The matching predicate and a filtering scheme based on matroids

Dimitris Magos
Department of Informatics, Technological Educational Institute of Athens, 12210 Egaleo , Greece,
Email: dmagos@teiath.gr

Ioannis Mourtos
Department of Economics, University of Patras, 26500 Rion, Patras, Greece,
Email: imourtos@upatras.gr

Leonidas Pitsoulis
Department of Mathematical and Physical Sciences, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece,
Email: pitsouli@gen.auth.gr

***Abstract*— Finding a maximum cardinality matching in a graph is a problem appearing in numerous settings. The problem asks for a set of edges of maximum cardinality, such that no two edges of this set have an endpoint in common. The variety of applications of this problem, along with the fact that several logic predicates can be modelled after it, motivates the study of a related global constraint in the context of Constraint Programming. In this work, we describe a filtering scheme for such a predicate based on matroids. Our method guarantees hyper-arc consistency in polynomial time. It is also applicable to any predicate expressed in terms of an independent system, and remains of polynomial complexity if there exists a polynomial time algorithm for finding a maximum cardinality basis of this independent system. Furthermore, we show that this filtering scheme can be employed to find a maximum cardinality matching.**

***Index Terms*— matching, hyper-arc consistency, matroid intersection**

I. INTRODUCTION

Let $G(V, E)$ denote an undirected graph with V, E being the sets of its nodes and edges, respectively. A *matching* in $G(V, E)$ is a subset of edges with no common endpoints. Finding a matching of maximum cardinality constitutes the maximum cardinality matching (MCM) problem. This is a fundamental problem in graph theory and one of the first problems analysed within the combinatorial optimisation literature. It has been extensively studied because of its inherent theoretical properties as well as its numerous applications in various fields, including Economics, Engineering, Computer Science and Artificial Intelligence. For a thorough theoretical discussion, including solution methods, we refer to [1], while several applications can be found in [2]. We briefly present three examples of such applications to motivate the definition of a corresponding *global* constraint (for the definition of a global constraint and related concepts, see [3]).

- A hotel manager wishes to assign pairs of roommates to rooms of her hotel. The sex, nationality, religion and cultural background are factors, which could determine compatible pairs of roommates. The problem of finding the maximum number of compatible pairs, thus identifying the maximum room requests, is an MCM problem.
- A set of processes are to be assigned to a set of processors. Depending on the computational power of each processor and the resource requirements of each process, compatible pairs of processes and processors are formed. Given that each process can only be assigned to one processor and each processor can be engaged by a single process, the problem asks for a set of pairs (process, processor) of maximum cardinality.
- The variables $x_1, ..., x_n$ with domains $D_1, ..., D_n$, respectively, must receive pairwise distinct values. This problem can be reduced to finding an MCM on a *bipartite* graph. Consider a bipartition of nodes into the sets V_1, V_2. The nodes of V_1 correspond to the variables and the nodes of V_2 to the values of the set $D = \bigcup_{i=1}^{n} D_i$. An edge connects a node corresponding to a variable $x_i, i \in \{1, ..., n\}$, to a node corresponding to a value v, if and only if $v \in D_i$. Solving the MCM problem in this graph either yields a feasible assignment of values to the variables or proves (in the case that the cardinality of the resulting matching is less than n) that no such assignment exists.

This paper is based on "Consistency of the matching predicate" by D. Magos, I. Mourtos and L.Pitsoulis which appeared in the Proceedings of the Fourth Hellenic Conference on AI, SETN 2006, LNAI 3955, pp. 555-558.

This research has been partially funded by the Greek Ministry of Education under the program Pythagoras II.

Apart from its applications, the MCM problem is of specific interest from a Constraint Programming (CP) point of view because it can model several logical constraints. The last example presented describes such a model for the *all_different* predicate [4]. Further examples are the *symmetric all_different* predicate [5] and the system of two *all_different* predicates [6]. Thus, the definition of a maximum cardinality matching constraint provides a framework encompassing other predicates. This has a potential impact on the development of CP solution strategies. Specifically, designing and implementing inference techniques for this constraint provides tools that can be used "out-of-the-box" on other predicates expressed as special cases of it.

Along this line of research, we study the global constraint with respect to the MCM, originally defined in [7], in terms of *hyper-arc* consistency. With respect to the graph $G(V, E)$, this amounts to identifying the edges not participating in any maximum cardinality matching. Such an identification, also called *filtering*, is carried out in [4], [8] for bipartite graphs only. The approach presented here applies to any graph. The wider applicability of our method stems from the exploitation of the matroidal structure of the problem. In short, we propose an algorithm for establishing hyper-arc consistency for problems defined as the *intersection* of matroids and, since matching can be regarded as such, the algorithm becomes applicable to the MCM problem. In this way, we achieve two goals. First, we provide a filtering scheme for both bipartite and non-bipartite matching through a unifying theoretical approach and, secondly, the procedure described applies to *any* predicate that could be expressed as a matroid intersection. Thus the broader research question being addressed is whether matroid theory could be valuable for designing efficient filtering algorithms. Finally, the paper also investigates the relation between the MCM problem and that of establishing hyper-arc consistency for MCM. It is shown that the two problems are equivalent.

II. Preliminaries and Problem Formulation

We refer to [3], [9] for formal definitions concerning the Constraint Satisfaction Problem (CSP). We only give a brief exposition of certain notions, in order to facilitate the following discussion. A global constraint $C(x_i : i \in I)$ imposes a condition on the set of variables $\cup_{i\in I}\{x_i\}$, each receiving values from a domain D_i. A *solution* to $C(x_i : i \in I)$ is a tuple $(d_1, \ldots, d_{|I|}) \in D_1 \times \cdots \times D_{|I|}$, which satisfies this constraint. A constraint is *hyper-arc* consistent, if and only if for each $i \in I$ and $d_j \in D_i$ there exist $d_1, \ldots, d_{j-1}, d_{j+1}, \ldots, d_{|I|}$ such that $d_1, \ldots, d_j, \ldots, d_{|I|}$ is a solution.

A formal definition of the MCM problem follows. Let V (E) denote the set of nodes (edges) of an undirected graph $G(V, E)$. As a matter of convention, we consider that each element of E is a set consisting of two elements (nodes) belonging to V. Hence, for $e \in E$ and $v \in V$, one can write $0 \leq |e \cap \{v\}| \leq 1$. Similarly, edges $e_i, e_j \in E$ are incident to the same node if and only if $|e_i \cap e_j| = 1$. A set $M \subseteq E$ is called a *matching* if, for every pair of edges $e_i, e_j \in M$, it holds that $|e_i \cap e_j| = 0$. The MCM problem is defined as

$$\max_{M\subseteq E:|e_i\cap e_j|=0,\forall e_i,e_j\in M} |M|$$

The *matching* constraint is presented in the global constraint format described above. The constraint is imposed on a set of variables, each corresponding to a node of the graph $G(V, E)$. That is, for each $v \in V$ there exists a variable x_v. Let $d : E \to \mathbb{Z}_+$ be a bijection, i.e. d maps each edge to a unique positive integer. For $v \in V$, let E_v denote the set of edges incident to node v, i.e. $E_v = \{e \in E : |e \cap \{v\}| = 1\}$. The domain of x_v is defined as $D_v = \{d(e) : e \in E_v\} \cup \{0\}$, where value 0 is reserved and its meaning is explained next. The variables x_u, x_w are said to form a pair if they share the same value (called *pair value*), which is also different than 0. Apparently, this value must belong to both D_u, D_w. This implies that nodes u, w are linked via the edge associated with the common value of the variables x_u, x_w.

The predicate

$$matching\{x_v : v \in V\},$$
$$x_v \in D_v, \forall v \in V,$$

asks for a maximum number of pairs with distinct pair values. The variables not selected to form a pair are assigned value 0. Hence, the above constraint can be equivalently thought of as minimising the number of variables with value 0.

Before proceeding, we should recall certain definitions related to matroids. Matroid theory is a very elegant approach used in combinatorial optimisation (see [2], [10]). The motivation for applying matroid theory to combinatorial optimisation problems could be traced back to the pioneering work of Edmonds, who proved that several important models associated with matching and matroids are solvable, and observed an apparent connection between polynomial solvability and the existence of a matroidal structure. After providing some basic definitions, we focus on the relationship of these structures to matchings in graphs.

Denote the collection of subsets of a set A as 2^A.

Definition 1: Given a finite set $\mathcal{E}$ and some family of subsets $\mathcal{F} \subset 2^{\mathcal{E}}$, the set system $\mathcal{M} = (\mathcal{E}, \mathcal{F})$ is a matroid if the following conditions hold

(I1) $\emptyset \in \mathcal{F}$,

(I2) If $X \in \mathcal{F}$ and $Y \subset X$ then $Y \in \mathcal{F}$,

(I3) If $X, Y \in \mathcal{F}$ and $|Y| < |X|$, then there exists an element e of $X \backslash Y$ such that $Y \cup \{e\} \in \mathcal{F}$.

If conditions (I1), (I2) hold, $\mathcal{M} = (\mathcal{E}, \mathcal{F})$ is called an *independent* system (IS). Hence, by Definition 1, a matroid is an IS for which (I3) also holds. The following definitions apply to any IS. $\mathcal{E}$ is called the *ground set*. A set $X \in \mathcal{F}$ is said to be an *independent set* of $\mathcal{M} = (\mathcal{E}, \mathcal{F})$. Any set which is not independent is dependent. A *maximal* independent set is called a *base*, and a *minimal* dependent set is called a *circuit*. The *rank*

$r(X)$ of a set $X \subseteq \mathcal{E}$ is the cardinality of a maximal independent subset of X. The *span* $sp(X)$ of a set $X \subseteq \mathcal{E}$ is the maximal superset of X having the same rank as X.

An IS can be described by a set of matroids on the same ground set. Consider the matroids $\mathcal{M}_1 = (\mathcal{E}, \mathcal{F}_1), \ldots, \mathcal{M}_k = (\mathcal{E}, \mathcal{F}_k)$. Their intersection, also called k-matroid intersection, is defined as $(\mathcal{E}, \bigcap_{i=1,\ldots,k} \mathcal{F}_i)$. Clearly this is an IS. Moreover,

Theorem 1 ([11]): Any independence system is the intersection of a finite number of matroids .

A critical implication of (I3) is that all the maximal independent subsets of any $X \subseteq E$ are of the same cardinality. This does not apply to independent systems in general. Consequently, the problem of finding a base of a matroid is quite different than that of finding a maximum cardinality basis (MCB) of an IS. In general, the latter problem remains polynomially solvable if the IS is the intersection of only two matroids. An algorithm for the MCB problem on a 2-matroid intersection system, proposed by Edmonds, is described in [2].

The connection between matchings in a graph and independence systems is direct: the set of matchings forms an IS, to be denoted by $M(G)$, on the ground set of edges E. It is not difficult to observe that properties (I1) and (I2) hold. An MCM is simply an MCB in $M(G)$. Therefore, when $M(G)$ can be expressed as a 2-matroid intersection, an algorithm identifying an MCB (e.g. Edmonds' algorithm), can provide an MCM. In an analogous manner, a filtering procedure identifying the elements of $\mathcal{E}$ that do not belong to any MCB is applicable to the MCM problem. Such a scheme, when applied to $M(G)$, produces the set of edges not participating in any maximum cardinality matching. In the following section, we describe algorithms of this kind, which achieve hyper-arc consistency, i.e. identify all "inconsistent" edges.

III. CONSISTENCY OF THE *matching* CONSTRAINT

Achieving hyper-arc consistency on the *matching* constraint implies identifying all edges of the graph which do not appear in any maximum cardinality matching. An additional step would be to identify the edges that participate in *every* such matching. The overall aim is to establish a partitioning of E to sets $E_{ALL}, E_{SOME}, E_{NONE}$, which include the edges participating to all, some or none maximum cardinality matchings, respectively.

This section resolves this question by first considering graphs for which the matching is given by the intersection of two matroids and subsequently describing a filtering procedure applicable to the general case. This distinction is necessary because in the first case there already exists an algorithm for producing the above partitioning [12], while a new algorithm is presented for the second case. Since the algorithms to be presented next apply to any IS, the notation used for the ground set is $\mathcal{E}$. Naturally, when we discuss application of these methods to $M(G)$, the ground set is denoted by E. The same convention applies to the sets forming the partition of the ground set.

A. A special case: the 2-matroid intersection

The matroid used to describe $M(G)$, in the special case where the graph G is m-partite, is the so called *partition matroid*. To formally define partition matroids, consider a partition of the ground set $\mathcal{E}$

$$\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \cdots \cup \mathcal{E}_n$$

and define the independence relationship as

$$\mathcal{F} := \{X \subseteq \mathcal{E} : |X \cap \mathcal{E}_i| \leq 1, \;\; i = 1, \ldots, n\}.$$

It is not difficult to show that the system $(\mathcal{E}, \mathcal{F})$ is a matroid. Given an m-partite graph $G(\bigcup_i V_i, E)$ each node set V_i imposes a partition of the edge set

$$E = E_1^{(i)} \cup E_2^{(i)} \cup \cdots \cup E_{|V_i|}^{(i)},$$

where

$$E_j^{(i)} := \{e \in E : e \text{ incident to node } v_j \in V_i\},$$

for $j = 1, \ldots, |V_i|$. Defining the partition matroids for each node set with the independence relationship

$$\mathcal{F}_i := \{X \subseteq E : |X \cap E_j^{(i)}| \leq 1, j = 1, \ldots, |V_i|\},$$

the independence system which is defined by the intersection of the m partition matroids $(E, \mathcal{F}_i)$ describes the set of matchings of the m-partite graph G. Evidently, $M(G)$ can be described as the intersection of two matroids if graph G is bipartite. Moreover, the following theorem provides a characterisation of the class of graphs, for which the set of matchings is the intersection of two matroids.

Theorem 2 ([13]): The set of matchings $M(G)$ of a graph $G(V, E)$ is the intersection of two matroids iff G contains no odd cycle of cardinality ≥ 5 and each triangle of G has at most one node with degree > 2 .

Observe that the class implied by the above theorem is slightly larger than the class of bipartite graphs. Given that the class of bipartite graphs includes those graphs which contain no odd cycles, the class of graphs in Theorem 2 includes also bipartite graphs with triangles attached at their nodes. An example of such a graph is shown in Figure 1.

The authors in [12] describe an $O(\max\{\Lambda_1, \Lambda_2\} \cdot |\mathcal{E}|^2)$ algorithm for constructing the partition $\mathcal{E}_{ALL} \cup \mathcal{E}_{SOME} \cup \mathcal{E}_{NONE}$ for the MCB of some IS, which is the intersection of two matroids, where Λ_1, Λ_2 are the computational complexities of computing the rank functions of the two matroids. Their algorithm extends the notion of *augmenting paths*, which is used to increase the size of an independent set by one in the intersection of two matroids, and is the main component of the 2-matroid intersection algorithm [2]. For the MCM problem, although there exists an algorithm for constructing the same partitioning of the edge set in bipartite graphs [8], the algorithm of [12] is more general since it applies to the graphs characterised in Theorem 2.

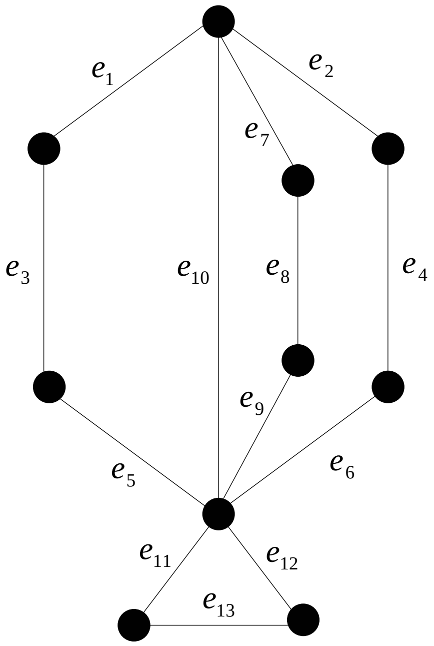

Figure 1. A graph whose $M(G)$ is an intersection of two matroids

B. The general case

We know that, for a given graph G, the set of matchings $M(G)$ is an IS and therefore, by Theorem 1, an intersection of a finite number m of matroids. If those m matroids were known, one could iteratively consider all $\binom{m}{2}$ 2-matroid intersections and apply the algorithm mentioned above [12]. In fact, it has been proved that the set of matchings in an arbitrary graph G is the intersection of at most $O(log|V|/loglog|V|)$ matroids [13]. Apparently, the above procedure terminates when all such intersections are tested and no additional edge is filtered out. Although this approach achieves a certain level of consistency, it definitely does not guarantee hyper-arc consistency.

The algorithm presented next achieves hyper-arc consistency on any IS under the condition that we can compute the rank of any subset of the ground set $\mathcal{E}$. An additional requirement is to be able to find a maximum cardinality base, which includes a specific $e \in \mathcal{E}$. We denote such a base by $B(e)$. Note that, in the case of $M(G)$, this problem as well as the problem of computing the rank of any subset of edges are polynomially solvable.

Theorem 3: Algorithm 1 correctly determines in $O((\Omega_1 + \Omega_2)\,|\mathcal{E}|)$ steps the partitioning $\mathcal{E}_{ALL}$, $\mathcal{E}_{SOME}$, $\mathcal{E}_{NONE}$, where Ω_1, Ω_2 are, respectively, the computational complexities of computing the rank function and $B(e)$, for $e \in \mathcal{E}$.

Proof: (Correctness) The algorithm consists of two loops. The first one separates the elements of $\mathcal{E}_{NONE}$ from those of $X = \mathcal{E}_{ALL} \cup \mathcal{E}_{SOME}$. The separation is achieved by identifying for each $e \in \mathcal{E}$ an MCB containing it. Apparently, if the rank of such a base is smaller than the rank of $\mathcal{E}$ then $e \in \mathcal{E}_{NONE}$; otherwise, all the elements of $B(e)$ belong to X. In the second loop, we identify those elements of X that belong to $\mathcal{E}_{ALL}$. Observe that $|\mathcal{E}_{ALL}| \leq r(\mathcal{E})$ and recall that for each $e \in X$ there is a basis containing it, whose rank is the same as $\mathcal{E}$. Thus, for each such element we check if its deletion from $\mathcal{E}$ decreases its rank. If this is true then the element belongs to $\mathcal{E}_{ALL}$; otherwise, it must belong to $\mathcal{E}_{SOME}$.

(Complexity) Let Ω_1, Ω_2 denote the computational complexity of computing the rank function and of finding $B(e)$, respectively. The first loop is performed at most $|\mathcal{E}|$ times. At each iteration, we need to compute $B(e)$, $r(B(e))$ and perform one comparison. Thus, the complexity of the first loop is $O((\Omega_1 + \Omega_2)\,|\mathcal{E}|)$. Similarly, the complexity of the second loop is $O(\Omega_1\,|\mathcal{E}|)$. ■

Algorithm 1 Achieving hyper-arc consistency for the MCB problem on an IS

$X := \emptyset$;
$\mathcal{E}_{NONE} := \emptyset$;
$\mathcal{E}_{ALL} := \emptyset$;
$rank\mathcal{E} := r(\mathcal{E})$;
for all $e \in \mathcal{E} \setminus X$ **do**
  Compute a basis $B(e)$;
  if $r(B(e)) < rank\mathcal{E}$ **then**
    $\mathcal{E}_{NONE} := \mathcal{E}_{NONE} \cup \{e\}$;
  else
    $X := X \cup \{B(e)\}$;
  end if
end for
for all $e \in X$ **do**
  if $r(\mathcal{E} \setminus \{e\}) < rank\mathcal{E}$ **then**
    $\mathcal{E}_{ALL} := \mathcal{E}_{ALL} \cup \{e\}$;
  end if
end for
$\mathcal{E}_{SOME} := \mathcal{E} \setminus \{\mathcal{E}_{ALL} \cup \mathcal{E}_{NONE}\}$;

Let us now examine in more detail how the critical steps of Algorithm 1 are implemented in the case of $M(G)$. The standard solution method for the maximum cardinality matching problem is the algorithm described in [14], which can be implemented in $O(|V|^3)$ steps. The proof, together with references to more recent algorithms of modestly lower complexity, can be found in [11]. For $X \subseteq E$, by deleting all the edges in $E \setminus X$ and subsequently applying Edmond's algorithm, we can determine the cardinality of a maximum matching consisting entirely of edges of X. In this way, we identify $r(X), \forall X \subseteq E$. Finally, the same algorithm can be used to determine $B(e)$, for $e \in E$. To achieve this for an edge $e \in E$, it is sufficient to delete all edges incident to nodes u, v, where $|e \cap \{u\}| = |e \cap \{v\}| = 1$. As a result, the graph decomposes into two components: one consisting of the edge e and its two incident nodes u, v and one consisting of the remaining nodes and all the edges not incident to either of u, v. Let B denote a maximum cardinality matching on the second component. It is easy to see that

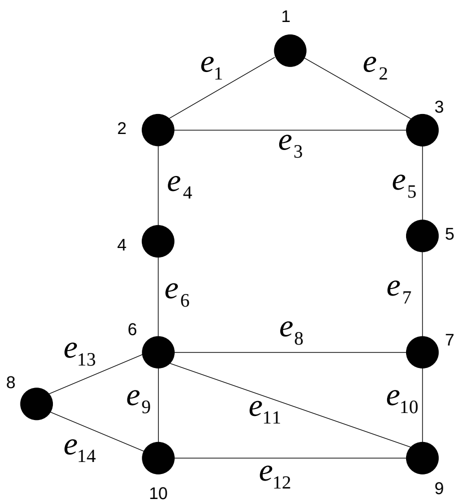

Figure 2. Filtering with respect to a matching on a arbitrary graph

$B(e) = B \cup \{e\}$.

To better clarify the above, we provide the following example.

Example 4: The maximum cardinality matching constraint for the graph of Figure 2 is $matching\{x_i : i \in \{1, ..., 10\}\}$. Assuming $d(e_i) = i$, for all e_i of E, the domains of the variables are $D_1 = \{0, 1, 2\}$, $D_2 = \{0, 1, 3, 4\}$, $D_3 = \{0, 2, 3, 5\}$, $D_4 = \{0, 4, 6\}$, $D_5 = \{0, 5, 7\}$, $D_6 = \{0, 6, 8, 9, 11, 13\}$, $D_7 = \{0, 7, 8, 10\}$, $D_8 = \{0, 13, 14\}$, $D_9 = \{0, 10, 11, 12\}$, $D_{10} = \{0, 9, 12, 14\}$.

A maximum cardinality matching is induced by the edge set $M = \{e_1, e_5, e_6, e_{10}, e_{14}\}$. Thus, $r(E) = 5$. We can set $B(e_1) = M$. Since $r(B(e_1)) = r(E) = 5$, $X = \{e_1, e_5, e_6, e_{10}, e_{14}\}$. The next element to consider is e_2. There exists $B(e_2) = \{e_2, e_4, e_7, e_{12}, e_{13}\}$. Again because $r(B(e_2)) = r(E) = 5$, $X = X \cup B(e_2) = \{e_1, e_2, e_4, e_5, e_6, e_7, e_{10}, e_{12}, e_{13}, e_{14}\}$. Observe that any maximum cardinality matching that includes either e_3, or e_8 consists of 4 edges. Thus, $r(B(e_3)) = r(B(e_8)) = 4$ and $E_{NONE} = \{e_3, e_8\}$. Finally $X = X \cup \{e_{11}\}$ because there exists the matching $B(e_{11}) = \{e_2, e_4, e_7, e_{11}, e_{14}\}$. For each of the elements of X we can find a matching of cardinality 5 that does not include it. Overall, $E_{ALL} = \emptyset$, $E_{SOME} = X = \{e_1, e_2, e_4, e_5, e_6, e_7, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}\}$ and $E_{NONE} = \{e_3, e_8\}$.

Conclusively, the *matching* constraint becomes hyper-arc consistent if we remove the value 3 from D_2, D_3 and the value 8 from D_6, D_7.

From the above presentation, it is evident that, given an algorithm for calculating an MCM, one can compute the partition $(E_{ALL}, E_{SOME}, E_{NONE})$ for the matching predicate. The inverse direction is also plausible, i.e. given an algorithm that identifies the edges that participate in all (none) MCMs, calculate an MCM. This is exactly achieved by Algorithm 2, presented next.

Algorithm 2 Finding a maximum cardinality matching X, using the routine *partition* which computes the partitioning of the edge set of a graph with respect to the MCM

$(E_{ALL}, E_{SOME}, E_{NONE})$:=*partition*(E);
$X := \emptyset$;
while $E_{SOME} \neq \emptyset \rightarrow$
    $X := X \cup E_{ALL} \cup \{e\}$, for some $e \in E_{SOME}$;
    $\bar{E} := E \setminus (X \cup E_{NONE})$;
    $(\bar{E}_{ALL}, \bar{E}_{SOME}, \bar{E}_{NONE})$:=*partition*$(\bar{E})$;
    $E := \bar{E}$;
    $E_{ALL} := \bar{E}_{ALL}$;
    $E_{SOME} := \bar{E}_{SOME}$;
    $E_{NONE} := \bar{E}_{NONE}$;
endwhile;
return$(X \cup E_{ALL})$;

Observe that the Algorithm 2 is polynomial in the cardinality of the set of edges E and the complexity of the routine *partition.*

IV. CONCLUSIONS

The *matching* constraint enjoys various applications and generalises other well-known predicates. In the current work, we propose a scheme for achieving hyper-arc consistency, for this constraint, by establishing an equivalence between consistency and a certain partitioning of the set of edges (the ground set). The algorithm proposed constructs the "consistency" partitioning of the ground set in polynomial time, thus identifying edges that cannot appear in any MCM and edges that appear in all MCM's. Moreover, the ideas presented in this paper could be applied to other predicates, which can be modeled as the intersection of matroids. The aim would be to obtain efficient implementations of filtering routines within Constraint Programming solvers. Furthermore, we have also proposed an algorithm which can solve the MCM problem, given an algorithm that computes the above-mentioned partitioning of the set of edges E. Consequently, the two problems are equivalent. An open question is whether this result also holds for any problem formulated as the maximum cardinality basis problem in the intersection of a finite number of matroids.

REFERENCES

[1] L. Lovasz and M. D. Plummer, *Matching theory*, ser. Annals of Discrete Mathematics. North Holland, 1986, vol. 29.

[2] E. Lawler, *Combinatorial Optimization: Networks and Matroids*. Dover, 2001, unabridged edition originally published by Holt, Rinehart, Winston in 1976.

[3] R. Dechter, *Constraint Processing*. Morgan Kaufmann, 2003.

[4] J. C. Régin, "A filtering algorithm for constraints of difference in csps," in *Proceedings of National Conference on Artificial Intelligence*, 1994, pp. 362–367.

[5] ——, "The symmetric alldifferent constraint," in *Proceedings of the IJCAI-99*, 1999, pp. 420–425.

[6] G. Appa, D. Magos, and I. Mourtos, "On the system of two all_different predicates," *Inform. Process. Lett.*, vol. 94, pp. 99–106, 2005.

[7] D. Magos, I. Mourtos, and L. Pitsoulis, "Consistency of the matching predicate," in *Proceedings of SETN'06, LNAI 3955*, 2006, pp. 555–558.

[8] M. Costa, "Persistency in maximum cardinality bipartite matchings," *Oper. Res. Lett.*, vol. 15, pp. 143–149, 1994.

[9] J. Hooker, *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*, ser. Wiley-Interscience Series in Discrete Mathematics and Optimization. J. Wiley, 2000.

[10] J. G. Oxley, *Matroid Theory*, ser. Oxford Graduate Texts in Mathematics, K. Donaldson, S., S. Hildenbrandt, J. Taylor, M., and R. Cohen, Eds. Oxford University Press, 1992, vol. 3.

[11] B. Korte and J. Vygen, *Combinatorial optimization: Theory and algorithms*, ser. Algorithms and Combinatorics. Springer, 1991, vol. 21.

[12] K. Cechlárova and V. Lacko, "Persistency in combinatorial optimization problems on matroids," *Discrete Appl. Math.*, vol. 110, pp. 121–132, 2001.

[13] S. P. Fekete, R. T. Firla, and B. Spille, "Characterizing matchings as the intersection of matroids," *Math. Methods Oper. Res.*, vol. 58, pp. 319–329, 2003.

[14] J. Edmonds, "Paths, trees and flowers," *Canad. J. Math.*, vol. 17, pp. 449–467, 1965.

Dimitris Magos received his Ph. D. from the Athens University of Economics and Business (Department of Informatics). He is currently an associate professor at the Department of Informatics of the Technological Educational Institute of Athens. His research interests include mathematical programming, constraint programming and combinatorial optimization.

Ioannis Mourtos is a lecturer at the Department of Economics, University of Patras. He obtained his Ph. D. from the London School of Economics, Operational Research Department. His research interests lie in the areas of integer programming and combinatorial optimisation.

Leonidas Pitsoulis received his Ph.D. degree at the Department of Industrial and Systems Engineering of the University of Florida and his is currently at the Department of Mathematical and Physical Sciences of the Aristotle University of Thessaloniki. His research interests include combinatorial optimization and computational methods.