Bounds of Graph Characteristics

Nicolas Beldiceanu[1], Thierry Petit[1], and Guillaume Rochart[2]

[1] École des Mines de Nantes, LINA FRE CNRS 2729, FR-44307 Nantes, France
{Nicolas.Beldiceanu, Thierry.Petit}@emn.fr
[2] Bouygues e-lab, 78061 St. Quentin en Yvelines, France
grochart@bouygues.com

Abstract. This article presents a basic scheme for deriving systematically a filtering algorithm from the graph properties based representation of global constraints. This scheme is based on the bounds of the graph characteristics used in the description of a global constraint. The article provides bounds for the most common used graph characteristics.

1 Introduction

Beldiceanu presented in [1] a systematic description of these global constraints in terms of graph properties: among the 224 constraints of the catalog of global constraints, about 200 constraints are described as a conjunction of graph properties where each graph property has a the form $P\ op\ V$, where P is a graph characteristics, op is a comparison operator in $\{\leq, \geq, =, \neq\}$, and V a domain variable[1].

Example 1. Consider the `nvalue`$(N, \{x_1, ..., x_m\})$ constraint [3], where $N, x_1, ..., x_m$ are domain variables. The `nvalue` constraint holds iff the number of distinct values assigned to the variables in $\mathcal{X} = \{x_1, ..., x_m\}$ is equal to N. It can been seen as enforcing the following graph property: the number of strongly connected components of the *intersection graph* $G(\mathcal{X}, E)$, where $E = \{x_i \in \mathcal{X}, x_j \in \mathcal{X} : x_i = x_j\}$, is equal to N.

In this context, Dávid Hanák made a preliminary exploitation of this description for designing filtering algorithms, for a particular graph property [4]. In this article we present a systematic approach which aims at providing generic filtering algorithms for the most used graph properties [1]: given a specification of a global constraint C in terms of graph properties, we can derive a filtering algorithm for C.

A global constraint C is represented as an initial digraph $G_i = (\mathcal{X}_i, E_i)$: to each vertex in $\mathcal{X}_i$ corresponds a variable involved in C, while to each arc e in E_i corresponds a binary constraint involving the variables at both extremities of e. To generate G_i from the parameters of C, the set of arcs generators described in [1] is used. When all variables of C are fixed, we remove from G_i all binary constraints which do not hold as well as isolated vertices, i.e., vertices which are not extremity of an arc. This final digraph is denoted by G_f. C is defined by a conjunction of graph properties which

[1] A *domain variable* is a variable that ranges over a finite set of integers; $dom(V)$, $min(V)$ and $max(V)$ respectively denote the set of possible values of variable V, the minimum value of V and the maximum value of V.

P. van Beek (Ed.): CP 2005, LNCS 3709, pp. 742–746, 2005.

should be satisfied by G_f. Each graph property has the form $P\ op\ V$; P is a graph characteristics, V is an domain variable and op is one of the comparison operator $\geq, \leq, =, \neq$. Within the global constraint catalog [1], common used graph characteristics are:

– **NARC** and **NVERTEX** denote the number of arcs and vertices: they are respectively used by 95 and 17 global constraints,
– **NCC** and **NSCC** denote the number of connected and strongly connected components; they are used in the description of 19 and 13 global constraints,
– **NSINK** (respectively **NSOURCE**) denotes the number of vertices which don't have any successor (resp. predecessor); they are respectively used by 16 and 15 global constraints; since **NSINK** and **NSOURCE** are similar, the rest of this article considers **NSINK**.

Example 2. Consider the `nvalue(`N`,`$\mathcal{X}$`)` constraint. Parts (A) and (B) of Fig. 1 respectively show the initial digraph G_i generated for the `nvalue` constraint with $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$ and the digraph G_f associated with the ground solution `nvalue`$(3, \{5, 8, 1, 5\})$. Each vertex of G_i depicts its corresponding variable. All arcs corresponding to equality constraints that are not satisfied are removed to obtain G_f from G_i. Each vertex of G_f depicts the value assigned to its corresponding variable. The `nvalue` constraint is defined by the graph property $\mathbf{NSCC} = N$. The `nvalue`$(3, \{5, 8, 1, 5\})$ constraint holds since G_f contains three strongly connected components, which can be interpreted as the fact that N is equal to the number of distinct values taken by the variables x_1, x_2, x_3 and x_4. Part (C) of Fig. 1 will be referenced in Example 3.

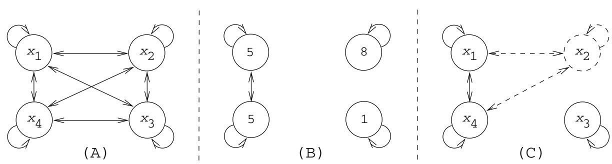

Fig. 1. (A) Initial digraph G_i associated with the `nvalue`$(N, \{x_1, x_2, x_3, x_4\})$ constraint. (B) Final digraph G_f of the ground solution `nvalue`$(3, \{5, 8, 1, 5\})$. (C) Intermediate digraph.

2 Filtering from Graph Properties

Given a graph property $P\ op\ V$ occurring in the description of a global constraint, this section first shows how to reduce the domain of V in order to enforce $P\ op\ V$. Finally, it discuss the case where several graph properties are used to define a global constraint. We first introduce the notion of *intermediate digraph* derived from the initial digraph G_i, where vertices and arcs can have different status as detailed below. The purpose of this *intermediate digraph* is to reflect the knowledge we currently have about the vertices and the arcs of G_i that may or may not belong to the final digraph G_f. This knowledge comes from two sources:

– Because of the current domain of its variables, a binary constraint associated to an arc of G_i does not hold (or is entailed),
– Because of an external reason, a given arc or vertex of G_i is forced to belong to G_f (or is forced to no belong to G_f).

When a global constraint C is posted the *intermediate digraph* corresponds to G_i, while when all variables of C are fixed the *intermediate digraph* is equal to G_f.

Notation 1. *Let $G_i = (X_i, E_i)$ be the initial digraph of a global constraint C, and $G_f = (X_f, E_f)$ its final digraph. At a given step corresponding to a partial assignment of values to the variables of C, we classify a vertex $v_j \in X_i$ and an arc $e_k \in E_i$:*

- *v_j is a T-vertex (true) iff $v_j \in X_f$; v_j is a F-vertex (false) iff $v_j \notin X_f$; otherwise v_j is a U-vertex (undetermined). X_T, X_F and X_U respectively denote the sets of T-vertices, of F-vertices and of U-vertices.*
- *e_k is a T-arc (true) iff $e_k \in E_f$; e_k is a F-arc (false) iff $e_k \notin E_f$; otherwise e_k is a U-arc (undetermined). E_T, E_F and E_U respectively denote the sets of T-arcs, of F-arcs and of U-arcs.*

The definition of the *intermediate digraph* takes into account the fact that the final graph will not contain any isolated vertex.

Definition 1. *The* intermediate digraph *is the digraph defined from G_i, X_T, X_F, X_U, E_T, E_F, E_U by applying the next rules while they induce some modifications:*

- *Remove all F-arcs,*
- *Any F-vertex which is not the extremity of at least one T-arc is removed; when a vertex is removed, we remove also all its ingoing and outgoing arcs which are turned to F-arcs,*
- *Any U-vertex which is not the extremity of at least one arc is removed,*
- *Any U-vertex which is an extremity of a T-arc is turned to a T-vertex,*
- *If a T-vertex is the extremity of exactly one U-arc e and not the extremity of any T-arc, then e is turned to a T-arc.*

When a vertex or an arc is removed, or when the status of a vertex or of an arc is changed by one of the previous rule the sets X_T, X_F, X_U, E_T, E_F, E_U are updated to reflect this change.

Example 3. Consider again the `nvalue(`N`,`$\mathcal{X}$`)` constraint presented in the introduction, and assume that not all variables of $\mathcal{X} = \{x_1, x_2, x_3, x_4\}$ are fixed: dom(x_1)=$\{5\}$, dom(x_2)=$\{5\}$, dom(x_3)=$\{5, 8\}$, dom(x_4)=$\{1\}$. Furthermore assume that, for an equality constraint *ec* associated to an arc of the initial digraph G_i of `nvalue`, entailment is only detected when all variables occurring in *ec* are fixed. This leads to partition the edges of G_i in the following three sets $E_T = \{(x_1, x_1),(x_1, x_4),(x_3, x_3),(x_4, x_1),(x_4, x_4)\}$, $E_U = \{(x_1, x_2),(x_2, x_1),(x_2, x_2),(x_2, x_4),(x_4, x_2)\}$ and $E_F = \{(x_1, x_2),(x_2, x_3),(x_3, x_1),(x_3, x_2),(x_3, x_4),(x_4, x_3)\}$. The status of the vertices is initially set *undetermined* (i.e. $X_U = \{x_1, x_2, x_3, x_4\}$) and we apply the rules of Definition 1 in order to obtain the *intermediate digraph* depicted by part (C) of Fig. 1. A plain line depicts a T-vertex or a T-arc, while a dashed line indicates a U-vertex or an U-arc. The same style will be used in all other figures of this article in order to depict T-vertices, T-arcs, U-vertices and U-arcs.

Property 1. From the definition of G_f, the global constraint C has no solution if the *intermediate digraph* contains a F-vertex or if it contains a T-vertex which is not the extremity of any arc.

Given a graph property $P\ op\ V$ associated to a global constraint C, the *intermediate digraph* will be used for evaluating a lower bound $\underline{P}$ and an upper bound $\overline{P}$ of the graph characteristics P. Section 4 provides the algorithms for computing $\underline{P}$ and $\overline{P}$ for different graph characteristics. It assumes that all U-vertices or arcs of the *intermediate digraph* can be freely turned into T-vertices or T-arcs (resp. F-vertices or F-arcs). According to the comparison operator op, the next table gives the different possible cases for reducing the domain of variable V according to $\underline{P}$ and $\overline{P}$.

$P \leq V$	$min(V) \geq max(\underline{P}, min(V))$
$P \geq V$	$max(V) \leq min(\overline{P}, max(V))$
$P = V$	$min(V) \geq max(\underline{P}, min(V)) \wedge max(V) \leq min(\overline{P}, max(V))$
$P \neq V$	$\underline{P} = \overline{P} \Rightarrow \underline{P} \notin dom(V)$

3 Bounds of Graph Characteristics

This section is devoted to the evaluation of lower and upper bounds of the graph characteristics introduced in Section 2. For this purpose, we will deal with graphs derived from the *intermediate digraph* with different sets of arcs and vertices which are described in the following notations.

Notation 2. *Let Q, R and S be non-empty words over the alphabet $\{T, U\}$.*

- *Given a word W, $w \in W$ denotes a letter of W.*
- *X_Q and E_Q respectively denote $\bigcup_{q\in Q} X_q$ and $\bigcup_{q\in Q} E_q$.*
- *$X_{Q,R}$ (resp. $X_{Q,\neg R}$) denotes $v \in X_Q$ such that there is at least one arc (resp. there is no arc) in E_R where v is an extremity.*
- *$E_{Q,R}$ denotes the set of arcs $(v_1, v_2) \in E_Q$ such that $v_1 \in X_R$ or $v_2 \in X_R$.*
- *$X_{Q,R,S}$ (resp. $X_{Q,R,\neg S}$) denotes $v \in X_{Q,R}$ such that, within the vertices which share an arc with v, there is at least one vertex in X_S (resp. no vertex is in X_S).*
- *$X_{Q,\neg R,\neg S}$ denotes $v \in X_{Q,\neg R}$ such that, within the vertices which share an arc with v, no vertex is in X_S.*

Based on the previous notations, we define four kind of graphs, where $\mathcal{X}$ is a set of vertices and $\mathcal{E}$ a set of arcs:

- $\overrightarrow{G}(\mathcal{X}, \mathcal{E})$ denotes the digraph defined by the vertex set $\mathcal{X}$ and the subset of arcs of $\mathcal{E}$ having their two extremities in $\mathcal{X}$.
- $\overrightarrow{G}(\mathcal{E})$ denotes the digraph defined by the set of arcs $\mathcal{E}$ and the set of vertices which are extremities of arcs in $\mathcal{E}$.
- $\overleftrightarrow{G}(\mathcal{X}, \mathcal{E})$ (resp. $\overleftrightarrow{G}(\mathcal{E})$) denotes the undirected graph derived from $\overrightarrow{G}(\mathcal{X}, \mathcal{E})$ (resp. $\overrightarrow{G}(\mathcal{E})$) by forgetting the orientation of the arcs and by keeping its eventual loops.

Example 4. We illustrate some sets of vertices and arcs previously introduced and some graphs on the *intermediate digraph* depicted by part (C) of Fig. 1:

- $X_{U,T} = \{x_1, x_4\}, \quad X_{U,\neg T} = \{x_3\}$,
- $E_{UT} = \{(x_1, x_1), (x_1, x_2), (x_1, x_4), (x_2, x_1), (x_2, x_2), (x_2, x_4), (x_3, x_3), (x_4, x_1), (x_4, x_2), (x_4, x_4)\}$,
- $E_{U,T} = \{(x_1, x_2), (x_2, x_1), (x_2, x_2), (x_2, x_4), (x_4, x_2)\}$,

- $\overrightarrow{G}(X_T, E_T) = \overrightarrow{G}(\{x_1, x_3, x_4\}, \{(x_1, x_1), (x_1, x_4), (x_3, x_3), (x_4, x_1), (x_4, x_1)\})$
- $\overrightarrow{G}(E_U) = \overrightarrow{G}(\{x_1, x_2, x_4\}, \{(x_1, x_2), (x_2, x_1), (x_2, x_4), (x_4, x_2)\})$.

Computing lower and upper bounds of graph characteristics can be seen as computing some graph characteristics on the graphs previously introduced. Some bounds are expressed in terms of graph characteristics that correspond to non-polynomial problems. However in such a case we provide bounds that are sharp. Note that many of the digraphs, which express a global constraint, belong to specific graph classes for which a non-polynomial problem becomes polynomial. Even when the computation is polynomial, we can get better worst-case complexity by exploiting the structure of the *intermediate digraph*.

Bound of graph characteristics	Sharpness	Polynomial
$\underline{\mathbf{NARC}} \geq \lvert E_T \rvert + \lvert X_{T,\neg T} \rvert - \mu(\overleftrightarrow{G}(X_{T,\neg T}, E_U))$	yes	yes
$\overline{\mathbf{NARC}} \leq \lvert E_{TU} \rvert$	yes	yes
$\underline{\mathbf{NVERTEX}} \geq \lvert X_T \rvert + h(\overleftrightarrow{G}((X_{T,\neg T,\neg T}, X_{U,\neg T,T}), E_{U,T}))$	yes	**no**
$\overline{\mathbf{NVERTEX}} \leq \lvert X_{TU} \rvert$	yes	yes
$\underline{\mathbf{NCC}} \geq ncc_T(\overrightarrow{G}(X_{TU}, E_{TU}))$	yes	yes
$\overline{\mathbf{NCC}} \leq ncc_{T_{ni}} + \mu_l(\overleftrightarrow{G}_{rem})$	yes	yes
$\underline{\mathbf{NSCC}} \geq nscc_T(\overrightarrow{G}(X_{TU}, E_{TU}))$	yes	yes
$\overline{\mathbf{NSCC}} \leq nscc(\overrightarrow{G}(X_{TU}, E_T))$	yes	yes
$\underline{\mathbf{NSINK}} \geq nsink_T(\overrightarrow{G}(X_{TU}, E_{TU}))$	**no**	yes
$\overline{\mathbf{NSINK}} \leq nsink(\overrightarrow{G}(X_T, E_T)) + \lvert X_U \rvert$	**no**	yes

The table provides a lower and an upper bound for the different graph characteristics. Proofs are available in [2]. $\mu(G)$ is the cardinality of a maximum matching of G. $\mu_l(G)$ is the maximum size of a set of edges of G, such that no two edges have a vertex in common, where G eventually contains loops. Given a bipartite graph $G((X, Y), E)$, a *hitting set* is a set of vertices in Y required to cover all vertices of X. $h(G)$ denotes the *cardinality of a minimum hitting set* of G. $nscc(G)$ and $nsink(G)$ respectively denote the number of strongly connected components and the number of sinks of G. $ncc_T(G)$ and $nscc_T(G)$ respectively denote the number of connected components, strongly connected components with at least one T-vertex. $nsink_T(G)$ denotes the number of sinks of G which are T-vertices. $ncc_{T_{ni}}(G)$ denotes the number of connected components formed only by T-arcs and T-vertices which are not isolated vertices.

References

1. N. Beldiceanu, M. Carlsson, and J.-X. Rampon. Global constraint catalog. Technical Report T2005-06, Swedish Institute of Computer Science, 2005.
2. N. Beldiceanu, T. Petit, and G. Rochart. Bounds of graph characteristics. Technical Report 05/2/INFO, École des Mines de Nantes, 2005.
3. C. Bessière, E. Hebrard, B. Hnich, Z. Kızıltan, and T. Walsh. Filtering algorithms for the *nvalue* constraint. In *CP-AI-OR 2005*, volume 3524 of *LNCS*, pages 79–93. Springer-Verlag, 2005.
4. D. Hanák. Implementing global constraints as structured graphs of elementary constraints. *Scientific Journal Acta Cybernetica*, 2003.