Propagating Conjunctions of ALLDIFFERENT Constraints

Christian Bessiere  
LIRMM, CNRS  
Montpellier, France  
bessiere@lirmm.fr

George Katsirelos  
CRIL-CNRS  
Lens, France  
gkatsi@gmail.com

Nina Naroditska  
NICTA and UNSW  
Sydney, Australia  
ninan@eese.unsw.edu.au

Claude-Guy Quimper  
Université Laval  
Québec, Canada  
equimper@gmail.com

Toby Walsh  
NICTA and UNSW  
Sydney, Australia  
toby.walsh@nicta.com.au

Abstract

We study propagation algorithms for the conjunction of two ALLDIFFERENT constraints. Solutions of an ALLDIFFERENT constraint can be seen as perfect matchings on the variable/value bipartite graph. Therefore, we investigate the problem of finding simultaneous bipartite matchings. We present an extension of the famous Hall theorem which characterizes when simultaneous bipartite matchings exists. Unfortunately, finding such matchings is NP-hard in general. However, we prove a surprising result that finding a simultaneous matching on a convex bipartite graph takes just polynomial time. Based on this theoretical result, we provide the first polynomial time bound consistency algorithm for the conjunction of two ALLDIFFERENT constraints. We identify a pathological problem on which this propagator is exponentially faster compared to existing propagators. Our experiments show that this new propagator can offer significant benefits over existing methods.

Introduction

Global constraints are a critical factor in the success of constraint programming. They capture patterns that often occur in practice (e.g. “these courses must occur at different times”). In addition, fast propagation algorithms are associated with each global constraint to reason about potential solutions (e.g. “these 4 courses have only 3 time slots between them so, by a pigeonhole argument, the problem is infeasible”). One of the oldest and most useful global constraints is the ALLDIFFERENT constraint (Laurière 1978). This specifies that a set of variables takes all different values. Many different algorithms have been proposed for propagating the ALLDIFFERENT constraint (Régis 1994; Leconte 1996; Puget 1998). Such propagators can have a significant impact on our ability to solve problems (Stergiou & Walsh 1999).

Problems often contain multiple ALLDIFFERENT constraints (e.g. “The CS courses must occur at different times, as must the IT courses. In addition, CS and IT have several courses in common”). Currently, constraint solvers ignore information about the overlap between multiple constraints (except for the limited communication provided by the domains of common variables). Here, we show the benefits of reasoning about such overlap. This is a challenging problem as finding a solution to just two ALLDIFFERENT constraints is NP-hard (Kutz et al. 2008) and existing approaches to deal with such overlaps require exponential space (Lardeux et al. 2008). Our approach is to focus on domains that are ordered, as often occurs in practice. For example in our time-tableing problem, values might represent times (which are naturally ordered). In such cases, domains can be compactly represented by intervals. Propagation algorithms can narrow such intervals using the notion of bound consistency. Our main result is to prove we can enforce bound consistency on two ALLDIFFERENT constraints in polynomial time. Our algorithm exploits a connection with matching on bipartite graphs. In particular, we consider simultaneous matchings. By generalizing Hall’s theorem, we identify a necessary and sufficient condition for the existence of such a matching and show that the this problem is polynomial for convex graphs.

Formal background

Constraint programming. We use capitals for variables and lower case for values. Values range over 1 to d. We write D(X) for the domain of values for X, lb(X) (ub(X)) for the smallest (greatest) value in D(X). A global constraint is one in which the number of variables n is a parameter. For instance, ALLDIFFERENT([X_1, ..., X_n]) ensures that X_i \neq X_j for any i < j. Constraint solvers prune search by enforcing properties like domain consistency. A constraint is domain consistent (DC) iff when a variable is assigned any value in its domain, there are compatible values in the domains of all other variables. Such an assignment is a support. A constraint is bound consistent (BC) iff when a variable is assigned the minimum or maximum value in its domain, there are compatible values between the minimum and maximum domain value for all other variables. Such an assignment is a bound support. A constraint is bound disentailed iff no possible assignment is a bound support.

Graph Theory. Solutions of ALLDIFFERENT correspond to matchings in a bipartite variable/value graph (Régis 1994).

Definition 1. The graph G = (V, E) is bipartite if V partitions into 2 classes, V = A \cup B and A \cap B = \emptyset, such that every edge has ends in different classes.

Definition 2. Let G = (A \cup B, E) be a bipartite graph. A matching that covers A is a set of pairwise non-adjacent edges M \subseteq E such that every vertex from A is incident to
If $M \subseteq G$ and $S \subseteq T$ such that $A = S \cup T$, $A \cap B = \emptyset$, and $S \cap T \neq \emptyset$.

**Definition 4.** Let $(A \cup B, E)$ and $S, T$ be an overlapping bipartite graph. A simultaneous matching is a set of edges $M \subseteq E$ such that $M \cap (S \times B)$ and $M \cap (T \times B)$ are matchings that cover $S$ and $T$, respectively.

In the following, we use the convention that a set of vertices $P$ is a subset of the partition $A$. We write $N(P)$ for the neighborhood of $P$, $P^S = P \cap (S \setminus T)$, $P^T = P \cap (T \setminus S)$ and $P^{ST} = P \cap S \cap T$. SIM-BM problems frequently occur in real-world applications like production scheduling and timetabling. We introduce here a simple exam timetabling problem that will serve as a running example.

**Running example.** We have 7 exams offered over 5 days and 2 students. The first student has to take the first 5 exams and the second student has to take the last 5 exams. Due to the availability of examiners, not every exam is offered each day. For example, the first exam cannot be on the last day of the week. Only one exam can be sat each day. This problem can be encoded as a SIM-BM problem. A represents the exams and contains 7 vertices $X_1$ to $X_7$. $B$ represents the days and contains the vertices $1$ to $5$. $S = \{X_1, X_2, X_3, X_4, X_5\}$ and $T = \{X_3, X_4, X_5, X_6, X_7\}$. We connect vertices between $A$ and $B$ to encode the availability restrictions of the examiners. The adjacency matrix of the graph is as follows:

|     | 1 | 2 | 3 | 4 | 5 |
|-----|---|---|---|---|---|
| $A^S = S \setminus T$ | $X_1$ | * | * | * | * |
|     | $X_2$ | * | | | |
| $A^{ST} = S \cap T$ | $X_3$ | * | * | * | * |
|     | $X_4$ | * | | * | * |
|     | $X_5$ | * | * | * | * |
| $A^T = T \setminus S$ | $X_6$ | * | * | * | * |
|     | $X_7$ | * | * | * | * |

Finding a solution for this SIM-BM problem is equivalent to solving the timetabling problem.

**Extension of Hall’s Theorem**

Hall’s theorem provides a necessary and sufficient condition for the existence of a perfect matching in a bipartite graph. 

**Theorem 1 (Hall Condition (Hall 1935)).** Let $G = (A \cup B, E)$ such that $A \cap B = \emptyset$. There exists a perfect matching if $|N(P)| \geq |P|$ for $P \subseteq A$.

Interestingly, we only need a small adjustment for simultaneous matching.

**Theorem 2 (Simultaneous Hall Condition (SIM-HC)).** Let $G = (A \cup B, E)$ and sets $S, T$ be an overlapping bipartite graph. There exists a SIM-BM, iff $|N(P)| + |N(P^S) \cap N(P^T)| \geq |P|$ for $P \subseteq A$.

**Proof.** We prove SIM-HC by induction on $|A|$. When $|A| = 1$, the statement holds. Let $|A| = k > 1$.

If $A^S = \emptyset$ or $A^T = \emptyset$ then SIM-HC reduces to the condition of Hall’s theorem and the statement is true for that reason. Hence, we assume $A^S \neq \emptyset$ and $A^T \neq \emptyset$. We show that there is an edge $(u, v)$ that can be chosen for a simultaneous matching and the graph $G'_{(u,v)}$ will satisfy SIM-HC. Following (Diestel 2006), page 37, we consider two cases. The first case when all subsets of $A$ satisfy the strict SIM-HC, namely, $|N(P)| + |N(P^S) \cap N(P^T)| > |P|$ and the second case when we have an equality.

**Case 1.** Suppose $|N(P)| + |N(P^S) \cap N(P^T)| > |P|$ for all sets $P \subseteq A$. As $A^S \neq \emptyset$ we select any edge $(u, v), u \in A^S$ and construct the graph $G'_{(u,v)}$ (the case $u \in A^T$ is symmetric). For any set $P \subseteq A \setminus \{u\}$ we consider two cases: either $v \notin N(P)$ or $v \in N(P)$. In the first case, the neighborhood of $P$ is the same in $G$ and $G'$, so the SIM-HC holds for $P$. In the case that $v \in N(P)$, then either $v$ is a shared neighbor of $P^S$ and $P^T$, which means that $|N(P^S) \cap N(P^T)| = |N(P^S)| \cap N(P^T)| - 1$ but $N(P) = N(P)$ by construction, or $v$ is a neighbor of $P^S$ but not of $P^T$. Therefore $N'(P) \geq N(P) - 1$. But $N'(P^S) \cap N'(P^T) = |N(P^S) \cap N(P^T)| - 1 \geq |P|$ for any set $P$ in $G'$. By the inductive hypothesis there exists a simultaneous matching in it.

**Case 2.** Suppose that there exists a set $P \subseteq A$ such that $|N(P)| + |N(P^S) \cap N(P^T)| = |P|$. Let $Q = \langle A' \cup B', E' \rangle$ such that $A' = A \setminus P, B' = B \setminus (N(P^S) \cap N(P^T))$ and $E' = \{(u, v) \in E \cap (A' \times B') | (u \in A^S \Rightarrow v \notin N(P) \setminus N(P^T)) \wedge (u \in A^T \Rightarrow v \notin N(P) \setminus N(P^S)) \wedge (u \in A^{ST} \Rightarrow v \notin N(P)) \}$.

There exists a simultaneous matching in $G - Q$ by the inductive hypothesis. We claim that the SIM-HC holds also for $Q$. This implies that, by the inductive hypothesis, there exists a simultaneous matching in $Q$. Suppose there exists a set $P' \subseteq A'$ that violates the SIM-HC in $Q$.

We denote as $N(P)$ the neighborhood of $P$ in $G$ and $N_Q(P^S)$ as the neighborhood of $P'$ in $Q$. We know that the sets $P'$ and $P$ are disjoint. We observe that $N(P \cup
Removing edges

To build a propagator, we consider how to detect edges that cannot appear in any simultaneous matching.

**Definition 5.** Let \( G = (A \cup B, E) \) and sets \( S, T \) be an overlapping bipartite graph. A set \( P, P \subseteq A, \) is

- a **simultaneous Hall set** if
  \[
  |N(P)| + |N'(P) \cap N'(P')| = |P|.
  \]
- an **almost simultaneous Hall set** if
  \[
  |N(P)| + |N'(P) \cap N'(P')| = |P| + 1.
  \]
- a **loose set** if
  \[
  |N(P)| + |N'(P) \cap N'(P')| \geq |P| + 2.
  \]

**Theorem 3.** \( G = (A \cup B, E) \) and sets \( S, T \) be an overlapping bipartite graph. Each edge \((u, v), u \in A \) and \( v \in B \) can be extended to a matching that covers \( S \) and \( T \) if

1. for each set \( P \):
   - \( |N(P)| + |N'(P) \cap N'(P')| \geq |P| \)
2. for each simultaneous Hall set \( P \):
   - \( |N(P)| + |N'(P) \cap N'(P')| \geq |P| \)
3. for each almost simultaneous Hall set \( P \):
   - \( |N(P)| + |N'(P) \cap N'(P')| \geq |P| \)

**Proof. Soundness.** The soundness of Rule 1a follows from Theorem 2. Let \((u, v)\) be an edge that we want to extend to a matching. Suppose that \((u, v)\) violates one of the rules for a **SIM-HALL-SET** or an **A-SIM-HALL-SET** \( P \) in \( G \). We show that if \((u, v)\) is selected to be in a matching, then \( P \) fails **SIM-HC** in \( G'_{(u,v)} \).

**Rule 2a:** If \((u, v)\) violates Rule 2a for a **SIM-HALL-SET** \( P \) then \(|N'(P) \cap N'(P')| = |N(P) \cap N(P')| - 1\) and \( N'(P) = N(P) \), so the **SIM-HC** is violated for \( P \) in \( G' \).

**Rule 2b:** If \((u, v)\) violates Rule 2b for a **SIM-HALL-SET** \( P \) then \(|N'(P)| = |N(P)| - 1\) and \( |N'(P) \cap N'(P')| = |N(P) \cap N(P')|\) so the **SIM-HC** is violated for \( P \) in \( G' \).

**Rule 2c:** Symmetric to Rule 2b.

**Rule 2d:** If \((u, v)\) violates Rule 2d for a **SIM-HALL-SET** \( P \) then \(|N'(P)| = |N(P)| - 1\) so the **SIM-HC** is violated for \( P \) in \( G' \).

**Rule 3a:** If \((u, v)\) violates Rule 3a for an **A-SIM-HALL-SET** \( P \) then \(|N'(P)| = |N(P)| - 1\) and \( |N'(P) \cap N'(P')| = |N(P) \cap N(P')|\) - 1, so \( |N'(P)| + |N'(P) \cap N'(P')| = |P| - 1\) and the **SIM-HC** is violated for \( P \) in \( G' \).

**Completeness.** Second, we show that Rules 2a- 3a are complete. We will show that we can use any edge \((u, v)\) in a matching by showing that the graph \( G'_{(u,v)} \) satisfies the **SIM-HC**, thus has a **SIM-BM**.

Suppose there is a set \( P \) that violates the **SIM-HC** in \( G' \) but not in \( G \) so that

\[
|N'(P)| + |N'(P) \cap N'(P')| < |P| \tag{1}
\]

and

\[
|N(P)| + |N'(P) \cap N'(P')| \geq |P| \tag{2}
\]
Note that \( |N'(P)| = N(P) \setminus \{v\} \) and \( |N'(PS) \cap N'(PT)| = N(PS) \cap N(PT) \setminus \{v\} \). Hence, \( |N'(P)| \geq |N(P)| - 1 \) and \( |N'(PS) \cap N'(PT)| \geq |N(PS) \cap N(PT)| - 1 \).

There are three cases to consider for \( P \) in \( G \), when \( P \) is a loose set, a SIM-HALLSET and an A-SIM-HALLSET in \( G \). These cases are similar, so we consider only the most difficult case. Let \( P \) be an A-SIM-HALLSET in \( G \). If \( u \in A^S \) then \( v \not\in N(PS) \cup N(PT) \) by Rule 3a. Hence \( |N'(PS) \cap N'(PT)| = N(PS) \cap N(PT) \), so \( |N'(P)| + |N'(PS) \cap N'(PT)| \geq |N(P)| + |N(PS) \cap N(PT)| - 1 \) and therefore (1) and (2) cannot both be true.

If \( u \in A^S \) (\( u \in A^T \) is symmetric) then \( v \in N(PS) \cup N(PT) \) or its complement. In the first case \( N'(P) = N(PS) \), while in the second \( N'(PS) \cap N'(PT) = N(PS) \cap N(PT) \). In both cases, \( |N'(P)| + |N'(PS) \cap N'(PT)| \geq |N(P)| + |N(PS) \cap N(PT)| - 1 \) so (1) and (2) cannot both be true.

**Running example.** Consider again our running example. We show that Rules 2a-3a remove every edge that can not be extended to a matching. Consider the set \( P = \{X_2, X_3, X_4, X_6\} \). This is a SIM-HALLSET as \( N(P) = \{1, 2, 3\}, N(PS) \cap N(PT) = \{2\} \text{ and } |P| = |N(P)| + |N(PS) \cap N(PT)| = 4 \). Hence, by Rule 2d we prune 1, 2, 3 from X_5 and by Rule 2a we prune 2 from X_1 and X_7. Now consider the set \( P = \{X_2, X_3, X_6\} \). This is an A-SIM-HALLSET. By Rule 3a we prune 2 from X_4. The set \( P = \{X_2, X_3, X_6\} \) is also an A-SIM-HALLSET and, by Rule 3a, we prune 2 from X_3. Next consider the set \( P = \{X_3, X_4\} \) which is a SIM-HALLSET. By Rules 2b and 2c we prune 1, 3 from X_1, X_2, X_4 and X_7. Now, \( \{X_1\} \) is a SIM-HALLSET and 4 is pruned from X_5 by Rule 2d. Finally, from the simultaneous Hall set \( \{X_5\} \), we prune 5 from X_7 using Rule 2c and we are now at the fixpoint.

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| \( A^S = S \setminus T \) | \( X_1 \) | \( X_2 \) | * | * |
| \( A^S = S \cap T \) | \( X_3 \) | * | * | * |
| \( A^T = T \setminus S \) | \( X_6 \) | * | * |

**The overlapping ALLDIFFERENT constraint**

We now use these results to build a propagator.

**Definition 6.** OVERLAPPINGALLDIFF(X, S, T) where \( S \subseteq X, T \subseteq X, S \cup T = X \) holds iff ALLDIFFERENT(S) and ALLDIFFERENT(T) hold simultaneously.

Enforcing DC on the OVERLAPPINGALLDIFF constraint is \( NP \)-hard (Bessiere et al. 2007). We consider instead enforcing just BC. This relaxation is equivalent to the simultaneous matching problem on a bipartite convex variable-value graph. Our main result is an algorithm that enforces BC on the OVERLAPPINGALLDIFF constraint in \( O(nd^2) \) time. The algorithm is based on the decomposition of the OVERLAPPINGALLDIFF constraint into a set of arithmetic constraints derived from Rules 2b–3a. It is inspired by a decomposition of ALLDIFFERENT (Bessiere et al. 2009). As there, we introduce Boolean variables \( a_{id, b_{id}} \) to represent which \( X_i \) takes a value in the interval \([l, u]\) and the variables \( CS, CST \) and \( CT \) to represent bounds on the number of variables from \( S \setminus T, T \setminus S \) and \( S \cap T \) that may take values in the interval \([l, u]\). We introduce the following set of constraints for \( 1 \leq i \leq n, 1 \leq l \leq u \leq d \) and \( u - l < n \):
Then, there exists a set \( P \) of real numbers \( a, b \) such that a simultaneous matching can be found in poly-open intervals \( [1, \ldots, n] \) of length \( x \). Using the sequence \( a, (g, 2, g, 3, g, 4, \ldots, g^{k+1}m-1, g^{k+1}m) \). We show that intervals \( 0 < l \leq x \) and \( l \neq \frac{x}{2} \) gives the contradiction. As \( l \) is an interval and \( \sum^{3} \) equals at most. There are \( O(d^{2}) \) intervals. Finding \( N(P^{S}) \cap N(P^{T}) \) takes \( O(n + d) \) time inside an interval. Enforcing the rule takes \( O(n) \) time. Hence, the total time complexity is \( O(n d^{3}) \).

From Theorems 6 and 7 it follows that

**Theorem 8.** \( BC \) on OVERLAPPINGALLDIFF can be enforced in \( O(n d^{3}) \) time.

**Running example.** We demonstrate the action of constraints (3)-(12). The interval \( [1, 4] \) contains a SIM-HALLSET \( P = \{X_2, X_3, X_4, X_6\} \). Rule 2d. \( ub(C_{13}^{P}) \geq 1 \) and \( lb(C_{23}^{P}) \geq 1 \) \( \Rightarrow \) (9), (10) \( \Rightarrow \) \( ub(C_{12}^{P}) \leq 1 \) and \( lb(C_{13}^{P}) \leq 1 \) \( \Rightarrow \) (8) \( \Rightarrow lb(C_{13}^{P}) \leq 2 \) \( \Rightarrow \) the interval \( [1, 3] \) is saturated, as \( lb(C_{13}^{P}) = ub(C_{13}^{P}) \). Hence, by (3)-(5), [1, 3] is removed from \( D(X_5) \). Rules 2b, 2c. As \( lb(C_{13}^{P}) = 2 \) \( \Rightarrow \) (9) \( \Rightarrow \) \( ub(C_{12}^{P}) \leq 1 \) \( \Rightarrow \) (12) \( \Rightarrow lb(C_{12}^{P}) \leq 1 \). The interval \( [1, 2] \) is saturated, as \( lb(C_{12}^{P}) = ub(C_{12}^{P}) \). Hence, by (3)-(4), (6), [1, 2] is removed from \( D(X_1) \). Similarly, [2, 3] is removed from \( D(X_5) \).

**Rule 2a.** This is satisfied as 2 is removed from all variables outside \( P \).

**Exponential separation.**

We now give a pathological problem on which our new propagator does exponentially less work than existing methods.

**Theorem 9.** There exists a class of problems such that enforcing \( BC \) on OVERLAPPINGALLDIFF immediately detects unsatisfiability while a search method that enforces \( BC \) on the decomposition into ALLDIFFERENT constraints explores an exponential search tree regardless of branching.

**Proof.** The instance \( T_n \) is defined as follows \( T_n = ALLDIFFERENT([X \cup Y] \land ALLDIFFERENT([Y \cup Z]), D(X_i) = [1, 2n - 1], i = 1, \ldots, n, D(Y_1) = [1, 4n - 1], i = 1, 2n, D(Z_1) = [2n, 4n - 1], i = 1, \ldots, n.\) OVERLAPPINGALLDIFF. Consider the interval \( [1, 4n - 1], |P| = 4n, |N(P^{S})| = 4n - 1 and |N(P^{T}) \cap N(P^{T})| = 0.\) By Theorem 2, we detect unsatisfiability.
Decomposition. Consider any ALLDIFFERENT constraint. A subset of $n$ or fewer variables has at least $2n - 1$ values in their domains and a subset of $n + 1$ to $3n$ variables has $4n - 1$ values in their domains. Thus, to obtain a Hall set and prune, we must instantiate at least $n - 1$ variables.

Experimental results

To evaluate the performance of our decomposition we carried out an experiment on random problems. We used Ilog 6.2 on an Intel Xeon 4 CPU, 2.0 GHz, 4GB RAM. We compare the performance of the $DC$, $BC$ (Lopez-Ortiz et al. 2003) propagators and our decomposition into constraints (3)-(12) for the OVERLAPPINGALLDIFF constraint (OBC). We use randomly generated problems with three global constraints: ALLDIFFERENT$(X \cup W)$, ALLDIFFERENT$(Y \cup W)$ and ALLDIFFERENT$(Z \cup W)$, and a linear number of binary ordering relations between variables in $X$, $Y$ and $Z$. We use a random variable ordering and run each instance with 50 different seeds. As Table 1 shows, our decomposition reduces the search space significantly, is much faster and solves more instances overall.

Table 1: Random problems. $n$ is the size of $X$, $Y$ and $Z$, $o$ is the size of $W$, $d$ is the size of variable domains. Number of instances solved in 300 sec out of 50 runs / average backtracks/average time to solve.

| n,d,o | $BC$ | $DC$ | OBC |
|-------|------|------|-----|
|       | $#s$ / $#bt$ / t | $#s$ / $#bt$ / t | $#s$ / $#bt$ / t |
| 4, 15, 10 | 14/2429411 / 61.8 | 41/1491341 / 52.1 | 42 / 17240 / 32.5 |
| 4, 16, 11 | 6/5531047 / 153.7 | 22 / 1745160 / 67.9 | 31 / 8421 / 19.5 |
| 4, 17, 12 | 1 / 17 / 0 | 6 / 2590427 / 100.9 | 24 / 8185 / 21.5 |
| 5, 16, 10 | 11 / 3052298 / 82.0 | 37 / 1434903 / 58.2 | 42 / 20482 / 48.5 |
| 5, 17, 11 | 2 / 3309113 / 94.5 | 19 / 2593819 / 114.6 | 26 / 4374 / 15.8 |
| 5, 18, 12 | 1 / 17 / 0 | 4 / 2666556 / 133.1 | 22 / 3132 / 12.2 |
| 6, 17, 10 | 11 / 2845367 / 79.1 | 31 / 1431671 / 66.3 | 40 / 6796 / 21.9 |
| 6, 18, 11 | 4 / 199357 / 66.6 | 16 / 1498128 / 80.2 | 31 / 4494 / 17.5 |
| 6, 19, 12 | 4 / 3183496 / 110.0 | 5 / 1035126 / 66.2 | 27 / 3362 / 15.5 |
| TOTALS | | | |
| sol/total | 54 / 450 | 181 / 450 | 285 / 450 |
| avg time for sol | 78.072 | 70.551 | 24.689 |
| avg bt for sol | 2818926 | 1666568 | 9561 |

Conclusions

We have generalized Hall’s theorem to simultaneous matchings in a bipartite graph. This generalization suggests a polynomial time algorithm to find a simultaneous matching in a convex bipartite graph. We applied this to a problem in constraint programming of propagating conjunctions of ALLDIFFERENT constraints. Initial experimental results suggest that reasoning about such conjunctions can significantly reduce the size of the explored search space. There are several avenues for future research. For example, the algorithmic techniques proposed in (Puget 1998) and (Lopez-Ortiz et al. 2003) may be generalizable to simultaneous bipartite matchings, giving more efficient propagators. Further, matchings are used to propagate other constraints such as NVALUE (Bessiere et al. 2006). It may be possible to apply similar insights to develop propagators for conjunctions of other global constraints, or to improve existing propagators for global constraints that decompose into overlapping constraints like SEQUENCE (Brand et al. 2005). Finally, we may be able to develop polynomial time propagators for otherwise intractable cases if certain parameters are fixed (Bessiere et al. 2008).

Acknowledgments

This research is supported by ANR UNLOC project (ANR 08-BLAN-0289-01), the Australian Government’s Department of Broadband, Communications and the Digital Economy and the ARC.

References

Bessiere, C.; Hebrard, E.; Hnich, B.; Kiziltan, Z.; and Walsh, T. 2006. Filtering algorithms for the NVALUE constraint. Constraints, 11 (4): 271–293.

Bessiere, C.; Hebrard, E.; Hnich, B.; Kiziltan, Z.; and Walsh, T. 2008. The Parameterized Complexity of Global Constraints. In Proc. of 23rd AAAI, 235–240.

Bessiere, C.; Hebrard, E.; Hnich, B.; and Walsh, T. 2007. The Complexity of Reasoning with Global Constraints. Constraints. 12(2): 239–259.

Bessiere, C.; Katsirelos, G.; Narodytska, N.; Quimper, C.-G.; and Walsh, T. 2009. Decompositions of all different, global cardinality and related constraint. In Proc. of 21st IJCAI, 419–424.

Brand, S.; Narodytska, N.; Quimper, C.-G.; Stuckey, P.; and Walsh, T. 2007. Encodings of the Sequence Constraint. In Proc. of CP-2007, 210–224.

Diestel, R. 2006. Graph Theory. Springer.

Hall, P. 1935. On representatives of subsets. J. of the London Math. Soc. 10:26–30.

Kutz, M.; Elbassioni, K.; Katriel, I.; and Mahajan, M. 2008. Simultaneous matchings: Hardness and approximation. J. of Computer and System Sciences 74(5):884–897.

Lardeux, F.; Monfroy, E.; and Saubion, F. 2008. Interleaved Alldifferent Constraints: CSP vs. SAT Approaches. In Proc. of 12th Nat. Conf. on AI, 359–366.

Laurière, J.-L. 1978. A language and a program for stating and solving combinatorial problems. Art. Intell. 10:29–127.

Leconte, M. 1996. A bounds-based reduction scheme for constraints of difference. In Proc. of 12th Int. Workshop on Constraint-based Reasoning.

Lopez-Ortiz, A.; Quimper, C.; Tromp, J.; and van Beek, P. 2003. A fast and simple algorithm for bounds consistency of the alldifferent constraint. Proc. of 18th IJCAI, 245–250.

Puget, J. 1998. A fast algorithm for the bound consistency of alldiff constraints. In 15th Nat. Conf. on AI, 359–366.

Régin, J.-C. 1994. A filtering algorithm for constraints of difference in CSPs. In 12th Nat. Conf. on AI, 362–367.

Stergiou, K., and Walsh, T. 1999. The difference all-difference makes. In Proc. of 16th IJCAI, 414–419.