Graph Properties Based Filtering

Nicolas Beldiceanu[1], Mats Carlsson[2], Sophie Demassey[1], and Thierry Petit[1]

[1] École des Mines de Nantes, LINA FRE CNRS 2729, FR-44307 Nantes, France.
{Nicolas.Beldiceanu,Sophie.Demassey,Thierry.Petit}@emn.fr
[2] SICS, P.O. Box 1263, SE-164 29 Kista, Sweden.
Mats.Carlsson@sics.se

May 4, 2006

SICS Technical Report T2006:10

ISRN: SICS-T–2006/10-SE

ISSN: 1100-3154

Abstract. This report presents a generic filtering scheme, based on the graph description of global constraints. This description is defined by a network of binary constraints and a list of elementary graph properties: each solution of the global constraint corresponds to a subgraph of the initial network, retaining only the satisfied binary constraints, and which fulfills all the graph properties. The graph-based filtering identifies the arcs of the network that belong or not to the solution subgraphs. The objective is to build, besides a catalog of global constraints, also a list of systematic filtering rules based on a limited set of graph properties. We illustrate this principle on some common graph properties and provide computational experiments of the effective filtering on the `group` constraint.
Keywords: Constraint Programming, Global Constraints, Filtering Algorithms, Graph Properties.

1 Introduction

The global constraint catalog [2] provides the description of hundreds of global constraints in terms of graph properties: The solutions of a global constraint are identified to the subgraphs of one initial digraph sharing several graph properties. Existing graph properties use a small set of graph parameters such as the number of vertices, or arcs, or the number of connected components.The most common graph parameters were considered in [6]. It showed how to estimate, from the initial digraph, the lower and upper values of a parameter in the possible solution subgraphs. Those bounds supply necessary conditions of feasibility for almost any global constraint.

This report goes one step further by introducing systematic filtering rules for those global constraints. The initial digraph describing a global constraint is indeed a network of constraints on pairs of variables. To each complete instantiation of the variables corresponds a final subgraph obtained by removing from the initial digraph all the arcs (i.e., the binary constraints) that are not satisfied. Since solution(s) of the global constraint correspond to final subgraphs fulfilling a given set of graph properties, filtering consists

in identifying and dropping elements of the initial digraph that do not belong to such subgraphs, or to force those elements that belong to any solution subgraphs.

A first way to achieve this identification might be to use shaving [12]. That is, fix the status of an arc or a vertex, and check if it leads to a contradiction. Since this is very costly in practice, we present in this report another way to proceed. The filtering rules proposed thereafter apply whenever a graph parameter is set to one of its bounds.

Last, the global constraints can be partitioned wrt. the class that their associated final digraphs belongs to. Taking into account a given graph class leads to a better estimation of the graph parameter bounds and then a more effective filtering.

The report is organized as follows: Section 2 recalls the graph-based description of global constraints and introduces a corresponding reformulation. Section 3 sets up a list of notations in order to formalize the systematic graph-based filtering. Section 4 presents the filtering rules related to the bounds of some graph parameters. Section 5 shows how the graph-based filtering relates to existing ad-hoc filtering for some global constraints. Section 6 illustrates how refining the filtering according to a given graph class and provides computational results on the `group` constraint, which belongs to the `path_with_loops` graph class.

2 Graph-based Description of Global Constraints

2.1 Graph-based Description

Let $C(V_1, \ldots, V_p, x_1, \ldots, x_n)$ be a global constraint with domain variables[3] $V_1, \ldots, V_p$, and domain or set variables[4] $x_1, \ldots, x_n$. When it exists, a *graph-based description* of C is given by one (or several) network(s) $G_{\mathcal{R}} = (X, E_{\mathcal{R}})$ of binary constraints over $X = \{x_1, \ldots, x_n\}$ in association with a set $\mathcal{GP}_{\mathcal{R}} = \{\mathcal{P}_l \ op_l \ V_l \mid l = 1, \ldots, p\}$ of graph properties and, optionally, a graph class $c_{\mathcal{R}}$, where:

- The constraints defining the digraph $G_{\mathcal{R}} = (X, E_{\mathcal{R}})$ share the same semantic (typically it is an equality, an inequality or a disequality). Let $x_j \mathcal{R} x_k$ denote the so-called *arc constraint* between the ordered pair of variables $(x_j, x_k) \in E_{\mathcal{R}}$ (or the unary constraint if $j = k$).
- $\mathcal{P}_l \ op_l \ V_l$ expresses a graph property comparing the value of a graph parameter $\mathcal{P}_l$ to the value of variable V_l. The comparison operator op_l is either $\geq$, $\leq$, $=$, or $\neq$. Among the most usual graph parameters $\mathcal{P}_l$, let $\mathbf{NARC}$ denote the number of arcs of a graph, $\mathbf{NVERTEX}$ the number of vertices, $\mathbf{NCC}$ the number of connected components, $\mathbf{MIN\_NCC}$ and $\mathbf{MAX\_NCC}$ the numbers of vertices of the smallest and the largest connected components respectively.
- $c_{\mathcal{R}}$ corresponds to recurring graph classes that show up for different global constraints. For example, we consider graphs in the classes `acyclic`, `symmetric`, `bipartite`.

[3] A *domain variable* D is a variable ranging over a finite set of integers $\mathrm{dom}(D)$. $\min(D)$ and $\max(D)$ respectively denote the *minimum* and *maximum* values in $\mathrm{dom}(D)$.

[4] A *set variable* S is a variable that will be assigned to a finite set of integer values. Its domain is specified by its *lower bound* $\underline{S}$, and its *upper bound* $\overline{S}$, and contains all sets that contain $\underline{S}$ and are contained in $\overline{S}$.

$G_{\mathcal{R}}$ is called the *initial digraph*. When all variables x are instantiated, the subgraph of $G_{\mathcal{R}}$, obtained by removing all arcs corresponding to unsatisfied constraints $x_j \mathcal{R} x_k$ as well as all vertices becoming isolated, is called a *final digraph* (associated to the instantiation) and is denoted by $G_f = (X_f, E_f)$.

The relation between C and its graph-based description is stated as follows:

Definition 1. *A complete assignment of variables $V_1, \ldots, V_p, x_1, \ldots, x_n$ is a solution of C iff the final digraph associated to the assignment of $x_1, \ldots, x_n$, satisfies all graph properties $\mathcal{P}_l$ op_l V_l in $\mathcal{GP}_{\mathcal{R}}$ and belongs to the graph class $c_{\mathcal{R}}$.*

Example 1. Consider the `proper_forest`(`NTREE`, `VER`) constraint introduced in [5]. It receives a domain variable `NTREE` and a digraph G described by a collection of n vertices `VER`: each vertex is labelled by an integer between 1 and n and is represented by a set variable whose lower and upper bounds are the sets of the labels of respectively its *mandatory neighbors* and its *mandatory or potential neighbors* in G. This constraint partitions the vertices of G into a set of vertex-disjoint proper trees (i.e., trees with at least two vertices each).

Part (A) of Figure 1 illustrates such a digraph G, where solid arrows depict mandatory arcs and dashed arrows depict potential arcs. Part (B) of the figure shows a possible solution on this digraph with three proper trees. In the graph-based description of `proper_forest`, the initial di-

i	dom(`VER`[i]) (A)	dom(`VER`[i]) (B)	i	dom(`VER`[i]) (A)	dom(`VER`[i]) (B)
1	$\{2,3\}$	$\{2\}$	6	$\{5,7,8\}$	$\{5,7\}$
2	$\{1,3,4\}$	$\{1\}$	7	$\{5,6,9\}$	$\{6,9\}$
3	$\{1,2,4,5\}$	$\{4\}$	8	$\{6,9\}$	$\{9\}$
4	$\{2,3\}$	$\{3\}$	9	$\{7,8\}$	$\{7,8\}$
5	$\{3,6,7\}$	$\{6\}$			

Table 1. Domains of the variables for the `proper_forest` constraint corresponding to parts (A) and (B) of Figure 1.

graph corresponds exactly to G and has no loop. Any final digraph G_f contains all the mandatory arcs of G and belongs to the `symmetric` graph class.[5] Moreover G_f has to fulfill the following graph properties: $\mathbf{NVERTEX} = n$ (since it is a vertex partitioning problem, G_f contains all the vertices of G), $\mathbf{NARC} = 2 \cdot (n - \texttt{NTREE})$ (2 since G_f is symmetric, and $n - \texttt{NTREE}$ since we have `NTREE` acyclic connected digraphs) and $\mathbf{NCC} = \texttt{NTREE}$.

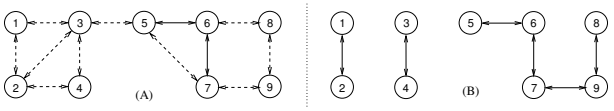

Fig. 1. (A) A digraph and (B) a solution with three proper trees for the `proper_forest` constraint

[5] A digraph is *symmetric* iff, if there is an arc from u to v, there is also an arc from v to u.

2.2 Graph-based Reformulation

According to Definition 1, any global constraint $C(V_1, \ldots, V_p, x_1, \ldots, x_n)$ holding a graph-based description can be reformulated as follows:

Proposition 1. *Define additional variables attached to each constraint network $G_{\mathcal{R}} = (X, E_{\mathcal{R}})$: to each vertex x_j and to each arc e_{jk} of $G_{\mathcal{R}}$ correspond* 0-1 *variables respectively denoted $vertex_j$ and arc_{jk}. $Vertex$ and Arc denote these sets of variables. Last, let G_f denote the subgraph of $G_{\mathcal{R}}$, whose vertices and arcs correspond to the variables $vertex_j$ and arc_{jk} set to 1. Then constraint C holds iff the following constraints hold:*

$$arc_{jk} = 1 \;\Leftrightarrow\; x_j \mathcal{R} x_k, \quad \forall e_{jk} \in E_{\mathcal{R}} \tag{1}$$

$$vertex_j = \min(1, \textstyle\sum_{\{k \mid e_{jk} \in E_{\mathcal{R}}\}} arc_{jk} + \sum_{\{k \mid e_{kj} \in E_{\mathcal{R}}\}} arc_{kj}), \quad \forall x_j \in X \tag{2}$$

$$ctr_{\mathcal{P}_l}(Vertex, Arc, P_l), \quad \forall (\mathcal{P}_l \; op_l \; V_l) \in \mathcal{GP}_{\mathcal{R}} \tag{3}$$

This constraint is satisfied when P_l is equal to the value of the corresponding parameter $\mathcal{P}_l$ in G_f.

$$P_l \; op \; V_l, \quad \forall (\mathcal{P}_l \; op_l \; V_l) \in \mathcal{GP}_{\mathcal{R}} \tag{4}$$

$$ctr_{c_{\mathcal{R}}}(Vertex, Arc) \tag{5}$$

This graph-class constraint *is satisfied if G_f belongs to the graph class $c_{\mathcal{R}}$.*

Proof. Proposition 1 is a simple rephrasing of Definition 1: given any complete assignment of the variables of this reformulation, the final digraph associated to the assignment of $x_1, \ldots, x_n$ is the digraph G_f defined above iff constraints (1) and (2) hold. P_l is the value of the parameter $\mathcal{P}_l$ in G_f iff constraint (3) holds. Definition 1 states that $V_1, \ldots, V_p, x_1, \ldots, x_n$ is a solution of C iff constraint (4) holds for each graph property. Last, G_f belongs to the graph class $c_{\mathcal{R}}$ according to constraint (5). □

Example 2. Consider again the `proper_forest` constraint previously introduced. Since its final digraph is symmetric and does not contain isolated vertices, the graph-class constraint (5) is the conjunction of the following constraints: $arc_{jk} = arc_{kj}$ for each arc e_{jk} and $vertex_j = \min(1, 2 \cdot \sum_{\{k \mid e_{jk} \in E_{\mathcal{R}}\}} arc_{jk})$ for each vertex x_j in $G_{\mathcal{R}}$.

Filtering domains of variables V and x according to C can be achieved by enforcing alternatively, and for each constraint network $G_{\mathcal{R}}$, the five sets of constraints of this reformulation. Enforcing constraints (1), (2), (4) and (5) is mostly trivial since these constraints are elementary arithmetic constraints. The generic graph-based filtering mainly lies then on maintaining consistency according to constraints (3), from the arc and $vertex$ variables to the bounds of the graph parameter variables P_l, and conversely. In [6] it was presented, for some usual graph parameters, how to estimate their minimal ($\underline{P_l}$) and maximal ($\overline{P_l}$) values in the final digraphs G_f, given the current state of the arcs and vertices of $G_{\mathcal{R}}$. Section 4 shows how in turn, the status of some arcs and vertices can be determined according to a graph parameter variable when it is set to one of its extreme values (i.e. $\mathrm{dom}(P_l) = \{\underline{P_l}\}$ or $\mathrm{dom}(P_l) = \{\overline{P_l}\}$).

Hence, the approach relies on identifying the possible final digraphs in $G_{\mathcal{R}}$ that minimize or maximize a given graph parameter. Any final digraph contains (resp. does not contain) the arcs and vertices corresponding to arc and $vertex$ variables fixed to 1 (resp. to 0). Since it has no isolated vertices, we assume that the *normalization constraints* (2) are enforced before estimating a graph parameter. Section 6 shows how

refining this estimation when the final digraphs must belong to a given graph class, by also first enforcing constraints (5).

3 Notations for a Systematic Filtering

As for the graph-based description of any global constraint, we aim at providing a catalog of generic filtering rules related to the bounds of graph parameters. In order to formalize this, we first need to introduce a number of notations.

Let $G_{\mathcal{R}}$ an initial digraph associated to the graph-based description of a global constraint. The current domains of variables arc and $vertex$ of the reformulation correspond to a unique partitioning of the arc and vertex sets of $G_{\mathcal{R}}$, denoted as follows:

Notation 1 *A vertex x_j or an arc e_{jk} of $G_{\mathcal{R}}$ is either* true *(T),* false *(F), or* undetermined *(U) whether the corresponding variable $vertex_j$ or arc_{jk} is fixed to* 1*, fixed to* 0 *or yet unfixed (with domain $\{0,1\}$). This leads to the partitioning of the vertex set of $G_{\mathcal{R}}$ into $X_T \dot\cup X_F \dot\cup X_U$ and to the partitioning of the edge set of $G_{\mathcal{R}}$ into $E_T \dot\cup E_F \dot\cup E_U$. For two distinct elements Q and R in $\{T,U,F\}$, let X_{QR} denote the vertex subset $X_Q \dot\cup X_R$, and E_{QR} denote the arc subset $E_Q \dot\cup E_R$.*

Once the normalization constraints are enforced, subgraph (X_T, E_T) is well defined and is included in any final digraph. (X_{TU}, E_{TU}) is also a subgraph of $G_{\mathcal{R}}$, called the *intermediate digraph*, and any final digraph is derived from this by turning each U-arc and U-vertex into T or F.[6] We aim at identifying the final digraphs in which a graph parameter P reaches its lower value $\underline{P}$ or its upper value $\overline{P}$. An estimated bound is said to be *sharp* if for any intermediate digraph, there exists at least one final digraph where the parameter takes this value. To estimate these bounds, we deal with different digraphs derived from the intermediate digraph:

Notation 2 *For any words Q, R and S on the alphabet $\{T,U,F\}$, X_Q and X_S are vertex subsets and E_R is an arc subset of the initial digraph, and:*

- *$X_{Q,R}$ (resp. $X_{Q,\neg R}$) denotes the set of vertices in X_Q that are extremities of at least one arc (resp. no arc) in E_R:*
  $X_{Q,R} = \{x \in X_Q \mid \exists e \in E_R \wedge x \in e\}$ and $X_Q = X_{Q,R} \dot\cup X_{Q,\neg R}$.
- *$X_{Q,R,S}$ (resp. $X_{Q,R,\neg S}$) denotes the set of vertices in $X_{Q,R}$ that are linked to at least one vertex (resp. to no vertex) in X_S by an arc in E_R:*
  $X_{Q,R,S} = \{x \in X_{Q,R} \mid \exists y \in X_S \wedge ((x,y) \in E_R \vee (y,x) \in E_R)\}$ and $X_{Q,R} = X_{Q,R,S} \dot\cup X_{Q,R,\neg S}$.
- *$X_{Q,\neg R,S}$ (resp. $X_{Q,\neg R,\neg S}$) denotes the set of vertices in $X_{Q,\neg R}$ that are linked to at least one vertex (resp. to no vertex) in X_S by an arc in E_{TU}:*
  $X_{Q,\neg R,S} = \{x \in X_{Q,\neg R} \mid \exists y \in X_S \wedge ((x,y) \in E_{TU} \vee (y,x) \in E_{TU})\}$ and $X_{Q,\neg R} = X_{Q,\neg R,S} \dot\cup X_{Q,\neg R,\neg S}$.

[6] In the context of CP(Graph) [10], these two digraphs respectively correspond to the lower and upper bounds of a graph variable. Note that, as a consequence, our approach can easily be adapted to providing a generic filtering for CP(Graph).

- *$E_{R,Q}$ is the set of arcs in E_R that are incident on at least one vertex in X_Q:* $E_{R,Q} = \{e \in E_R \mid \exists x \in X_Q \wedge x \in e\}$.
- *$E_{R,Q,S}$ is the set of arcs in E_R that are incident on one vertex in X_Q and on one vertex in X_S:* $E_{R,Q,S} = \{e \in E_R \mid \exists x \in X_Q \wedge \exists y \in X_S \wedge (e = (x,y) \vee e = (y,x))\}$.

Notation 3 *Given a digraph G and subsets $\mathcal{X}$ of vertices and $\mathcal{E}$ of arcs:*

- *$\overrightarrow{G}(\mathcal{X}, \mathcal{E})$ denotes the induced subgraph of G containing all vertices in $\mathcal{X}$ and all arcs of $\mathcal{E}$ having their two extremities in $\mathcal{X}$.*
- *$\overleftrightarrow{G}(\mathcal{X}, \mathcal{E})$ denotes the corresponding undirected graph: to one edge (u, v) corresponds at least one arc (or loop) (u, v) or (v, u) in $\overrightarrow{G}(\mathcal{X}, \mathcal{E})$. Note that $\mathcal{E}$ still refers to the set of arcs of digraph $\overrightarrow{G}(\mathcal{X}, \mathcal{E})$ and not to the derived sets of edges.*
- *$vertex(G)$ denotes the set of vertices of G.*
- *$cc(G)$ denotes the set of connected components of G and $cc_{[cond]}(G)$ denotes the subset of connected components that satisfy a given condition $cond$.*

Note that $\overrightarrow{G}(X_{TU}, E_{TU})$ is the intermediate digraph, and that several notations may represent the same subset of arcs or vertices. For example, according to the normalization constraints (2), $X_{U,T}$ and $E_{T,U,T}$ are always the empty set and $X_{U,\neg T}$ is always equal to X_U.

Example 3. We illustrate some sets of vertices and arcs previously introduced and some graphs on the intermediate digraph depicted by Part (A) of Figure 1:

- $X_T = \{1, 2, \ldots, 9\}$, $E_T = \{(5,6), (6,5), (6,7), (7,6)\}$.
- $X_U = \emptyset$, $E_U = \{ (1,2), (2,1), (1,3), (3,1), (2,3), (3,2), (2,4), (4,2), (3,4), (4,3), (3,5), (5,3), (5,7), (7,5), (6,8), (8,6), (7,9), (9,7), (8,9), (9,8)\}$.
- $X_{T,T} = \{5, 6, 7\}$ and $X_{T,\neg T} = \{1, 2, 3, 4, 8, 9\}$ forms a partition of the set of T-vertices depending on whether or not a vertex is the extremity of a T-arc.
- $\overrightarrow{G}(X_{T,T}, E_U) = (\{5, 6, 7\}, \{(5,7), (7,5)\})$ denotes the digraph defined by the set $X_{T,T}$ of vertices (possibly isolated) and the arcs of E_U that have their two extremities in $X_{T,T}$.

Example 4. $cc_{[|E_T| \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))$ denotes the set of connected components of the intermediate digraph containing at least one T-arc.

Notation 4 *$\overleftrightarrow{G}_{rem}$ denotes the (undirected) induced subgraph of $\overleftrightarrow{G}(X_{TU}, E_U)$ obtained by removing all vertices present in $cc_{[|E_T| \geq 1]}(\overrightarrow{G}(X_T, E_T))$ and then by removing all vertices becoming isolated in the remaining undirected graph.*

Last, we recall some graph theoretic terms(other standard graph term definitions used throughout this report can be found for instance in [7]):

Definition 2. – *Given a digraph G, a sequence $(a_1, a_2, \ldots, a_k)$ of arcs of G such that, for each arc a_i $(1 \leq i < k)$ the end of a_i is equal to the start of the arc a_{i+1}, is called a* path. *A path where all vertices are distinct is called an* elementary path. *Each equivalence class of the relation "a_i is equal to a_j or there exists a path between a_i and a_j" is a* strongly connected component *of G. The* reduced digraph

of G is defined as follows: to each strongly connected component of G corresponds a vertex of the reduced digraph, and to each arc of G connecting different strongly connected components corresponds an arc in the reduced digraph (multiple arcs between the same pair of vertices are merged).

- *Given an undirected graph G, a sequence $(e_1, e_2, \ldots, e_k)$ of edges of G such that each edge has a common vertex with the previous edge, and the other vertex common to the next edge is called a* chain*. A chain where all vertices are distinct is called an* elementary chain*. An* articulation point *(resp. a* bridge*) of an undirected graph G is a vertex (resp. an edge) whose removal increases the number of connected components. Similarly a* strong articulation point *of a digraph G is a vertex whose removal increases the number of strongly connected components.*
- *A* matching *of an undirected graph G is a subset of edges, excluding loops, such that no two edges have a vertex in common. A* maximum matching *is a matching of maximum cardinality. $\mu(G)$ denotes the* cardinality of a maximum matching *of G. If loops are allowed in the matching then it is called a l-*matching *and the maximum cardinality of an l-matching in G is denoted by $\mu_l(G)$.*
- *Given a bipartite graph $G((Y, Z), E)$, a* hitting set *of $G((Y, Z), E)$ is a subset Z' of Z such that for any vertex $y \in Y$ there exists an edge in E connecting y to a vertex in Z'. $h(G)$ denotes the* cardinality of a minimum hitting set *of G.*

4 Filtering from Bounds of Graph Parameters

This section illustrates on the examples of $\mathbf{NARC}$, $\mathbf{NVERTEX}$ and $\mathbf{NCC}$, how to filter according to a graph-parameter constraint $ctr_{\mathcal{P}_l}(Vertex, Arc, P_l)$ (Constraint (3) introduced in Proposition 1. Table 2 first recalls the generic formula to estimate the bounds of these three parameters according to the current instantiation of $Vertex$ and Arc. These results were previously given in [6]. Note that all these bounds are sharp. Then we present a reverse filtering when $\mathrm{dom}(P) = \{\underline{P}\}$ or $\mathrm{dom}(P) = \{\overline{P}\}$. The next rules allow to determine the status of U-vertices and U-arcs of the intermediate digraph whenever any final digraph must contain exactly the minimal or the maximal number of arcs, of vertices or of connected components. Before stating each theorem, we give an intuition of the corresponding formula.

4.1 Filtering from $\underline{\mathbf{NARC}}$

Theorem 1 corresponds to the case where $\mathbf{NARC}$ is equal to the lower bound $\underline{\mathbf{NARC}}$ of Table 2. It is based on the fact that $\underline{\mathbf{NARC}}$ is equal to the current number of T-arcs, $|E_T|$, plus the minimum number of additional U-arcs that should be turned into T-arcs to derive from $\overrightarrow{G}(X_{TU}, E_{TU})$ a final digraph where no T-vertex is isolated. This estimation is based on the computation of the cardinality of a minimum edge cover. Therefore, U-arcs (resp. U-vertices) that do not belong to any minimum edge cover can be turned into F-arcs (resp. F-vertices). Conversely, relatively to minimum edge covers, some U-arcs are required in any final digraph stemming from $\overrightarrow{G}(X_{TU}, E_{TU})$.

Theorem 1. *If* $\mathrm{dom}(\mathbf{NARC}) = \{\underline{\mathbf{NARC}}\}$ *then:*

Graph parameters	Bound
$\underline{\mathbf{NARC}}$	$\lvert E_T\rvert + \lvert X_{T,\neg T}\rvert - \mu(\overleftrightarrow{G}(X_{T,\neg T}, E_U))$
$\overline{\mathbf{NARC}}$	$\lvert E_{TU}\rvert$
$\underline{\mathbf{NVERTEX}}$	$\lvert X_T\rvert + h(\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$
$\overline{\mathbf{NVERTEX}}$	$\lvert X_{TU}\rvert$
$\underline{\mathbf{NCC}}$	$\lvert cc_{[\lvert X_T\rvert \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))\rvert$
$\overline{\mathbf{NCC}}$	$\lvert cc_{[\lvert E_T\rvert \geq 1]}(\overrightarrow{G}(X_T, E_T))\rvert + \mu_l(\overleftrightarrow{G}_{rem})$

Table 2. Bounds of the different graph parameters.

1. *Any U-arc connecting two connected components, not necessarily distinct, in $cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))$ is turned into an F-arc.*
2. *Any U-arc in $E_{U,U,U}$ is turned into an F-arc.*
3. *Any U-vertex in $X_{U,\neg T,\neg T}$ (i.e., not linked to any T-vertex) is turned into an F-vertex.*
4. *For any edge $e = (u, v)$, $u \neq v$, of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$ such that $u, v \in X_{T,\neg T}$ and e does not belong to any maximum matching of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$, the corresponding arcs (u, v) and (v, u) of $\overrightarrow{G}(X_{T,\neg T}, E_U)$ are turned into F-arcs.*
5. *For any edge $e = (u, v)$, $u \neq v$, of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$ such that $u, v \in X_{T,\neg T}$ and e belongs to all maximum matchings of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$, if a unique arc (u, v) or (v, u) in $\overrightarrow{G}(X_{T,\neg T}, E_U)$ corresponds to e then this arc is turned into a T-arc.*
6. *Any U-arc $e = (u, v)$ such that $u \in X_{T,\neg T}$ is saturated in all maximum matchings of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$ and $v \in X_U$ is turned into an F-arc.*
7. *Any U-loop $e = (u, u)$ such that $u \in X_{T,\neg T}$ is saturated in all maximum matchings of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$ is turned into an F-arc.*

Proof. [Theorem 1.] The quantity $|E_T| + |X_{T,\neg T}| - \mu(\overleftrightarrow{G}(X_{T,\neg T}, E_U))$ is equal to the current number of T-arcs, plus the minimum number of arcs required to connect all currently isolated T-vertices: for a given maximum matching, we count one arc per edge of this maximum matching (covering two T-vertices), plus one arc per remaining T-vertex. Setting a U-arc connecting two T-arcs increases by 1 the number of T-arcs without covering any additional T-vertex (Item 1). With respect to $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$, no edge that does not belong to a minimum edge cover can correspond to an arc that belongs to the final digraph G_f, and also no vertex that is not the extremity of at least one edge of one such minimum edge cover can belong to G_f (Items 2, 3 and 4).

Moreover, some edges are present in all minimal edge covers of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$. Let $e = (u, v)$ be such an edge. This edge e belongs to any maximum matching of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$ because its two extremities are T-vertices. If no arc corresponding to e belongs to G_f then this digraph would have a number of arcs strictly greater than $|E_T| + |X_{T,\neg T}| - \mu(\overleftrightarrow{G}(X_{T,\neg T}, E_U))$. Therefore at least one arc between u and v must be a T-arc in G_f (Item 5). As a consequence, no additional U-arc joining u or v to a U-vertex can belong to G_f (Item 6). Moreover, the loops (u, u) and (v, v) can also not belong to G_f (Item 7).

Example 5. Part (A) of Figure 2 corresponds to an intermediate digraph $\overrightarrow{G}(X_{TU}, E_{TU})$ chosen for illustrating the different items of Theorem 1 under the assumption that $\mathbf{NARC} = 6$. On this intermediate digraph we have that $|E_T| = 0$, $|X_{T,\neg T}| = 11$ and $\mu(\overleftrightarrow{G}(X_{T,\neg T}, E_U)) = 5$ and as a consequence the precondition of Theorem 1 holds. Part (B) is the corresponding undirected graph $\overleftrightarrow{G}(X_{TU}, E_{TU})$ introduced in the lower bound of $\mathbf{NARC}$: thick edges depict an instance of maximum matching of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$, while gray vertices represent T-vertices belonging to all maximum matchings of $\overleftrightarrow{G}(X_{T,\neg T}, E_U)$.

Within all figures of the report `T:i` (resp. `F:i`) depicts U-arcs or U-vertices that should be turned into T-arcs (resp. F-arcs) or T-vertices (resp. F-vertices) according to Item `i`. For instance, within Part (A) of Figure 2, the arc from 9 to 13 is labeled by `F:1` since, according to Item 1 of Theorem 1 it should be turned into an F-arc.

4.2 Filtering from $\overline{\mathbf{NARC}}$

Theorem 2 corresponds to the case where $\mathbf{NARC}$ is equal to the upper bound $\overline{\mathbf{NARC}}$ of Table 2. In this case, if one U-arc is turned into an F-arc then $\mathbf{NARC} = E_{TU}$ cannot hold.

Theorem 2. *If* $\mathrm{dom}(\mathbf{NARC}) = \{\overline{\mathbf{NARC}}\}$ *then any* U*-arc of* $\overrightarrow{G}(X_{TU}, E_{TU})$ *is turned into a* T*-arc.*

Proof. [Theorem 2.] Suppose that one arc $e \in E_U$ will not belong to a final digraph G_f. The number of arcs of G_f will be at most equal to $|E_{TU}| - 1$, which is contradictory with $\mathbf{NARC} = |E_{TU}|$.

Example 6. Part (C) of Figure 2 illustrates Theorem 2 according to the hypothesis that the final digraph should contain at least 3 arcs. The two U-arcs $(3, 1)$ and $(3, 4)$ are turned into T-arcs.

4.3 Filtering from $\underline{\mathbf{NVERTEX}}$

$\underline{\mathbf{NVERTEX}}$ is equal to the current number of T-vertices, $|X_T|$, plus the minimum number of U-vertices that should be turned into T-vertices to avoid isolated T-vertices. This estimation is based on the computation of the cardinality of a minimum hitting set. Therefore, to reach this bound, U-vertices that do not belong to any minimum hitting set can be turned into F-vertices. Conversely, some U-vertices and U-arcs are required in any final digraph stemming from $\overrightarrow{G}(X_{TU}, E_{TU})$.

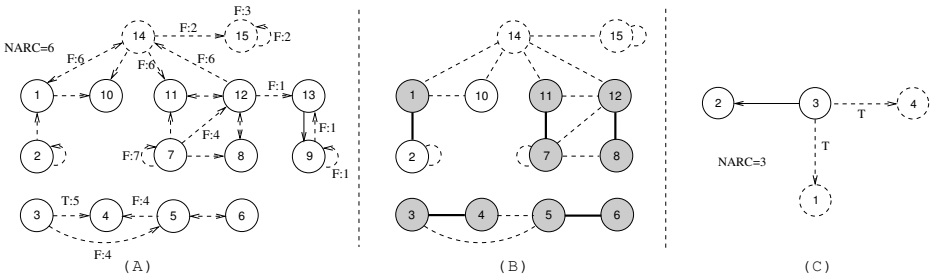

Fig. 2. Filtering according to $\mathbf{NARC}$: Illustration of Theorems 1 and 2.

Theorem 3. *If* $\mathrm{dom}(\mathbf{NVERTEX}) = \{\underline{\mathbf{NVERTEX}}\}$ *then*

1. *Any U-vertex in $X_{U,\neg T,\neg T}$ is turned into an F-vertex.*
2. *Any U-vertex in $X_{U,\neg T,T}$ that do not belong to any minimum hitting set of $\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T})$ is turned into an F-vertex (notably if it is not linked to any vertex in $X_{T,\neg T,\neg T}$).*
3. *Any U-vertex in $X_{U,\neg T,T}$ that belong to all minimum hitting sets of $\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T})$ is turned into a T-vertex.*
4. *For all edges $e = (u, v)$ such that $u \in X_{T,\neg T,\neg T}$ and $v \in X_{U,\neg T,T}$, if all minimum hitting sets are such that v is the only vertex that can be associated with u, and if a unique arc corresponds to e in $\overrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T})$, then this arc can be turned into a T-arc.*

Proof. [Theorem 3.] $h(\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$ is the minimum number of U-vertices that should be turned into T-vertices to make all isolated T-vertices in $\overrightarrow{G}(X_T, E_T)$ linked by at least one arc in the final digraph G_f. If $\mathbf{NVERTEX} = |X_T| + h(\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$ then the final digraph will contain all current T-vertices plus $h(\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$ vertices from $X_{U,\neg T,T}$. Note that any of these U-vertices belong to at least one arc such that the other extremity is a T-vertex. As a consequence, the filtering stated in Item 1 holds. Moreover if one U-vertex that does not belong to a minimum hitting set of $\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$ is turned into a T-vertex in G_f then this digraph will contain strictly more than $|X_T| + h(\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$ vertices (a contradiction). Therefore the filtering stated in Item 2 holds. If one U-vertex belongs to all minimum hitting sets then it must be taken, to prevent there from being too many vertices in G_f (Item 3). Finally, as a consequence of Item 3, if some U-arcs a required to connect such required U-vertices to current T-vertices in $\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$ then they should be turned into T-arcs (Item 4).

Example 7. Part (A) of Figure 3 illustrates Theorem 3 according to the hypothesis that the final digraph should not contain more than 7 vertices[7]:

[7] As in Figure 1, solid arrows/circles depict T-arcs/vertices and dashed arrows/circles depict U-arcs/vertices in the intermediate digraph.

– The U-vertices 1 and 10 are turned into F-vertices according to Item 1.
– Since they do not belong to any minimum hitting set, the U-vertices 4, 5 and 12 are turned into F-vertices according to Item 2.
– Since the U-vertex 9 belongs to all minimum hitting sets, it is turned into a T-vertex according to Item 3.
– Last, the U-arc $(8, 9)$ is turned into a T-arc according to Item 4.

Part (B) depicts the corresponding bipartite graph $\overleftrightarrow{G}((X_{T,\neg T,\neg T},\ X_{U,\neg T,T}),\ E_{U,T})$ used for computing the cardinality of a minimum hitting set. Thick lines correspond to a minimum hitting set.

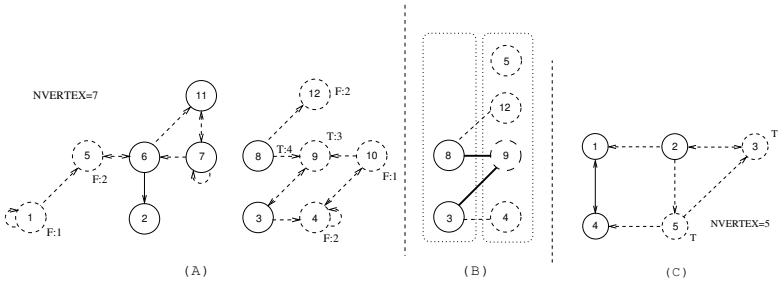

Fig. 3. Filtering according to $\mathbf{NVERTEX}$: Illustration of Theorems 3 (A, B) and 4 (C).

Since Theorem 3 involves computing the cardinality of a minimum hitting set, which is exponential, we provide a weaker form of Theorem 3.

Corollary 1. *If* $\mathrm{dom}(\mathbf{NVERTEX}) = \{|X_T|\}$ *then all* U*-vertices can be turned into* F*-vertices.*

4.4 Filtering from $\overline{\mathbf{NVERTEX}}$

$\overline{\mathbf{NVERTEX}}$ corresponds to final digraphs derived $\overleftrightarrow{G}(X_{TU}, E_{TU})$ by turning all U-vertices into T.

Theorem 4. *If* $\mathrm{dom}(\mathbf{NVERTEX}) = \{\overline{\mathbf{NVERTEX}}\}$ *then any* U*-vertex of* $\overleftrightarrow{G}(X_{TU}, E_{TU})$ *is turned into a* T*-vertex.*

Proof. [Theorem 4.] If one vertex $v \in X_U$ does not belong to the final digraph then the number of vertices of that final digraph is at most equal to $|X_{TU}| - 1$, which is contradictory with $\mathbf{NVERTEX} = |X_{TU}|$.

Example 8. Part (C) of Figure 3 illustrates Theorem 4 according to the hypothesis that the final digraph should contain at least 5 vertices. Theorem 4 turns all U-vertices into T-vertices.

4.5 Filtering from $\underline{\mathbf{NCC}}$

The minimal number of connected components in any final digraph is equal to the number of connected components in the intermediate digraph that contain at least one T-vertex.

Theorem 5. *If* $\mathrm{dom}(\mathbf{NCC}) = \{\underline{\mathbf{NCC}}\}$ *then*

1. *Any U-arc or U-vertex of any connected component in $|cc_{[|X_T|=0]}(\overrightarrow{G}(X_{TU}, E_{TU}))|$ is turned into an F-arc or an F-vertex.*
2. *Any U-vertex that is an articulation point of $\overleftrightarrow{G}(X_{TU}, E_{TU})$ such that its removal disconnects two T-vertices[8] is turned into a T-vertex.*
3. *For any edge e of $\overleftrightarrow{G}(X_{TU}, E_{TU})$ that is a bridge such that its removal disconnects two T-vertices, if a unique U-arc in $\overrightarrow{G}(X_{TU}, E_{TU})$ corresponds to e then this U-arc is turned into a T-arc.*

Proof. [Theorem 5.] Since all T-vertices appear in any final digraph G_f, a subpart of each connected component in $cc_{[|X_T|\geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))$ will necessarily belong to a distinct connected component of G_f. By definition, no additional connected component of $\overrightarrow{G}(X_{TU}, E_{TU}))$ can belong to G_f. As a consequence, Item 1 holds. Moreover, existing connected components containing a T-arc should not be split. Therefore some U-vertices and U-arcs should necessarily belong to G_f. Items 2 and 3 hold.

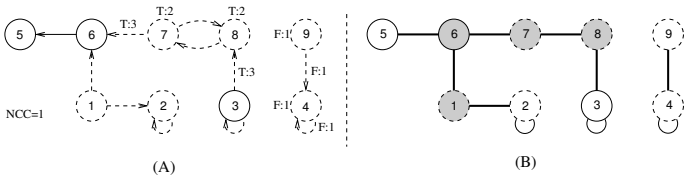

Fig. 4. Filtering according to $\underline{\mathbf{NCC}}$: Illustration of Theorem 5.

Example 9. Part (A) of Figure 4 illustrates Theorem 5 according to the hypothesis that the final digraph should contain no more than one connected component. Part (B) represents the undirected graph $\overleftrightarrow{G}(X_{TU}, E_{TU})$, where grey vertices correspond to articulation points and thick lines correspond to bridges. Since $\overrightarrow{G}(X_{TU}, E_{TU})$ contains one single connected component involving T-vertices, the precondition of Theorem 5 holds and we get the following filtering:

- Since the connected component of $\overrightarrow{G}(X_{TU}, E_{TU})$ with vertices $\{4, 9\}$ belongs to $cc_{[|X_T|=0]}(\overrightarrow{G}(X_{TU}, E_{TU}))$, then, from Item 1, U-vertices 4 and 9 as well as U-arcs $(4, 4)$ and $(9, 4)$ are respectively turned into F-vertices and F-arcs.
- From Item 2, the two U-vertices 7 and 8, which are articulation points of $\overleftrightarrow{G}(X_{TU}, E_{TU})$ belonging to an elementary chain between two T-vertices (3 and 6), are turned into T-vertices.
- From Item 3, among the 3 bridges of $\overleftrightarrow{G}(X_{TU}, E_{TU})$ belonging to an elementary chain between two T-vertices (3 and 6), $(3, 8)$ and $(7, 6)$ are turned into T-arcs since their respective counterparts $(8, 3)$ and $(6, 7)$ do not belong to $\overrightarrow{G}(X_{TU}, E_{TU})$.

[8] The two T-vertices do not belong any more to the same connected component.

4.6 Filtering from $\overline{\mathbf{NCC}}$

$\overline{\mathbf{NCC}}$ is equal to the current number of connected components of $\overrightarrow{G}(X_T, E_T)$ containing at least one T-arc (that is, $|cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))|$) plus the cardinality of a maximum matching $\mu_l(\overleftrightarrow{G}_{rem})$, which is the maximum possible number of new connected components that could be present in a final digraph stemming from $\overrightarrow{G}(X_{TU}, E_{TU})$, in addition to $|cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))|$. Then all U-arcs (and U-vertices) that may reduce the number of connected components if they would belong to the final digraph have to be turned into F-arcs (and F-vertices). For instance, a U-arc joining two connected components with T-arcs. Moreover, some U-vertices are present in any maximum matching $\mu_l(\overleftrightarrow{G}_{rem})$. They should be turned into T-vertices since they will be required in any final digraph.

Theorem 6. *If* $\mathrm{dom}(\mathbf{NCC}) = \{\overline{\mathbf{NCC}}\}$ *then*

1. *Any U-arc of $\overrightarrow{G}(X_T, E_{TU})$ joining two T-vertices belonging to two distinct connected components in $cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))$ is turned into an F-arc.*
2. *For any edge in $\overleftrightarrow{G}_{rem}$ that does not belong to any maximum l-matching of $\overleftrightarrow{G}_{rem}$, the corresponding U-arc(s) are turned into F-arcs.*
3. *Any U-arc $e = (u, v)$ such that u belongs to a connected component in $cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))$ and v is saturated in every maximum l-matchings of $\overleftrightarrow{G}_{rem}$ is turned into an F-arc.*
4. *Any U-vertex of $\overleftrightarrow{G}_{rem}$ belonging to all maximum l-matchings of $\overleftrightarrow{G}_{rem}$ is turned into a T-vertex.*
5. *For all edges e belonging to all maximum l-matchings of $\overleftrightarrow{G}_{rem}$, if a unique U-arc in $\overrightarrow{G}(X_{TU}, E_U)$ corresponds to e then this arc is turned into a T-arc.*
6. *Any U-vertex of $\overleftrightarrow{G}_{rem}$ that does not belong to any maximum l-matching of $\overleftrightarrow{G}_{rem}$ is turned into an F-vertex.*

Proof. [Theorem 6.] If one additional arc joins two connected components in $cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))$ then the number of connected components of the final digraph will be strictly less than $|cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))| + \mu_l(\overleftrightarrow{G}_{rem})$, a contradiction. Therefore Item 1 holds.
Similarly U-arcs corresponding to edges in $\overleftrightarrow{G}_{rem}$ that do not belong to at least one maximum l-matching of $\overleftrightarrow{G}_{rem}$ cannot belong to the final digraph because their presence would decrease the number of new connected components. Therefore Item 2 holds.
Furthermore, no U-arc that joins a connected component in $cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))$ with a vertex saturated in all maximum matchings of $\overleftrightarrow{G}_{rem}$ can be turned into a T-arc. Therefore Item 3 holds.
On the contrary, U-vertices saturated in all maximum matchings of $\overleftrightarrow{G}_{rem}$ and U-arcs belonging to all such maximum matchings should necessarily belong to G_f. Items 4 and 5 hold.

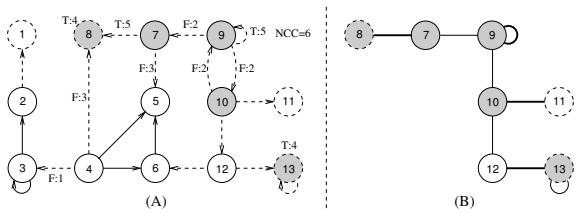

Fig. 5. Filtering according to $\overline{\mathbf{NCC}}$: Illustration of Theorem 6.

Example 10. Part (A) of Figure 5 illustrates Theorem 6 according to the hypothesis that the final digraph should contain at least 6 connected components. $cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))$ consists of the following two connected components, respectively corresponding to the sets of vertices $\{2, 3\}$ and $\{4, 5, 6\}$. Part (B) illustrates the corresponding undirected graph $\overleftrightarrow{G}_{rem}$, where thick lines correspond to a maximum l-matching of cardinality 4, and grey vertices are vertices that are saturated in all maximum l-matchings. Since the precondition $\mathbf{NCC} = |cc_{[|E_T|\geq 1]}(\overrightarrow{G}(X_T, E_T))| + \mu_l(\overleftrightarrow{G}_{rem}) = 6$ of Theorem 6 holds, Items 1, 2 and 3 respectively turn the U-arcs of $\{(4, 3)\}$, of $\{(9, 7), (9, 10), (10, 9)\}$ and of $\{(4, 8), (7, 5)\}$ into F-arcs. Item 4 turns the U-vertices $\{8, 13\}$ into T-vertices. Finally, Item 5 turns the U-arcs $\{(7, 8), (9, 9)\}$ into T-arcs.

4.7 Complexity Results

Table 3 provides complexity results for the triggering conditions as well as for each item of the theorems that were previously introduced. Most of these items correspond directly to an existing graph problem that we mention in the third column of the table. We only describe how to adapt depth first search in order to handle Item 2 of Theorem 5. The complexity stated for each item of a theorem assumes that the corresponding triggering condition was already computed: for instance, assuming that a maximum cardinality matching was already computed, identifying vertices that are saturated in every maximum cardinality matching is linear in the number of edges of the graph [5].

Implementing Item 2 of Theorem 5. We need to identify all U-vertices that are articulation points of the intermediate digraph such that there removal disconnect two T-vertices. For this purpose, we adapt the depth first search algorithm used for computing the articulation points of a connected graph. This algorithm characterizes the articulation points by one of the following conditions:

1. A vertex v, different from the root of the depth first search tree, is not an articulation point if every child c in the depth first search tree has some vertex lower in the tree connected to a vertex higher in the tree than v.
2. The root v of the search tree is an articulation point if it has two or more children.

In order to find out whether an articulation point disconnects two T-vertices or not, we have to associate to each vertex w the following information:

– The number, $T_{before}[w]$, of T-vertices on the path from the root of the depth first search tree to w.

Theorem Parts	Complexity	Graph Related Problems
Theorem 1		
• Triggering	$O(m\sqrt{n})$	*maximum cardinality matching* [13]
• Items 1,2	$O(m)$	*iterating through the arcs*
• Item 3	$O(n)$	*iterating through the vertices*
• Items 4,5	$O(m \cdot n)$	*identifying edges that do not belong to any maximum cardinality matching* [15]
• Items 6,7	$O(m)$	*identifying vertices that are saturated in every maximum cardinality matching* [5]
Theorem 2		
• Triggering	$O(m)$	*iterating through the arcs*
• Item 1	$O(m)$	*iterating through the arcs*
Theorem 3		
• Triggering	NP	*cardinality of a minimum hitting set* [11]
• Item 1	$O(m)$	*iterating through the arcs*
• Items 2,4	NP	*identifying vertices that do not belong to any minimum hitting set*
• Item 3	NP	*identifying vertices that belong to every minimum hitting set*
Corollary 1		
• Triggering	$O(n)$	*iterating through the vertices*
• Item 1	$O(n)$	*iterating through the vertices*
Theorem 4		
• Triggering	$O(n)$	*iterating through the vertices*
• Item 1	$O(n)$	*iterating through the vertices*
Theorem 5		
• Triggering	$O(n)$	*iterating through the vertices*
• Item 1	$O(m)$	*iterating through the arcs*
• Items 2,3	$O(m)$	*depth first search*
Theorem 6		
• Triggering	$O(m\sqrt{n})$	*maximum cardinality matching* [13]
• Item 1	$O(m)$	*computing the connected components*
• Items 2,5,6	$O(m \cdot n)$	*identifying edges that do not belong to any maximum cardinality matching* [15]
• Items 3,4	$O(m)$	*identifying vertices that are saturated in every maximum cardinality matching* [5]

Table 3. Complexity of each theorem. m and n respectively denote the number of arcs and the number of vertices in the intermediate digraph.

– The number, $T_{after}[w]$, of T-vertices in the descendants of w[9] in the depth first search tree.

With this information, each time we finish exploring all the descendants of a children c of a U-vertex v, we check condition (1.) and test if $T_{before}[v] \geq 1$ and $T_{after}[c] \geq 1$ both hold. If this is the case, we have identified an articulation point whose removal disconnects two T-vertices. Finally, if a U-vertex v is an articulation point with at least two children c_1 and c_2 that cannot both be connected to a vertex higher in the tree than v and such that $T_{after}[c_1] \geq 1$ and $T_{after}[c_2] \geq 1$ both hold, then v is also an articulation point whose removal disconnects two T-vertices.[10]

5 Relating to Ad-Hoc Filtering

At this point one may wonder whether our generic graph-based filtering is not too theoretical in order to have any practical interest. We show that we can obtain rational reconstructions of several ad-hoc algorithms that were constructed for specific global constraints. For this purpose, we first consider the `proper_forest` constraint, which was introduced in Example 1 and for which a specialized filtering algorithm was recently proposed in [5].After recalling the different steps of this algorithm, we "deconstruct" this algorithm and reinterpret its parts in terms of our generic graph-based filtering. We finally also consider the `among` constraint, which was introduced in [4] and for which a specialized filtering algorithm was proposed by Bessière *et al.* in [8].

5.1 Retrieving the filtering algorithm of `proper_forest`

The specialized filtering algorithm of the `proper_forest` constraint is made up from the following steps:

1. Renormalize $\overrightarrow{G}(X_{TU}, E_{TU})$ according to the fact that the final digraph has to be symmetric.
2. Check the feasibility of the `proper_forest` constraint:
   (a) The intermediate digraph has no isolated vertex.
   (b) There is no cycle made up from T-arcs.
   (c) `NTREE` has at least one value in $[$`MINTREE`, `MAXTREE`$]$ where `MINTREE` is the number of connected components of the intermediate digraph and `MAXTREE` is the number of connected components of $\overrightarrow{G}(X_T, E_T)$ plus the cardinality of a maximum cardinality matching in the subgraph induced by the vertices that are not linked to any T-vertices.
3. Every U-arc that would create a cycle of T-vertices, is turned into an F-arc.
4. The minimum and maximum values of `NTREE` are respectively adjusted to `MINTREE` and `MAXTREE`.
5. When `NTREE` is fixed to `MINTREE` all U-arcs corresponding to bridges of $\overrightarrow{G}(X_{TU}, E_{TU})$ are turned into T-arcs.
6. When `NTREE` is fixed to `MAXTREE` each U-arc (u, v) satisfying one of the following conditions is turned into an F-arc:

[9] w is considered as a descendant of itself.

[10] Remember that, when we build a depth first search tree on an undirected graph, we cannot have any edge between two subtrees of a given vertex.

(a) Both u and v belong to two distinct connected components of $\overrightarrow{G}(X_T, E_T)$ involving more than one vertex.
(b) (u, v) does not belong to any maximum matching in the subgraph induced by the vertices that are not extremities of any T-arc.
(c) u is the extremity of a T-arc and v is saturated in every maximum matching in the subgraph induced by the vertices that are not linked to any T-vertices.

7. Every U-arc involving a source or a sink is turned into a T-arc.

By considering the generic graph-based reformulation of Proposition 1 on the graph property $\texttt{NTREE} = \mathbf{NCC}$ we retrieve almost all the steps of the previous algorithm (except steps 2(b) and 3, which come from the invariant $\mathbf{NARC} = 2 \cdot (n - \mathbf{NCC})$ linking the two graph parameters $\mathbf{NARC}$ and $\mathbf{NCC}$):

- Item 1 corresponds to posting the graph-class constraints (5) (in the context of `proper_forest`, the final digraph has to be symmetric).
- Item 2(a) corresponds to posting the normalization constraints (2).
- Item 2(c) corresponds to checking the graph property constraint $\mathbf{NCC} = \texttt{NTREE}$ (4). `MINTREE` and `MAXTREE` respectively correspond to the general lower and upper bounds of $\mathbf{NCC}$ given in Table 2 and deduced from constraint (3).
- Item 4 is the propagation induced by the graph property constraint (4).
- Item 5 and Item 6 correspond to the propagation of constraint (3) on the graph parameter $\mathbf{NCC}$: Item 5 is the specialization of Theorem 5, namely its third item (since the first two items of Theorem 5 are irrelevant because $\overrightarrow{G}(X_{TU}, E_{TU})$ does not contain any U-vertex). Item 6 corresponds to Theorem 6.
- Finally, Item 7 corresponds to the propagation obtained by the graph-class constraint (5), which avoids the creation of any isolated vertex in the symmetric graph (see Example 2).

5.2 Retrieving the filtering algorithm of `among`

Consider the `among`(`NVAR`, `VARIABLES`, `VALUES`) constraint, introduced in [4], which receives a domain variable `NVAR`, a collection of domain variables `VARIABLES` and a collection of distinct integer values `VALUES`. The `among`(`NVAR`, `VARIABLES`, `VALUES`) constraint enforces `NVAR` to be equal to the number of variables of the collection `VARIABLES` assigned to a value of `VALUES`. We now show that we can retrieve the filtering algorithm for `among` recently introduced by Bessière *et al.* in [8]. The specialized filtering algorithm of the `among` constraint is made up from the following steps:

1. Set low to the number of variables var of `VARIABLES` such that $\mathrm{dom}(var) \subseteq$ `VALUES`.
2. Set up to the total number of variables of `VARIABLES` minus the number of variables var of `VARIABLES` such that $\mathrm{dom}(var) \cap$ `VALUES` $= \emptyset$.
3. Update the minimum value of `NVAR` to low.
4. Update the maximum value of `NVAR` to up.
5. If `NVAR` is fixed to low then, for all variables var of `VARIABLES` such that $\mathrm{dom}(var) \not\subseteq$ `VALUES`, enforce $\mathrm{dom}(var) = \mathrm{dom}(var) \setminus$ `VALUES`.
6. If `NVAR` is fixed to up then, for all variables var of `VARIABLES` such that $\mathrm{dom}(var) \cap$ `VALUES` $\neq \emptyset$, enforce $\mathrm{dom}(var) = \mathrm{dom}(var) \cap$ `VALUES`.

We now give the graph-based representation of the among constraint. To each variable var of VARIABLES corresponds a vertex of the initial digraph of the among constraint. In addition each vertex var has a loop corresponding to the unary arc-constraint $var \in$ VALUES. By considering the generic graph-based reformulation of Proposition 1 on the graph property NVAR $= \mathbf{NARC}$ we retrieve all the steps of the previous algorithm:

- The lower bound of NVAR corresponds to a simplified form of the lower bound of $\mathbf{NARC}$ depicted by Table 2, namely $|E_T| + |X_{T,\neg T}| - \mu(\overleftrightarrow{G}(X_{T,\neg T}, E_U))$. Since the initial digraph of the among constraint contains only loops, $X_{T,\neg T}$ is empty and we retrieve the lower bound $|E_T|$.
- The upper bound of NVAR corresponds to the upper bound of $\mathbf{NARC}$ depicted by Table 2, namely $|E_{TU}|$.
- Item 5 is the specialization of Theorem 1, namely its second and third items.
- Item 6 corresponds to Theorem 2.

6 Specializing the Filtering According to Graph Classes

Quite often the final digraph of a global constraint has a regular structure inherited from the initial digraph or stemming from some property of the arc constraint. This translates as extra elementary constraints, the *graph-class constraints* (5), in the graph-based reformulation of the global constraint. Enforcing these constraints before evaluating the graph parameter bounds in the intermediate digraph allows to refine the bound formula of Table 2 and the bound-based filtering (Section 4), both in terms of sharpness and of algorithmic complexity.

This section illustrates this principle on the path_with_loops graph class for the four graph parameters $\mathbf{NVERTEX}$, $\mathbf{NCC}$, $\mathbf{MIN\_NCC}$, and $\mathbf{MAX\_NCC}$. The filtering is then experimentally evaluated on the example of the group constraint, which belongs to the path_with_loops graph class and which involves these four parameters in its graph-based description.

6.1 The path_with_loops Graph Class

The path_with_loops graph class groups together global constraints with the following graph-based description:

- The initial digraph uses the $PATH$ and the $LOOP$ arc generators. It consists of a sequence of vertices $X = \{x_1, \ldots, x_n\}$ with one arc $(x_j, x_{j+1}) \in E_{\mathcal{R}}$, $j \in \{1, \ldots, n-1\}$, for each pair of consecutive vertices, and one loop $(x_j, x_j) \in E_{\mathcal{R}}$, $j \in \{1, \ldots, n\}$, on each vertex (see Part (A) of Figure 6).
- In any final digraphs, each remaining vertex has its loop and two consecutive vertices remain linked by an arc (see Part (B) of Figure 6). These conditions correspond to the following graph-class constraints in the reformulation of Proposition 1:

$$vertex_j = arc_{j,j} \tag{6}$$

$$\min(vertex_j, vertex_{j+1}) = arc_{j,j+1} \tag{7}$$

Among the global constraints belonging to the path_with_loops graph class, the catalog mentions for example group [9] and stretch [14]. Such constraints enforce sequences of variables to satisfy given patterns.

Example 11. Consider the group (NGROUP, MIN_SIZE, MAX_SIZE, MIN_DIST, MAX_DIST, NVAL, VARIABLES, VALUES) constraint, where the first six parameters are domain variables, while VARIABLES is a sequence of n domain variables and VALUES a finite set of integers.

Let $x_i, x_{i+1}, \ldots, x_j$ $(1 \leq i \leq j \leq n)$ be consecutive variables of the sequence VARIABLES such that all the following conditions simultaneously apply: (1) all variables $x_i, \ldots, x_j$ take their value in the set of values VALUES, (2) either $i = 1$, or x_{i-1} does not take a value in VALUES, (3) either $j = n$, or x_{j+1} does not take a value in VALUES. We call such a set of variables a *group*. The constraint group is fulfilled if all the following conditions hold:

- there are exactly NGROUP groups of variables,
- MIN_SIZE and MAX_SIZE are the number of variables of the smallest and largest groups,
- MIN_DIST and MAX_DIST are the minimum and maximum number of variables between two consecutive groups or between one border and one group,
- NVAL is the number of variables taking their value in the set VALUES.

For instance, group$(2, 2, 4, 1, 2, 6, \langle 0, 0, 1, 3, 0, 2, 2, 2, 3\rangle, \{1, 2, 3\})$ holds since the sequence $\langle 0, 0, 1, 3, 0, 2, 2, 2, 3\rangle$ contains 2 groups $\langle 1, 3\rangle$ and $\langle 2, 2, 2, 3\rangle$ of nonzero values of size 2 and 4, 2 groups $\langle 0, 0\rangle$ and $\langle 0\rangle$ of zeros, and 6 nonzero values. The graph-based description of the group constraint uses two graph constraints which respectively mention the graph properties $\mathbf{NCC}$ = NGROUP, $\mathbf{MIN\_NCC}$ = MIN_SIZE, $\mathbf{MAX\_NCC}$ = MAX_SIZE, $\mathbf{NVERTEX}$ = NVAL and $\mathbf{MIN\_NCC}$ = MIN_DIST, $\mathbf{MAX\_NCC}$ = MAX_DIST. Figure 6 depicts the initial digraph of well as the two final digraphs associated to the two graph constraints of the example given for the group constraint.

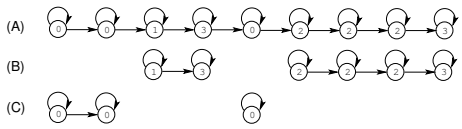

Fig. 6. Initial (A) and final digraphs (B,C) of group.

6.2 Bounds and Filtering for the path_with_loops Graph Class

The path_with_loops properties highlight well the interest of specializing the parameter bound formula and the filtering rules. Firstly, in this context, the path structure of the considered digraphs naturally makes the different algorithms polynomial. The status of vertices and arcs can be determined and fixed during filtering in linear time by just following the path from vertex x_1 to vertex x_n.

Secondly, some general bounds are not sharp anymore in this context because of the additional graph-class constraints. Refining those bounds then leads to a more accurate filtering. This is the case for $\overline{\mathbf{NVERTEX}}$, $\overline{\mathbf{NCC}}$, $\underline{\mathbf{MIN\_NCC}}$, $\overline{\mathbf{MIN\_NCC}}$, $\underline{\mathbf{MAX\_NCC}}$, and $\overline{\mathbf{MAX\_NCC}}$. Lastly, the graph-class constraints allow to simplify some bounds that are sharp both in the general and the path_with_loops contexts. This is the case for $\underline{\mathbf{NVERTEX}}$ and $\underline{\mathbf{NCC}}$.

Graph parameters	Bound
$\underline{\mathbf{NVERTEX}}$	$\lvert X_T \rvert$
$\overline{\underline{\mathbf{NVERTEX}}}$	$\lvert X_{TU} \rvert - \sum_{i \in cc(\overrightarrow{G}(X_U, E_F))} \lfloor \frac{l_i}{2} \rfloor$
$\underline{\mathbf{NCC}}$	$\lvert cc_{[\lvert X_T \rvert \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU})) \rvert$
$\overline{\underline{\mathbf{NCC}}}$	$\lvert cc(\overrightarrow{G}(X_T, E_T)) \rvert +$ $\sum_{i \in cc_{[\lvert X_{U,U,T} \rvert = 0]}(\overrightarrow{G}(X_U, E_{UF}))} \lceil \frac{l_i}{2} \rceil +$ $\sum_{i \in cc_{[\lvert X_{U,U,T} \rvert = 1]}(\overrightarrow{G}(X_U, E_{UF}))} \lfloor \frac{l_i}{2} \rfloor +$ $\sum_{i \in cc_{[\lvert X_{U,U,T} \rvert = 2]}(\overrightarrow{G}(X_U, E_{UF}))} (\lceil \frac{l_i}{2} \rceil - 1)$
$\underline{\mathbf{MIN\_NCC}}$ if $\lvert X_T \rvert = 0$ if $\lvert X_T \rvert \geq 1 \wedge \lvert X_{U,U,\neg T} \rvert \geq 1$ if $\lvert X_T \rvert \geq 1 \wedge \lvert X_{U,U,\neg T} \rvert = 0$	 0 1 $\min_{i \in cc(\overrightarrow{G}(X_T, E_T))} l_i$
$\overline{\underline{\mathbf{MIN\_NCC}}}$ if $\lvert X_T \rvert \geq 1$ if $\lvert X_T \rvert = 0$	 $\min_{i \in cc_{[\lvert X_T \rvert \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i - \epsilon$ $\max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i$
$\underline{\mathbf{MAX\_NCC}}$	$\max_{i \in cc(\overrightarrow{G}(X_T, E_T))} l_i$
$\overline{\underline{\mathbf{MAX\_NCC}}}$	$\max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i$

Table 4. Bounds of the different graph parameters in the context of the `path_with_loops` graph class.

Notation 5 *Let $\overrightarrow{G}(\mathcal{X}, \mathcal{E})$ be a subgraph of the initial digraph $\overrightarrow{G}(X, E_{\mathcal{R}})$ and $i \in cc(G)$, then the vertices of i form a subsequence of $X = \{x_1, \ldots, x_n\}$, pairwise linked by arcs in $\mathcal{E}$. The length of this subsequence is denoted by $l_i \in \{1, \ldots, n\}$ and the first index is denoted by $j_i \in \{1, \ldots, n - l_i + 1\}$. To ease notations, we assume that x_0 and x_{n+1} denote two dummy F-vertices (with two F-arcs (x_0, x_1) and (x_n, x_{n+1})) representing respectively the source and the sink of the initial graph path.*

Table 4[11] summarizes the bounds of these graph parameters in the `path_with_loops` graph class. Note that all these bounds can be evaluated in $O(n)$ time by iterating once through the n vertices of the initial digraph. Furthermore all the bounds are sharp when considering the graph-class constraints. As in the general case, they are derived by considering those final digraphs that minimize or maximize the corresponding graph parameter. Again identifying U-arcs and U-vertices belonging or not to these digraphs yields filtering rules that apply to the intermediate digraph when a given bound has to be reached.

We give thereafter, for each of the eight considered bounds, the proofs of the formula as well as the filtering rules applying when the bound has to be reached in any final digraph. Note that, similarly to the bounds, all the following filtering theorems can be implemented in $O(n)$ time.

Filtering from $\underline{\mathbf{NVERTEX}}$

Corollary 2. $\underline{\mathbf{NVERTEX}} = |X_T|$ *and this bound is sharp.*

Proof. This bound is equal to the one of the general case $\underline{\mathbf{NVERTEX}} = |X_T| + h(\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$. Indeed, because of Constraint (6), each T-vertex is linked to itself, then $X_{T,\neg T,\neg T} = \emptyset$ and the second term of the formula becomes 0. Furthermore, this bound remains sharp because subgraph (X_T, E_T) is a possible final digraph (i.e. (i) any arc in E_T has its two extremities in X_T, (ii) this graph contains all T-elements and (iii) it satisfies Constraints (2), (6) and (7)).

Because bound $\underline{\mathbf{NVERTEX}}$ is identical to the general case, the corresponding filtering is a simplified reformulation of Theorem 3:

Corollary 3. *If* $\mathrm{dom}(\mathbf{NVERTEX}) = \{\underline{\mathbf{NVERTEX}}\}$ *then any U-vertex of $\overrightarrow{G}(X_{TU}, E_{TU})$ is turned into an F-vertex.*

Filtering from $\overline{\mathbf{NVERTEX}}$

Theorem 7. $\overline{\mathbf{NVERTEX}} = |X_{TU}| - \sum_{i \in cc(\overrightarrow{G}(X_U, E_F))} \lfloor \frac{l_i}{2} \rfloor$ *and this bound is sharp.*

[11] By convention in these formulas, a maximum value over the empty set is zero. In the formula for $\overline{\mathbf{MIN\_NCC}}$, $\epsilon = 1$ if there exist two adjacent (linked by an F-arc) connected components of minimum size in $cc_{[|X_T| \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))$, $\epsilon = 0$ otherwise.

Proof. Because of the graph-class constraints, the general bound $\overline{\mathbf{NVERTEX}} = |X_{TU}|$ is not sharp in this context: Constraints (7) imply that two consecutive U-vertices linked by an F-arc cannot both be turned into T-vertices, while any other U-vertex can be turned into a T-vertex or an F-vertex independently of its neighbors. By definition, graph $\overrightarrow{G}(X_U, E_F)$ is the subgraph $(X_U, E_{F,U,U})$ of the initial digraph. When turning all U-vertices into T-vertices except every other vertex in each element of $cc(\overrightarrow{G}(X_U, E_F))$ and then by turning all U-arcs incident on two T-vertices into T-arcs, we obtain a possible final digraph. In order to maximize $\mathbf{NVERTEX}$, each connected component $i \in cc(\overrightarrow{G}(X_U, E_F))$ with an odd number of vertices is processed by turning vertices $x_{j_i}, x_{j_i+2}, \ldots, x_{j_i+l_i-1}$ into T-vertices, and vertices $x_{j_i+1}, x_{j_i+3}, \ldots, x_{j_i+l_i-2}$ into F-vertices. In such a graph, $\mathbf{NVERTEX}$ is equal to $\overline{\mathbf{NVERTEX}}$, so the bound is sharp.

Theorem 8. *If* $\mathrm{dom}(\mathbf{NVERTEX}) = \{\overline{\mathbf{NVERTEX}}\}$ *then consider each connected component* $i \in cc(\overrightarrow{G}(X_U, E_F))$. *If* l_i *is odd then each vertex* $x_{j_i+2\cdot k}$ *with* $(k \in [0, \lfloor \frac{l_j}{2} \rfloor])$ *is turned into a* T*-vertex and each vertex* $x_{j_i+2\cdot k-1}$ *with* $(k \in [1, \lfloor \frac{l_j}{2} \rfloor])$ *into an* F*-vertex.*

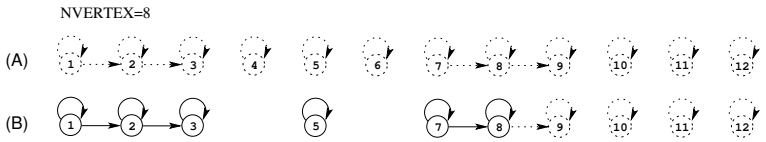

Fig. 7. Initial (A) and final digraphs (B) after applying Theorem 8 when $\mathbf{NVERTEX} = 8$.

Example 12. Figure 7 illustrates Theorem 8 according to the hypothesis that the final digraph should contain 8 vertices. The connected components of $\overrightarrow{G}(X_U, E_F)$ respectively correspond to the vertex sets $\{1\}$, $\{2\}$, $\{3, 4, 5, 6, 7\}$, $\{8\}$ and $\{9, 10, 11, 12\}$. The bound $\overline{\mathbf{NVERTEX}}$ is then equal to $12 - \lfloor \frac{1}{2} \rfloor - \lfloor \frac{1}{2} \rfloor - \lfloor \frac{5}{2} \rfloor - \lfloor \frac{1}{2} \rfloor - \lfloor \frac{4}{2} \rfloor = 8$ and the precondition of Theorem 8 holds. Since the last connected component contains an even number of vertices, nothing can be deduced about their status. On the other hand, the first connected component contains an odd number of vertices, so vertices $1, 2, 3, 5, 7, 8$ are turned into T-vertices and vertices $4, 6$ are turned into F-vertices.

Filtering from $\underline{\mathbf{NCC}}$

Corollary 4. $\underline{\mathbf{NCC}} = |cc_{[|X_T| \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))|$ *and this bound is sharp.*

Proof. Bound $\underline{\mathbf{NCC}}$ is identical to the general case presented in Table 2. It remains sharp since it is reached in a possible final digraph: consider for instance, the subgraph (X_T, E_T) of the intermediate digraph obtained after turning into a T-vertex each vertex in any connected component of $\overrightarrow{G}(X_U, E_U)$ that is linked to two T-vertices, and then after enforcing Contraints (2), (6) and (7).

Theorem 5 applies in the same way as in the general case. Nevertheless, the articulation points disconnecting T-vertices (see Item 2 of Theorem 5) are more easily identified in the context of the `path_with_loops` graph class. There is also no need here to check the status of the bridges (see Item 3) since, according to the graph-class constraints 6 and 7, these arcs are turned into T-arcs once the corresponding articulation points are turned into T-vertices. This leads to the following corollary of Theorem 5:

Corollary 5. *If* $\mathrm{dom}(\mathbf{NCC}) = \{\underline{\mathbf{NCC}}\}$*:*

1. *Any U-vertex of any connected component in $cc_{[|X_T|=0]}(\overrightarrow{G}(X_{TU}, E_{TU}))$ is turned into an F-vertex.*
2. *In any connected component of $\overrightarrow{G}(X_U, E_U)$ that is linked to two distinct T-vertices, any U-vertex is turned into a T-vertex.*

Filtering from $\overline{\mathbf{NCC}}$

Theorem 9. $\overline{\mathbf{NCC}} = |cc(\overrightarrow{G}(X_T, E_T))| +$
$\sum_{i \in cc_{[|X_{U,U,T}|=0]}(\overrightarrow{G}(X_U, E_{UF}))} \lceil \frac{l_i}{2} \rceil +$
$\sum_{i \in cc_{[|X_{U,U,T}|=1]}(\overrightarrow{G}(X_U, E_{UF}))} \lfloor \frac{l_i}{2} \rfloor +$
$\sum_{i \in cc_{[|X_{U,U,T}|=2]}(\overrightarrow{G}(X_U, E_{UF}))} (\lceil \frac{l_i}{2} \rceil - 1)$ *and this bound is sharp.*

Proof. Because of Constraints (6), each U-vertex and its loop forms a potential connected component in the final digraphs deriving from the intermediate digraph at its current state. But due to Constraints (7), two consecutive vertices in a final digraph must belong to the same connected component. As a consequence, maximizing the number of connected components involves to turn at least one out of two consecutive U-arcs into an F-arc and to consider at least one T-vertex for each connected component obtained thereafter.

Hence, consider each connected component $i \in cc(\overrightarrow{G}(X_U, E_{UF}))$ together with the status (T or F) of its neighbor vertices x_{j_i-1} and $x_{j_i+l_i}$. Turn one out of two consecutive vertices in i into a T-vertex and the other one into an F-vertex so that x_{j_i} is turned into a T-vertex (resp. an F-vertex) if x_{j_i-1} is an F-vertex (resp. a T-vertex). Then turn each loop (x_j, x_j) in i into a T-arc (resp. an F-arc) if x_j is a T-vertex (resp. an F-vertex) and turn all arcs (x_j, x_{j+1}) in i into F-arcs. If (x_{j_i-1}, x_{j_i}) is a U-arc, then turn it into an F-arc (note that otherwise, it is already an F-arc). Last, if $x_{j_i+l_i-1}$ and $x_{j_i+l_i}$ are both T-vertices, then turn $(x_{j_i+l_i-1}, x_{j_i+l_i})$ into a T-arc (it is necessarily a U-arc), otherwise if $(x_{j_i+l_i-1}, x_{j_i+l_i})$ is a U-arc, then turn it into an F-arc. By construction, all elements of the intermediate digraph are now either T or F and the subgraph (X_T, E_T) satisfies Contraints (2), (6) and (7). This subgraph is then a final digraph, and its number of connected components is equal to $\overline{\mathbf{NCC}}$: the sum terms in the formula correspond respectively to the cases where x_{j_i-1} and $x_{j_i+l_i}$ are both F-vertices, where one is a T-vertex and the other one is an F-vertex, where both are T-vertices.

As in Theorem 8, the second graph-class constraints (7) allows a strong filtering from property $\mathrm{dom}(\mathbf{NCC}) = \{\overline{\mathbf{NCC}}\}$ since no two consecutive T-vertices are linked by

an F-arc. Depending on the parity of C and on the status of its direct neighbor vertices (either in X_T or X_F), we can fix the final status of each vertex in C (vertices are alternatively set to T and F: case 1 of Theorem 10) or, at least, we can remove half the arcs of C (every other arc is turned into an F-arc: case 2 of Theorem 10).

Theorem 10. *If* $\mathrm{dom}(\mathbf{NCC}) = \{\overline{\mathbf{NCC}}\}$ *then consider each connected component* i *of* $\overrightarrow{G}(X_U, E_U)$.

1. *If* l_i *is odd and* x_{j_i-1} *and* $x_{j_i+l_i}$ *have the same status (i.e. both in* X_T *or both in* X_F*), or if* l_i *is even and* x_{j_i-1} *and* $x_{j_i+l_i}$ *have different status (i.e. one in* X_T *and the other in* X_F*), then:*
   - *each vertex* $x_{j_i-1+\lambda+2k}$ *is turned into an* F*-vertex, with* $\lambda = 1$ *if* $x_{j_i-1} \in X_T$, $\lambda = 0$ *otherwise, and* $(k \in [1-\lambda, \lfloor \frac{l_i-\lambda}{2} \rfloor])$
   - *all other vertices in* i *are turned into* T*-vertices.*
2. *Otherwise, each arc* $(x_{j_i-1+\lambda+2k}, x_{j_i+\lambda+2k})$ *is turned into an* F*-arc, with* $\lambda = 1$ *if* $x_{j_i-1} \in X_T$, $\lambda = 0$ *otherwise, and* $(k \in [1-\lambda, \lfloor \frac{l_i-\lambda-1}{2} \rfloor])$

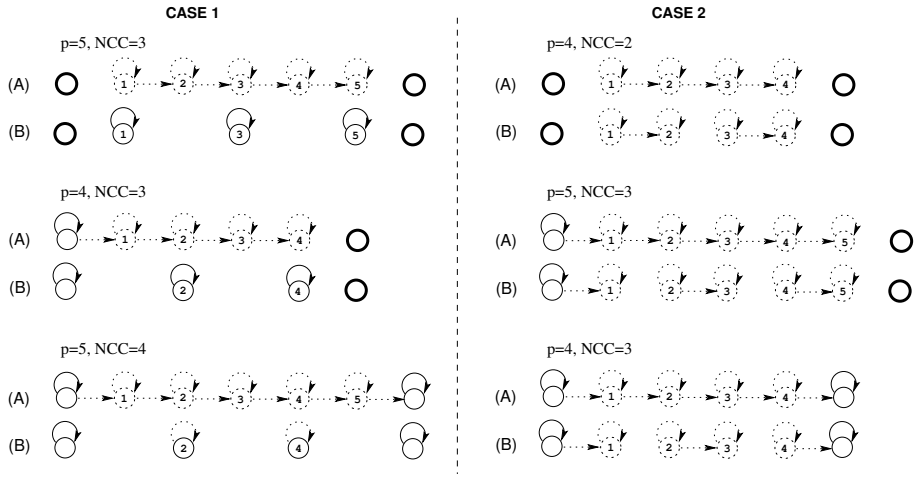

Fig. 8. Initial (A) and final digraphs (B) after applying Theorem 10 for various configurations (thick circles denote F-vertices).

Filtering from $\underline{\mathbf{MIN\_NCC}}$

Theorem 11.

$$\underline{\mathbf{MIN\_NCC}} = \begin{cases} 0 & \textit{if} |X_T| = 0 \\ 1 & \textit{if} |X_T| \geq 1 \wedge |X_{U,U,\neg T}| \geq 1 \\ \min_{i \in cc(\overrightarrow{G}(X_T, E_T))} l_i & \textit{if} |X_T| \geq 1 \wedge |X_{U,U,\neg T}| = 0 \end{cases}$$

and this bound is sharp.

Proof. If the intermediate digraph does not contain any T-vertex we can set all U-vertices to F and we get a lower bound of 0. Otherwise, if we have at least one U-vertex that is not linked to any T-vertex, then we can turn it into T, and turn into F any neighboring U-vertex. As a consequence we get a lower bound of 1. Finally, if every U-vertex is linked to at least one T-vertex, and since consecutive T-vertices can not disappear from any final digraph, a lower bound corresponds to the smallest connected components of $\overrightarrow{G}(X_T, E_T)$. In this context this lower bound can be achieved by turning all U-vertices into F.

Theorem 12 presents the filtering when $\mathbf{MIN\_NCC}$ is set at its lower bound. The principle consists in finding potential candidates for a minimal connected component of size $\underline{\mathbf{MIN\_NCC}}$. Depending on the number of candidates and on their respective positions, it assumes that at least one of the candidate components will not be extended to the neighbor vertices in any final digraph.

Definition 3. *The* candidate minimal components *of the intermediate digraph* $\overrightarrow{G}(X_{TU}, E_{TU})$ *are:*

- *any (minimal) connected components of* $\overrightarrow{G}(X_T, E_T)$ *of size* $\underline{\mathbf{MIN\_NCC}}$*,*
- *and any component, of size 1, made of one single vertex of* $X_{U,U,\neg T}$ *and its loop.*

Observe from Table 2 that $X_{U,U,\neg T}$ is the empty set when $\mathbf{MIN\_NCC} > 1$. Furthermore, candidates are only constituted of one T-vertex or U-vertex (without neighbor in X_T) together with its loop when $\mathbf{MIN\_NCC} = 1$.

In the following theorem, i and i' denote respectively the first and the last candidate minimal components in the path formed by the intermediate digraph ($j_i + l_i \leq j_{i'}$). Items 3(a), 4 and 5 assume the existence of two adjacent candidates. By definition of $X_{U,U,\neg T}$ and according to graph-class constraints (7), this implies that these candidates are singleton components of $X_{U,U,\neg T}$ and that, consequently, $\mathbf{MIN\_NCC} = 1$ (all candidates are of size 1). For the same reasons, the condition of Item 3(b) (i and i' separated by at most one vertex $x_{j_i+l_i}$) holds only if $x_{j_i+l_i}$ is an F-vertex or if it is a U-vertex and i or i' belongs to $cc(\overrightarrow{G}(X_T, E_T))$. The condition of Item 3(c) (i and i' separated by at most two vertices $x_{j_i+l_i}$ and $x_{j_i+l_i+1}$) holds only if $x_{j_i+l_i}$ or $x_{j_i+l_i+1}$ are F-vertices or if they both are U-vertices and i and i' both belong to $cc(\overrightarrow{G}(X_T, E_T))$.

Theorem 12. *If* $\mathrm{dom}(\mathbf{MIN\_NCC}) = \{\underline{\mathbf{MIN\_NCC}}\}$ *then at most one of these conditions holds:*

1. *If* $|X_T| = 0$*, then all* U*-vertices are turned into* F*-vertices.*
2. *If there exists exactly one candidate minimal component* i*, then:*
   (a) *Any* U*-vertices of* i *are turned into* T*-vertices.*
   (b) *The* U*-vertices linked to* i *are turned into* F*-vertices.*
3. *If there exist exactly two candidate minimal components (*i *and* i'*) then:*
   (a) *If* i *and* i' *are adjacent (i.e.* $\mathbf{MIN\_NCC} = 1$ *and* $j_i + 1 = j_{i'}$*), then* U*-arcs* (x_{j_i-1}, x_{j_i})*,* (x_{j_i}, x_{j_i+1})*,* (x_{j_i+1}, x_{j_i+2}) *are all turned into* F*-arcs.*
   (b) *If* i *and* i' *are linked by one* U*-vertex (i.e.* $j_{i'} = j_i + l_i + 1$ *and* $x_{j_i+l_i} \in X_U$*), then this vertex* $x_{j_i+l_i}$ *is turned into an* F*-vertex.*

(c) *If i and i' are linked by two adjacent U-vertices (i.e. $j_{i'} = j_i + l_i + 2$ and $x_{j_i+l_i}, x_{j_i+l_i+1} \in X_U$), then the U-arc $(x_{j_i+l_i}, x_{j_i+l_i+1})$ linking these two vertices is turned into an F-arc.*

4. *If there exist exactly three candidate minimal components (i and i' the first and the last one respectively) and if they are adjacent (i.e. $\mathbf{MIN\_NCC} = 1$ and $j_i + 2 = j_{i'}$) then (x_{j_i}, x_{j_i+1}) and (x_{j_i+1}, x_{j_i+2}) are turned into F-arcs.*
5. *If there exist exactly three or four candidate minimal components (i and i' the first and the last one respectively) and if $j_i + 3 = j_{i'}$ then (x_{j_i+1}, x_{j_i+2}) is turned into an F-arc.*

Filtering from $\overline{\mathbf{MIN\_NCC}}$

Theorem 13.

$$\overline{\mathbf{MIN\_NCC}} = \begin{cases} \min_{i \in cc_{[|X_T| \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i - \epsilon & \text{if } |X_T| \geq 1 \\ \max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i & \text{if } |X_T| = 0 \end{cases}$$

with $\epsilon = 1$ if there exist two adjacent (linked by an F-arc) connected components of minimum size in $cc_{[|X_T| \geq 1]}(\overrightarrow{G}(X_{TU}, E_{TU}))$, and $\epsilon = 0$ otherwise, and this bound is sharp.

Proof. First assume that $|X_T| = 0$. Since two vertices belonging to two distinct connected components of the intermediate digraph $\overrightarrow{G}(X_{TU}, E_{TU})$ cannot belong to the same connected component of any final digraph, the quantity $\max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i$ is an upper bound on the size of the largest connected components of any final digraph. As a consequence it is also an upper bound of the size of the smallest connected components of any final digraph.
This upper bound is sharp since it can effectively be reached by selecting a connected component $c_{\max}$ of $\overrightarrow{G}(X_{TU}, E_{TU})$ containing $\max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i$ vertices and by first setting all U-arcs between two vertices of $c_{\max}$ to T-arcs and then by setting all remaining U-arcs of $\overrightarrow{G}(X_{TU}, E_{TU})$ to F-arcs. Since, by the hypothesis $|X_T| = 0$, the final digraph consists of one single connected component corresponding to $c_{\max}$.
Now assume that $|X_T| > 0$. Since all T-arcs will be part of any final digraph and since each connected component of $\overrightarrow{G}(X_{TU}, E_{TU})$ involving only U-vertices can disappear, the largest possible size for the smallest connected components cannot exceed the size of the smallest connected components of $\overrightarrow{G}(X_{TU}, E_{TU})$ containing at least one T-vertex. Moreover, because of the graph-class normalization constraints, which prevent any two vertices connected by an F-arc from both belonging to any final digraph, we have to subtract 1 if two minimum-size connected components of $\overrightarrow{G}(X_{TU}, E_{TU})$ are linked by an F-arc.

Theorem 14 presents the filtering when $\mathbf{MIN\_NCC}$ is set at its upper bound. Each connected component of size lower than $\mathbf{MIN\_NCC}$ in the intermediate graph can be eliminated from the intermediate digraph (Item 1(a) and 2(a)). Furthermore, each connected component i' of $\overrightarrow{G}(X_T, E_T)$ belongs to a connected component of size greater

than $\mathbf{MIN\_NCC}$ in any final digraph. Then the compulsory part, i.e. the vertices that belong to all the possible final components of size $\mathbf{MIN\_NCC}$ including i', can be turned into T-vertices (Item 1(b)). Last, if the graph contains no T-vertex and only one maximum connected component, then the only possible final subgraph is exactly this component alone (Item 2(b)).

Theorem 14. *If* $\mathrm{dom}(\mathbf{MIN\_NCC}) = \{\overline{\mathbf{MIN\_NCC}}\}$*:*

1. *If* $|X_T| \geq 1$ *then:*
   (a) *Any U-vertex of a connected component in* $cc_{[|X_{TU}|<\mathbf{MIN\_NCC} \wedge |X_T|=0]}$ $(\overrightarrow{G}(X_{TU}, E_{TU}))$ *is turned into an F-vertex.*
   (b) *Consider each connected component i in* $cc_{[|X_T|\geq 1]}$ $(\overrightarrow{G}(X_{TU}, E_{TU}))$*, then consider each connected component i' in cc* $(\overrightarrow{G}(X_T, E_T))$ *included in i, then any U-vertex x_j with* $j \in [\min(j_i + l_i - \mathbf{MIN\_NCC}, j_{i'}), \max(j_i + \mathbf{MIN\_NCC} - 1, j_{i'} + l_{i'} - 1)]$ *is turned into a T-vertex.*
2. *If* $|X_T| = 0$ *then:*
   (a) *All vertices of any connected component in* $cc_{[|X_U|<\mathbf{MIN\_NCC}]}$ $(\overrightarrow{G}(X_U, E_U))$ *are turned into F-vertices.*
   (b) *If* $|cc_{[|X_U|=\mathbf{MIN\_NCC}]}(\overrightarrow{G}(X_U, E_U))| = 1$ *then all vertices of this single connected component are turned into T-vertices.*

Filtering from $\underline{\mathbf{MAX\_NCC}}$

Theorem 15. $\underline{\mathbf{MAX\_NCC}} = \max_{i \in cc(\overrightarrow{G}(X_T, E_T))} l_i$ *and this bound is sharp.*

Proof. Since every connected component of $\overrightarrow{G}(X_T, E_T)$ will be part of a connected component of any final digraph, the quantity $\max_{i \in cc(\overrightarrow{G}(X_T, E_T))} l_i$ is a lower bound on the size of the largest connected components of any final digraph.

This lower bound is sharp since it can effectively be reached by setting all U-arcs of $\overrightarrow{G}(X_{TU}, E_{TU})$ to F-arcs.

Theorem 16. *If* $\mathrm{dom}(\mathbf{MAX\_NCC}) = \{\underline{\mathbf{MAX\_NCC}}\}$*:*

1. *Any U-vertex linked to a connected component in* $cc_{[|X_T|=\mathbf{MAX\_NCC}]}$ $(\overrightarrow{G}(X_T, E_T))$ *is turned into an F-vertex.*
2. *Any U-arc between two distinct U-vertices, such that one of its extremities is linked to a connected component of $\overrightarrow{G}(X_T, E_T)$ of size s such that $s+2 > \mathbf{MAX\_NCC}$, is turned into an F-arc.*
3. *Any U-vertex linked to two connected components of $\overrightarrow{G}(X_T, E_T)$ of respective size s_1 and s_2 such that $s_1 + s_2 \geq \mathbf{MAX\_NCC}$ is turned into an F-vertex.*
4. *Any U-arc between two distinct U-vertices linked to two connected components of $\overrightarrow{G}(X_T, E_T)$ of respective size s_1 and s_2 such that $s_1 + s_2 + 2 > \mathbf{MAX\_NCC}$ is turned into an F-arc.*
5. *If* $\mathbf{MAX\_NCC} = 1$ *then any U-arc linking two distinct U-vertices is turned into an F-arc.*
6. *If* $\mathbf{MAX\_NCC} = 0$ *then any U-vertex is turned into an F-vertex.*

Filtering from $\overline{\mathbf{MAX\_NCC}}$

Theorem 17. $\overline{\mathbf{MAX\_NCC}} = \max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i$ *and this bound is sharp.*

Proof. Since two vertices belonging to two distinct connected components of the intermediate digraph $\overrightarrow{G}(X_{TU}, E_{TU})$ cannot belong to the same connected component of any final digraph, the quantity $\max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i$ is an upper bound on the size of the largest connected components of any final digraph.

This upper bound is sharp since it can effectively be reached by selecting a connected component $i_{\max}$ of $\overrightarrow{G}(X_{TU}, E_{TU})$ containing $\max_{i \in cc(\overrightarrow{G}(X_{TU}, E_{TU}))} l_i$ vertices and by first setting all U-arcs between two vertices of $i_{\max}$ to T-arcs and then by setting all remaining U-arcs of $\overrightarrow{G}(X_{TU}, E_{TU})$ to F-arcs.

Theorem 18. *If* $\mathrm{dom}(\mathbf{MAX\_NCC}) = \{\overline{\mathbf{MAX\_NCC}}\}$ *then if* $|cc_{[|X_{TU}|=\mathbf{MAX\_NCC}]}(\overrightarrow{G}(X_{TU}, E_{TU}))| = 1$*, any U-vertex of this single connected component is turned into a T-vertex.*

6.3 Performance

In order to evaluate the effectiveness of graph-based filtering, we performed two experiments, generating random instances of the `group` constraint. `VARIABLES` was chosen as a sequence of n domain variables ranging over $[0, 1]$, and `VALUES` as the singleton set $\{1\}$. A constraint instance was generated by setting the initial domain of each domain variable to a randomly chosen interval. Furthermore, with 10% probability, each variable in `VARIABLES` was randomly fixed.

The experiments compare the effect of graph-based filtering with the approach of constructing an automaton for each graph characteristic and by reformulating that automaton as a conjunction of constraints as described in [1]. We call this approach automata-based filtering. For both methods, graph invariants, providing auxiliary constraints [3], were also posted.

In the first experiment, we computed the number of labeling choices made during search for all solutions for $n = 10$. In the second experiment, we computed the number of labeling choices made during search for the first solution for $n = 20$. We chose to count labeling choices as opposed to measuring runtime, as a fair runtime comparison would require a more polished implementation of graph-based filtering than we currently have. Note however that all filtering rules run in $O(n)$ time.

The results are presented in two scatter plots in Figure 10. Each point represents a random instance, its X (resp. Y) coordinate corresponding to automata-based (resp. graph-based) filtering. Feasible and infeasible instances are distinguished in the plots.

From these experiments, we observe that most of the time, but not always, the graph-based method dominates the automata-based one. One would expect domination, as the graph method reasons about arc variables in addition to vertex variables. The graph method is currently limited by our approach to only apply the filtering when a graph parameter reaches one of its bounds. We observe no significant difference between the patterns for feasible vs. infeasible instances.

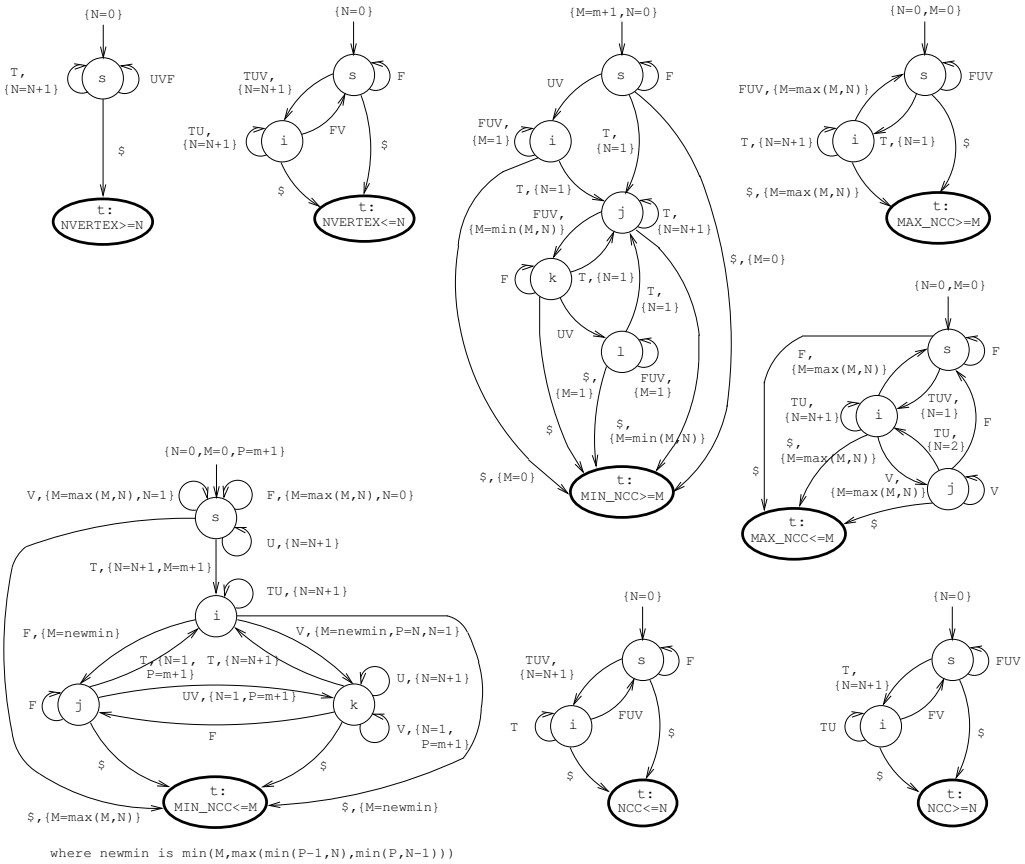

Fig. 9. Automata for computing the lower and upper bounds of **NVERTEX**, **NCC**, **MIN_NCC** and **MAX_NCC** given by Table 4. V, U, F, T and $ respectively correspond to a U-vertex entered via an F-arc, a U-vertex not entered via an F-arc, an F-vertex, a T-vertex and the end of string.

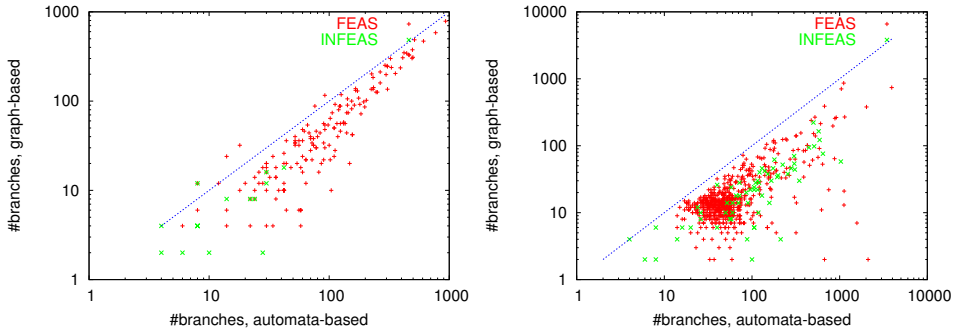

Fig. 10. Scatter plots of random instances. Left: comparing labeling choices for finding all solutions. Right: comparing labeling choices for finding the first solution.

7 Conclusion

This report provided a first generic filtering scheme stemming from lower and upper bounds for common graph parameters used in the graph-based reformulation of global constraints. Moreover, it shows how we could retrieve existing specialized filtering algorithms solely from the graph-based description. Our experiments on the example of the `path_with_loops` graph class point to an enhancement of the approach: filtering before a graph parameter reaches one of its bounds.

References

1. N. Beldiceanu, M. Carlsson, and T. Petit. Deriving Filtering Algorithms from Constraint Checkers. In M. Wallace, editor, *Principles and Practice of Constraint Programming (CP'2004)*, volume 3258 of *LNCS*, pages 107–122. Springer-Verlag, 2004.
2. N. Beldiceanu, M. Carlsson, and J.-X. Rampon. Global constraint catalog. Technical Report T2005-06, Swedish Institute of Computer Science, 2005.
3. N. Beldiceanu, M. Carlsson, J.-X. Rampon, and C. Truchet. Graph Invariants as Necessary Conditions for Global Constraints. In P. van Beek, editor, *Principles and Practice of Constraint Programming (CP'2005)*, volume 3709 of *LNCS*, pages 92–106. Springer-Verlag, 2005. Preprint available as SICS Tech Report T2005-07.
4. N. Beldiceanu and E. Contejean. Introducing Global Constraints in CHIP. *Mathl. Comput. Modelling*, 20(12):97–123, 1994.
5. N. Beldiceanu, I. Katriel, and X. Lorca. Undirected Forest Constraints. In *CP-AI-OR'06*, volume 3990 of *LNCS*, pages 29–43, 2006.
6. N. Beldiceanu, T. Petit, and G. Rochart. Bounds of Graph Characteristics. In P. van Beek, editor, *CP'2005*, volume 3709 of *LNCS*, pages 742–746, 2005.
7. C. Berge. *Graphes*. Dunod, 1970. In French.
8. C. Bessière, E. Hebrard, B. Hnich, Z. Kızıltan, and T. Walsh. *Among*, *common* and *disjoint* Constraints. In *CSCLP 2005*, pages 223–235, 2005.
9. COSYTEC. *CHIP Reference Manual*, release 5.1 edition, 1997.
10. G. Dooms, Y. Deville, and P. Dupont. CP(Graph): Introducing a Graph Computation Domain in Constraint. In P. van Beek, editor, *Principles and Practice of Constraint Programming (CP'2005)*, volume 3709 of *LNCS*, pages 211–225. Springer-Verlag, 2005.
11. M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W.H.Freeman and co., San Francisco, 1979.
12. P. Martin and D. B. Shmoys. A New Approach to Computing Optimal Schedules for the Job-Shop Scheduling Problem. In *IPCO'96*, volume 1084 of *LNCS*, pages 389–403, 1996.
13. S. Micali and V. V. Vazirani. An $\mathcal{O}(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In *FOCS 1980*, pages 17–27, New York, 1980. IEEE.
14. G. Pesant. A Filtering Algorithm for the *stretch* Constraint. In *CP'2001*, volume 2239 of *LNCS*, pages 183–195, 2001.
15. J.-C. Régin. The Symmetric *alldiff* Constraint. In *16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, pages 420–425, 1999.