Domain Reduction for the Circuit Constraint

Student: Latife Genc Kaya
Supervisor: Prof. John Hooker

Carnegie Mellon University Pittsburgh PA 15213 USA

Abstract. We present an incomplete filtering algorithm for the circuit constraint. The filter removes redundant values by eliminating non-Hamiltonian edges from the associated graph (i.e., edges that are part of no Hamiltonian cycle). We prove a necessary condition for an edge to be Hamiltonian, which provides the basis for eliminating edges of a smaller graph defined on a separator of the original graph.

1 Introduction

The circuit constraint, $circuit(y_1, \ldots, y_n)$, where $y_j \in \{1, \ldots, n\}$, is true if and only if for each $j \in \{1, \ldots, n\}$, y_j is the successor of j in some permutation of $1 \ldots n$ and $y_j \in D_j$, where D_j is the domain of variable j. In this paper we consider designing a filtering algorithm for the circuit constraint.

On a graph of vertices $1, \ldots, n$, the circuit constraint can be thought as defining a directed Hamiltonian cycle. Nodes of the graph represent the variables. A directed edge (i, j) exists if and only if j is in the domain of variable i. Moreover, elimination of an edge (i, j) from the graph means elimination of the value j from the domain of variable i. With this representation, the problem of domain reduction for the circuit constraint reduces to identifying and eliminating non-Hamiltonian edges on a digraph (i.e., edges that belong to no Hamiltonian cycle).

Shufelt and Berliner (see [3]) describe a set of patterns to identify Hamiltonian and non-Hamiltonian edges on an undirected graph. These patterns can be adapted for digraphs and used to eliminate non-Hamiltonian edges when some part of the Hamiltonian cycle has been constructed. It is not intended for the elimination of non-Hamiltonian edges for the general case in which arbitrary variable domains are given. Also the analysis of [3] relies on the special structure of the problem for which it was developed, namely the construction of a knight's tour on a chessboard.

In this paper, a necessary condition for an edge to be Hamiltonian on a (undirected or directed) graph is given. By this condition, we present a recursive algorithm that eliminates non-Hamiltonian edges from the graph via graph separators. A much smaller but denser multi-graph is constructed from a separator of the original graph. Then by applying a filtering algorithm for the global cardinality constraint together with in and out-vertex degree constraints, non-Hamiltonian edges are identified and eliminated from the graph.

2 Preliminaries

Let $G = (V, A)$ be a directed graph. A pair of vertices v_i and v_j are *neighbors* if (v_i, v_j) or $(v_j, v_i) \in A$. A *directed path* P between vertices $v_1, v_m \in V$ is a sequence of edges $(v_1, v_2), (v_2, v_3), \ldots, (v_{m-1}, v_m) \in A$. P is a *simple path* if $v_1, \ldots, v_{m-1}$ are distinct. The endpoints v_1, v_m are *connected* by P.

G is *connected* if any two vertices of G are connected by some directed path. For convenience we will say that an edge (u, w) connects two node sets V_1, V_2 if $v \in V_1$ and $w \in V_2$. An edge connects V_1 with graph (V_2, A_2) if it connects V_1 with V_2.

A nonempty vertex set $V' \subset V$ induces a *connected component* of G if V' induces a connected subgraph, and no edge of G connects V' with $V \setminus V'$. A set $S \subset V$ *separates* a connected graph G into connected components $C_1, \ldots, C_p$ if $V \setminus S$ induces a subgraph $\bar{G}_S$ with connected components $C_1, \ldots, C_p$. We say S is a *(vertex) separator* of G if it separates G into at least two connected components. [1]

A *directed cycle* of G is a directed path of which every vertex is an endpoint. A *Hamiltonian cycle* is a simple cycle whose vertices are precisely those in V. An edge (v, w) of G is *Hamiltonian* if (v, w) belongs to a Hamiltonian cycle.

3 Basic Idea

The *separator graph* G_S for a separator S of $G = (V, A)$ is defined as follows. The node set of G_S is S. G_S contains a directed edge (v, w) with *label* C if C is a connected component of $\bar{G}_S$ and (v, c_i) and (c_j, w) are edges of G for some pair of vertices c_i, c_j in C (possibly $c_i = c_j$). G_S contains an *unlabeled* directed edge (v, w) when $(v, w) \in A$. m_S is the number of edges (labeled or unlabeled) in the separator graph.

A Hamiltonian cycle of G_S is *permissible* if it contains at least one edge of each label. The following theorem may allow one to identify a graph (or edge) as non-Hamiltonian by looking for a certain kind of Hamiltonian cycle in a much smaller graph.

Theorem 1. *Suppose S separates graph G into p connected components. Then G is Hamiltonian only if the separator graph G_S has a permissible Hamiltonian cycle. Furthermore, an edge e connecting vertices of G_S is Hamiltonian only if G_S has a permissible Hamiltonian cycle that contains e.*

Proof. Consider an arbitrary Hamiltonian cycle H of G. We can construct a permissible Hamiltonian cycle H_S for G_S as follows. Let $v_1, \ldots, v_n, v_1$ be the sequence of vertices in H. Consider any pair of vertices v_j, v_k of H such that $v_j, v_k \in S$ and no vertices of S lie on the portion of H between v_j and v_k. If v_j, v_k are adjacent in H then (v_j, v_k) is an unlabeled edge of G_S. In this case, let

[1] If S separates G into at least three components, S is a *shredder*.

unlabeled edge (v_j, v_k) belong to H_S. If v_j, v_k are not adjacent then all vertices between v_j, v_k lie in the same component C and (v_j, v_k) is an edge of G_S with label C. So let (v_j, v_k) with label C belong to H_S. Note that H_S is a Hamiltonian cycle of G_S. Since at least two edges of H connect any given component to S, there is at least one edge of H_S with label C for every component C. Therefore, H_S is permissible.

Corollary 1. *If S separates G into more than $|S|$ components, then G is non-Hamiltonian.*

Proof. The separator graph G_S has $|S|$ vertices and therefore cannot have a Hamiltonian cycle with more than $|S|$ edges.

Corollary 2. *If S separates G into $|S|$ components, then no edge connecting vertices of S is Hamiltonian.*

Proof. An edge e that connects vertices in S is unlabeled in G_S. If e is Hamiltonian, some Hamiltonian cycle in G_S that contains e must have at least $|S|$ labelled edges. But since the cycle must have exactly $|S|$ edges, all the edges must be labelled and none can be identical to e.

4 The Algorithm

Given circuit constraint $circuit(y_1, \ldots, y_n)$ with variable domains $D_1, \ldots, D_n$, we present the following domain reduction algorithm:

1. Construct the corresponding graph $G = (V, A)$.
2. Find a vertex separator S in G
3. Construct the separator graph G_S
4. Find a set, A^N, of edges of G_S that do not satisfy the necessary condition of Theorem 1.
   (a) If G_S is not Hamiltonian, then STOP. G is not Hamiltonian.
   (b) Else, set $G = (V, A \setminus A^N)$ and go to 2.

4.1 Finding A Separator

One straightforward heuristic for identifying a vertex separator uses Breadth First Search (BFS). To construct BFS tree, initially all nodes of G are unlabeled. An arbitrary node v is chosen as root of the tree and it is labelled level 0. Neighbors of v are labelled level 1. At step k, all unlabeled nodes that are neighbors to some node in level $(k-1)$ are labelled level k and so on. When there is no remaining unlabeled node in the graph, the BFS tree is constructed. Assuming the highest level is at least 2, every intermediate level in the BFS tree is a vertex separator of the graph. We are currently investigating more effective heuristics for identifying small separators of a graph.

4.2 Filtering for the Separator Graph

In step 4 of the algorithm, to find a set of non-Hamiltonian edges on G_S, we will view the condition that G_S contains a permissible Hamiltonian cycle as a constraint and construct a filter for a relaxation of this constraint.

The constraint can be written *per-circuit*$(G_S, z_1, \ldots, z_{|S|})$ where z_i is the i^{th} edge of a permissible Hamiltonian cycle. We will filter *per-circuit* by filtering a relaxation of it, consisting a Global Cardinality Constraint and vertex degree constraints.

A *Global Cardinality Constraint* (gcc) has the form (X, V, l, u) where $X = \{x_1, \ldots, x_m\}$ is a set of variables which take their values in a subset of $V = \{v_1, \ldots, v_p\}$. It constrains the number of times a value $v_i \in V$ is assigned to a variable in X to be in an interval $[l_i, u_i]$.

In this case, let $X = \{(v_i, v_j) | v_i \textit{ and } v_j \textit{ are adjacent in } G_S\}$ and $V = \{C_1, \ldots, C_p, U, D\}$ where $C_1, \ldots, C_p$ are labels corresponding to the components of $\bar{G}_S$, U corresponds to "unlabeled" and D is a dummy value. Let l and u be $(l_{c_i}, u_{c_i}) = (1, |S| - p + 1)$ for each C_i, $(l_U, u_U) = (0, |S| - p)$ for U and $(l_D, u_D) = (m_S - |S|, m_S - |S|)$ for D.

A solution $(z_1, \ldots, z_{|S|})$ of per-circuit defines a permissible Hamiltonian cycle H_S. H_S corresponds to a solution of the above gcc constraint in which (v_i, v_j) receives the value:

C_k when (v_i, v_j) is an edge of H_S with label C_k

U when (v_i, v_j) is an unlabeled edge of H_S

D when (v_i, v_j) is not an edge of H_S

We can use Regin's filtering algorithm for gcc (see [2]) to find and eliminate edges that can be part of no permissible Hamiltonian cycle in G_S.

4.3 Global cardinality Constraint with Degree Constraints

We can further constrain the solution of gcc with out and in-degree constraints:

1. Given any (v_i, v_j) and (v_i, v_k), at least one must have label D.
2. Given any (v_j, v_i) and (v_k, v_i), at least one must have label D.

We can incorporate (1) or (2), but not both, in the filtering algorithm for gcc as follows. Suppose we incorporate (1). Define the value graph [2] and the value network as follows:

Definition 1. *The value graph is the bipartite graph* $GV = ((v_i, v_j)|(v_i, v_j) \in A_S, \{C_1, \ldots, C_p, U\}, E)$ *where* $((v_i, v_j), C_k) \in E$ *if and only if* (v_i, v_j) *is a labeled edge on* G_S*,* $((v_i, v_j), U) \in E$ *if and only if* (v_i, v_j) *is an unlabeled edge on* G_S*..*

We add a third set of vertices, $v_1, v_2, \ldots, v_{|S|}$ to the value graph. These vertices are connected to the variables part of the value graph by the set of edges, $\{(v_i, (v_i, v_j))|(v_i, v_j) \in A_S\}$ [2].

The value network, GN is obtained from the modified value graph by:

- orienting each edge of GV from values to variables, and orienting each additional edge from variables to new vertices. For both kind of arcs the lower bound is 0, and the capacity is 1
- adding a vertex s, and an arc from s to each value,i.e., labels. For such an arc, if the value is U, then the lower bound is 0, and the capacity is $|S| - p$. For all other values, as we require each label to appear at least once in the solution, the lower bound is 1, and the capacity is $|S| - p + 1$
- adding a vertex t, and an arc from each new vertex to t. For such an arc, the lower bound is 0, and the capacity is 1.
- adding an arc from t to s with the lower bound 0 and the capacity ∞

We will use Regin's filtering algorithm on this value network to filter gcc combined with out-vertex degree constraints. When (2) is incorporated in gcc, the resulting value network will be used to filter gcc combined with in-vertex degree constraints. We will use both of these filters and the filter for gcc itself to filter per-circuit constraint on the separator graph.

5 Conclusion

In this work, we have introduced a necessary condition for an edge to be Hamiltonian in a graph. Moreover, we have presented a partial filter for the circuit constraint.

Currently, besides investigating effective heuristics for identifying small separators that will allow us to eliminate non-Hamiltonian edges on a directed graph, we are working on testing the efficiency of the proposed filter.

References

1. Hooker, J.: Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction. *John Wiley Sons* (2000).
2. Regin, J.-C.: Generalized Arc Consistency for Global Cardinality Constraint. *AAAI*, (1996).
3. Shufelt J., Berliner, H.: Generating Hamiltonian Circuits Without Backtracking from Errors. *Theoretical Computer Science*, **132(1-2)** (1994) 347-375

[2] When (2) is incorporated, edges $(v_i, (v_j, v_i))$ are added to the value graph.