Propagation Redundancy for Permutation Channels *

C.W. Choi and J.H.M. Lee
Dept. of Comp. Sci. & Eng.
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong SAR, China
{cwchoi,jlee}@cse.cuhk.edu.hk

P. J. Stuckey
Dept. of Comp. Sci. & Soft. Eng.
University of Melbourne, 3010
Australia
pjs@cs.mu.oz.au

1 Introduction

Finding a good model of a constraint satisfaction problem (CSP) is a challenging task. A modeller must specify a set of constraints that capture the definitions of the problem, and the model should also have strong propagation. In other words, the model should be able to quickly reduce the domains of the variables of the problem, *and* the implementation of these propagators should be efficient, *and* the search space should not be too large.

A problem can be modelled differently from two viewpoints using two different sets of variables. In redundant modelling [Cheng *et al,* 1999], we connect the two different models with channelling constraints, which relates valuations in the two different models stronger propagation behaviour can be observed. However, the additional variables and constraints impose extra computation overhead may outweigh the gain of reduction in search space.

In this paper we consider redundant models connected by permutation channels, which commonly arise when the underlying problem is some form of assignment problem. Since each model is complete and only admits the solutions of the problem, each model is logically redundant with respect to the other model plus the permutation channel. In order to keep the benefits of redundant modelling without paying all the costs, We give a theorem which allows us to determine when we can eliminate constraints in the mutually redundant models that do not give extra propagation. Due to space limitations, we state the theorem without proof.

2 Reasoning about Domain Propagation

We consider integer constraint solving with constraint propagation and tree search.

An *integer valuation* θ is a mapping of variables to integer values, written $\{x_1 \mapsto d_1, \ldots, x_n \mapsto d_n\}$. Let *vars* be the function that returns the set of variables appearing in a constraint or valuation. A *constraint c* defines a set of valuations *solns{c)* each mapping the same set of variables *vars(c).* We call *solns(c)* the *solutions* of *c.* A constraint *c* is *logically redundant* with respect to a set of constraints *C* if $\models C \rightarrow c$.

A *domain D* is a complete mapping from a fixed (countable) set of variables $\mathcal{V}$ to finite sets of integers. *A false domain D* is a domain with *D(x)* $= \emptyset$ for some *x.* A domain D_1 is *stronger* than a domain *I)n,* written $D_1 \sqsubseteq D_2$, if D_1 is a false domain or $D_1(x) \subseteq D_2(x)$ for all variables *x.* The initial domain D_{init} gives the initial values possible for each variable, allows us to restrict attention to domains *D* such that $D \sqsubseteq D_{init}$.

We adopt the notion of *propagation solver* and *domain consistency*[1] from Schulte and Stuckey [2001]. *A propagator f* is a monotonically decreasing function from domains to domains. A *propagation solver* for a set of propagators *F* and current domain *D, solv(F, D),* repeatedly applies all the propagators in *F* starting from domain *D* until there is no further change in resulting domain. A domain *D* is *domain consistent* if *D* is the least domain containing all solutions of *r* in *D.* Define the *domain consistency propagator dom(c)* for a constraint *c* such that *solv(dom(c), D)* is always domain consistent for r.

For all domains $D \sqsubseteq D_{init}$, a set of propagator F_2 is made *propagation redundant* by a set of propagators *F\,* written $F_1 \gg F_2$, if $solv(F_1, D) \sqsubseteq solv(F_2, D)$, and is *equivalent* to F_u written *F1~F2,* if *solv{Fi, D) = solv(F2,D).*

It is well known that in general the domain propagation of a conjunction of constraints is not equivalent to applying the domain propagators individually. But there are cases where propagation of a conjunction is equivalent to propagation on the individual conjuncts.

Lemma 1 If c_1 *and* c_2 *share at most one variable x, then {dom[c1), dom[c2)}* $\approx$ *{dom{*c_1Ar$_2$)}.

An *atomic constraint* is one of X_i = *d* or $x_i \neq d$ where X_i £ V and *d* is an integer. An atomic constraint represents the basic changes in domain that occur during propagation.

A *propagation rule* is of the form $C \mapsto c$ where *C* is a conjunction of atomic constraints, *c* is an atomic constraint and $\not\models C \rightarrow c$. Note our propagation rules are similar to the "membership rules" of Apt and Monfroy [2001] except we allow equations on the right hand side.

A propagator / *implements* a propagation rule $C \mapsto$ c if for each $D \sqsubseteq D_{init}$ whenever $\models D \rightarrow$ C, then $\models f(D) \rightarrow$ r.

*We thank Barbara Smith and the anonymous referees for their constructive comments. The work described in this paper was substantially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project no. CUHK4204/01E).

[1] Equivalently, hyper-arc or generalized arc consistent.

We can characterize a propagator / in terms of the propagation rules that it implements. Let *rules(f)* be the set of rules implemented by /. Then *prop(f)* $\subseteq$ *rulcs(f)* are a set of propagation rules such that every $r \in$ *rulcs(f)* is subsumed by a rule $r' \in$ *prop(f).*

3 Permutation Channels

A common form of redundant modelling is when we consider two viewpoints to a permutation problem. We can view the problem as finding a bipartite matching between two sets of objects of the same size. For notational convenience, let the two viewpoints as having the set of variables $X = \{x_0, \ldots, x_n\}$, and V = $\{y_0, \ldots, y_n\}$ respectively.

The *permutation channel* $C_{\bowtie}$ is defined by the conjunction of constraints $\bigwedge_{i=0}^{n} \bigwedge_{j=0}^{n} (x_i = j \Leftrightarrow y_j = i)$. The *permutation channel propagator* $F_{\bowtie}$ maintains domain consistency of each individual bi-implication, that is $\bigcup_{i=0}^{n} \bigcup_{j=0}^{n} \{dom(x_i = j \Leftrightarrow y_j = i)\}$.

Smith [2000] first observes that the permutation channel makes each of the disequations between variables in either model propagation redundant. Walsh [2001] proves this holds for other notions of consistency.

Lemma 2 (Walsh, 2001) $F_{\bowtie} \gg \{dom(x_i \neq x_k)\}$

Related to $C_{\bowtie}$ is the *permutation channel function* $\bowtie$ which is a bijection between atomic constraints in *X* to atomic constraints in Y, $\bowtie(x_i = j) = (y_j = i)$, and $\bowtie(x_i \neq j) = (y_j \neq i)$. We extend $\bowtie$ to map conjunctions of constraints in the obvious manner $\bowtie(C_1 \wedge C_2) = \bowtie(C_1) \wedge \bowtie(C_2)$.

The fundamental theorem states that a constraint in *Y* is propagation redundant if there exist a constraint in *X* when conjuncts with $C_{\bowtie}$ logically imply every propagation rules implemented by the constraint in *Y'*. Since $\bowtie$ is bijective, the theorem is valid when *X* and *Y* are reversed.

Theorem 3 *Let f_Y be a propagator on Y, and c_X be a constraint on X. If* $\models (D_{init} \wedge c_X \wedge \bowtie(C)) \rightarrow \bowtie(c)$ *for all* $(C \rightarrowtail c) \in prop(f_Y)$, *then* $\{dom(c_X)\} \cup F_{\bowtie} \gg \{f_Y\}$.

Example 4 Smith [2000] suggests two ways to model the Langford's problem as a permutation problem and how to combine them with the permutation channel. She points out that the so-called *minimal combined model,* which includes only *X* model and the permutation channel, gives as much pruning as the full combined model. This is proved in an ad hoc manner by Choi and Lee [2002]. We prove this formally using our generic approaches.[2]

The disequality constraints are propagation redundant by Lemma 2. Consider the separation constraints $c_Y \equiv y_j = 3i \Leftrightarrow y_{j+(i+2)} = 3i + 1$. The propagation rules for $dom(c_Y)$ are (r1) $y_j = 3i \rightarrowtail y_{j+(i+2)} = 3i + 1$, (r2) $y_{j+(i+2)} = 3i + 1 \rightarrowtail y_j = 3i$, (r3) $y_j \neq 3i \rightarrowtail y_{j+(i+2)} \neq 3i + 1$, and (r4) $y_{j+(i+2)} \neq 3i + 1 \rightarrowtail y_j \neq 3i$. We have that for $c_X \equiv x_{3i+1} = x_{3i} + (i+2)$, $\models (D_{init} \wedge c_X \wedge \bowtie(C)) \rightarrow \bowtie(c)$ for all propagation rules above. For example (r1) is mapped to $x_{3i} = j \rightarrowtail x_{3i+1} = j + i + 2$. Hence the conditions of Theorem 3 hold and $dom(c_Y)$ is propagation redundant.

[2]The complete description of the two permutation models for the Langford's Problem can be found in [Choi and Lee, 2002].

Note the importance of *equational* propagation rules such as $y_j = 3i \rightarrowtail y_{j+(i+2)} = 3i + 1$. If we only allowed disequations on the right hand side, we would replace this with rules $y_j = 3i \rightarrowtail y_{j+(i+2)} \neq k, 0 \leq k \neq 3i + 1 \leq n$. It is impossible to prove that the translated versions are implied by $D_{init} \wedge c_X$.

We can similarly show that each of the constraints $c'_Y \equiv y_j = 3i \Leftrightarrow y_{j+2(i+2)} = 3i + 2$ are propagation redundant using $c_X \wedge c'_X$, where $c'_X \equiv x_{3i+2} = x_{3i+1} + (i + 2)$ by Theorem 3. Although model M_X does not include a domain propagator for $c_X \wedge c'_X$, we can still show propagation redundancy since $\{dom(c_X), dom(c'_X)\} \approx \{dom(c_X \wedge c'_X)\}$ by Lemma 1.

Similar reasoning applies to show that each constraint $y_j \neq 3i$ where $0 \leq i \leq 8$ and $27 - 2(i + 2) \leq j \leq 26$ is made propagation redundant by $(x_{3i+1} = x_{3i} + (i+2)) \wedge (x_{3i+2} = x_{3i+1} + (i + 2)) \wedge (0 \leq x_{3i+2} \leq 26)$. □

4 Conclusion

We have extend our approach to other types of channelling constraints and lead to significantly faster models that do not increase the search space. Although we have illustrated the use of the theorems herein by hand, the approach can clearly be automated. We can constructs the propagation rules automatically using the approach of Abdennadher and Rigotti [2002]. We are interested in extending the work to reason bounds propagation. Another direction is to study a weaker notion of propagation redundancy which allows removal of constraints without affecting the search space given a specific search heuristic.

References

[Abdennadher and Rigotti, 2002] S. Abdennadher and C. Rigotti. Automatic generation of rule-based solvers for intentionally defined constraints. *IJTA1,* 11(2):283—302, 2002.

[Apt and Monfroy, 2001] K. Apt and E. Monfroy. Constraint programming viewed as rule-based programming. *TPLP,* 1(6):713-750,2001.

[Cheng *et al,* 1999] B.M.W. Cheng, K.M.F. Choi, J.H.M. Lee, and J.C.K. Wu. Increasing constraint propagation by redundant modelling: an experience report. *Constraints,* 4(2): 167-192,1999.

[Choi and Lee, 2002] C.W. Choi and J.H.M. Lee. On the pruning behaviour of minimal combined models for permutation CSPs. In *International Workshop on Reformulating Constraint Satisfaction Problems,* 2002.

[Schulte and Stuckey, 2001] C. Schulte and P. J. Stuckey. When do bounds and domain propagation lead to the same search space. In *PPDP 2001,* pages 115-126,2001.

[Smith, 2000] B. M. Smith. Modelling a permutation problem. Research Report 2000.18, School of Computer Studies, University of Leeds, 2000.

[Walsh, 2001] T. Walsh. Permutation problems and channelling constraints. In *LPAR 2001,* pages 377-391,2001.