OPTIMAL DISCRETE STRUCTURES AND ALGORITHMS

ODSA 2006

Rostock (Germany), September 04 - 06, 2006

Matching Theory and Economic Model Building

By André A. Keller, Univ. Haute Alsace (France)

Rostock, September 4 - 6, 2006

Contents

- Perfect bipartite matching
- Existence and counting problems
- Maximal matching and network solutions, and algorithms
- Application to economic model building
- Conclusion : enumeration problem

  and large size application

Perfect bipartite matching

- Definition 1: A matching M in a graph $G=(V,E)$ is a set $M \subseteq E$ of pairwise non-adjacent edges.

- Definition 2: *(i)* A bipartite matching M is a set of pairwise non-adjacent edges in a bipartite graph $B=(U \cup W, E)$, where U and W are the color classes of G. *(ii)* A perfect matching $\pi(M)$ of graph B is a pairing of the set U to the set W which uses each element of U and each element of W once and only once. Such a matching covers all the vertices of the graph.

- Definition 3: A path $P=\{v_1,\ldots,v_m\}$ is an alternating path with respect to the matching M, if $(v_i,v_{i+1}) \in M$ then $(v_{i+1},v_{i+2}) \notin M$ for $1 \le i \le m-2$. An M – augmenting path begins and ends at M – unsaturated vertices.

The existence problem

- P. Hall’s Theorem :Let (V,W) be the bipartition of B. Let G(X) be all vertices which are adjacent two at least one vertex of X. Then B has a complete matching of U into W iff |G(S)| ³ |S| holds for every S í U.
- Corollary : (The Marriage Theorem of Frobenius). A bipartite graph B:(U,W) has a perfect matching iff |U|=|W| and for each X í U, |X|£|G(X)|.
- Tutte’s Theorem : Let c_0(G) be the number of odd components of the graph G=(V,E). G has a perfect matching iff c_0(G-S) £ |S| for all S í V(G).

The enumeration problem

<u>Permanent</u> : let $A=(a_{ij})$ be a n x n matrix. The permanent of A is

$$\text{per } A = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)} ,$$

where the sum is computed over all permutations σ of the numbers {1,…,n}. If A is the bi-adjacency matrix of the graph B, each non-zero term corresponds to a perfect matching. Then we have

$$\textbf{per } \mathbf{A} = \Phi(\mathbf{G}).$$

<u>Pfaffian</u> : let B be a 2n x 2n skew symmetric matrix. For each partition form a_α = sgn σ $b_{i_1,j_1} \ldots b_{i_n j_n}$. The Pfaffian is defined by

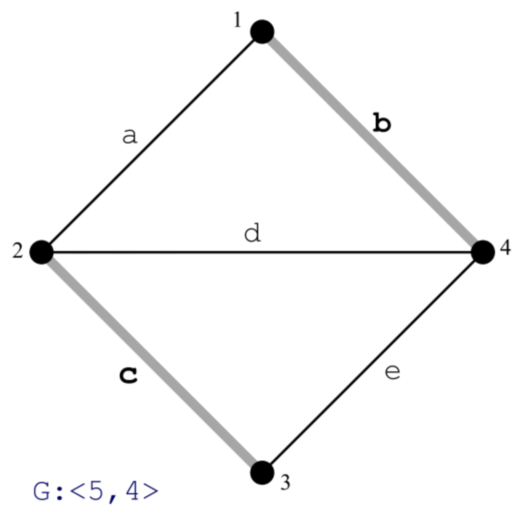
$$\text{Pf } B = \sum_{\alpha} a_{\alpha}$$

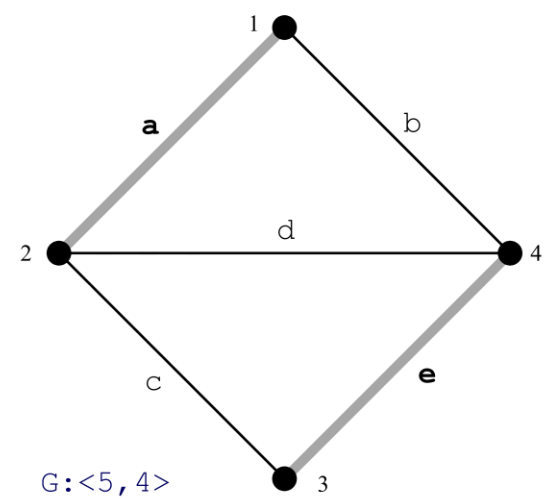
Lemma : If B is a skew symmetric matrix then det B = $(\text{Pf } B)^2$.

Example

B

0	a	0	b
-a	0	c	d
0	-c	0	e
-b	-d	-c	0

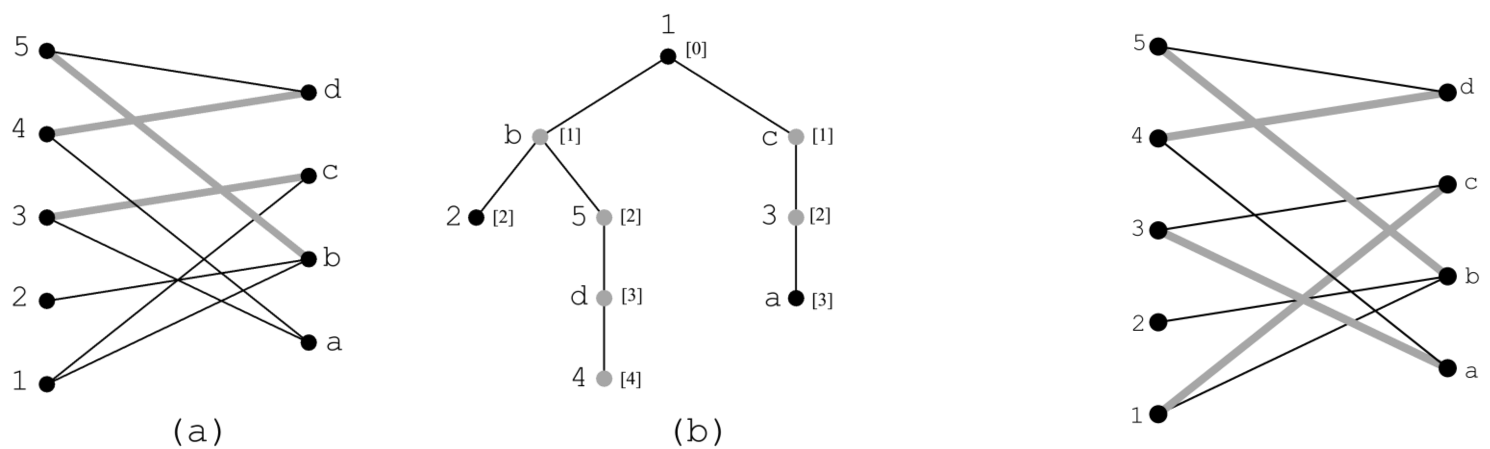




Each non-zero term of the Pfaffian Pf N = b c + a e refers to a perfect matching.

The maximum matching solution

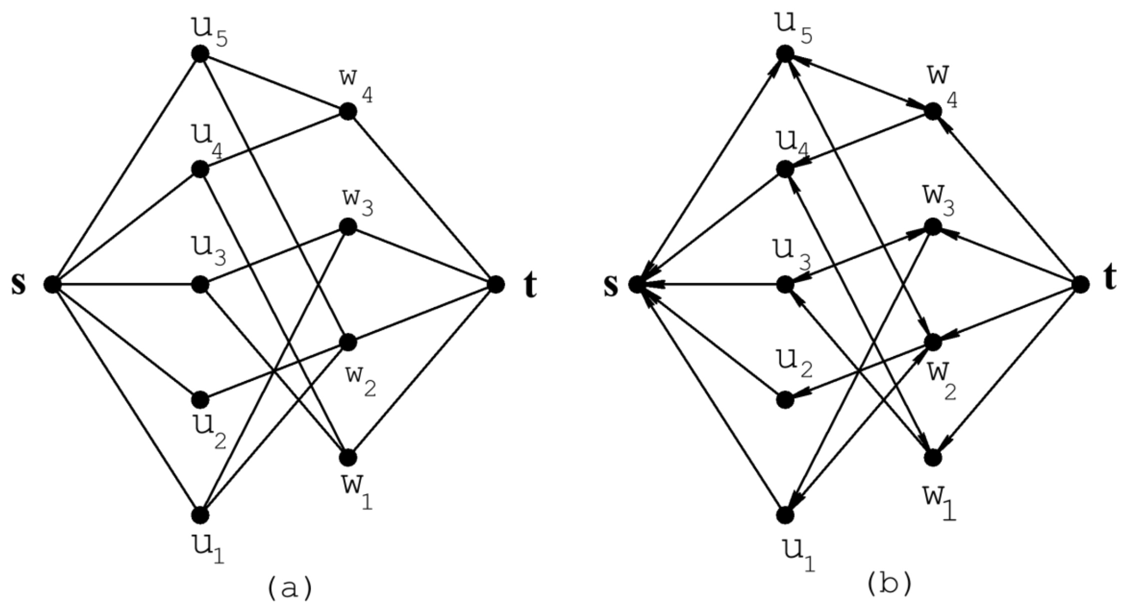
The Ford – Fulkerson algorithm



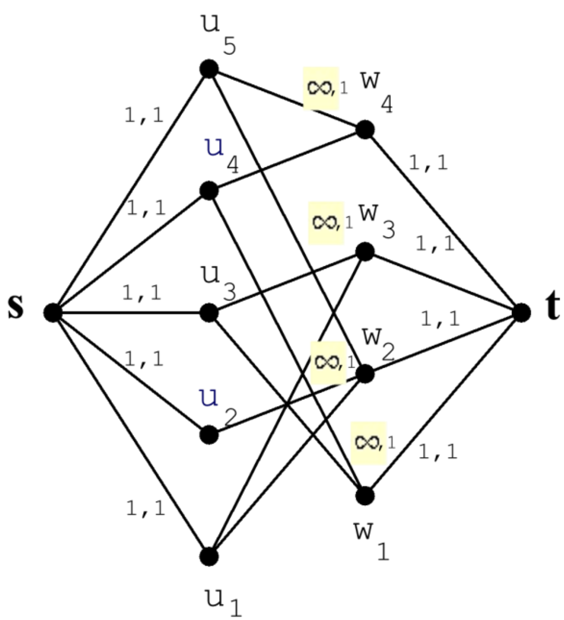
Let G be a graph with bipartition (U,W) and let M be any arbitrary matching in G. Suppose U_1 and W_1 are the sets of unmatched vertices. We aim to find an □ -augmenting path, if any, connecting U_1 to W_1 . We have : (a) an arbitrary matching at the begining, (b) augmenting path trees, (c) the maximal matching.

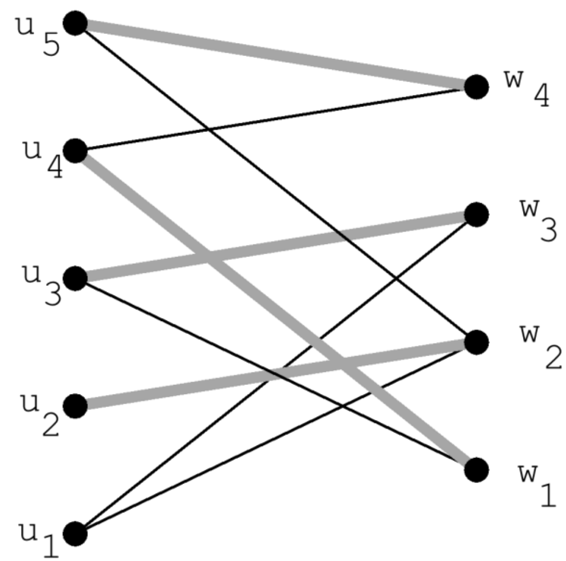
The network flow solution (1)

A Network is a graph G(V,E) with a non-negative capacity function C: E(G) ® Â+. A flow network is a flow network is a network with two addtionnal vertices **s** and **t**. The objective is to determine the maximum amount to carry in G from **s** to **t**. The problem of Maximum matching may be solved with a complexity of $O(|V|^{1/2}.E)$.



The network flow solution (2)





Residual network (a)

(b) matching

The duality Theorem

The maximum matching problem : Let G be a bipartite graph

(U,W) . A 0-1 vector x in $R^{E(G)}$ is the incidence vector of a matching in G iff $x(\delta(v)) \le 1$ for every point $v \in V(G)$. Hence the primal : *Maximize* $\mathbf{1}\, x$

subject to $A x \le \mathbf{1},\ x \ge 0$

The vertex cover problem : A 0-1 vector y is the incidence vector of a point cover iff it satisfies $y_u + y_v \ge 1$ for every $(uv) \in V(G)$.
Hence the dual : *Minimize* $\mathbf{1}\, y$

Subject to $A^T y \ge \mathbf{1},\ y \ge 0$

Theorem : For any cover (u,v) and perfect matching M, c(u,v) $\ge$ w(M). Furthermore c(u,v) = w(M) iff every edge (i j) in M satisfies $u_i + v_j = w_{ij}$. In this case M is a maximum matching and (u,v) a minimum vertex cover.

The equations of a growth model

$$(1- g) \log v + d\, r = a + b(\underline{G},v,r) \qquad (1)$$

$$D \log P + m\, D \log v - e\, DR = D \log \underline{M} - D \log \underline{Y}^* \qquad (2)$$

$$R = r + D\, P^e \qquad (3)$$

$$D \log (D\, P^e) = l\, (D \log P - D \log P^e) \qquad (4)$$

$$D \log P = w \log v + D\, P^e \qquad (5)$$

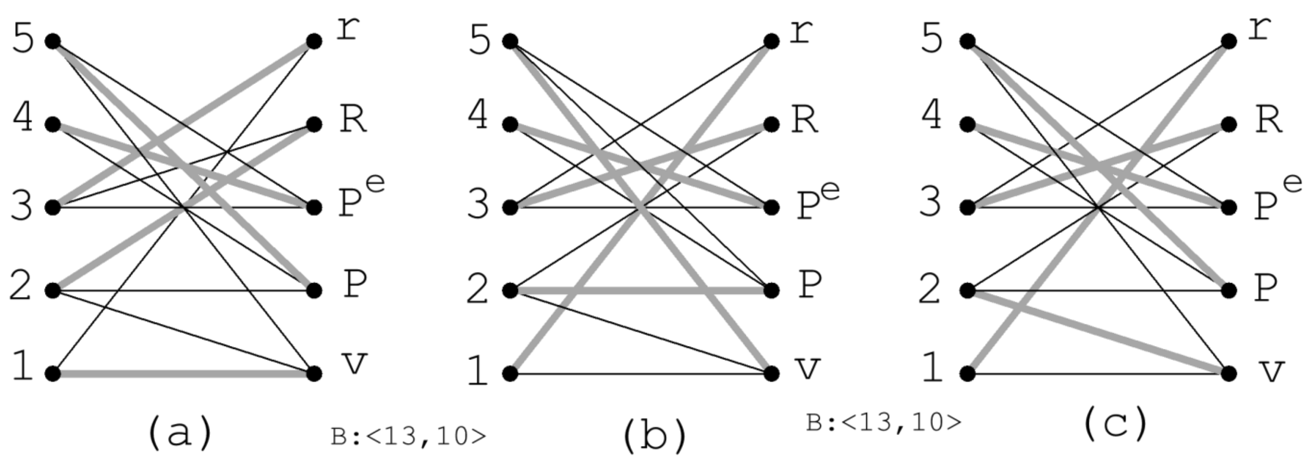
Note: **D** is the time derivative operator d/dt, **log** is the Neperian logarithm, the exogenous variables are underlined.

The endogenous variables are : P price of goods, P^e expected prices, R nominal interest rate, r real interest rate, v transitory component of national product.
The exogenous variables are : G government expenditures, M nominal money supply, Y* normal revenue.
The parameters in the list {a, d, e, g, l, m, w} are all taken positive, b(G,v,r) is a logaritm expression.

Economic solutions

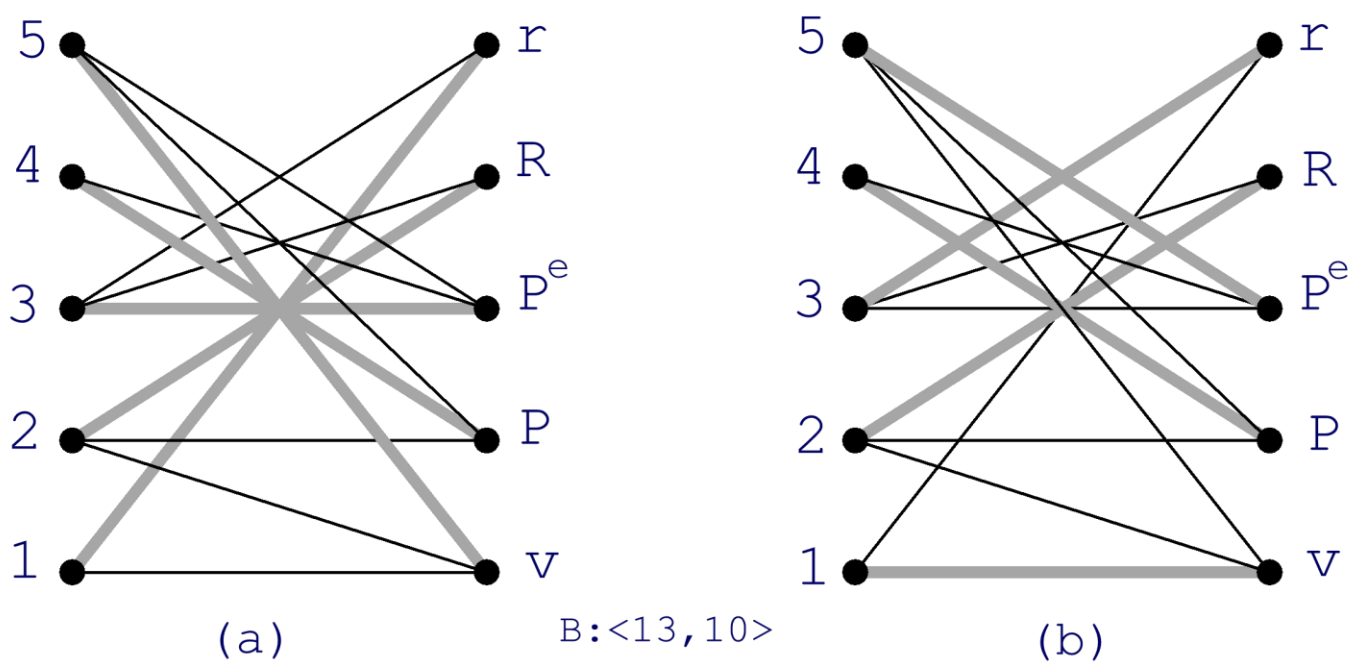
Interpretation in matching terms

	v	P	P^e	R	r
1	1	0	0	0	1
2	1	1	0	1	0
3	0	0	1	1	1
4	0	1	1	0	0
5	1	1	1	0	0



(a) is a wicksellian interpretation , (b) is a friedmanian interpretation and (c) is an extreme monetarist interpretation.

Technical solutions



(a) Is the maximal matching, (b) is the minimal cost assignment

Conclusion 1 : finding all minimum-cost perfect matchings in a bipartite graph

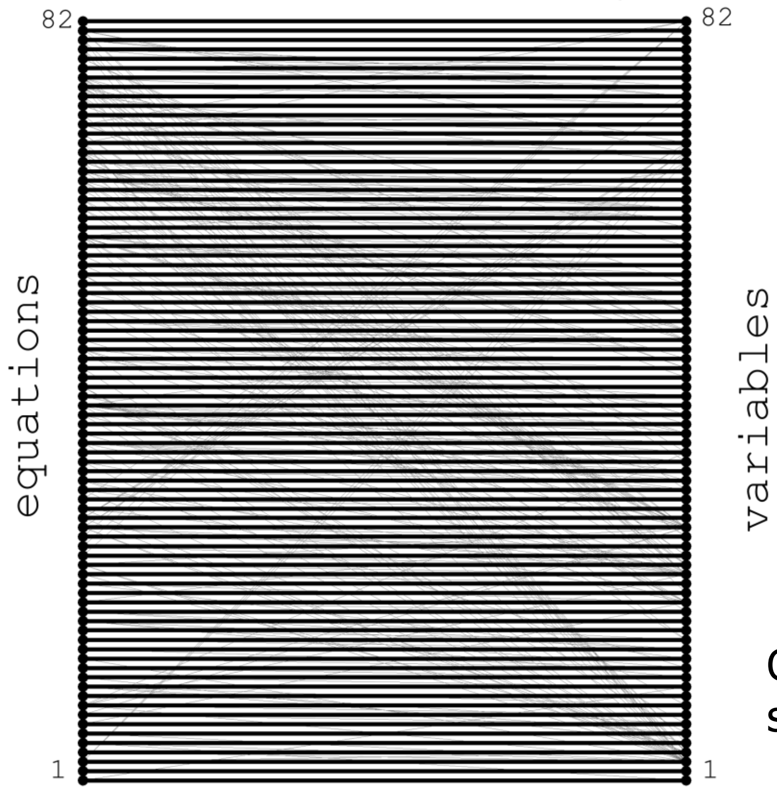
- The algorithm of Fukuda & Matsui (92') uses the *K* th-best solution of assignment problems (AP) developed by Murty (68') and Chegireddy & Hamacher (87'). The computational time is O(n(n+m)) and it requires O(n+m) memory storage for each additional matching. Their recent algorithm (95') requires $O(e(n+m)+n^{5/2})$ computational time and O(nm) memory storage, where e is | F(G) |.

  Method : first solve the AP by the Hungarian method and then generate each additional perfect matching in a lexicographic order. The procedure is based on a binary partitionning where the enumeration problem can be partitionned into two subproblems. It generalizes the Murty's algorithm for ranking the solutions of APs.

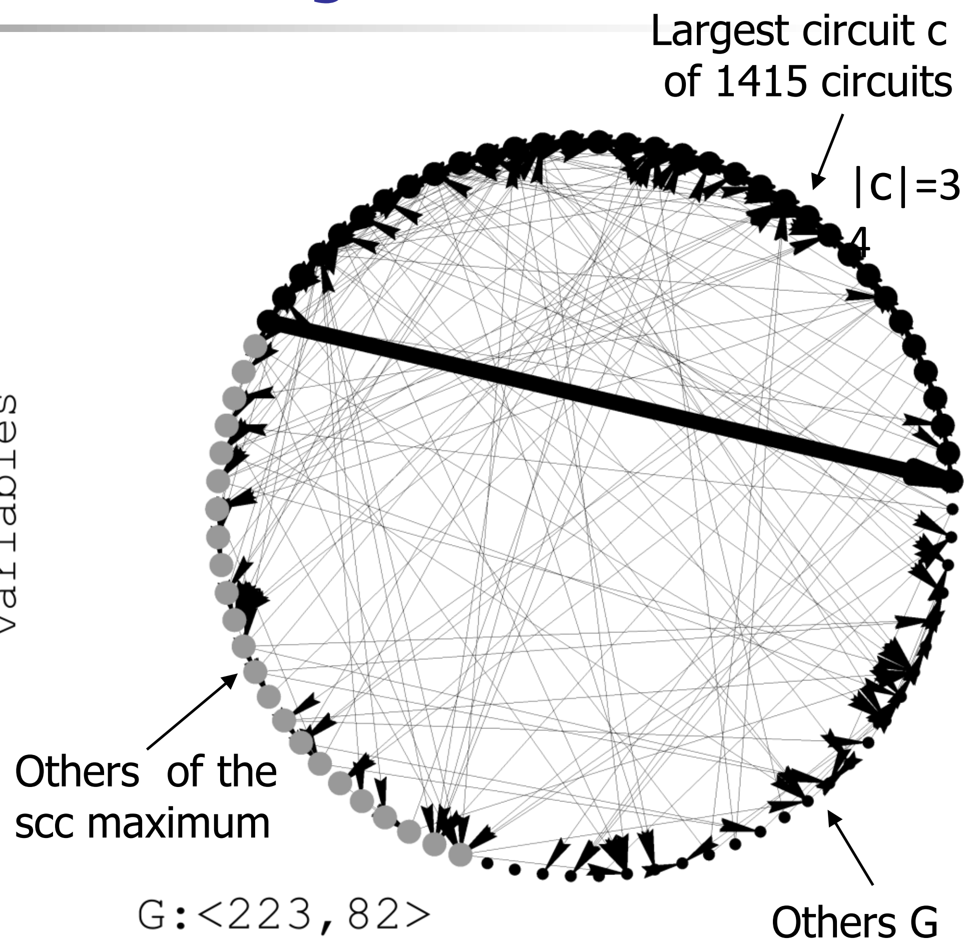
  Uno (97') proposed a new approach in two phases called trimming and balancing.

Conclusion 2 : further results
- the DMS forecasting model -

Maximal matching



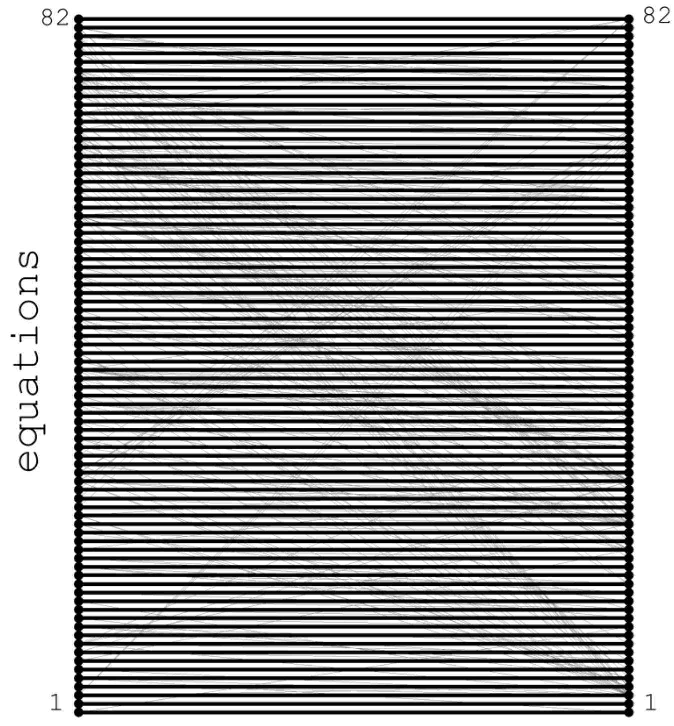
B:<305,164>



G:<223,82>

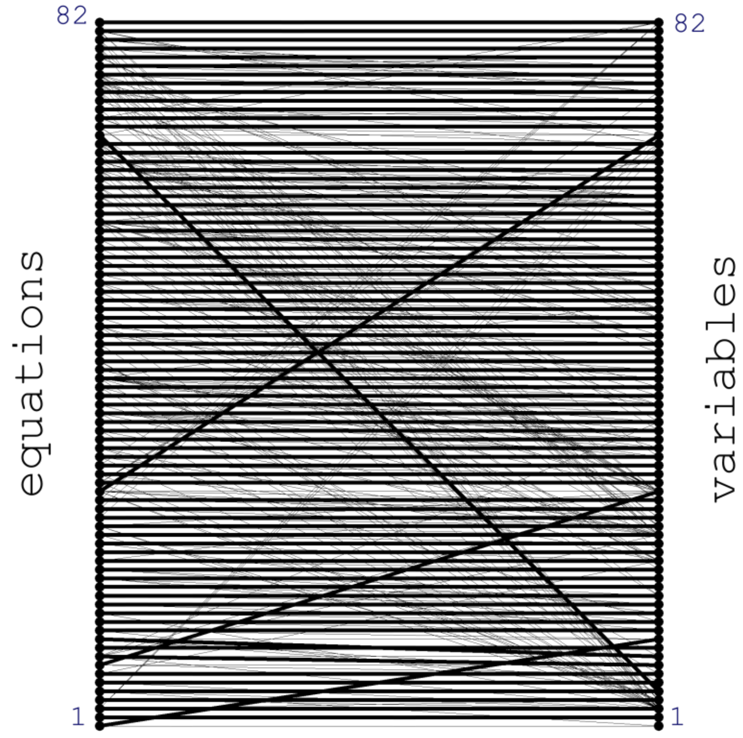
Technical solutions of DMS

Maximal matching



B:<305,164>

Minimal cost assignment



B:<305,164>

Other slides

The solution of linear programming

- The explicit form of the continous program is

$$\text{Maximize } \sum_i \sum_j c_{ij} x_{ij},$$

$$\text{Subject to: } \sum_j x_{ij} \le 1,$$

$$\sum_i x_{ij} \le 1,$$

$$0 \le x_{ij} \le 1.$$

where c_{ij} (= 1 or 0) is the cost of assigning the equation i to the variable j and where we have x_{ij} = 1 if equation 1 is assigned to variable j.

The assignment problem : LP solving

- The total variables is 13, the total constraints is 11
- There are 39 nonzeros
- A global optimal solution is found
- The objective value is 5

	x16	x110	x26	x27	x29	x38	x39	x310	x47	x48	x56	x57	x58
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1											
3			1	1	1								
4						1	1	1					
5									1	1			
6											1	1	1
7	1		1								1		
8				1					1			1	
9						1				1			1
10					1		1						
11		1						1					

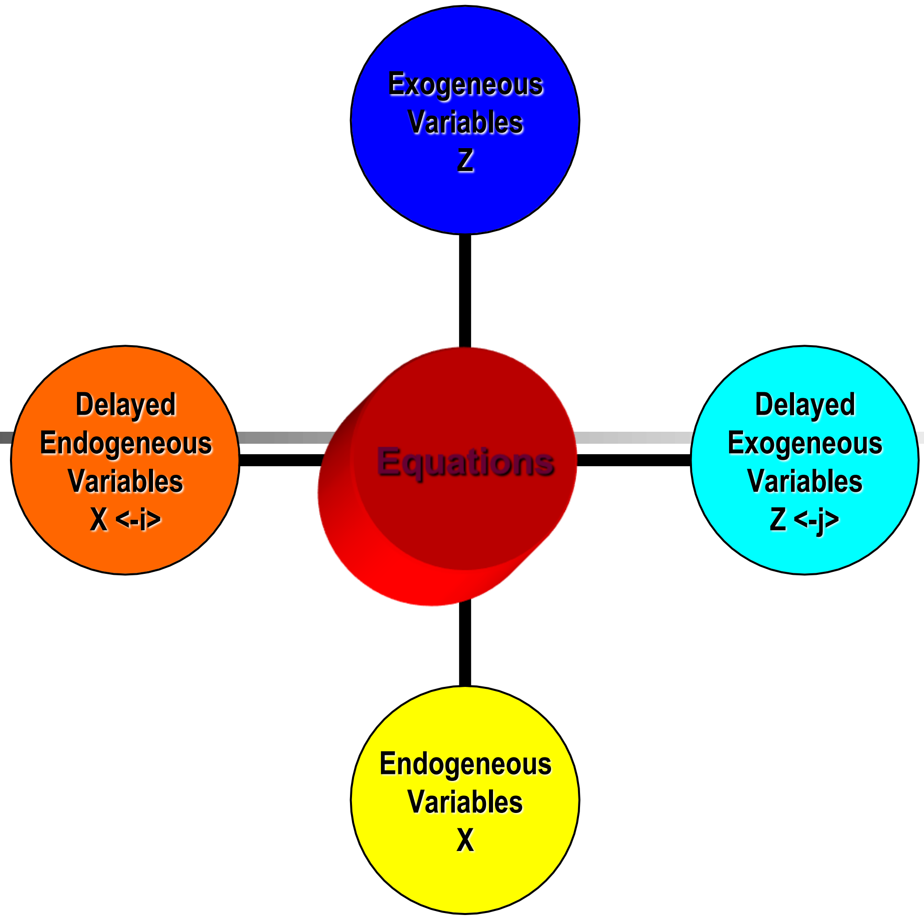
A square states for 1

Determining Matchings

Author	Algorithm	Performance	
		Computational time	Memory storage
Chegireddy /Hamacher		***O((c+1) n^3)***	***O((c+1) n^2***
Fukuda / Matsui	***Binary partitionning***	***O(c(n+m)+ $n^{5/2}$)***	***O((c+1)n+ m)***
Murty	***K^{th} – best solution***	***O((c+1) n^3)***	***O(n.m)***

Introduction to the economic models

What are the elements of such models ?



The normalized form of the model :

$$X_i = f_i(X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n, Z_0, Z_1, \ldots, Z_m), \; i = 1, n.$$