Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)

Variety Reasoning for Multiset Constraint Propagation*

Y.C. Law, J.H.M. Lee, and M.H.C. Woo
Department of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
{yclaw,jlee,hcwoo}@cse.cuhk.edu.hk

Abstract

Set variables in constraint satisfaction problems (CSPs) are typically propagated by enforcing set bounds consistency together with cardinality reasoning, which uses some inference rules involving the cardinality of a set variable to produce more prunings than set bounds propagation alone. Multiset variables are a generalization of set variables by allowing the elements to have repetitions. In this paper, we generalize cardinality reasoning for multiset variables. In addition, we propose to exploit the variety of a multiset—the number of distinct elements in it—to improve modeling expressiveness and further enhance constraint propagation. We derive a number of inference rules involving the varieties of multiset variables. The rules interact varieties with the traditional components of multiset variables (such as cardinalities) to obtain stronger propagation. We also demonstrate how to apply the rules to perform variety reasoning on some common multiset constraints. Experimental results show that performing variety reasoning on top of cardinality reasoning can effectively reduce more search space and achieve better runtime in solving multiset CSPs.

1 Introduction

Many combinatorial design problems can be modeled as *constraint satisfaction problems* (CSPs) using *set variables*, which can take collections of distinct elements as their values. The domain of a set variable is typically represented by its set upper and lower bounds [Gervet, 1997] and propagated by enforcing set bounds consistency [Gervet, 1997] together with cardinality reasoning [Azevedo and Barahona, 2000]. By considering also the cardinality of a set variable during propagation, more prunings can be produced than set bounds propagation alone and further reduce the search space.

Multiset variables are a generalization of set variables by allowing the elements to have repetitions. Consider the template design problem (prob002 in CSPLib) which is to assign some designs to printing templates subject to some constraints. Each template has a fixed number of slots for the designs. One possible modeling is to use an integer variable for each slot in a template. However, this model introduces unnecessary symmetries as the slots are indistinguishable. Since a design can appear multiple times in one template, a more "natural" model is to use a multiset variable for each template to avoid the symmetries. The domain of each variable is the set of all possible multisets of designs that can be assigned to the template. Other than this problem, Frisch et al. listed a collection of ESSENCE specifications[1] containing many problems that can be modeled using multiset variables.

The cardinality of a set reveals the *total* number of elements in it. Incorporating a cardinality variable to a set variable [Azevedo and Barahona, 2000] enjoys success in enhancing propagation for set constraints. On the other hand, the number of *distinct* elements, which we call *variety*, is a property specific to multisets. In this paper, we propose a multiset variable representation which is an improvement over the occurrence representation [Kiziltan and Walsh, 2002; Walsh, 2003]. We incorporate a cardinality variable as well as a variety variable to the representation which do not just allow to express certain problem constraints much more easily (i.e., better modeling expressiveness), but also increase the opportunities to infer more domain prunings for better solving efficiency. We derive a number of inference rules involving the varieties of multiset variables and show how the traditional components of multiset variables (such as cardinalities) interact with the varieties to achieve stronger constraint propagation. We also apply our rules to perform variety reasoning on some common multiset constraints. Experimental results confirm that performing variety reasoning on top of cardinality reasoning can further reduce the search space and give a better runtime in solving multiset CSPs.

2 Background

A *constraint satisfaction problem* (CSP) is a triple $\mathcal{P} = (\mathcal{X}, \mathcal{D}, \mathcal{C})$, where $\mathcal{X} = \{X_1, \ldots, X_n\}$ is a finite set of *vari-*

*We thank the anonymous referees for their constructive comments. The work described in this paper was substantially supported by grants (CUHK413207 and CUHK413808) from the Research Grants Council of Hong Kong SAR.

[1] Available at `http://www.cs.york.ac.uk/aig/constraints/AutoModel/Essence/specs120/`

Equality	$X = Y$ iff $occ(i, X) = occ(i, Y)$
Subset	$X \subseteq Y$ iff $occ(i, X) \leq occ(i, Y)$
Union	$X \cup Y = Z$ iff $occ(i, Z) = \max(occ(i, X), occ(i, Y))$
Union-Plus	$X \uplus Y = Z$ iff $occ(i, Z) = occ(i, X) + occ(i, Y)$
Intersection	$X \cap Y = Z$ iff $occ(i, Z) = \min(occ(i, X), occ(i, Y))$

Table 1: Some common multiset operations

ables, $\mathcal{D} = \{D_{x_1}, \ldots, D_{x_n}\}$ is a set of finite *domains* of possible values, and $\mathcal{C}$ is a set of *constraints*. Each constraint involves a subset of the variables in $\mathcal{X}$, limiting the combination of values that the variables in the subset can take. A *solution* of $\mathcal{P}$ is to assign a value to every variable $X_i \in \mathcal{X}$ from its domain D_{X_i} such that all the constraints in $\mathcal{C}$ are satisfied.

A *set* is an unordered list of elements *without repetitions*. The *cardinality* of a set S is the number of elements in S, denoted as $|S|$. Gervet [1997] proposed to represent the domain of a set variable S with an interval $[RS(S), PS(S)]$ such that $D_S = \{m | RS(S) \subseteq m \subseteq PS(S)\}$. The *required set* $RS(S)$ contains all the elements which *must exist* in the set, while the *possible set* $PS(S)$ contains any element which *may exist* in the set. S is said to be *bound* when its lower bound equals its upper bound (i.e., $RS(S) = PS(S)$).

Traditional domain reasoning for integer variables is not practical for set variables, as their domains are exponential to the size of possible sets. Gervet [1997] proposed using bounds reasoning to maintain consistency on set variables. A set variable S with set interval domain $[RS(S), PS(S)]$ is *set bounds consistent* with respect to a constraint C if and only if $RS(S) = \bigcap dom_S(C)$ and $PS(S) = \bigcup dom_S(C)$, where $dom_S(C)$ denotes the domain values of S that satisfy C.

A *multiset* is a generalization of set by allowing the elements to have repetitions. We denote a multiset S as $S = \{\!\{\cdots\}\!\}$ and its cardinality as $|S|$. For example, if $S = \{\!\{1, 1, 2, 2, 3\}\!\}$, then $|S| = 5$. In this paper, without loss of generality, we assume that multiset elements are positive integers from 1 to n. We shall use $\emptyset$ to denote both the empty set and the empty multiset. Since the number of occurrences of an element in a multiset variable can be more than one, enumerating all possible multisets to represent a multiset variable domain is even more impractical. Thus, Kiziltan and Walsh [2002; 2003] suggested to represent a multiset variable S with n elements by a vector of occurrence (integer) variables $\langle occ(1, S), \ldots, occ(n, S)\rangle$. Each variable $occ(i, S)$ models the number of occurrences of an element i in S. Its domain is denoted as the interval $D_{occ(i,S)} = [occ_r(i, S), occ_p(i, S)]$. We also define s_r as the multiset whose occurrence of each element i is $occ_r(i, S)$. Similarly, s_p is the multiset whose occurrence of each element i is $occ_p(i, S)$. The multisets s_r and s_p are in fact the required multiset $RS(S)$ and the possible multiset $PS(S)$ of S respectively (i.e., $D_S = [RS(S), PS(S)] = [s_r, s_p]$). This occurrence representation is compact but cannot represent all forms of disjunctions. Most set constraints can be generalized to their multiset counterparts. Table 1 gives some common multiset constraints, in which X, Y, and Z are multiset variables and i is an element.

Multiset variables are usually propagated by enforcing *multiset bounds consistency*, which can be defined using the occurrence representation. A multiset variable S consisting of a vector of occurrence variables $occ(i, S)$ with interval domain $[occ_r(i, S), occ_p(i, S)]$ is *multiset bounds consistent* with respect to a constraint C if and only if $occ_r(i, S) = \min(dom_{occ(i,S)}(C))$ and $occ_p(i, S) = \max(dom_{occ(i,S)}(C))$, where $dom_{occ(i,S)}(C)$ denotes the domain values of the number of occurrences of element i in S that satisfy C. This definition is equivalent to the notion BC by Kiziltan and Walsh [2002; 2003].

3 Variety

A multiset S allows repeated elements. Its cardinality $|S|$ reveals only the total number of elements, but not the number of *distinct* elements, which we define as *variety* and denoted as $\|S\|$. Note that $\|S\|$ can never be greater than $|S|$. In fact, S degenerates to a set when $|S| = \|S\|$ (i.e., the number of occurrences of all elements equals one). In this section, we shall formally define a multiset variable in terms of variety. Then, we describe a naive approach to model variety using meta-constraints, which can hinder propagation. Subsequently, we propose some inference rules which improve over the naive approach. We also demonstrate how variety can help increase propagation for multiset constraints by variety reasoning.

3.1 Multiset Variable with Cardinality and Variety

We define a multiset variable S consisting of three components. The first is a vector of finite domain variables $\langle occ(1, S), \ldots, occ(n, S)\rangle$ [Kiziltan and Walsh, 2002; Walsh, 2003] modeling the number of occurrences of each element in S. On top of the occurrence representation, the second component is a *cardinality variable* C_S [Azevedo and Barahona, 2000] whose domain is denoted as the interval $D_{C_S} = [c_r, c_p]$, modeling the total number of elements in S (denoted as $|S|$). The third is a *variety variable* V_S whose domain is denoted as the interval $D_{V_S} = [v_r, v_p]$, modeling the number of *distinct* elements in S (denoted as $\|S\|$).

For example, suppose $n = 4$ and consider a multiset variable S whose components have the following domains: $D_{occ(1,S)} = [0, 1]$, $D_{occ(2,S)} = [0, 2]$, $D_{occ(3,S)} = [0, 3]$, $D_{occ(4,S)} = [0, 1]$, $D_{C_S} = [0, 7]$, and $D_{V_S} = [0, 4]$. Then, we have (1) $s_r = \emptyset$, as the lower bounds of all the occurrence variables are 0; and (2) $s_p = \{\!\{1, 2, 2, 3, 3, 3, 4\}\!\}$, as the upper bounds of the occurrence variables of elements 1, 2, 3, and 4 are 1, 2, 3, and 1 respectively. The domain of S is in fact the multiset interval $[\emptyset, \{\!\{1, 2, 2, 3, 3, 3, 4\}\!\}]$.

Introducing a variety variable to the representation allows us to model the domain of a multiset variable in a more precise way (although still inexact). Consider S in the previous example and suppose we are interested in only the domain values whose varieties are 1. Without the variety variable V_S, we can only set D_{C_S} to $[1, 3]$. This domain accepts, for example, the multiset $\{\!\{1, 2\}\!\}$, which obviously should not be included. However, with V_S, we can simply set $D_{V_S} = [1, 1]$ to further remove the multisets which contain more than one kind of elements. This essentially models $D_S = \{\{\!\{1\}\!\}, \{\!\{2\}\!\}, \{\!\{2, 2\}\!\}, \{\!\{3\}\!\}, \{\!\{3, 3\}\!\}, \{\!\{3, 3, 3\}\!\}, \{\!\{4\}\!\}\}$, a much more precise representation.

Another advantage is to improve modeling expressiveness by allowing to post *variety constraints* to limit the number of distinct elements in multiset variables. For example, in the template design problem, we may want to restrict a template T to have at most three distinct designs in its slots. Using our representation, posting a variety constraint $\|T\| \leq 3$ simply means to propagate the simple unary constraint $V_T \leq 3$. Without V_T, we need to post a number of meta-constraints to model the requirement, which may hinder propagation. The advantage becomes more obvious when the form of the variety constraints is more complicated, e.g., $\|T_1\| + \|T_2\| \leq 4$. The next subsection describes the naive approach of using meta-constraints to model variety.

3.2 Naive Approach

Consider a multiset variable S with its three components: occurrence $occ(i,S)$ where i is the possible elements in S, cardinality C_S, and variety V_S. We can model the cardinality C_S and the variety V_S using the constraints $C_S = \sum_i occ(i,S)$ and $V_S = \sum_i (occ(i,S) > 0)$ for each distinct element i in S respectively. Note that the latter one is a meta-constraint. Its propagation is usually weak in most constraint solvers. In fact, propagating the meta-constraints neglects the direct relationship between V_S and C_S, and also the more complicated relationship among $occ(i,S)$, C_S, and V_S, as can be shown in the following example.

Consider two multiset variables S_1 and S_2 where $D_{S_1} = [\{\!\{1\}\!\}, \{\!\{1,2,3,3\}\!\}]$ (i.e., $D_{occ(1,S_1)} = [1,1]$, $D_{occ(2,S_1)} = [0,1]$, $D_{occ(3,S_1)} = [0,2]$), $D_{C_{S_1}} = [3,3]$, $D_{V_{S_1}} = [1,3]$, and $D_{S_2} = [\{\!\{1\}\!\}, \{\!\{1,4,5,5\}\!\}]$ (i.e., $D_{occ(1,S_2)} = [1,1]$, $D_{occ(4,S_2)} = [0,1]$, $D_{occ(5,S_2)} = [0,2]$), $D_{C_{S_2}} = [3,3]$, $D_{V_{S_2}} = [1,3]$. Suppose we now post a constraint in which the variety of the union-plus of S_1 and S_2 is not greater than 3 (i.e., $\|S_1 \uplus S_2\| \leq 3$). Reasoning on this constraint reveals that S_1 and S_2 cannot contain elements 2 and 4 respectively, because the cardinalities of both S_1 and S_2 must be 3 and their union-plus can contain at most three different elements. S_1 and S_2 *should* then be bound to $\{\!\{1,3,3\}\!\}$ and $\{\!\{1,5,5\}\!\}$ respectively. However, using the meta-constraint $\sum_{i=1}^{5}(occ(i,S_1) + occ(i,S_2) > 0) \leq 3$, the domains of S_1 and S_2 remain unchanged.

By exploiting the relationships between the three components of a multiset variable, we propose a number of inference rules to strengthen propagation. In the next subsection, we shall systematically enumerate the possible relationships.

3.3 Inferences within One Multiset Variable

Upon creation of a multiset variable S, the vector of occurrence variables $\langle occ(1,S), \ldots occ(n,S)\rangle$, the cardinality variable C_S, and the variety variable V_S will also be created. A number of inference rules are subsequently maintained. Inferences occur between any two kinds of variables (i.e., between $occ(i,S)$ and C_S, between $occ(i,S)$ and V_S, or between C_S and V_S), or among all three of them. Inference rules will be formally described as rewriting rules as in the following schematic figure:

$$\text{(trigger condition)}\ \frac{\text{conditions(which can be nil)}}{\text{changes in constraint store}}$$

(1) Inferences between $occ(i,S)$ and C_S

The cardinality C_S must always remain inside the limits given by the multiset bounds s_r and s_p [Azevedo and Barahona, 2000]. (Recall that s_r and s_p can be computed using the occurrence variables.)

$$(S \text{ changed bounds})\ \frac{\alpha = |s_r|, \beta = |s_p|}{\{\} \mapsto \{C_S \geq \alpha, C_S \leq \beta\}}$$

The cardinality C_S is the sum of the number of occurrences of all elements in a multiset variable. Thus, the number of occurrences of each element is updated when there are changes in the lower bound c_r and the upper bound c_p of the cardinality variable. Here, we only concern with the elements that have not yet been included in the lower bound (i.e., $s_p \setminus s_r$, which is defined as $occ(i, s_p \setminus s_r) = \max\{0, occ(i,s_p) - occ(i,s_r)\}$ for all elements i).

$$(C_S \text{ changed bounds})\ \frac{\alpha = c_r, \beta = c_p}{\{\} \mapsto \{|K| - occ(i,K) \geq \alpha - |s_r|, occ(i,K) \leq \beta - |s_r|\}}$$
where $K = s_p \setminus s_r$.

For example, consider a multiset variable S where $D_S = [\{\!\{1,1\}\!\}, \{\!\{1,1,1,2,2,3\}\!\}]$ (i.e., $D_{occ(1,S)} = [2,3]$, $D_{occ(2,S)} = [0,2]$, $D_{occ(3,S)} = [0,1]$), $D_{V_S} = [1,3]$, and c_p is updated from 6 to 3 (i.e., $D_{C_S} = [2,3]$). Since S can now have at most three elements of at most two different kinds, and there are already two 1s in the required set, only one more element (1, 2, or 3) may exist in S. Based on the inference rule, $K = s_p \setminus s_r = \{\!\{1,1,1,2,2,3\}\!\} \setminus \{\!\{1,1\}\!\} = \{\!\{1,2,2,3\}\!\}$ and $occ(i,K) \leq \beta - |s_r| = 3 - 2 = 1$ for $i = 1,2,3$. Thus, one of the 2s is removed, resulting $D_{occ(2,S)} = [0,1]$ and $D_S = [\{\!\{1,1\}\!\}, \{\!\{1,1,1,2,3\}\!\}]$.

(2) Inferences between $occ(i,S)$ and V_S

The variety V_S must always remain inside the limits given by the multiset bounds. This is generalized from the inferences between $occ(i,S)$ and C_S [Azevedo and Barahona, 2000].

$$(S \text{ changed bounds})\ \frac{\alpha = \|s_r\|, \beta = \|s_p\|}{\{\} \mapsto \{V_S \geq \alpha, V_S \leq \beta\}}$$

The occurrence of each element $occ(i,S)$ will be updated only when V_S is bound and equals either $\|s_r\|$ or $\|s_p\|$. When the variety V_S is fixed and equals $\|s_r\|$, any elements i that are not in s_r (i.e, $occ_r(i,S) = 0$) have to be removed (i.e., $occ(i,S) = 0$ for those i). On the other hand, if $V_S = \|s_p\|$, then each element in s_p (i.e, $occ_p(i,S) > 0$) must occur at least once in S (i.e., $occ(i,S) > 0$ for those i).

$$(V_S \text{ is bound})\ \frac{V_S = \|s_r\|, occ_r(i,S)) = 0}{\{\} \mapsto \{occ(i,S) = 0\}} \quad \frac{V_S = \|s_p\|, occ_p(i,S)) > 0}{\{\} \mapsto \{occ(i,S) > 0\}}$$

For example, consider a multiset variable S where $D_S = [\{\!\{1,1\}\!\}, \{\!\{1,1,2,2,3\}\!\}]$ (i.e., $D_{occ(1,S)} = [2,2]$, $D_{occ(2,S)} = [0,2]$, $D_{occ(3,S)} = [0,1]$), $D_{C_S} = [2,5]$, and V_S is bound to 1. Since $V_S = \|s_r\|$ (i.e., $1 = \|\{\!\{1,1\}\!\}\|$) and the elements 2 and 3 are not yet in s_r, they will not exist in S, resulting $occ(2,S) = occ(3,S) = 0$.

Consider the same multiset variable S but V_S is now bound to 3. Since $V_S = \|s_p\|$ (i.e., $3 = \|\{\!\{1,1,2,2,3\}\!\}\|$) and the elements 2 and 3 are not yet in s_r, at least one occurrence of 2 and 3 has to be added to their lower bound, resulting $occ_r(2, S) = occ_r(3, S) = 1$. Here, $occ_r(1, S)$ remains unchanged because the element 1 is already in its lower bound.

(3) Inferences between C_S and V_S

The variety must always be smaller than or equal to the cardinality at both limits because cardinality counts same elements but variety does not.

$$(C_S \text{ changed upper bound}) \; \frac{\beta = c_p}{\{\} \mapsto \{V_S \leq \beta\}} \qquad (V_S \text{ changed lower bound}) \; \frac{\alpha = v_r}{\{\} \mapsto \{C_S \geq \alpha\}}$$

(4) Inferences among $occ(i, S)$, C_S, and V_S

When any two of occurrences $occ(i, S)$, cardinality C_S, and variety V_S change their bounds, the remaining one has to be updated as well. This kind of inferences lead to stronger constraint propagation than those between the pairwise ones (i.e., between $occ(i, S)$ and C_S, between $occ(i, S)$ and V_S, and between C_S and V_S).

When the occurrences $occ(i, S)$ and the variety V_S change their bounds, the cardinality C_S will be adjusted accordingly to fulfill the requirements on V_S based on the elements existing in $s_p \setminus s_r$.

$$(S \text{ or } V_S \text{ changed bounds}) \; \frac{\alpha = v_r, \beta = v_p}{\{\} \mapsto \{C_S \geq |s_r| + (\alpha - \|s_r\|), C_S \leq |s_r| + a\}}$$
where $a = \max(|b| : b \subseteq (s_p \setminus s_r) \wedge \|b \uplus s_r\| = \beta)$.

For example, consider a multiset variable S which updates its bounds to $D_S = [\{\!\{1, 1\}\!\}, \{\!\{1, 1, 1, 2, 2, 3\}\!\}]$ (i.e., $D_{occ(1,S)} = [2, 3]$, $D_{occ(2,S)} = [0, 2]$, $D_{occ(3,S)} = [0, 1]$), $D_{C_S} = [2, 6]$, and $D_{V_S} = [2, 3]$. Since S must contain at least two different kinds of elements, besides element 1, either element 2 or 3 has to be included in S. This lead to an increase in c_r although the exact addition has not yet taken place. Based on the inference rule, $C_S \geq |s_r| + (\alpha - \|s_r\|) = 2 + 2 - 1 = 3$. Thus, D_{C_S} is updated to $[3, 6]$.

To find a, the subset $s_p \setminus s_r$, which fulfills the condition $\|b \uplus s_r\| = \beta$, is first extracted. The possible elements are then ordered. Thus, the complexity is bounded by the sorting procedure $O(n \log n)$, where n is the number of distinct elements in S.

Similarly, when the occurrences $occ(i, S)$ and the cardinality C_S change their bounds, the variety V_S will be adjusted accordingly to fulfill the requirements on C_S based on the elements existing in $s_p \setminus s_r$.

$$(S \text{ or } C_S \text{ changed bounds}) \; \frac{\alpha = c_r, \beta = c_p}{\{\} \mapsto \{V_S \geq a, V_S \leq \|s_r\| + c\}}$$
where $a = \min(\|s_r \uplus b\| : b \subseteq (s_p \setminus s_r) \wedge |b \uplus s_r| = \alpha)$, and $c = \max(\|d\| : d \subseteq (s_p \setminus s_r) \wedge \|d \uplus s_r\| > \|s_r\| \wedge |d \uplus s_r| = \beta)$.

For example, consider a multiset variable S which updates its bounds to $D_S = [\{\!\{1, 1\}\!\}, \{\!\{1, 1, 1, 2, 2, 3\}\!\}]$ (i.e., $occ(1, S) = [2, 3]$, $occ(2, S) = [0, 2]$, $occ(3, S) = [0, 1]$), $D_{V_S} = [1, 3]$, and c_r is updated from 2 to 4 (i.e., $D_{C_S} = [4, 6]$). Here, S must contain at least four elements. With the current v_r, S can only have at most three 1s and one more element is needed to reach c_r. Other elements, which lead to a minimal change in v_r, are selected from their upper bounds. Based on the inference rule, b refers to a subset of $(s_p \setminus s_r)$ where $D_{(s_p \setminus s_r)} = [\emptyset, \{\!\{1, 2, 2, 3\}\!\}]$ and the cardinality of $b \uplus s_r$ equals c_r (i.e., 4). Thus, v_r equals the minimum variety of the union of s_r and a possible b (i.e., $\|\{\!\{1, 1\}\!\} \cup \{\!\{1, 2\}\!\}\| = 2$). V_S is updated to $[2, 3]$.

a and c can be obtained using the same way as finding a in the previous inference rule, but with different conditions. Thus, the complexity for this inference rule as a whole is also bounded by the sorting procedure $O(n \log n)$, where n is the number of distinct elements in S.

When the cardinality C_S is fixed and equals either $|s_r|$ or $|s_p|$, S can be set to the corresponding bound (i.e., all occurrences $occ(i, S)$ can be fixed) and V_S can also be bound accordingly.

$$(C_S \text{ is bound}) \; \frac{C_S = |s_r|}{\{\} \mapsto \{S = s_r, V_S = \|s_r\|\}} \qquad \frac{C_S = |s_p|}{\{\} \mapsto \{S = s_p, V_S = \|s_p\|\}}$$

When both the cardinality C_S and the variety V_S are bound to the same value α, S degenerates to a set. Thus, the occurrence of each element $occ(i, S)$ will be at most one.

$$(C_S \text{ and } V_S \text{ are bound and equal}) \; \frac{C_S = V_S = \alpha, occ_p(i, S) > 1}{\{\} \mapsto \{occ(i, S) \leq 1\}}$$

(5) Failure

A failure can be detected when any one of the conditions is true: (1) the lower bound s_r is not included in the upper bound s_p; or (2) the domain of the cardinality variable D_{C_S} becomes empty; or (3) the domain of the variety variable D_{V_S} becomes empty.

$$(S, C_S, \text{ or } V_S \text{ changed bounds}) \; \frac{\text{not}(s_r \subseteq s_p)}{\{\} \mapsto \text{fail}} \quad \frac{D_{C_S} = \emptyset}{\{\} \mapsto \text{fail}} \quad \frac{D_{V_S} = \emptyset}{\{\} \mapsto \text{fail}}$$

The inference rules described in this subsection are incomplete in the sense that they only enforce bounds consistency on the component variables. They do not enforce the strongest possible consistency, as it is intractable in general.

Theorem 1. *Enforcing GAC on a multiset variable consisting of occurrence, cardinality, and variety variables is NP-hard.*

Proof. Enforcing GAC on any general constraints on integer variables is NP-hard [Bessiere *et al.*, 2007]. An integer variable is a special case of a multiset variable, which degenerates to an integer variable when both cardinality and variety equal 1. Hence the result. □

In fact, our primary aim is not for completeness, but for inference rules that are efficiently implementable. Nonetheless, the inference rules as a whole maintain more than multiset bounds consistency.

Theorem 2. *The inference rules (1) between $occ(i, S)$ and C_S, (2) between $occ(i, S)$ and V_S, (3) between C_S and V_S, (4) among $occ(i, S)$, C_S, and V_S, and (5) for failure collectively enforce a consistency level strictly stronger than multiset bounds consistency.*

Proof. Due to space limitations, we skip the proof that our inference rules are at least as strong as multiset bounds consistency. For strictness, the two examples under "(4) Inferences among $occ(i, S)$, C_S, and V_S" show that given a domain which is already multiset bounds consistent, the inference rules can further tighten the bounds of C_S or V_S. Hence the result. □

3.4 Multiset Constraints

The previous subsection describes the inferences within one multiset variable. In this subsection, we focus on propagation that occurs across different multiset variables. We give some constraint propagation rules that enforce bounds consistency on some common multiset constraints. Performing inferences on the cardinality and variety variables are known as *cardinality reasoning* and *variety reasoning* respectively. For each multiset constraint, we use an example to show how they are useful in increasing constraint propagation. In the rules, the changes in the constraint store involving the cardinality variables are adopted from Azevedo and Barahona [2000], but those involving the variety variables are more generalized.

Equality Constraint ($X = Y$)
If X and Y are told to be equal, then their cardinalities and varieties are also equal respectively.

$$\{X = Y\} \mapsto \{occ(i, X) = occ(i, Y), C_X = C_Y, V_X = V_Y\}$$

For example, consider the equality constraint $X = Y$, where $n = 3$, $D_{occ(1,X)} = [0, 2]$, $D_{occ(2,X)} = [0, 2]$, $D_{occ(3,X)} = [0, 2]$, $D_{C_X} = [4, 4]$, $D_{V_X} = [2, 2]$, $D_{occ(1,Y)} = [0, 2]$, $D_{occ(2,Y)} = [0, 2]$, $D_{occ(3,Y)} = [0, 2]$, $D_{C_Y} = [4, 4]$, and $D_{V_Y} = [3, 3]$. Without the variety variables V_X and V_Y (and thus without variety reasoning), there are no prunings available. However, with variety reasoning, the problem fails immediately because when $X = Y$ (i.e., $occ(i, X) = occ(i, Y)$ for all elements i), $V_X = V_Y$ is obviously violated.

Subset Constraint ($X \subseteq Y$)
If Y contains X, then C_Y is greater than or equal to C_X, and V_Y is also greater than or equal to V_X.

$$\{X \subseteq Y\} \mapsto \{occ(i, X) \leq occ(i, Y), C_X \leq C_Y, V_X \leq V_Y\}$$

Consider the subset constraint $X \subseteq Y$, where $D_X = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3, 3\}\!\}]$ with cardinality 5 and variety 3 (i.e., $D_{occ(1,X)} = [0, 2]$, $D_{occ(2,X)} = [0, 2]$, $D_{occ(3,X)} = [0, 3]$), $D_{C_X} = [5, 5]$, $D_{V_X} = [3, 3]$, and $D_Y = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3, 3\}\!\}]$ with cardinality 5 and variety 2 (i.e., $D_{occ(1,Y)} = [0, 2]$, $D_{occ(2,Y)} = [0, 2]$, $D_{occ(3,Y)} = [0, 3]$), $D_{C_Y} = [5, 5]$, $D_{V_Y} = [2, 2]$. With variety reasoning, the problem fails immediately because V_X can never be smaller than or equal to V_Y (i.e., $3 \not\leq 2$). Again, without variety reasoning, there are no available prunings.

Union Constraint ($X \cup Y = Z$)
Union takes the maximum number of occurrences of each element. When Z is the union of X and Y, $occ(i, Z) = \max(occ(i, X), occ(i, Y))$ for all elements i. C_Z (resp. V_Z) is smaller than or equal to $C_X + C_Y$ (resp. $V_X + V_Y$). On the other hand, the lower bound of C_Z (resp. V_Z) can be obtained from the maximum of the following two cases: (1) Suppose S_Z contains S_X (i.e., $S_X \subseteq S_Z$), S_Z will have at least C_X elements (resp. V_X distinct elements). We can safely add the elements which appear in S_Y but not in S_X (i.e., $y_r \setminus x_p$) to S_Z because S_Z is the multiset union and it takes all elements in both S_X and S_Y. Thus, $C_Z \geq C_X + |y_r \setminus x_p|$ (resp. $V_Z \geq V_X + \|y_r \setminus x_p\|$). (2) Similarly, we can add the elements in $(x_r \setminus y_p)$ to S_Z if S_Z contains S_Y. Thus, $C_Z \geq C_Y + |x_r \setminus y_p|$ (resp. $V_Z \geq V_Y + \|x_r \setminus y_p\|$).

$$\begin{aligned}
&\{X \cup Y = Z\} \\
\mapsto \quad &\{occ(i, Z) = \max(occ(i, X), occ(i, Y)), \\
&occ(i, X) \leq occ(i, Z), occ(i, Y) \leq occ(i, Z), \\
&C_Z \leq C_X + C_Y, V_Z \leq V_X + V_Y, \\
&C_Z \geq \max(C_X + |y_r \setminus x_p|, C_Y + |x_r \setminus y_p|), \\
&V_Z \geq \max(V_X + \|y_r \setminus x_p\|, V_Y + \|x_r \setminus y_p\|)\}
\end{aligned}$$

Consider the union constraint $X \cup Y = Z$, where $D_X = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3\}\!\}]$ (i.e., $D_{occ(1,X)} = [0, 2]$, $D_{occ(2,X)} = [0, 2]$, $D_{occ(3,X)} = [0, 2]$), $D_{C_X} = [1, 2]$, $D_{V_X} = [1, 1]$, $D_Y = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3\}\!\}]$, (i.e., $D_{occ(1,Y)} = [0, 2]$, $D_{occ(2,Y)} = [0, 2]$, $D_{occ(3,Y)} = [0, 2]$), $D_{C_Y} = [1, 2]$, $D_{V_Y} = [1, 1]$, and $D_Z = [\{\!\{1, 2, 3\}\!\}, \{\!\{1, 1, 2, 2, 3, 3\}\!\}]$, (i.e., $D_{occ(1,Z)} = [1, 2]$, $D_{occ(2,Z)} = [1, 2]$, $D_{occ(3,Z)} = [1, 2]$), $D_{C_Z} = [3, 6]$, $D_{V_Z} = [3, 3]$. With variety reasoning, the problem fails immediately because V_Z can never be smaller than or equal to the sum of V_X and V_Y. Without reasoning on the three variety variables, the problem will not fail even when $3 \not\leq 1 + 1$.

Union-Plus Constraint ($X \uplus Y = Z$)
When Z is the union-plus of X and Y, C_Z equals $C_X + C_Y$ because union-plus sums up all the elements in both X and Y. For all elements i, $occ(i, Z) = occ(i, X) + occ(i, Y)$. However, V_Z is smaller than or equal to $V_X + V_Y$ because X and Y can contain the same kind of elements (i.e., $\|X\| + \|Y\| \neq \|X \uplus Y\|$). For the lower bound of V_Z, it can be obtained in the same way as in the union constraint.

$$\begin{aligned}
&\{X \uplus Y = Z\} \\
\mapsto \quad &\{occ(i, Z) = occ(i, X) + occ(i, Y), \\
&occ(i, X) \leq occ(i, Z), occ(i, Y) \leq occ(i, Z), \\
&C_Z = C_X + C_Y, V_Z \leq V_X + V_Y, \\
&V_Z \geq \max(V_X + \|y_r \setminus x_p\|, V_Y + \|x_r \setminus y_p\|)\}
\end{aligned}$$

Consider a union-plus constraint $X \uplus Y = Z$, where $D_X = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3\}\!\}]$ (i.e., $D_{occ(1,X)} = [0, 2]$, $D_{occ(2,X)} = [0, 2]$, $D_{occ(3,X)} = [0, 2]$), $D_{C_X} = [1, 2]$, $D_{V_X} = [1, 1]$, $D_Y = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3\}\!\}]$ (i.e., $D_{occ(1,Y)} = [0, 2]$, $D_{occ(2,Y)} = [0, 2]$, $D_{occ(3,Y)} = [0, 2]$), $D_{C_Y} = [1, 2]$, $D_{V_Y} = [1, 1]$, and $D_Z = [\{\!\{1, 2, 3\}\!\}, \{\!\{1, 1, 2, 2, 3, 3\}\!\}]$ (i.e., $D_{occ(1,Z)} = [1, 2]$, $D_{occ(2,Z)} = [1, 2]$, $D_{occ(3,Z)} = [1, 2]$), $D_{C_Z} = [3, 6]$, $D_{V_Z} = [3, 3]$. Variety reasoning fails the problem immediately because V_Z can never be smaller than or equal to the sum of V_X and V_Y. Without reasoning on the three variety variables, the problem will not fail even when $3 \not\leq 1 + 1$.

Intersection Constraint ($X \cap Y = Z$)
If Z is the intersection of X and Y, then C_Z is smaller than or equal to both C_X and C_Y, and V_Z is also smaller than or equal to both V_X and V_Y. This is because intersection takes the minimum number of occurrence of each element between X and Y (i.e., $occ(i, Z) = \min(occ(i, X), occ(i, Y))$ for all elements i). The upper bound of C_Z (resp. V_Z) can be obtained from the minimum of the following two cases. (1) For the elements existing only in x_r but not in y_p (i.e., $x_r \setminus y_p$), they must not be part of the intersection. We can safely subtract these elements from C_X (resp. subtract these kinds of elements from V_X), resulting $C_Z \geq C_X - |x_r \setminus y_p|$ (resp. $V_Z \geq V_X - \|x_r \setminus y_p\|$). (2) Similarly, we can subtract the elements that exist in y_r but not in x_p (i.e., $y_r \setminus x_p$) from C_Y (resp. subtract these kinds of elements from V_Y), resulting $C_Z \geq C_Y - |y_r \setminus x_p|$ (resp. $V_Z \geq V_Y - \|y_r \setminus x_p\|$).

$$\begin{array}{ll} \multicolumn{2}{l}{\{X \cap Y = Z\}} \\ \mapsto & \{occ(i, Z) = \min(occ(i, X), occ(i, Y)), \\ & occ(i, Z) \leq occ(i, X), occ(i, Z) \leq occ(i, Y), \\ & C_Z \leq \min(C_X, C_Y), V_Z \leq \min(V_X, V_Y), \\ & C_Z \geq \min(C_X - |x_r \setminus y_p|, C_Y - |y_r \setminus x_p|), \\ & V_Z \geq \min(V_X - \|x_r \setminus y_p\|, V_Y - \|y_r \setminus x_p\|)\} \end{array}$$

Consider an intersection constraint $X \cap Y = Z$, where $D_X = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3, 3\}\!\}]$ (i.e., $D_{occ(1,X)} = [0, 2]$, $D_{occ(2,X)} = [0, 2]$, $D_{occ(3,X)} = [0, 3]$), $D_{C_X} = [1, 3]$, $D_{V_Z} = [1, 1]$, $D_Y = [\emptyset, \{\!\{1, 1, 2, 2, 3, 3, 3\}\!\}]$ (i.e., $D_{occ(1,X)} = [0, 2]$, $D_{occ(2,X)} = [0, 2]$, $D_{occ(3,X)} = [0, 3]$), $D_{C_Y} = [1, 3]$, $D_{V_Y} = [1, 1]$, and $D_Z = [\emptyset, \{\!\{1, 2, 3, 3, 3\}\!\}]$ (i.e., $D_{occ(1,X)} = [0, 1]$, $D_{occ(2,X)} = [0, 1]$, $D_{occ(3,X)} = [0, 3]$), $D_{C_Z} = [2, 4]$, $D_{V_Z} = [2, 2]$. With variety reasoning, the problem fails immediately because V_Z can never be smaller than or equal to both V_X and V_Y. The problem will not fail without variety reasoning even when $2 \not\leq 1$.

4 Experimental Results

To verify the feasibility and efficiency of our proposal, we implement our multiset variable representation, the inference rules, and the multiset constraints in ILOG Solver 6.0 [ILOG, 2003]. We use the extended Steiner system and the template design problem as the benchmark problems. While the standard Steiner system is only set-based, the extended version is an important and practical multiset problem in the area of information retrieval [Johnson and Mendelsohn, 1972; Bennett and Mendelsohn, 1980; Park and Blake, 2008]. Solving the extended Steiner system can provide solutions to the problem of a multiset batch code.

The extended Steiner system $ES(t, k, v)$ is a collection of b blocks. Each block is a k-element multiset drawn from a v-element set whose elements can be drawn multiple times. For every two blocks in the collection, the cardinality of their intersection must be smaller than t. For example, one possible solution for $ES(2, 3, 3)$ in 3 blocks is $\{\{\!\{1, 1, 2\}\!\}, \{\!\{2, 2, 3\}\!\}, \{\!\{3, 3, 1\}\!\}\}$. The intersection of $\{\!\{1, 1, 2\}\!\}$ and $\{\!\{2, 2, 3\}\!\}$ is $\{\!\{2\}\!\}$; the intersection of $\{\!\{1, 1, 2\}\!\}$ and $\{\!\{3, 3, 1\}\!\}$ is $\{\!\{1\}\!\}$; the intersection of $\{\!\{2, 2, 3\}\!\}$ and $\{\!\{3, 3, 1\}\!\}$ is $\{\!\{3\}\!\}$. All of them have size smaller than $t = 2$. In our experiments, we adapt the extended Steiner system to an optimization problem which maximizes the sum of the varieties of the multisets in a solution. To further increase problem difficulty, we also constrain each multiset to have at least certain varieties.

		SB		SB+CR		SB+CR+VR	
(t,k,v)	b	fails	runtime	fails	runtime	fails	runtime
(2, 3, 4)	2	307	0.01	83	**0**	**20**	**0**
	3	2266	0.04	518	0.02	**135**	**0**
	4	9530	0.2	2232	0.1	**737**	**0.04**
(2, 3, 5)	2	800	0.01	222	**0.01**	**12**	0.02
	3	13812	0.23	3079	0.11	**156**	**0.01**
	4	166064	3.56	33679	1.58	**3539**	**0.21**
	5	1185644	31.37	244547	14.41	**39930**	**3**
	6	4744639	152.77	1095106	78.97	**244430**	**22.67**
(2, 4, 4)	2	867	0.01	204	**0**	**70**	**0**
	3	6246	0.1	1330	0.04	**581**	**0.03**
	4	20425	0.42	4980	0.21	**2757**	**0.18**
(2, 4, 5)	2	2800	0.04	638	0.03	**202**	**0**
	3	64458	1.12	12636	0.46	**4603**	**0.21**
	4	627704	14.13	124611	5.91	**57329**	**3.66**
	5	2800951	79.37	637199	38.06	**356785**	**29.27**

Table 2: Maximization results of the extended Steiner systems. The variety of each multiset is at least 2.

		SB		SB+CR		SB+CR+VR	
(t,k,v)	b	fails	runtime	fails	runtime	fails	runtime
(2, 3, 6)	2	1751	0.02	480	0.01	**8**	**0**
	3	14895	0.28	2992	0.12	**13**	**0.01**
	4	57449	1.33	9548	0.49	**17**	**0**
(2, 4, 6)	2	6072	0.09	1219	0.04	**204**	**0.02**
	3	184841	3.55	33165	1.34	**11709**	**0.59**
	4	848172	20.62	132400	7.04	**56762**	**3.89**
(3, 4, 4)	2	446	0.02	160	0.02	**26**	**0**
	3	2549	0.04	793	0.02	**135**	**0.01**
	4	9615	0.18	2646	0.11	**634**	**0.04**
(3, 4, 5)	2	1475	0.02	537	**0.01**	**67**	**0.01**
	3	35582	0.58	10913	0.4	**1831**	**0.07**
	4	591668	12.07	160724	7.65	**29938**	**1.66**
	5	6565175	160.67	1630805	96.76	**312397**	**22.7**
	6	48187790	1370.32	11387223	812	**2108410**	**194.99**
	7	-	-	-	-	**9813128**	**1125.01**
(3, 4, 6)	2	4122	0.05	1537	0.05	**24**	**0.01**
	3	80815	1.4	27386	1.17	**63**	**0.01**
	4	3994239	87.12	1187476	66.63	**4134**	**0.25**
	5	-	-	48272955	3358.97	**370386**	**26.52**
	6	-	-	-	-	**17854829**	**1562.45**

Table 3: Maximization results of the extended Steiner systems. The variety of each multiset is at least 3.

The experiments are run on a Sun Blade 2500 (2×1.6GHz US-IIIi) workstation with 2GB memory. We report the number of fails (i.e., the number of backtracks occurred in solving a model) and CPU time in seconds to find and prove the optimal solution for each instance. Comparisons are made among set bounds consistency (SB), set bounds consistency with cardinality reasoning (SB+CR), and set bounds consistency with cardinality and variety reasoning (SB+CR+VR) proposed in this paper. We use the naive approach mentioned in Section 3.2 to model the cardinality variables in SB, and the variety variables in both SB and SB+CR. The meta-constraints are enforced by the built-in propagation algorithms in ILOG Solver instead of the inference rules. In the tables, the best number of fails and CPU time among the results for each instance are highlighted in bold. A cell labeled with "-" denotes a timeout after one hour.

Tables 2 and 3 show the experimental results of the maximization and the variety of each multiset is at least 2 and 3 respectively. Among the three propagation approaches, SB+CR+VR always achieves the fewest number of fails. There are more than 90% reduction in number of fails when compared to SB alone, and more than 50% reduction when compared to SB+CR. This confirms that variety (and cardinality) reasoning is highly effective in reducing search space. The extra prunings are so significant that they compensate the overhead of extra computational effort spent for variety (and cardinality) reasoning. For runtime, SB+CR+VR is also always the fastest, although the proportion of reduction is less than that for the number of fails. The reduction of SB+CR+VR over SB+CR in Table 2 is moderate, but that in Table 3 is significant. There are even instances in which both SB and SB+CR cannot finish execution within the time limit, but SB+CR+VR can. This also shows that the usefulness of variety reasoning sometimes depends on the tightness of the variety constraints in a problem.

For the template design problem, each multiset variable represents a template and its domain values are the possible combinations of designs which allow repetitions. Like the extended Steiner system, we further impose a restriction to constrain the varieties of each multiset. Due to space limitations, we do not show the tables, but from the experimental results, SB+CR+VR always achieves the fewest number of fails. There are at least 20% reduction in the number of fails when compared to SB and SB+CR. When the problem instance has no solutions, enforcing SB+CR+VR can even reduce the search space by up to 70%. The savings in runtime, however, is not as significant as those in the extended Steiner system. The performance difference of SB+CR+VR between the satisfiable and unsatisfiable instances is an interesting phenomenon which we are still investigating.

Note that our current implementation of the multiset variable representation and the rules is only prototypical. There are still rooms for improvement. For example, it is known that adjusting the triggering order of the rules (depending on the computational cost of the rules) can affect the performance [Azevedo and Barahona, 2000]. We expect that our implementation can be optimized in the future.

5 Related Work

Conjunto [Gervet, 1997] is the first constraint solver developed in which a set variable is represented by set intervals. Other set constraint solvers include Oz [Müller and Müller, 1997], Mozart [Müller, 2001], and ROBDD [Lagoon and Stuckey, 2004; Hawkins *et al.*, 2005]. Azevedo and Barahona [2000] further proposed *cardinality reasoning* on set constraints and developed another set constraint solver, *Cardinal*, to handle the set cardinality more actively and improve the performance in solving CSPs with set variables.

Kiziltan and Walsh [2002; 2003] suggested three multiset representations: bounds, occurrence, and fixed cardinality representations. They proved that occurrence representation is more expressive than bounds representation when maintaining domain consistency, and is as expressive as when maintaining bounds consistency. Fixed cardinality representation is incomparable to the other two. All three of them are compact but cannot represent all forms of disjunctions.

6 Conclusion

We have introduced cardinality and variety variables to multiset variables based on the occurrence representation. While the introduction of cardinality variable is a straightforward generalization to the set variable counterpart, the idea of variety variable is a new concept. We have exploited the variety property to introduce new inference rules to increase pruning opportunities, and improved the propagation of some common multiset constraints through variety reasoning.

There can be plenty of scope for future research for multiset CSPs. An example is to perform variety reasoning on other multiset global constraints like the disjoint and partition constraints. Other multiset-specific properties can also be incorporated into multiset variable representations to even further enhance constraint propagation.

References

[Azevedo and Barahona, 2000] F. Azevedo and P. Barahona. Modelling digital circuits problems with set constraints. In *Proc. of CL'00*, pages 414–428, 2000.

[Bennett and Mendelsohn, 1980] F.E. Bennett and E. Mendelsohn. Extended (2, 4)-designs. *J. Comb. Theory, Ser. A*, 29(1):74–86, 1980.

[Bessiere *et al.*, 2007] C. Bessiere, E. Hebrard, B. Hnich, and T. Walsh. The complexity of reasoning with global constraints. *Constraints*, 12(2):239–259, 2007.

[Gervet, 1997] C. Gervet. Interval propagation to reason about sets: Definition and implementation of a practical language. *Constraints*, 1(3):191–244, 1997.

[Hawkins *et al.*, 2005] P. Hawkins, V. Lagoon, and P.J. Stuckey. Solving set constraint satisfaction problems using ROBDDs. *JAIR*, 24:109–156, 2005.

[ILOG, 2003] ILOG. *ILOG Solver 6.0 Reference Manual*, 2003.

[Johnson and Mendelsohn, 1972] D.M. Johnson and N.S. Mendelsohn. Extended triple systems. *Aequat. Math.*, 8(3):291–298, 1972.

[Kiziltan and Walsh, 2002] Z. Kiziltan and T. Walsh. Constraint programming with multisets. In *Proc. of SymCon'02*, 2002.

[Lagoon and Stuckey, 2004] V. Lagoon and P.J. Stuckey. Set domain propagation using ROBDDs. In *Proc. of CP'04*, pages 347–361, 2004.

[Müller and Müller, 1997] T. Müller and M. Müller. Finite set constraints in Oz. In *13. Workshop Logische Programmierung*, pages 104–115, 1997.

[Müller, 2001] T. Müller. *Constraint Propagation in Mozart*. Doctoral dissertation, Universität des Saarlandes, Germany, 2001.

[Park and Blake, 2008] E.Y. Park and I. Blake. Construction of extended steiner systems for information retrieval. *Rev. Mat. Complut.*, 21(1):179–190, 2008.

[Walsh, 2003] T. Walsh. Consistency and propagation with multiset constraints: A formal viewpoint. In *Proc. of CP'03*, pages 724–738, 2003.