Matching Theory and Economic Model Building

André A. Keller [1]

Groupe de Recherche sur l'Apprentissage, l'Innovation et la Connaissance dans les Organisations, Université de Haute Alsace Mulhouse, France

Abstract

While building and by solving an economic model, the economists are confronted with the problem of knowing how to assign the variables of the model to the equations. By which equation does have one to calculate each endogenous variable of the model? Among the whole set of the possible and numerous solutions, only few of them will be eligible for an economic interpretation. In this study, we will show that the same economic model may have several "readings" according to economic theory or use for an economic policy purpose. Just take one familiar example : an empirical model for forecasting will have in the short run a "keynesian reading" with a production determination by demand components,while it will have "a classic reading" with a supply driven explanation, in the long run. The objective of this study is to deal with matching problem using the concepts and algorithms of graph theory [LoPl86]. Within this framework, one can look at the existence of matchings (Tutte's theorem), at counting the matchings(Ryser's formula) and at finding all the perfect matchings (Fukuda and Matsui's algorithm). Two types of applications are proposed for an illustration. One application is an economic growth model of small size, with 5 equations [Ve81]. The other application is a larger size empirical model, with 82 equations [Br97]. The computations have been carried out using the softwares Mathematica 5.1 and Lindo, as well as other Fortran 77 and C++ source programs. The computer software *Mathematica*® 5.1 contains specialized packages such as DecisionAnalysis‘Combinatorica and ‘GraphPlot.

Keywords: matching theory, bipartite graph, maximal matching, perfect matching, counting, enumeration, algorithm, macroeconomic model building.

This study [2] provides an useful insight into the economic model building and analysis. This knowledge is essential for economic model building. The interest of such an approach is shown using two types of macroeconomic models. Three major but simple proposals are given in this paper. Firstly, an exhaustive list of perfect matchings for a small theoretical economic model. Secondly the determination of the maximal and minimal network solutions for a large empirical model. Thirdly, an improved embedding of the graphs is proposed. The computer calculations have been done using the software *Mathematica*® 5.1 and its specialized packages DecisionAnalysis‘Combinatorica, ‘GraphPlot at `http://library.wolfram.com/infocenter/`. Other computer programs has been used to enumerate the circuits and the softwares LINDO [Sc97] for solving linear programs.

1 Introduction to the matching problem

1.1 Presentation of the model

The macrodynamic model of Vedel [Ve81] is an attempt to conciliate fundamental elements of the macroeconomic theory : the standard IS - LM model of Hicks, the natural unemployment theory of Friedman and Phelps, the theory of inflation of Cagan, the equation of Fisher and the analysis of Wicksell. The equations have been rewritten in a more readable form. The variables are continuous and derivable functions of time and parameters are all positive with $\gamma, \mu \in [0,1]$. A doted variable states for its first time derivative and log is the logarithm with base e. We have

$$Y = Y^{\gamma}\, \tilde{Y}^{(1-\gamma)}\, e^{\alpha-\delta r} + G \tag{1.1}$$

$$\frac{M}{P} = k\, Y^{\mu}\, \tilde{Y}^{(1-\mu)}\, e^{-\epsilon r} \tag{1.2}$$

[1] Email: `andre.keller@uha.fr`

[2] Presented at the Conference “Optimal Discrete Structures and Algorithms”, *Universität* Rostock, Institut *für* Mathematik, Institut *für* Informatik, September 04-6, 2006.

$$R = r + \frac{\dot{P}^e}{P^e} \tag{1.3}$$

$$\frac{\ddot{P}^e}{\dot{P}^e} = \lambda\,(\frac{\dot{P}}{P} - \frac{\dot{P}^e}{P^e}) \tag{1.4}$$

$$\frac{\dot{P}}{P} = \omega\ \log\ \frac{Y}{\bar{Y}} + \frac{\dot{P}^e}{P^e} \tag{1.5}$$

The endogenous variables are : P the price of goods, P^e the expected price, R the nominal interest rate, r the real interest rate, Y the effective national income, $\bar{Y}$ the expected normal income. The exogenous variables are : G the government expenditures and M the nominal money supply. Equation 1.1 is the equilibrium condition on the market of goods : the private demand depends on an geometric average between the current and the expected income, as well as on real interest rates. In equation 1.2 the real money demand has the same set of explanatory variables (except the government expenditures). Equation 1.3 is the equation of Fisher [3], where the real and nominal rates are related. The equation 1.4 describes the formation of price expectation following Cagan [4]. Equation 1.5 expresses the Phillips curve with the natural unemployment of Friedman and Phelps [5].

1.2 The assignment problem

This model may be transformed [Ve81] introducing the variable $Y/\tilde{Y}$ which is the transitory component of the national product and the two parameters of budgetary policy $z = \ 1+G/Y^{\gamma}\ \tilde{Y}^{1-\ \gamma} \times e^{\alpha-\ \delta\ r}$ and $b = log\ z$ [6]. All variables are functions of time. The model can then be rewritten as

$$(1-\gamma)\ \log v + \ \delta\ r = \ \alpha + b$$

$$\frac{\dot{p}}{p} + \ \mu\ \frac{\dot{v}}{v} - \ \epsilon\ \Delta R = \ \frac{\dot{M}}{M} - \frac{\dot{\tilde{Y}}}{\tilde{Y}}$$

---

[3] I. Fisher, *The Theory of Interest*, in A.M. Kelley, "Reprints of Economic Classics", New-York (1961).

[4] P. Cagan, *The Monetary Dynamic of Hyperinflation,* in M. Friedman (editor), "Studies in the Quantity Theory of Money", Chicago, Chicago University Press (1956).

[5] M. Friedman, *the role of monetary policy,* American Economic Review, **58**, (1968) 193–194 ; E. S. Phelps and al., "Microeconomic Foundations of Employment and Inflation", New-York, Norton 1975.

[6] These parameters are comparable to the ratio of public to private expenditures.

$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ \end{array} \begin{array}{c} \begin{array}{ccccc} v & P & P^e & R & r \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \\ \text{variable \#} \end{array} \text{equation \#}$$

Fig. 1.1. Equations and variables

$$R = r + \dot{P}^e$$
$$\frac{\ddot{P}^e}{\dot{P}^e} = \lambda \left(\frac{\dot{P}}{P} - \frac{\dot{P}^e}{P^e}\right)$$
$$\frac{\dot{P}}{P} = \omega \log v + \dot{P}^e$$

The relation between the equations 1 to 5 and the variables $\{v,\ P,\ P^e,\ R,\ r\}$ is shown in matrix $M = (m_{ij})$ of the Fig.1.1, where the rows are the equations and the columns j the variables. If $m_{ij} = 1$ the variable j is a variable of the equation i. When solving this model , the economist will be confronted to an assignment problem : how the endogenous variables will be calculated ? Is it the unique solution for associating the variables and the equations ? These questions that refer to the normalization in model building are relative to perfect matchings in a bipartite graph.

2 Perfect bipartite matching

Matching is a graph optimization problem. Let us introduce some definitions of graph theory about matchings on bipartite graphs.

Definition 2.1 . A graph $G = (U \bigcup W, E)$ is bipartite if its set of vertices can be partitioned into two sets U and W such that every edge in U has one endpoint in W. The sets U and W are the color classes of G and (U, W) a bipartition [7] of G.

[7] In a k-partite graph $G(V, E)$, the set $V(G)$ can be partitioned into the k partite sets $V_1, V_2, \ldots, V_k$ such that any edge (uv) belongs to different partite sets.

Definition 2.2 **1.** A bipartite matching $\mathcal{M}$ is a set of pairwise non-adjacent edges in a bipartite graph $G = (U \bigcup W, E)$. That is, $\mathcal{M} \subseteq E(G)$ such that $(e_1, e_2) \in \mathcal{M}$, $e_1 = (i_1 j_1)$, $e_2 = (i_2 j_2)$ and $i_1 = i_2 \Leftrightarrow j_1 = j_2$.
2. A perfect (or complete) matching $p(\mathcal{M})$ of the bipartite graph G 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.

Proposition 2.3 . *Every matching consists of at most $\frac{n}{2}$ edges, where n denotes the order of the graph. Every graph has a maximum matching, but not all graphs have a perfect matching whose edge number is exactly $\frac{n}{2}$.*

3 The existence problem and criteria

3.1 The existence problem

Frobenius' Theorem characterizes those bipartite graphs that have a perfect matching. Hall's Theorem characterizes the bipartite graph which have a matching of U into W. König 's Theorem gives a formula for the matching number $\nu(G)$ which is the size of the maximum matching, when a graph has no perfect matching. For this problem there is initially $n!$ possible pairings. According to Hall's Theorem, a condition for the existence of a matching is that every subset of the set U has enough neighbors in the set W [8] .

Theorem 3.1 .*(P. Hall's Theorem 1935). Let $G = (U \bigcup W, E)$ be a bipartite graph . Let X be any set in V(G) and $|\Gamma(X)|$ be all vertices which are adjacent to at least one vertex of X [9] . Then G has a complete matching of U into W if and only if $|\Gamma(X)| \geq |X|$ holds for every $X \subseteq U$.*

Proof of Theorem 3.1 . The demonstration [10] is done by induction on the cardinality $|U|$. When $|U| = 0_{\texttt{or}}\ 1$, the theorem holds immediately. Suppose that $|U| \geq 2$ and $|\Gamma(X)| > |X|$, for all $X \subset U$. Let (uw) be an edge of G. Let $G' = G - u - w$ be a bipartite graph with color classes U' and W'. For any $X' \subseteq U'$ we have $|\Gamma_{G'}(X')| \geq |\Gamma_G(X)| - 1 > |X'| - 1$ so that $|\Gamma_{G'}(X')| \geq |X'|$ in G'. Then there is a complete matching from U' to W', which along with (u,w) forms a complete matching from U to W. Now suppose that $|\Gamma(X)| = |X|$. Let G_1 be a subgraph induced by $X \bigcup \Gamma(X)$ and

---

[8] It is not an efficient algorithm since the Hall's condition would require to look at 2^n subsets.

[9] $\Gamma_G(X)$ denotes the set of neighbors of X in a graph G.

[10] The proof is given by M. Aigner and G.M. Ziegler, "Proofs from the Book", second edition, Springer Verlag, Berlin, 2001.

$G_2 = G - X - \Gamma(X)$. Let U_i, W_i $(i = 1, 2)$ be the color classes. One can easily verify that G_1 and G_2 satisfy the matching conditions. The two matchings together form a complete matching from U to W. □

Corollary 3.2 *.(The Marriage Theorem of Frobenius). A bipartite graph G= (U,W) has a perfect matching if and only if* $|U| = |W|$ *and for each* $X \subseteq U$, $|X| \leq |\Gamma(X)|$.

Definition 3.3 . An alternating path has edges that are alternately free (or unmatched) such as $e \in E - \mathcal{M}$ and matched such as $e \in \mathcal{M}$. An augmenting path starts at an unmatched vertex and ends at another free vertex.

Theorem 3.4 *.(Berge's Theorem 1957). Let* $\mathcal{M}$ *be a matching in a graph G. Then* $\mathcal{M}$ *is a maximum matching if and only if there exists no augmenting alternating path in G relative to* $\mathcal{M}$.

Proof of Theorem 3.4.(⇒) If there exists an augmenting alternating path $\mathcal{P}$ we would obtain a new matching $\mathcal{M}' = (\mathcal{M} \bigcup P) - (\mathcal{M} \bigcap P)$ of cardinality $|\mathcal{M}| + 1$.
(⇐) Let $\mathcal{M}'$ be a maximum matching. There cannot be an augmenting alternating path. Hence the partial graph H(V,$\mathcal{M}' \bigoplus \mathcal{M}$) has no odd component whose edges are alternately in $\mathcal{M}$ and in $\mathcal{M}'$. Then we have $|\mathcal{M} - \mathcal{M}'| = |\mathcal{M}' - \mathcal{M}|$. Hence the equality $|\mathcal{M}| = |\mathcal{M}'|$ proves that $\mathcal{M}$ is a maximum matching. □

Tutte's Theorem generalize the marriage Theorem to the characterization of perfect matchings. The condition that the cardinality of every subsets $|X|$ exceeds the number of odd connected component in the subgraph produced by $V \setminus X$.

Theorem 3.5 *.(Tutte's Theorem, Lovász* [11] *). A graph G has a perfect matching p(*$\mathcal{M}$*) if and only the number of odd components* $c_0(G - X)$ ***for all*** $X \subseteq V(G)$ *does not exceed the cardinality of X. Then we have*

$$c_0(G - X) \leq | X |, \ for\ all\ X \subseteq V(G).$$

Proof of Theorem 3.5.(⇒) Consider a graph with a perfect matching. Let X be an arbitrary subset of V $(= U \bigcup W)$. Consider an arbitrary odd component $\mathcal{C}$ in $G - X$. Then on edge at least in $\mathcal{C}$ must be matched to a vertex in X. Thus we have $c_0(G - X) \leq | X |$.

[11] Honsberger,R and R. Lovácz, *Proof of a Theorem of Tutte*, in Mathematical Gems II, Math. Assoc. Amer. (1976) 147-157.

($\Leftarrow$) *(i)* Suppose a graph G having a perfect matching [LoPl86] p($\mathcal{M}$) and $X \subseteq G$. Every odd-component of $G - X$ must have one edge of p($\mathcal{M}$) to X but every vertex in X have at most one edge. Then $G - X$ has at most $|X|$ odd components.
(ii) Suppose a graph G of even order, having no perfect matching. Consider the edge saturated graph G' by adding edges as long as we have no perfect matching. Let S' the set of vertices adjacent to every vertex of G' and $H' = V(G) - S'$. Moreover $H' = G'_1 \bigcup \ldots G'_k$ where the G'_i's are edge-disjoint complete subgraph and $k = |S'| + 2$. Then G'-S' has more than $|S'|$ odd components. □

Consider now our application to economic models with the sets Y and X of equations and variables respectively. Let Y_i be the set of equations that calculates the ith variable. Then the marriage theorem states that each variable can be calculated by one equation iff the collection of sets $\{Y_i\}$ satisfies the following marriage condition : for any subset of variables, the number of possible assignments to the equations must be at least as large as its cardinality.

Theorem 3.6 *.(König's Minimax Theorem 1931).The maximum size of a matching in a bipartite graph G is the minimum cardinality of a vertex cover in G.*

Proof of Theorem 3.6 . Let $U \subseteq V$ be a vertex cover of E, in which every edge of G is incident with a vertex in U. Let $\mathcal{M}$ in G be a matching of maximum cardinality. Let (ab) not be in $\mathcal{M}$. It contains an edge (a'b') with a = a' or $b = b'$. Assume that a = a'. If a is unmatched and b = b' then (ab)is alternating path P and so the end of $(a'b') \in \mathcal{M}$ chosen for U was the vertex b' = b. If a' = a is not in U, then $b' \in U$ and some alternating path P ends in b. There is also an alternating path P ending in b, either P := P b (if $b \in P$) or P' = P b' a' b. By maximality of $\mathcal{M}$, P' is not an augmenting path. So b must be matched and was chosen for U from the edge of $\mathcal{M}$ containing it. □

3.2 A circular implication of equivalent Theorems

Let us demonstrate the circular implication

$$König \Rightarrow Hall \Rightarrow Frobenius \Rightarrow König$$

Let $\nu(G)$ be the matching number and $\tau(G)$ be the vertex covering number [12].
Regarding the implication *König* $\Rightarrow$ *Hall* the necessity proof is obvious. For the sufficiency condition assume that for all $X \subseteq A$ where A is a vertex cover we have $|\Gamma(X)| \geq |X|$—. Let $\mathcal{C}$ be a cover vertex such that $|\mathcal{C}||| = \tau(G) \leq |A|$. Assume that— $|\mathcal{C}| < |A|$. Then we have

$$|\Gamma(A - \ \mathcal{C})| > |A - \ \mathcal{C}| = |A| - \ |A \bigcap \mathcal{C}| > |B \bigcap \mathcal{C}|.$$

The fact that an edge exists from A - $\mathcal{C}$ to B - $\mathcal{C}$contradicts that $\mathcal{C}$ is a cover vertex.
Concerning the implication *Hall* $\Rightarrow$ *Frobenius* it appears that Frobenius Theorem is a special case of Hall's Theorem .
About the implication *Frobenius* $\Rightarrow$ *König* the fact that $\nu(G) \leq \tau(G)$ is obvious for a matching in order to prove the necessary condition. Let $\mathcal{C}$ be a vertex cover of G of size $\tau(G)$. To show that $|\mathcal{C}| \leq \nu(G)$ we need to find a matching $\mathcal{M}$ such as $|\mathcal{C}| = |\mathcal{M}|$. The matching $\mathcal{M}$ is formed by the union of two sub-matchings $\mathcal{M}_1$ from A - $\mathcal{C}$ into B - $\mathcal{C}$ and $\mathcal{M}_2$ from B - $\mathcal{C}$ into A - $\mathcal{C}$. For every $X \subseteq A \bigcap C$ we have $\Gamma(X) \geq |X|$. This implies $|A \bigcap \mathcal{C}| \leq |B - \mathcal{C}|$. Then let us add dummy vertices to $|A \bigcap \mathcal{C}|$ until its size equals $|B - \mathcal{C}|$. Connect all dummy vertices to all vertices in $|B - \mathcal{C}|$ and verify that the Frobenius conditions are satisfied.

3.3 Criteria for a perfect matching

Definition 3.7 . Let G a bipartite graph with bipartition (U,W). For $X \subseteq U$ define the deficiency of X by $\mathtt{def}(X) = |X| - |\Gamma(X)|$, where $|\Gamma(X)|$ denotes all vertices which are adjacent to at least one element of X. Since def$(\varnothing) = 0$, we have def$(G) \geq 0$.

The Theorem 3.8 is a consequence of the König or P. Hall Theorems [LoPl86].

Theorem 3.8 . *The matching number of a bipartite graph G is* $|U| - |def(G)|$.

[12] The matching number is the size of the largest matching. The vertex covering number equals the cardinality of the smallest vertex cover in which each edge of G has at least one end-vertex .

Permanent

Definition 3.9 . Let G be a bipartite graph with bipartition sets (U, W) of the same size $|U| = |W| = n$ where $U = \{u_1, \ldots, u_n\}$ and $W = \{w_1, ..., w_n\}$. The bi-adjacency matrix $A = (a_{ij})$ is defined by

$$a_{ij} := \begin{cases} 1 & \text{if } (u_i w_j) \in E(G), \\ 0 & \text{otherwise.} \end{cases}$$

The a_{ij} denote the number of edges connecting u_i to w_j.

Let $A = (a_{ij})$ be a $n \times n$ matrix. Define the permanent of A by

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

where the sum is computed over all permutations σ of the numbers $\{1, \ldots, n\}$. The only difference with the Leibniz formula of the determinant is that all terms have the same sign [13] . This formula [14] per contains n ! summands. If A is the bi-adjacency matrix, each non-zero term corresponds to a perfect matching in the bipartite graph G. Then we have

$$\texttt{per } A = \Phi(G),$$

where $\Phi(G)$ denotes the number of perfect matchings in G.

---

[13] The determinant is defined by

$$\det A = \sum_{\sigma \in S_n} \texttt{sgn}(\sigma). \prod_{i=1}^{n} a_{i,\sigma(i)},$$

where sgn(σ) is -1 if σ is an odd number of inversions, and +1 otherwise.

[14] We also have the Ryser formula

$$\text{per } A = (-1)^n \sum_{s \subseteq \{1,\ldots,n\}} (-1)^{|s|} \prod_{j \in s} a_{i,j},$$

where the sum is over all the subsets of $\{1, \ldots, n\}$ and $|s|$ the number of elements in s .

Pfaffian

Let B be a $2n \times 2n$ skew symmetric matrix [15]. For each partition $\alpha = \{\{i_1, j_1\}, \ldots, \{i_n, j_n\}\}$ of the set $\{1, ..., 2n\}$ into pairs and form the following expression

$$b_\alpha = \texttt{sgn} \begin{pmatrix} 1 & 2 & \ldots & 2n-1 & 2n \\ i_1 & j_1 & \ldots & i_n & j_n \end{pmatrix} b_{i_1 j_1} \ldots b_{i_n j_n},$$

where $\texttt{sgn}(\sigma)$ is -1 if σ has an odd number of inversions , and +1 otherwise. The Pfaffian of matrix B is then define by

$$\texttt{pf}\ B = \sum_\alpha b_\alpha$$

.

Lemma 3.10 *.If B is a skew symmetric matrix, then* $\texttt{det}\ B = \texttt{pf}(B)^2$*. The determinant of a real skew-symmetric matrix is then always non negative.* [16]

Example 3.11 . For example, with the 4×4 matrix

$$B = \begin{pmatrix} 0 & b_{12} & b_{13} & b_{14} \\ -b_{12} & 0 & b_{23} & b_{24} \\ -b_{13} & -b_{23} & 0 & b_{34} \\ -b_{14} & -b_{24} & -b_{34} & 0 \end{pmatrix}$$

The Pfaffian of a skew-symmetric matrix B is then defined by

$$\texttt{pf}\ B = \sum_{\mathcal{M}} p(\mathcal{M}),$$

where the sum ranges over all perfect matchings $\mathcal{M}$. In the example above,we have pf B $= b_{14}\ b_{23} - b_{13}\ b_{24} + b_{12}\ b_{34}$.

---

[15] B skew symmetric matrix is a square matrix B whose transpose is its negative. It then satisfies $B^T = -B$. The determinant equals $(-1)^n \texttt{det}\ B$. According to the Jacobi's Theorem, the determinant vanishes when n is odd.

[16] This is a consequence of the Cayley's Theorem (1857). Let A be a matrix of even dimension which skew-symmetric after deletion of its rth row and column. Let R and C be two matrices formed by changing he rth row and column. Then the Theorem states that $\texttt{det}\ A = \texttt{pf}\ R \times \texttt{pf}\ C$.

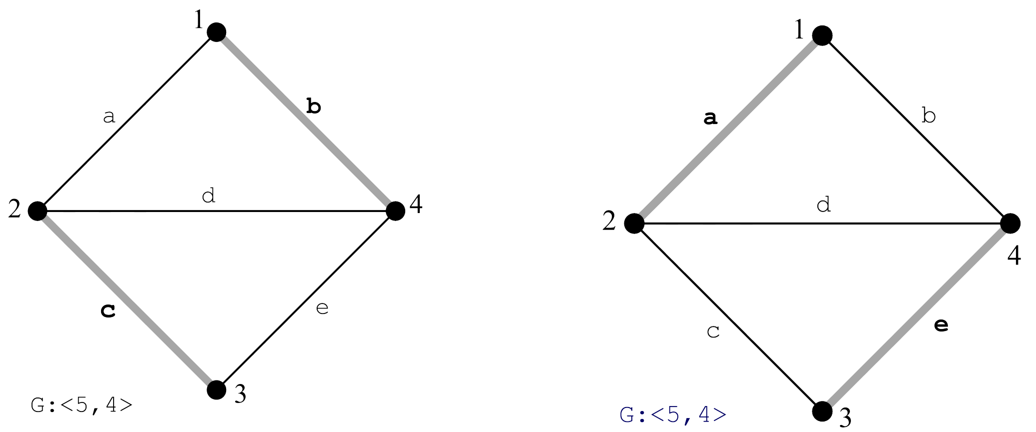


Fig. 3.1. Perfect matchings of an undirected graph

Example 3.12 . Given the following matrix B

$$B = \begin{pmatrix} 0 & a & 0 & b \\ -a & 0 & c & d \\ 0 & -c & 0 & e \\ -b & -d & -e & 0 \end{pmatrix}$$

Each non-zero term of the Pfaffian $\texttt{pf}\ B = b\ c_+\ a\ e$ refers to a perfect matching. The matchings are shown in heavy gray edges in Fig.3.1.

4 The counting problem

4.1 Estimates of the number $\Phi(G)$

Let us have some estimation of the number of perfect matchings [17] .

Theorem 4.1 . *Given a k-connected graph G containing at least one perfect matching. Then the number of perfect matchings in G is at least the double factorial number*

$$k\ !! = \prod_{i=0}^{(k-2)/2} (k - 2\ i).$$

[17] Lovász and Plummer [LoPl86] give more formulas, proofs and results for any types of graph pp. 345–349.

4.2 Permanent and Pfaffian

The following inequality may be useful for an estimation of the permanent

$$\text{per } A \leq (r_1!)^{1/r_1} \ldots (r_n!)^{1/r_n},$$

where $r_1 \ldots r_n$ are the row sums.

Definition 4.2 . Assign orientations to the edges of a graph G to have the directed graph $\boldsymbol{G} = (a_{ij})_{n\times n}$. The associated Tutte matrix is defined by the following skew symmetric adjacency matrix

$$A_s(\boldsymbol{G})_{ij} = \begin{cases} 1 & \text{if edge } (u_i u_j) \in E(\boldsymbol{G}), \\ -1 & \text{if edge } (u_j u_i) \in E(\boldsymbol{G}), \\ 0 & \text{otherwise.} \end{cases}$$

When we have

$$B = \begin{pmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & 1 & -1 \\ 0 & -1 & 0 & -1 \\ -1 & 1 & 1 & 0 \end{pmatrix}$$

then pf B = 2.

4.3 A probabilistic estimate

A probabilistic method is presented in Lovász and Plummer [LoPl86] for counting perfect matchings on a graph.

Theorem 4.3 . *Let G be a graph. Let* $\boldsymbol{G}$ *be a random orientation of G obtained after orienting each edge independently with equal probability. Let* $A_s(\boldsymbol{G})$ *be the skew symmetric adjacency matrix. Then the matching number* $\Phi(G)$ *is the expected value of* `det` $A_s(\boldsymbol{G})$ [18] .

Example 4.4 A Monte Carlo type algorithm is suggested by Lovász and Plummer [LoPl86] . The orientation of edges is done randomly to calculate the matching number $\Phi(G)$. In this study, the number of experiments has been fixed at N = 10000. Then N random orientations of G : $G^1, \ldots, G^N$ are

[18] A proof of the Theorem 4.3 is in Lovász and Plummer [LoPl86], page 330

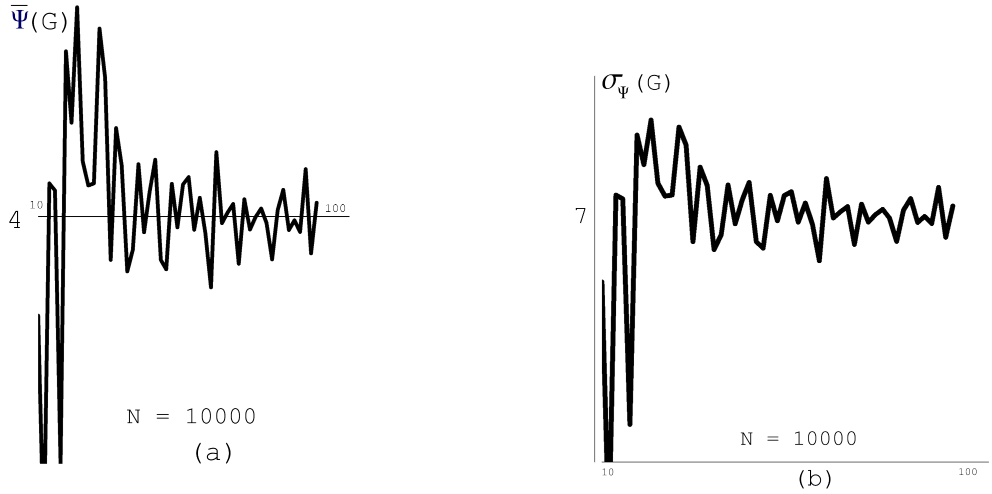


Fig. 4.1. Convergence of estimates of $\Phi(G)$ and $\sigma_{\Phi(G)}$

generated. The estimate of $\Phi(G)$ is

$$\Psi(G) = \frac{1}{N}\sum_{k=1}^{N} \texttt{det}\ A_s(G^k).$$

As N is a large number, we expect that $\Psi(G)$ will be close to the matching number with a large probability. A $\Psi(G) = 4$ has been obtained. The Fig.4.1 shows a rapid convergence of the mean and standard error evaluations when N is varying from 10 to 100.

The application of the method of variables (section 7) allows an identification of these four matchings. We have

$$\begin{aligned}
\mathcal{M}_1 &= \{(1a), (2c), (3e), (4b), (5d)\}\\
\mathcal{M}_2 &= \{(1b), (2a), (3e), (4c), (5d)\}\\
\mathcal{M}_3 &= \{(1a), (2c), (3d), (4b), (5e)\}\\
\mathcal{M}_4 &= \{(1b), (2a), (3d), (4c), (5e)\}
\end{aligned}$$

The exhaustive list of perfect matchings is shown in Fig.4.2.

5 Maximal matching and network flow solutions

Two solutions are considered here : the maximal matching and the minimal cost solutions.

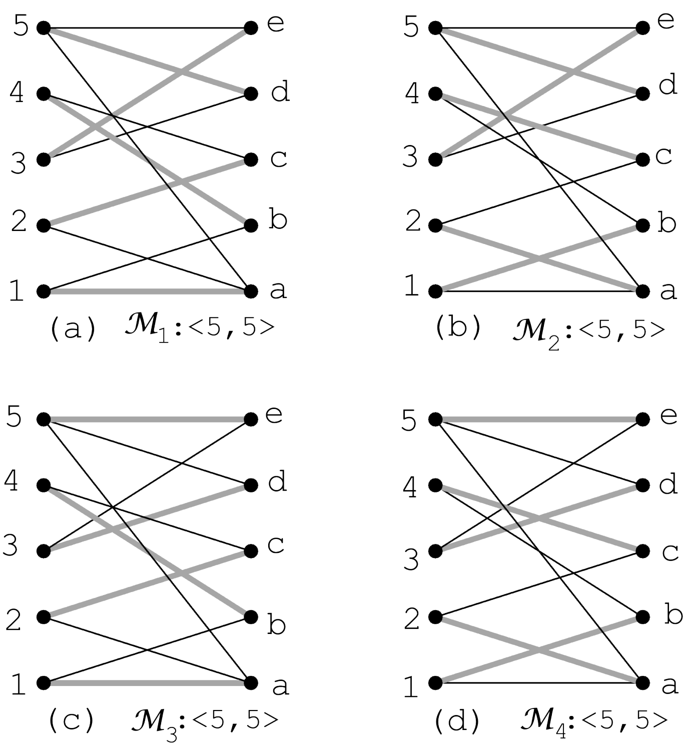


Fig. 4.2. All perfect matchings

5.1 The maximal matching solution

Given a bipartite graph $G = (U \bigcup W, E)$ A matching matches each vertex in U to one in W. A polynomial-time matching algorithm follows the statement of Berge's Theorem : a matching is maximum if and only if it contains no augmenting path. The algorithm starts with an arbitrary matching. This matching may then be improved by finding, if any, an $\mathcal{M}$-augmenting path that starts and ends at an unmatched vertex. Then $\mathcal{M}$ is replaced with the symmetric difference $(\mathcal{M} - P) \bigcup (P - \mathcal{M})$. The matching is maximum when it contains no augmenting path. How to find a maximum matching with as many edges as possible ? Let us take an arbitrary matching in G(V,E) with bipartition (U,W) say $\mathcal{M} = \{(u_1 w_1), (u_3 w_3)\}$. Then start at an unmatched vertex and look for a path P where edges from $E \setminus \mathcal{M}$ and edges from $\mathcal{M}$ alternate. Here

Example 5.1

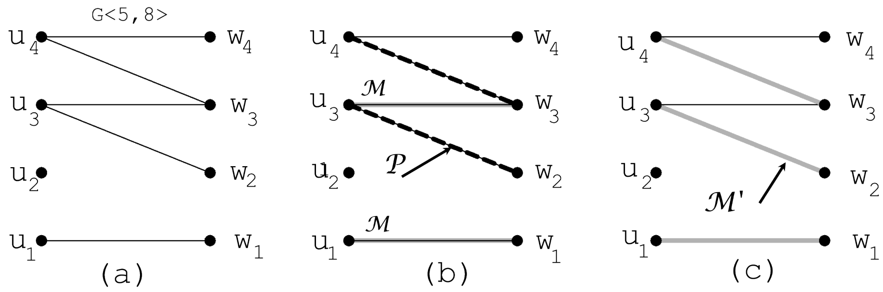


Fig. 5.1. Augmented matching

we clearly have $\mathcal{M} = \{(u_4 w_3), (u_3 w_3), (u_3 w_2)\}$. The symmetric difference [19] of $\mathcal{M}$ with $\mathcal{P}$ produces the new matching $\mathcal{M}' = \{\{u_1, w_1\}, \{u_3, w_2\}, \{u_4, w_3\}\}$ with augmented cardinality $|\mathcal{M}| = |\mathcal{M}| + 1$. The result is shown in Fig.5.1.

The calculations are described in the algorithm 5.1 for finding bipartite matchings [20].

5.2 The network flow solution

Definition 5.2 . Given an undirected graph G(V,E) with bipartition (U,W), a network flow (G, c, s, t) introduces two additional vertices. The source s is connected to every vertices of set U, and a sink t which is connected to all vertices of set W. A non-negative capacity function is $c : E(G) \mapsto \mathbb{R}_+$.

The objective of the maximal flow problem is to determine the maximum amount to carry in G from s to t. A pseudoflow is a function $f : E(G) \mapsto \mathbb{R}_+$ in the network (G, c, s, t) if it satisfies the capacity constraint (i)and the flow antisymmetric constraint (ii).

Definition 5.3 . The value of a flow is represented by the expression

$$\mathrm{val}(f) = \sum_u f(s, u) - \sum_u f(u, s).$$

Definition 5.4 .An excess function is such that $e_f : V(G) \mapsto \mathbb{R}$. The net

[19] The symmetric difference is calculated by $\overline{\mathcal{M}} \bigcup \overline{\mathcal{P}}$ or $\overline{\mathcal{M} \bigcap \mathcal{P}}$ according to the second Morgan law.
[20] adapted from [PaSt82] page 224.

Algorithm 5.1 : BIPARTITE MATCHING ALGORITHM
input: a bipartite graph G(V,E) with partition (U,W)
output: a maximum matching $\mathcal{M}$ represented by the array *mate[ ]*
label[stage]
begin
for all $u \in U \bigcup W$ **do** $mate[u] := 0$; \\ initialization
begin
for all $u \in U \bigcup W$ **do:**
expoxed[v] $:= 0$;
$A := \emptyset$; \\ construction of the auxiliary graph
for all $[u, w] \in E$ **do:**
if *mate[w]* $== 0$ **then** *exposed[u]* := w **else:**
if $mate[w] \neq u$ **then** $A := A \bigcup (u, mate[w])$
$Q := \emptyset$;
for all $u \in U$ **do if** mate[u] == 0 **then:**
$Q := Q \bigcup \{u\}$, *label[u]* $:= 0$
while $Q \neq \emptyset$ **do:**
begin
let w be a vertex in Q;
remove w from Q;
if $exposed[u] \neq O$ **then** augment[u], **go to** [stage]
else
for all unlabeled u' such that $(u, u') \in A$
label[u'] := u, $Q := Q \bigcup \{u'\}$;
end
end
end
procedure *augment[u]*
if *label[u]* $= 0$ **then:**
$mate[u] := exposed[u]$, $mate[exposed[u]] := u$;
else begin:
$exposed[label[u]] := mate[u]$;
$mate[u] := exposed[u]$;
$mate[exposed[u]] := u$;
$augment(label[u])$;
end

flow into v is defined by

$$e_f(v) = \sum_{u \in V} f(u,v) \; - \sum_{w \in V} f(u,v).$$

(i) $f(u,w) \leq c(u,w)$ for all (uw) in $E(G)$
(ii) $f(u,w) = -f(w,u)$

The conservation flow constraint states that

$$e_f(v) = 0 \texttt{ or } \sum_u f(u,v) = \sum_w f(v,w), \texttt{for all } v, w \in V(G)$$

.

Definition 5.5 .The residual flow graph is $G_f = (V, E_f)$ where $E_f = \{(v,w) \in E | c_j(v,w) > 0\}$. It may contain two edges
(i) an edge (ij) with weight $c(i,j) - f(i,j)$ if $c(i,j) - f(i,j) > 0$
(ii) an edge (ji) with weight f(i,j) if $f(i,j) > 0$.

$$\bar{V} \;= \{s,t\} \bigcup V$$

$$\bar{E} \;= \{s \rightarrow u | u \in U\} \; \bigcup \; \{w \rightarrow t | w \in W\}\} \bigcup \{u \rightarrow\rightarrow w | u - w \; in \; G\}$$

The problem of a maximum matching in a bipartite graph $G = (V, E)$ may be solved by a network flow with a complexity of $\mathcal{O}(|V|^{1/2} \times |E|)$ [HoKa75] [21] . A cut splits the vertices into two sets S and T, such that $s \in S$ and $t \in T$. There are $2^{|V|-2}$ possible cuts. The capacity of a cut (S,T) is

$$c(S,T) = \sum_{(uv) \in E c(u,v)} \;, \; \forall \; u \in S \text{ and } v \in T$$

.

Lemma 5.6 .*In every network* (G, c, s, t) *a maximum exists.*

Proof [LoPl86] . Each flow is a point of $\mathbb{R}^m$ where m is the size of the graph. The set of points being compact and the function val(f) being a continuous

[21] The big O notation refers to the following property : $\mathcal{O}(g(n)) = \{f(n) : \exists (c, N) \in \mathbb{R}_{*+} | 0 \leq f(n) \leq c.g(n), \forall n \geq N\}$. Thus any linear function as $f(n) = \sum_{i=1}^{n} i$ is in $\mathcal{O}(n^2)$, since

$$f(n) = \frac{n(n+1)}{2} < n^2 + n \leq 2 \; n^2 < c \; n^2, \; \forall n.$$

.

function of m variables, the network reaches its maximum on the set by the Bolzano-Weierstrass Theorem. □

Menger's Theorem is an essential result about connectivity in finite undirected graphs. Let us present the edge-connectivity version of the Theorem. It states that the size of the minimum edge cut for two non adjacent vertices u and w is equal to the maximum number of pairwise vertex-independent paths from u to w.

Theorem 5.7 *.(Menger's Theorem 1927). Let G be an undirected graph with two distinguished vertices s and t. Then the minimum size of any s-t separating set of edges is equal to the maximum number of edge-disjoint s-t paths.*

Proofs are given by Lovász and Plummer [LoPl86]. The max-flow min-cut Theorem is a generalization which states that the maximal amount of a flow is equal to the capacity of a minimal cut.

Theorem 5.8 *.(Max-flow min-cut Theorem). If G is a digraph with source s and sink t, the maximum value of any s-t flow equals the minimum capacity of any s-t cut*[22] .

Lovász and Plummer [LoPl86] prove this theorem using two lemmas : the first lemma states that in every network (G, c, s, t) a maximum flow exists and the second that if f is any flow in G and c is a s-t cut the value of flow $\texttt{Val}(f)$ is less or equal to capacity of C $\texttt{Cap}(C)$. Then a flow f in a network is a maximum flow iff an f-augmenting path does not exists.

Example 5.9 One example is an application of the flow theory to matching problems. Fig.5.2 shows the original directed graph and the residual graph. The original graph has been constructed, using the undirected graph G whose bipartition is (U,W), adding the source-vertex s and the sink-vertex t, and orienting the edges from s to t. The edges joining s to the $u_i's$ have a capacity of 1. The edges joining the $w_j's$ to t have also a capacity of 1. In the residual graph, there is no directed path from s to t, since it would mean that some flow could carry more. Then we could not have a maximum. The source - sink partition is $S = \{s, u_1, u_2, u_3, u_4, u_5, w_1, w_2, w_3, w_4\}$ and $T = \{t\}$. All edges from S to T are saturated as they carry the maximum flow and all edges from T to S carry no flow. The maximum flow from s to t will correspond to a maximum matching. Indeed it will find a largest set of vertex-disjoint paths

[22] The König' s theorem and Hall's theorem can be derived from this max-flow min-cut Theorem.

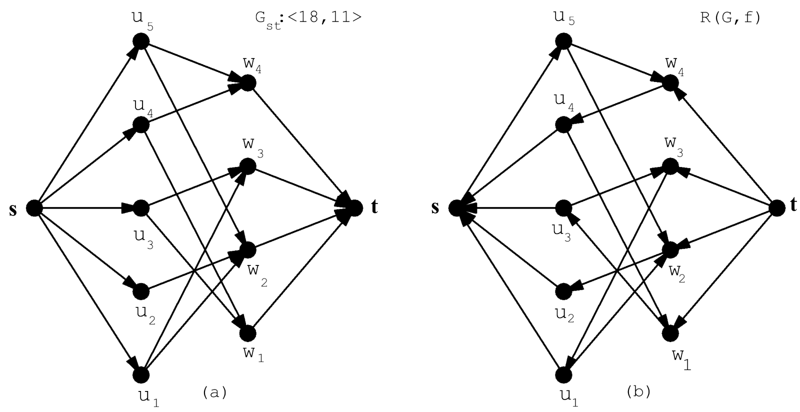


Fig. 5.2. Original and residual flow graph

which consist od disjoint edges from G. Finally the maximum matching for that example is

$$\mathcal{M} = \{(u_1 w_3), (u_2 w_2), (u_3 w_1), (u_4 w_4)\}$$

6 The combinatorial algorithms

We will examine some major algorithms and methods together with simple and economic examples : the Ford - Fulkerson algorithm, the Edmonds - Karp algorithm, the hungarian method and linear programming.

6.1 The Ford - Fulkerson algorithm

Let $G(U \bigcup W, E)$ be a bipartite graph and let $\mathcal{M}$ be any matching in G. Suppose U_1 and W_1 are the sets of unmatched (or exposed) vertices. We aim to find an $\mathcal{M}$-augmenting path [23], if any, connecting U_1 to W_1. We consider the set of vertices in U accessible from U_1 on an $\mathcal{M}$-alternating set. The following algorithm [24] uses the graph form of G. [25]

[23] A path $P = \{v_1, \ldots, v_m\}$ is an alternating path with respect to the matching $\mathcal{M}$ if $(v_i v_{i+1}) \in \mathcal{M}$ iff $(v_{i+1} v_{i+2}) \notin \mathcal{M}$ for $1 \leq i \leq m-2$.
[24] adapted from [GrYe03] page 1109.
[25] Another algorithm proposes to use the matrix form. Such algorithm consists in two phases the labeling and the flipping phase.

Algorithm 6.1 : FORD-FULKERSON ALGORITHM
input: a bipartite graph $G(U \bigcup W, E)$ and any arbitrary matching $\mathcal{M}$
output: a maximum matching $\mathcal{M}$
begin
$\mathcal{M} := \emptyset$;
done := False;
while not done **do:**
let *free*; \\ the set of unmatched vertices
$S_U := U \bigcap free$;
$seen := \emptyset$;
$stillooking := True$;
while *stillooking* **do** : \\ for an augmenting path
$S_W := \{w \mid w \notin seen$ and (uw) ∈ E, u ∈ $S_U\}$;
if $S_W \bigcap free \neq \varnothing$ **then** $\mathcal{M} := \mathcal{M} \bigoplus P$;
\\ an augmenting path exists
\\ constructing an augmenting path P
$stillooking := False$;
else \\ continue looking for an augmenting path
$seen := seen \bigcup S_W$;
$S_U := \{u \mid (w, u) \in \mathcal{M}, w \in S_W\}$;
if $S_U == \emptyset$ **then:**
$stillooking := False$;
$done := True$;
end
end
end

Example 6.1 .In the example of Fig.6.1 an arbitrary matching is shown $\mathcal{M} = \{(3c), (4d), (5b)\}$ in gray heavy lines. At the next iteration we have the set of free vertices $free = \{1, 2, a\}$. Then

$$S_U = U \bigcap free = \{1, 2, 3, 4, 5\} \bigcap \{1, 2, a\} = \{1, 2\}.$$

We have $seen = \emptyset$ and deduce

$$\begin{aligned} S_W &= \{w \mid w \notin seen \text{ and } (u, w, (c3), (3a)\}) \in E, u \in S_U\} \\ &= \{b, c\}. \end{aligned} \tag{6.1}$$

Since the vertices b and c are both matched the algorithm continues. We have

$$S_W \bigcap free = \{b, c\} \bigcap \{1, 2, a\} = \emptyset.$$

Then $seen = \emptyset \bigcup S_W = \{\emptyset, b, c\}$. We have

$$S_U = \{u|(wu) \in \mathcal{M}, w \in S_W = \{3, 5\},$$

and according to 6.1 we have $S_U = \{a, d\}$. The vertices a and d are free (or unmatched). We then calculate

$$S_W \bigcap free = \{a, d\} \bigcap \{1, 2, a\} = \{a\} \neq \emptyset.$$

We deduce that an augmenting path exists. In Fig.6.1 the Breadth-First Search (BFS) tree has been rooted at the unmatched source 1. The augmenting path P is $P = \{(1c), (3a), (4d), (5b)\}$ with two unmatched endpoints. The new matching is

$$\begin{aligned} \mathcal{M} = \ & \mathcal{M} \bigoplus P \\ = \ & \{(3c), (4d), (5b)\} \bigoplus \{(1c) \\ = \ & \{(1c), (3a), (4d), (5b)\}. \end{aligned}$$

At the next iteration, we have $free = \{2\}$ and

$$S_U = U \bigcap free = \{1, 2, 3, 4, 5\} \bigcap \{2\} = \{2\}.$$

Following 6.1 we deduce $S_W = \{b\}$ and

$$S_W \bigcap free = \{b\} \bigcap \{2\} = \emptyset,$$

such that there no augmenting path. Looking for a further augmenting path, we calculate

$$seen = seen \bigcup S_W = \emptyset \bigcup \{b\} = \{\emptyset, b\}.$$

Then we have $S_U = \{5\}$ and $S_W = \{d\}$. Hence $S_W \bigcap free = \emptyset$, such that no augmenting path exists. We continue looking for an augmenting path with

$$seen = \{b\} \bigcup \{d\} = \{b, d\},$$

and finally $S_W = \emptyset$, $S_U = \emptyset$. No further augmenting path has been found and the algorithm terminates with the maximum matching

$$\mathcal{M} = \{(1c), (3a), (4d)\}.$$

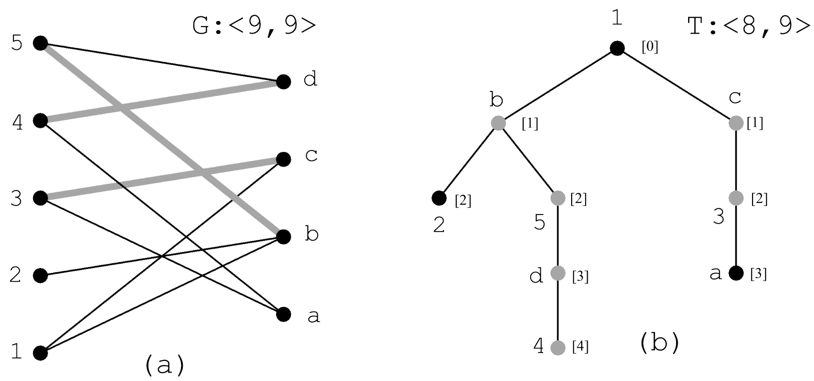


Fig. 6.1. Arbitrary matching and augmenting path tree

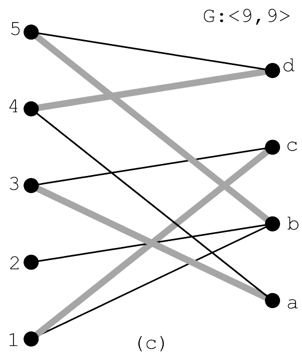


Fig. 6.2. Final maximum matching

This maximum matching is shown in Fig.6.2.

The complexity of the Ford-Fulkerson algorithm [26] is of $\mathcal{O}(\mathrm{E} \times maxflow)$ [27].

[26] The complexity of an algorithm is captured by the time or the number of steps it takes to complete a problem of size n.

[27] The specialized algorithm of Edmonds-Karp finding paths with breadth-first search is of $\mathcal{O}(|V| \times |E|^2)$ time complexity. There are many other ways to solve this maximum flow problem.

6.2 The Edmonds - Karp algorithm

A simple implementation is taken from electronic encyclopedia Wikipedia [28]. This algorithm is similar to the Ford-Fulkerson algorithm except that the augmenting path must be the shortest. The matching problem can be solved in $\mathcal{O}(|V|^2 \times |E|)$.

6.3 The hungarian method

Definition 6.2 . A vertex cover V_1 of a graph $G = (V, E)$ is a subset of vertices such that every edge in E is incident on a vertex in V_1.

If $\mathcal{M}$ is a matching in a graph G , then any vertex cover of G must contain at least $|\mathcal{M}|$ vertices. Thus taking both endpoint of all edges produces a vertex cover since if any edge not covered could be added to $\mathcal{M}$ to give a larger matching. There is a duality between the weighted matching problem and the weighted vertex cover problem which can be exploited to produce polynomial-time solutions. A vertex cover seeks to find the costs $u_i, 1 \le i \le n$ and $v_j, 1 \le j \le n$ such that for each i, j the sum of costs is minimum and $u_i + vj \ge 1$. The cost of a cover (u, v) is

$$c(u, v) = \sum_i u_i + \sum_j v_j.$$

Let p($\mathcal{M}$) be a perfect matching $w(\mathcal{M})$ is its cost. The duality Theorem shows the interconnection between the minimum weight matching problem and the minimum cost vertex cover problem.

Theorem 6.3 .*(The duality Theorem). For any vertex cover (u,v) and any perfect matching $|\mathcal{M}|$ we have $c(u, v) \ge w(\mathcal{M})$. Furthermore $c(u, v) = w(\mathcal{M})$ iff every edge (i, j) in $\mathcal{M}$ satisfies $u_i + v_j = w_{ij}$. $\mathcal{M}$ is then the maximum matching and (u,v) a minimum vertex cover.*

Kuhn [Ku55]proposed a polynomial-time Hungarian algorithm.

6.4 Linear programming

The primal - dual linear programming problems

Given a graph G with bipartition $V(G) = (U, W)$. Let us denote $\nabla(v)$ the set of edges incident to v. A 0-1 vector x in $\mathbb{R}^{E(G)}$ is the incidence vector of a

[28] Adapted from Wikipedia, the free encyclopedia
http://en.wikipedia.org/wiki/Edmonds-Karpalgorithm

Algorithm 6.2 : EDMONDS-KARP ALGORITHM
input: a network N(G) of bipartite graph G
output: the maximum flow in a flow network

```
begin
    n = len(C); \\ the capacity matrix
    F = [[0] * n for i in xrange(n)]
    \\ the residual capacity from u to v is C[u][v] − F[u][v]
    while True :
        path := bfs(C, F, source, sink);
        while \\ for an augmenting path
        if not path :
            break
        flow := float("infinity");
        \\ traverse path to find smallest capacity
            flow := min(flow,C[u][v] - F[u][v]) ;
        for (u,v) in path :
        \\ traverse path to update flow
        for (u,v) in path :
            F[u][v] += flow ;
            F[v][u] -= flow ;
    return sum([F[source][i] for i in xrange(n)])
end
procedure bfs(C,F,source,sink
    queue = [source];
    paths = {source: [ ]}
    while queue :
        u := queue.pop(0)
    for v in xrange(len(C)):
        if C[u][v] − F[u][v] > 0 and v not in paths :
            paths[v] := paths[u] + [(u, v)];
            if v == sink :
                if v == sink :
                    return paths[v]
                queue.append(v)
end
```

$$
\begin{array}{c|c}
 & \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 & e_9 \end{matrix} \\
\begin{matrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{matrix} &
\begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{pmatrix}
\end{array}
\text{ vertex \#}
$$

edge #

Fig. 6.3. The incidence matrix

matching iff $x(\nabla(v)) \leq 1$ for every vertex $v \in V(G)$. The linear programming problem is

$$
\begin{aligned}
\text{maximize } & \mathbf{1} \,.\mathbf{x} \\
\text{subject to } & A \,.\mathbf{x} \leq \mathbf{1} \\
& \mathbf{x} \geq 0,
\end{aligned}
$$

where $A = (a_{ve})$ is the incidence matrix of G defined by

$$
a_{ve} \; = \begin{cases} 1 & \text{if the vertex v is incident on edge e,} \\ 0 & \text{otherwise.} \end{cases}
$$

Example 6.4 . The set of the inequations forms a polytope $\mathcal{M}(G)$. The solutions are those which maximize the objective function $\mathbf{1}.\mathbf{x}$. In this example the incidence matrix is in Fig.6.3 has the following edges : $e_1 = (1b),\ e_2 = (1c),\ e_3 = (2b),\ e_4 = (3a),\ e_5 = (3c),\ e_6 = (4a),\ e_7 = (4d), \mathrm{e}_8 = (5b),\ e_9 = (5d)$.

The system of inequality constraints is

$$\begin{array}{llllll}
x_1 & +x_2 & & \leq 1 & x_1 & \geq 0 \\
x_3 & & & \leq 1 & x_2 & \geq 0 \\
x_4 & +x_5 & & \leq 1 & x_3 & \geq 0 \\
x_6 & +x_7 & & \leq 1 & x_4 & \geq 0 \\
x_8 & +x_9 & & \leq 1 & x_5 & \geq 0 \\
x_4 & +x_6 & & \leq 1 & x_6 & \geq 0 \\
x_1 & +x_3 & +x_8 & \leq 1 & x_7 & \geq 0 \\
x_2 & +x_5 & & \leq 1 & x_8 & \geq 0 \\
x_7 & +x_9 & & \leq 1 & x_9 & \geq 0
\end{array}$$

Among the optimal solutions of that system there will be a 0-1 vector.

The problem of finding minimum vertex cover is similar. A 0-1 vector $\mathbf{y}$ is the incidence vector of a vertex cover iff it satisfies

$$y_u + \ y_v \ \geq 1$$

for every $(uv) \in E(G)$. The dual linear program is

$$\begin{aligned}
\text{minimize } & \mathbf{1} \ .\mathbf{y} \\
\text{subject to } & A^T \ .\mathbf{y} \geq \mathbf{1} \\
& \mathbf{y} \geq 0,
\end{aligned}$$

The assignment LP-problem

The assignment LP-problem consists in minimizing the total cost of assigning the u to w under constraints. In condensed form, we have

$$\begin{aligned}
\text{minimize } & \sum_{e \ \in \delta(u)} c(e) \ x(e) \\
\text{subject to } & \sum_{e \ \in\delta(u)} x(e) = 1, \ \forall u \in U \\
& \sum_{e \ \in\delta(w)} x(e) = 1, \ \forall w \in W \\
& x_e \geq 0
\end{aligned}$$

where $v \in U \bigcup W$, $\delta(v)$ denotes the set of incident edges to the vertex v, x is a vector of $\mathbb{R}^E$, and where $c_e(= 0 \ or \ 1)$is the cost of assigning the extremities of edge e=(uw). We have $x_{uw} = 1$ if u is assigned to w. The dual is of the form

$$\begin{aligned} \text{maximize} \sum_{u\ \in U} y(u) + \sum_{w\ \in W} y(w) \\ \text{subject to } y(u) + \ y(w) \ \leq \ w(e), \forall e \ = (uw) \in E \end{aligned}$$

where y is a real vector in $\mathbb{R}^{U \cup W}$.

7 Finding all the perfect matchings

7.1 The method of variable

The method of variable is presented by Lovász and Plummer [LoPl86]. For each edge of a bipartite graph $e \in E(G)$, let x_e be a variable such that $A(x) = (a_{ij})$ with

$$a_{ij} \ = \begin{cases} x_e & \text{if } u_i \text{ and } w_j \text{ are adjacent and } e = (u_i w_j), \\ 0 & \text{otherwise.} \end{cases}$$

`det` A(x) is a polynomial of the variables x_e. Every expansion term corresponds to a perfect matching p($\mathcal{M}$)

Example 7.1 . Let us consider the growth model. The six matchings of Fig.7.1 have been obtained using the method of variables. Among these matchings three of them may have an theoretical interpretation in economics. Thus the matchings (b), (d) and (e) of the Fig.7.1 correspond to a wicksellian or a friedmanian or an extreme monetarist interpretation. [Ve81] [29] .

The directed graphs that correspond to such assignments are shown in Fig.7.2.

7.2 Finding another perfect matching

Example 7.2 The procedure for going from a perfect matching to another is shown in Fig.7.3. It consists first in starting with a perfect matching like

[29] Briefly according to the wicksellian conception the economy fluctuates as the consequence of a gap the real and the natural interest rates. According to the friedmanian conception the level of activity is depending on the Phillips curve : the fluctuations around the equilibrium path are due to forecasting errors on inflation rate, it determines the nominal interest rate and the real interest rates adjust the market of goods. The Fisher relation calculates the nominal interest rate. According to the extreme monetarist interpretation the economic fluctuations are imputable to the money market. The real interest rate equilibrates the market of goods.

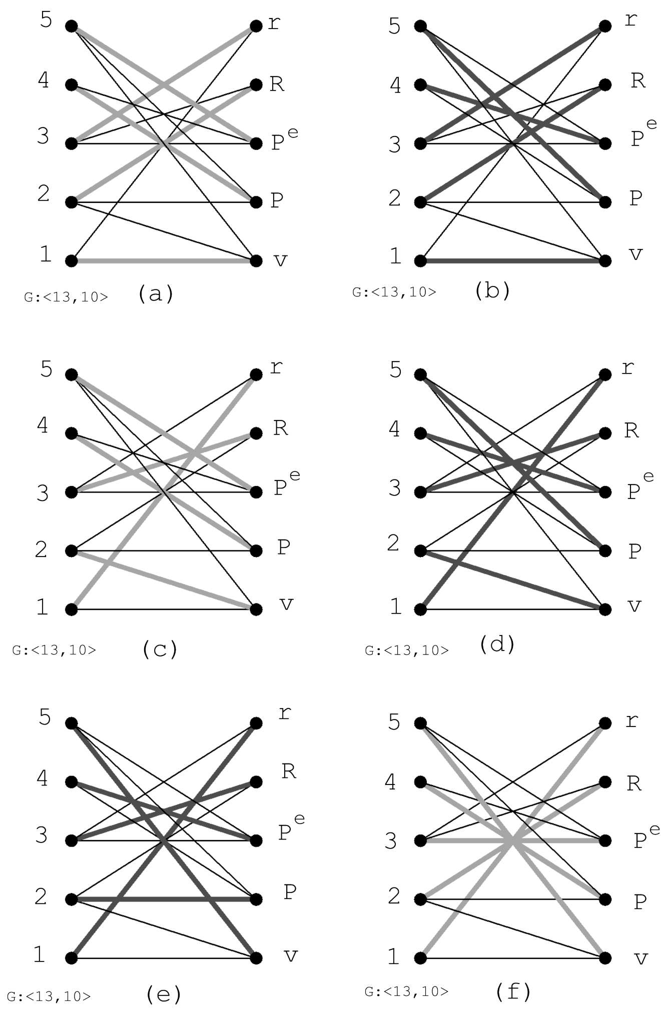


Fig. 7.1. Perfect matchings of the growth model

matching (a) of Fig.7.1 for the growth model, in looking for a shortest alternating cycle like $(4P), (P5), (5P^e), (P^e)5$ and finally take a symmetric difference to obtain a new matching (b) of the Fig.7.1.

7.3 Finding all the perfect matchings

The algorithm of Fukuda and Matsui [FuMa92] uses the Kth best solution of assignment problems (AP) developed by Murty (68') and Chegireddy and Hamacher (87'). The computational time is $\mathcal{O}(n \times (n+m))$ and it requires $\mathcal{O}(n+m)$ memory storage for each additional matching. Their recent algo-

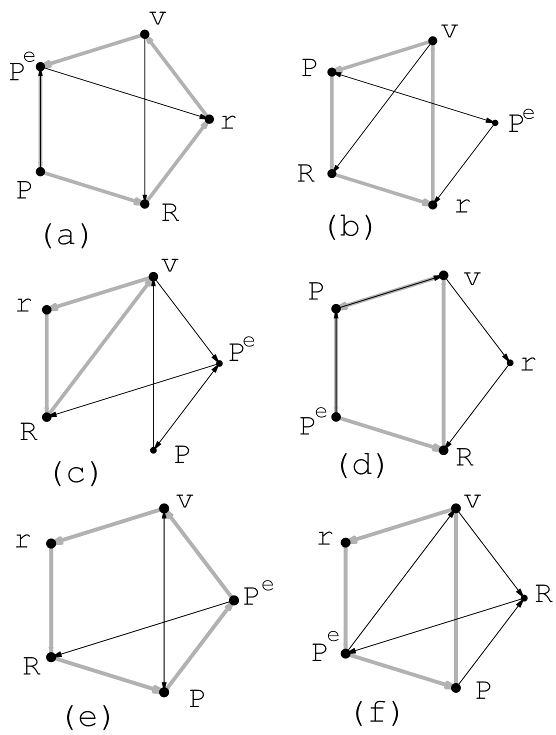


Fig. 7.2. Directed graphs of the growth model

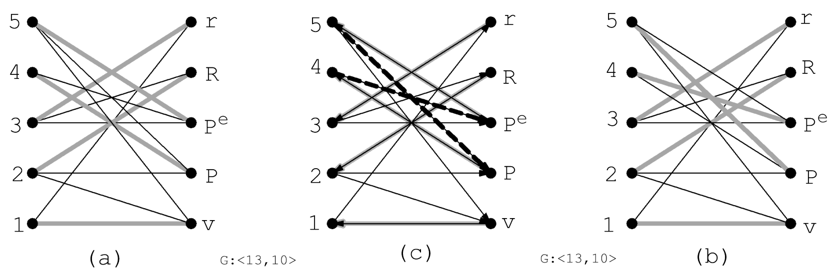


Fig. 7.3. Finding another perfect matching

Author	Method	Complexity	Storage
	Hungarian method	$\mathcal{O}(n^{1/2}m)$	$\mathcal{O}(n+m)$
Murty (1968)	Kth- best solution	$\mathcal{O}(n\ (n\ log\ n+\ m))$	$\mathcal{O}(K\ n^2)$
Edmonds and Karp (1972)	network max flow	$\mathcal{O}(n \times m^2)$	
Chegireddy and Hamacher (1987)		$\mathcal{O}(n\ (n\ log\ n+\ m))$	$\mathcal{O}(Kn)$
Fukuda and Matsui (1993)	Binary partitioning	$\mathcal{O}(n^{1/2}m + N(n+m))$	$\mathcal{O}(Kn^2)$
Uno (1997)		$\mathcal{O}(n^{1/2}\ m + N\ (n+\ m))$	

Table 1
The performances of some algorithms

rithm requires $\mathcal{O}(N(n+m)+n^{5/2})$ computational time and $\mathcal{O}(n \times m)$ memory storage, where N is $\Phi(G)$. First we solve the AP by the Hungarian method and then generate each perfect matching in a lexicographic order. The procedure is based on a binary partitioning where the enumeration problem can be partitioned into two subproblems. It generalizes the Murty's algorithm algorithm for ranking the solutions of APs. Uno (97') proposed a new approach in two phases called trimming and balancing.

The Table 1 shows the method and the performances of some algorithms used for finding all perfect matchings. The order of the graph is denoted by n and its size is m. The computational time is given for each additional perfect matching. The number of perfect matchings is N.

8 Application to a large size empirical model

8.1 Economic description

The annual macroeconomic model Micro-DMS (Dynamic Multi-Sector) for the French economy consists in 82 equations with 22 behavioral equations and 69 exogenous variables. This model involves different theoretical options. The following elements have been retained in this study : a Cobb - Douglas production function(a Putty-Clay technology [30]), an accelerator - substitution function type for investments, a saving equation, exports and imports functions depending on demand, disposable capacity and price competitiveness, a Phillips type wage equation, production prices depending on wage costs, profits and capacities, exogenous interest rates and exchange rate.

8.2 Matchings

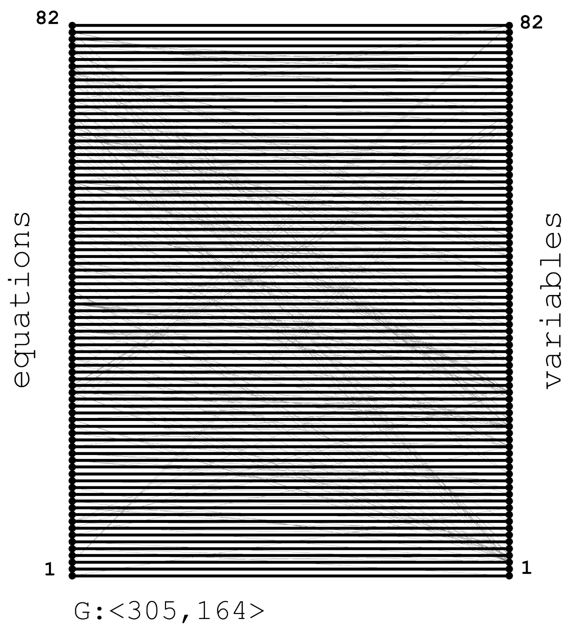
A maximal matching and a minimal matching have been calculated for this model. The maximal matching is at the top of Fig.8.1 corresponds exactly to the chosen solution of the model builders. This minimal solution is at the bottom of Fig.8.1. It differs but this solution cannot be retained because of causalities that have no economic sense.

8.3 Graph representations

The circular embedding is drawn with the longest circuit : big black points are those of the longest circuit, big gray points the remaining vertices of the SC, small black points the remaining vertices of the graph [31]. The model has 143 circuits with a maximal length of 23 vertices.The directed graph embedding is the one obtained with the maximal matching. The acyclic DAG clarifies the whole structure and classifies the variables between those which stay upwards or downward are the interdependent variables. The resolution and the analysis of the model will take advantages from this information. Both graph embeddings are shown in Fig.8.2.

[30] In this case, capital goods embody the technology at the time of their creation. Ex ante the choice of capital intensity is based on a Cobb - Douglas production function. Ex post the production function has the Leontief form with fixed coefficients.

[31] This graph which has been obtained applying permutation matrices is isomorphic to the initial one.



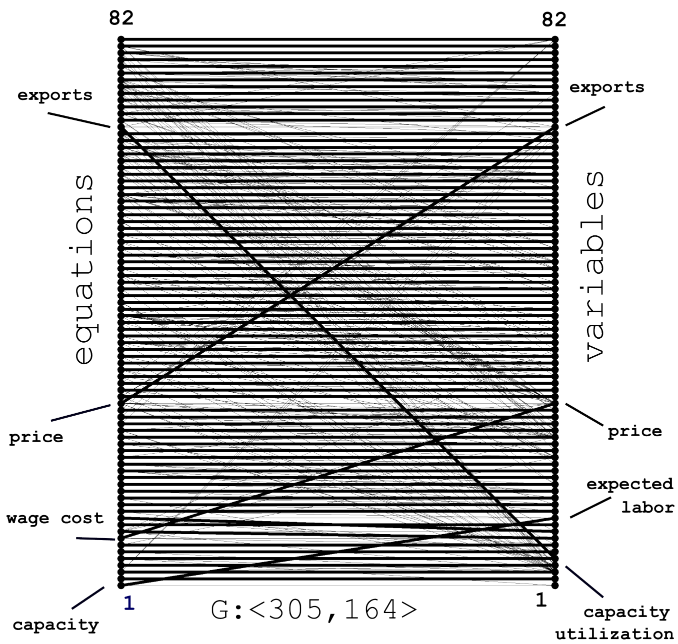


Fig. 8.1. Maximal and minimal matchings of the model DMS

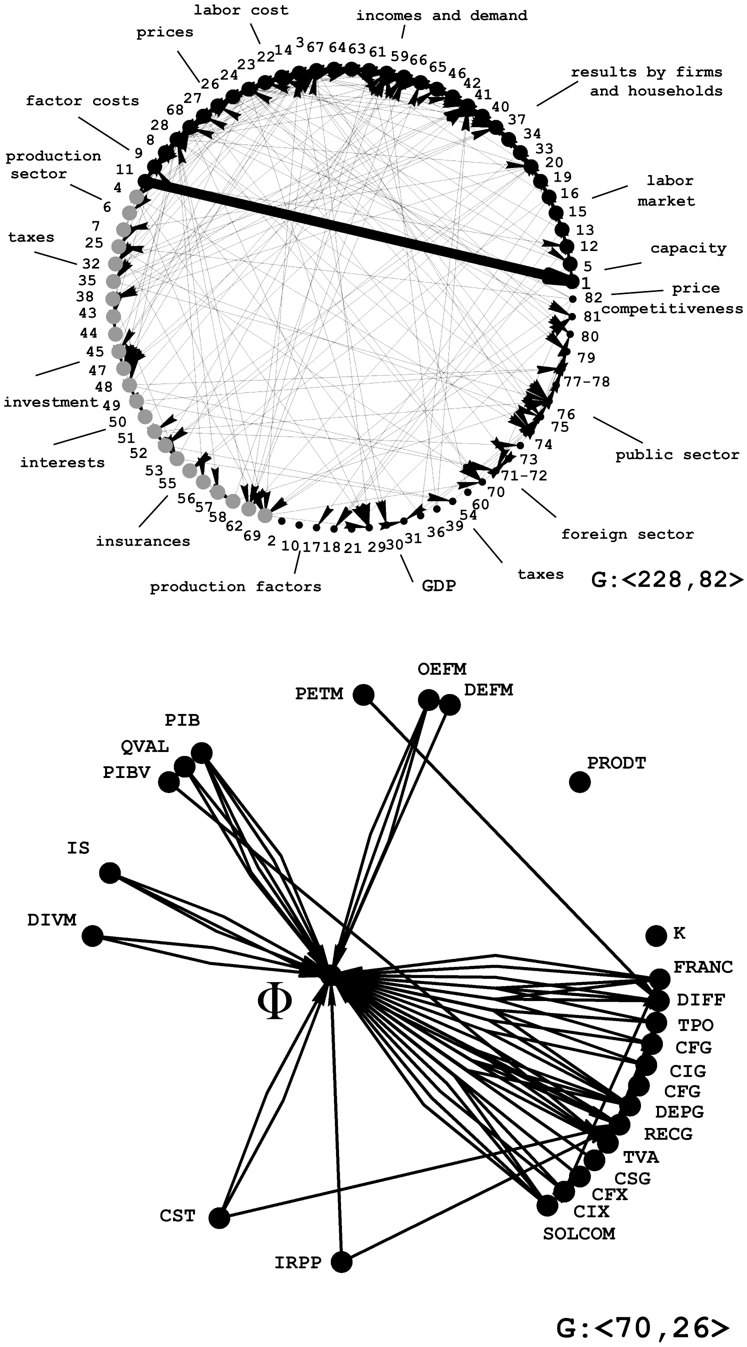


Fig. 8.2. Directed graph and DAG of the model DMS

References

[Br97] Brillet,J. L., "Win-MCD. Simulation sous Windows de la maquette économique Micro-DMS", INSEE Guides N^0 5–6, INSEE, Paris, 1997.

[ChZh05] Chartrand, G. and P. Zhang, "Introduction to Graph Theory", McGraw Hill Higher Education, New York, 2005.

[ChHa87] Chegireddy, C.R. and H.W. Hamacher,*Algorithms for Finding K-best Perfect Matchings*, Discrete Appl. Math. **18** (1987) 155-165.

[Di05] Diestel, R., "Graph Theory", third edition, Springer, Graduate Texts in Mathematics, Berlin, Heidelberg, 2005.

[Di70] Dinic, E.A., *Algorithm for solution of a problem of maximum flow in a network with power estimation*, Soviet Math. Dokl. **11** (1970) 1277– 1280.

[EdKa72] Edmonds, J. and R.M. Karp, *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems*, J. ACM, **19**(2) (1971) 248-264. With file Edmonds.pdf at the internet address
http://www.akira.ruc.dk/ kel/teaching/algorithmdesignf05/Atikler/08/

[Eve79] Even, S., "Graph algorithms", Computer Science Press, Rockville (Ma), UK, 1979.

[FoFu62] Ford, L. R. and D.R. Fulkerson, "Flows in Networks", Princeton University Press, Princeton, NJ, 1962.

[FuMa92] Fukuda, K. and T. Matsui, *Finding All Minimum-Cost Perfect Matchings in Bipartite Graphs*, Networks **22** (1992), 461–468.

[FuMa94] Fukuda, K. and T. Matsui, *Finding All the Perfect Matchings in Bipartite Graphs*, Appl. Math. Lett. **7**(1) (1994), 15–18.

[GoTaTa90] Goldberg, A. V., Tardos, É. and E. Tarjan, Network Flow Algorithms in Algoritms and Combinatorics, **9**, B., Lovász, H.J. Prömel and A. Schrijver (editors), Berlin, Heidelberg, Springer Verlag (1990), 101–164. With internet address
http://www.es.cornell.edu/People/eva/Network.Flow.Algorithms.pdf

[GoMi84] Gondran, M. and M. Minoux, "Graphs and Algorithms", John Wiley, New York, N.Y.,1984.

[GrYe03] Gross,J.L. and J. Yellen (editors), "Handbook of Graph Theory", in Discrete Mathematics and Its Applications, series Editor Kenneth H. Rosen,CRC Press, New York, N.Y.,2003.

[GrYe06] Gross,J.L. and J. Yellen , “Graph Theory and its Application”, Chapman-Hall.CRC, 2006.

[HoKa75] Hopcroft, J. and R.M. Karp, *An $O(n^{5/2})$ algorithm for maximum matching in general graphs*, SIAM J. Comput. (1975) 225–231.

[Hu04] Hudson, I., “The maximum flow problem”, Denison University, 2004. With internet address
http://www.denison.edu/mathsci/projects/cs272S04/hudson.pdf

[Ke06a] Keller, A. A.,“The Structure of Economic Models : Graph Theory with Mathematica”,in Y. Papegay (editor), Applied Mathematica Proceedings of the 8th International Mathematica Symposi ISBN 2-7261-1289-7.

[Ke06b] Keller, A. A.,“Graph Theory and Economic Models : from Small to Large Applications”, presented at the Sixth Czech-Slovak International Symposium on “Combinatorics, Graph Theory, Algorithms and Applications”,Prague, July 10-15, 2006.

[Ku55] Kuhn, H. W., *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart. **2** (1955), 83–97.

[LoPl86] Lovász L. and M.D. Plummer “Matching Theory”,Ann. of Discrete Math. **29**, North- Holland, New York, 1986.

[MaSuVi99] Mahajan, M., Subramanya, P.R. and V. Vinay, A Combinatorial Algorithm for Pfaffians, in the Proceedings of the Fifth Annual International Computing and Combinatorics Conference COCOON, in the Springer-Verlag Lecture Notes in Computer Science series Volume 1627 (1999) 134–143.

[Mu68] Murty, K.G, *An Algorithm for Ranking all the Assignments in Order of Increasing Cost*, Oper. Res. **16**(3)(1968) 682–687.

[PaSt82] Papadimitriou C. H. and K. Steiglitz, “Combinatorial Optimization”, Prentice-Hall, Englewood Cliffs, NJ,1982.

[PeSk82] Pemmaraju, S. and S. Skiena, “Combinatorics, and Graph Theory with *Mathematica*®”, Cambridge University Press, Cambridge, UK, 2003.

[Sc97] Schrage, L., “Optimization modeling with LINDO”, 5th edition, THOMSON Brooks /Cole New York, 1997.With internet address
http://www.lindo.com

[Un97] Uno, T., *Algorithms for Enumerating All Perfect, Maximum and Maximal Matchings in Bipartite Graphs*, Lecture Note in Computer Science 1350, Springer Verlag, Algorithms Comput. Math (1997) 92–101.

[Va79] Valiant, L.G, *The Complexity of Enumeration and Reliability Problems*, SIAM J. Comput. **8** (1979) 410-421.

[Ve81] Vedel, Cl., *Politique Monétaire et Politique Budgétaire dans un Modèle Dynamique d'une Economie en Croissance*, Institut international de finances publiques, unpublished, Tokyo, 1981.

[Wo03] Wolfram, S., "The *MATHEMATICA*® Book", 5 th Ed., Wolfram Media Inc., Champain, IL, 2003. With associated web site http://www.wolfram.com

Appendix

The following supplement shows the symbol, the number and definition of the endogenous variables of the Micro-DMS model. The exogenous variables are not considered in this study. The number of a variable refers to the equation that calculates this variable. The variables refer to levels, unless otherwise indicated.

Symbol	Number	Definition
AII	32	other indirect taxes
AUTOF	41	corporate financing from reserves
BFS	47	corporate financing requirements
CAP	1	potential output
CFG	77	nominal government deficit
CFGP	79	ratio of CFG to PIBV
CFM	65	households financing capacities
CFX	72	balance of payments

Table .1
The list of symbols (to be continued)

Symbol	Number	Definition
CI	4	intermediate goods consumption
CIG	78	interest payments of government
CIM	66	interest payments of households
CIS	50	interest payments of firms
CIX	71	interest payments of foreign sector
CO	64	consumer expenditures
CSE	34	employer social insurance contributions
CSG	73	government social insurance contributions
CSS	53	salaries social insurance contributions
CST	54	total social insurance contributions
CSUP	22	unit wage cost
DEFM	17	labor supply
DEPG	76	government expenditures
DI	67	domestic demand
DIFF	81	inflation discrepancy between France and foreign
DIVM	39	household dividends
DSTOC	45	inventories formation
DWB	8	increase rate of wage costs
EBEM	40	household activity results
FFCEI	46	investment of individual firms
FRANC	82	exchange rate of Franc
I	44	capacity investment of firms
ID	43	expected capacity investment
ILOG	62	dwelling investment

Table .2
The list of symbols (to be continued)

Symbol	Number	Definition
IRPP	60	personal income tax
IS	36	profit tax
K	2	capital stock
M	68	imports
MSE	33	wage payments by firms
MSG	51	wage payments by government
MST	52	total amount of wages
N	15	total employment
ND	12	expected labor by firms
NID	11	expected labor on last generation
NE	13	effective labor
OEFM	18	stock of labor demand
PDRE	16	unemployment
PETM	21	foreign production prices
PEX	25	export price
PIB	29	real gross domestic product
PIBV	31	nominal gross domestic product
PIM	27	import price
PIMHE	26	import price excluding energy
PP	24	price of production
PROD	14	observed labor productivity
PRODT	10	underlining labor productivity
PSAUT	57	other social insurance benefits
PSCHO	55	unemployment insurance benefits
PSOC	58	national insurance benefits
PSRET	56	retirement insurance benefits
PU	28	price of domestic demand

Table .3
The list of symbols (to be continued)

Symbol	Number	Definition
PVA	23	price of added value
Q	3	total added value
QD	6	expected variation of production capacities
QVAL	30	nominal added value
RBEI	38	gross profit of individual firms
RDM	61	disposable income of households
RECG	75	government receipts
RM	59	taxable income of households
SOLCOM	70	commercial balance
SUBV	35	subsidies of firms
TCHO	19	unemployment rate
TEPA	63	saving rate of households
TI	48	interest rate
TIM	49	average interest rate
TMARG	37	profit margin rate of firms
TPO	80	tax rate
TPRO	42	corporate profit
TVA	74	TVA receipts
UT	5	capacity utilization rate
UC	9	capital cost
W	20	wage rates
WB	7	wage rate and employer contributions
X	69	exports

Table .4
The list of symbols end)