Enforcing Domain Consistency on the Extended Global Cardinality Constraint is NP-hard

Claude-Guy Quimper
School of Computer Science
University of Waterloo
Waterloo, Canada

1 Introduction

We consider a set of variables \( X = \{x_1, \ldots, x_n\} \) and a set of values \( D \). Each variable \( x_i \) is associated to a domain \( \text{dom}(x_i) \subseteq D \) and each value \( v \in D \) is associated to a cardinality set \( K(v) \). An assignment satisfies the extended global cardinality constraint (extended-GCC) if each variable \( x_i \) is instantiated to a value in its domain \( \text{dom}(x_i) \) and if each value \( v \in D \) is assigned to \( k \) variables for some \( k \in K(v) \). Extended-GCC differs from normal GCC by its sets of cardinality \( K(v) \) that can be any set of values. In normal GCC, as introduced by Régis [2], these cardinality sets are restricted to intervals.

Enforcing domain consistency consists in verifying for each value \( v \) in a variable domain \( \text{dom}(x_i) \) if there is an assignment satisfying the extended-GCC such that \( x_i = v \). This is equivalent to determining if the extended-GCC is satisfiable when the domain of the variable is bounded to a single value, i.e. \( \text{dom}(x_i) = \{v\} \). We show that determining if the extended-GCC is satisfiable is NP-complete by reduction to the SAT problem and therefore enforcing domain consistency on the extended-GCC is NP-hard.

2 Extended-GCC as a Matching in a Graph

As demonstrated by Régis [1], an extended-GCC instance can be represented by a bipartite graph \( G = (L \cup R, E) \). Let the left-nodes of the bipartite graph be \( L = X \) the variables of the problem. Let the right-nodes of the bipartite graph be \( R = D \) the values of the problem. There is an edge \((x_i, v) \in E \) if and only if \( v \in \text{dom}(x_i) \).

A generalized matching [4] \( M \) is a subset of \( E \) such that all variables \( x_i \in L \) is adjacent to one edge in \( M \) and each node \( v \in R \) is adjacent to \( k \) edges in \( M \) for some \( k \in K(v) \).

A generalized matching \( M \) represents a solution of the extended-GCC. There is obviously a matching \( M \) if and only if the extended-GCC is satisfiable. In the next section, we show that determining if a generalized matching exists is NP-complete.
3 Reduction to the SAT problem

Consider a 3-SAT problem defined by a list of variables \( X = \{ X_1, \ldots, X_n \} \), a list of literals \( \mathcal{L} = \{ x_i, \neg x_i \mid x_i \in X \} \) and a list of clauses \( C = \{ C_1, \ldots, C_m \} \) where \( C_i \subseteq \mathcal{L} \) are the set of literals of the clause. We want to assign the value true or false to the literals in \( \mathcal{L} \) such that all clauses have at least one literal assigned to true.

From a SAT problem, we construct the bipartite graph \( G = (L \cup R, E) \) as follows. For each literal \( l_j \) in a clause \( C_i \), we create one left-node \( S(C_i, l_j) \in L \) and one right-node \( d(C_i, l_j) \in R \). For each clause \( C_i \) we create a left-node \( C_i \in L \) and for each variable \( X_i \) we create another left-node \( X_i \in L \). Finally, we add to the graph a right-node \( l_i \in R \) for each literal \( l_i \).

We connect the left-nodes in \( L \) to the right-nodes in \( R \) as follows. We start with an empty set of edges \( E = \emptyset \). For each clause \( C_i \) and each literal \( l_j \in C_i \), we add the edges \( (C_i, d(C_i, l_j)) \), \( (S(C_i, l_j), d(C_i, l_j)) \) and \( (S(C_i, l_j), l_j) \). For each variable \( x_i \in X \) we add the edges \( (X_i, x_i) \) and \( (X_i, \neg x_i) \). Finally, we set the cardinality of each right-node in \( L \) as follows: \( K(d(C_i, l_j)) = \{ 0, 1 \} \) and \( K(l_i) = \{ 0, k_i + 1 \} \) where \( k_i \) is equal to the number of clauses containing the literal \( l_i \) or more formally \( k_i = |\{ C_j \in C \mid l_i \in C_j \}| \). Figure 1 shows the part of graph \( G \) that is related to variable \( X_i \).

The intuition of the reduction is simple. A generalized matching in \( G \) corresponds to a solution to the SAT problem. If \( (X_i, x_i) \in M \) then \( x_i = \text{true} \) and if \( (X_i, \neg x_i) \in M \) then \( x_i = \text{false} \). All clause nodes \( C_i \) must be matched to another node. They can only be matched with an edge \( (C_i, d(C_i, l_j)) \) if \( l_j = \text{true} \).

**Lemma 1.** Let \( l_i \in \{ x_i, \neg x_i \} \), the edge \( (X_i, l_i) \) belongs to \( M \) if and only if \( S(C_j, l_i) \in M \) for all \( C_j \).

**Proof.** The nodes \( S(C_j, l_i) \in E \) and the node \( X_i \) are the only nodes connected to node \( l_i \). Since we have \( K(l_i) = \{ 0, k_i + 1 \} \) and \( k_i + 1 \) is equal to the number of nodes connected to \( l_i \), either all edges adjacent to \( l_i \) belong to \( M \) or no edges adjacent to \( l_i \) belong to \( M \). Therefore for all nodes \( S(C_j, l_i) \) we have \( (X_i, l_i) \in M \iff S(C_j, l_i) \in M \). \( \square \)

**Lemma 2.** Let \( l_j \in \{ x_j, \neg x_j \} \). If the edge \( (C_i, d(C_i, l_j)) \) belongs to a generalized matching then \( (X_j, l_j) \) also belongs to this generalized matching.

**Proof.** Suppose the edge \( (C_i, d(C_i, l_j)) \) belongs to the generalized matching \( M \). Since the cardinality of node \( d(C_i, l_j) \) is \( \{ 0, 1 \} \) and edge \( (C_i, d(C_i, l_j)) \) is adjacent to this node, no more edges in \( M \) can be adjacent to node \( (C_i, d(C_i, l_j)) \). Therefore the edge \( S(C_i, l_j) \) has no other choice to be matched with node \( l_j \). By Lemma 1 we obtain that \( (X_j, l_j) \) belongs to \( M \). \( \square \)
Clauses that contain $X_i$

Belongs to the matching if $X_i$ is true

Belongs to the matching if $X_i$ is false

Clauses that contain $\neg X_i$

Edges connected to nodes $d(C4, x_i)$ for $i > 1$

This edge may or may not belong to a matching if $x_i$ is true. It cannot be part of the matching if $x_i$ is false.

There must be 0 or 1 edge adjacent to this node in the matching.

Fig. 1. Part of graph $G$ related to variable $X_i$. 
Lemma 3. SAT is satisfiable if and only if there exists a generalized matching $M$ in graph $G$.

Proof. (⇒) Suppose SAT is satisfiable, we construct a matching by pointing each node $C_i$ to a node $d(C_i, l_i)$ such that literal $l_i$ is true in the SAT solution. Other left-nodes in $L$ are matched according to Lemma 2 and Lemma 1.

(⇐) Consider a generalized matching $M$. For all variables $X_i \in X$, we have either the edge $(X_i, x_i)$ or $(X_i, \neg x_i)$ in $M$. We say that literal $l_i$ is true if the edge $(X_i, l_i)$ belongs to $M$ and false if the edge does not belong to $M$. For all clause $C_i$, we have an edge $(C_i, d(C_i, l_j))$ in $M$ for some $l_j \in \{x_j, \neg x_j\}$. This implies by Lemma 2 that $l_j$ is true and therefore clause $C_i$ is satisfied. Therefore all clauses are satisfied by the variable assignments given by the edges $(X_i, l_i)$.

4 Conclusion

Lemma 3 shows that determining the satisfiability of extended-GCC is NP-complete and therefore enforcing domain consistency on the extended-GCC is NP-hard.

Acknowledgments

The author would like to thank Alejandro López-Ortiz for proof-reading this technical report. Thanks also to Alexander Golynski who pointed out that the graph presented above could be simplified by merging the $C_i$ nodes and the $d(C_i, l_j)$ nodes together and fixing their cardinality to the interval $[0, k_i - 1]$.

References