_i}\right\}$.

References