A Generalization of Generalized Arc Consistency: From Constraint Satisfaction to Constraint-Based Inference

Le Chang and Alan K. Mackworth

Department of Computer Science, University of British Columbia
2366 Main Mall, Vancouver, B.C. Canada V6T 1Z4
{lechang, mack}@cs.ubc.ca

Abstract. Arc consistency and generalized arc consistency are two of the most important local consistency techniques for binary and non-binary classic constraint satisfaction problems (CSPs). Based on the Semiring CSP and Valued CSP frameworks, arc consistency has also been extended to handle soft constraint satisfaction problems such as fuzzy CSP, probabilistic CSP, max CSP, and weighted CSP. This extension is based on an idempotent or strictly monotonic constraint combination operator. In this paper, we present a weaker condition for applying the generalized arc consistency approach to constraint-based inference problems other than classic and soft CSPs. These problems, including probability inference and maximal likelihood decoding, can be processed using generalized arc consistency enforcing approaches. We also show that, given an additional monotonic condition on the corresponding semiring structure, some of constraint-based inference problems can be approximately preprocessed using generalized arc consistency algorithms.

1 Introduction

The notion of local consistency plays a central role in constraint satisfaction. Given a constraint satisfaction problem (CSP), local consistency can be characterized as deriving new constraints based on local information. The derived constraints simplify the representation of the original CSP without the loss of solutions. Among the family of local consistency enforcing algorithms or filtering algorithms, arc consistency [1] is one of the most important techniques for binary classic CSP. It is straightforward to extend it as generalized arc consistency [2, 3] to handle non-binary classic CSPs.

To represent over-constrained and preference-based problems in the real world, researchers in the constraint processing community are increasingly interested in so-called soft constraint satisfaction problems. Fuzzy CSP, probabilistic CSP, max CSP, and weighted CSP have been proposed to address these requirements. Semiring CSP [4] and Valued CSP [5] are two of the most widely studied generalized frameworks. Based on the two frameworks, arc consistency is also extended as soft arc consistency to handle soft constraints [6, 7]. The soundness and completeness of soft arc consistency, within the Semiring CSP framework, relies on the idempotency of the constraint combination operator. Moreover, the c-semiring used in the Semiring CSP framework has the special requirement of idempotency of the additive operator. The Valued CSP framework extends soft arc consistency in the Semiring CSP framework. Soft arc consistency in Valued CSP depends on the strictly monotonic constraint combination operator or the fair valuation structure.

For most soft constraint proposals, the success of soft arc consistency in the Semiring CSP framework and the Valued CSP framework has been proven [6, 7]. However, because of the requirements of idempotent constraint combination operators or the fair valuation structure, soft arc consistency does not work on some problems with representations analogical to hard or soft CSPs. For example, the probability assessment problem in Bayesian networks can be represented by a commutative semiring $\mathbf{S_{prob}} = \langle \mathbb{R}^+ \cup \{0\}, +, \times \rangle$ [8]. Both the additive operator $+$ and the multiplicative operator $\times$ are not idempotent, so the problem can not be embedded into the Semiring CSP framework. The neutral (or identity) element 1 is not the minimum element of $\mathbb{R}^+$ and there exists no maximum difference of any non-zero element $\beta \in \mathbb{R}^+$ and 0 w.r.t. the constraint combination operator $\times$, so the problem can not be solved by soft arc consistency in the Valued CSP framework.

Given the representation analogues of constraint-based inference (CBI) problems, including probabilistic inferences, decision-making under uncertainty, constraint satisfaction problems, propositional satisfiability, decoding problems, and possibility inferences, we present in this paper a weaker condition for applying local consistency approaches to general constraint-based inference problems based on the commutative semiring structure. The weaker condition proposed here depends only on the existence and property of the combination absorbing element and does not depend on other semiring properties. We show that arc consistency and its variants, the most widely studied local consistency approaches, can be extended to handle problems from other research fields without modification, if the condition is satisfied. We also show that, by satisfying an additional monotonic condition on the semiring structure characterizing the problem, generalized arc consistency can also be used as an approximate local consistency enforcing technique for constraint-based inference problems.

2 Background

There are two essential operators in real world CBI problems: (1) combination, which corresponds to an aggregation of constraints, and (2) marginalization, which corresponds to focusing of a specified constraint to a narrow domain. These two operators allow us to use algebraic structures to generalize CBI problem representations. More specifically, both the abstract CBI representation framework and the generalized arc consistency approach in this paper are based on the semiring structure, an important notion in abstract algebra. This section introduces the definition of the semiring and related properties.

Definition 1 (Semiring). *Let $\mathbf{A}$ be a set. Let $\oplus$ and $\otimes$ be two closed binary operators defined on $\mathbf{A}$. Here we define operator $\otimes$ as taking precedence over operator $\oplus$. $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ is a semiring if the operators satisfy the following axioms:*

- *Additive associativity: $\forall a, b, c \in \mathbf{A}$, $(a \oplus b) \oplus c = a \oplus (b \oplus c)$;*
- *Additive commutativity: $\forall a, b \in \mathbf{A}$, $a \oplus b = b \oplus a$;*
- *Multiplicative associativity: $\forall a, b, c \in \mathbf{A}$, $(a \otimes b) \otimes c = a \otimes (b \otimes c)$;*
- *Left and right distributivity: $\forall a, b, c \in \mathbf{A}$, $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$ and $(b \oplus c) \otimes a = b \otimes a \oplus c \otimes a$.*

To capture computational properties of various inference approaches, we use commutative semiring, an extended algebraic notion of semiring, to formally represent CBI problems in this paper.

Definition 2 (Commutative Semiring). *A commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ is a semiring that satisfies the following additional conditions:*

- *Multiplicative commutativity: $\forall a, b \in \mathbf{A}$, $a \otimes b = b \otimes a$;*
- *Multiplicative identity: there exists a multiplicative identity element $\mathbf{1} \in \mathbf{A}$, such that $a \otimes \mathbf{1} = \mathbf{1} \otimes a = a$ for any $a \in \mathbf{A}$;*
- *Additive identity: there exists an additive identity element $\mathbf{0} \in \mathbf{A}$, such that $a \oplus \mathbf{0} = \mathbf{0} \oplus a = a$ for any $a \in \mathbf{A}$;*

We will show in the following sections that the application of local consistency techniques depend on the existence of a multiplicative (or combination) absorbing element. It is easy to show the uniqueness of the multiplicative absorbing element given the multiplicative commutativity of a commutative semiring, according the definition below.

Definition 3 (Multiplicative Absorbing Element). *An element $\alpha_\otimes \in \mathbf{A}$ is the multiplicative absorbing element of a commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ if $e \otimes \alpha_\otimes = \alpha_\otimes \otimes e = \alpha_\otimes$ for any element $e \in \mathbf{A}$.*

Similarly the additive absorbing element $\alpha_\oplus$ is defined as:

Definition 4 (Additive Absorbing Element). *An element $\alpha_\oplus \in \mathbf{A}$ is the additive absorbing element of a semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ if $e \oplus \alpha_\oplus = \alpha_\oplus \oplus e = \alpha_\oplus$ for any element $e \in \mathbf{A}$.*

Furthermore, we say that $\oplus$ is idempotent if $a \oplus a = a$, and $\otimes$ is idempotent if $a \otimes a = a$. For some semirings, we can define a partial order over the elements of S if $\oplus$ is idempotent.

Definition 5 (Partial Order $\leq_{\mathbf{S}}$ [7]). *Given a semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$, there exist a partial order $\leq_{\mathbf{S}}$ over S such that $a \leq_{\mathbf{S}} b$, $\forall a, b \in \mathbf{A}$ if:*

No.	S	$\oplus$,$\mathbf{0}$	$\otimes$,$\mathbf{1}$	$\alpha_\otimes$	$\alpha_\oplus$	Eliminative	Monotonic
1	$\{true, false\}$	$\vee$, $false$	$\wedge$, $true$	$false$	$true$	Yes	No
2	$[0, 1]$	max, 0	min, 1	0	1	Yes	Yes
3	$\mathbb{R}^+ \cup \{0\}$	max, 0	min, ∞	0	∞	Yes	Yes
4	$[0, 1]$	max, 0	$\times$, 1	0	1	Yes	Yes
5	$\mathbb{R}^- \cup \{0\}$	max, $-\infty$	$+$, 0	$-\infty$	0	Yes	Yes
6	$\mathbb{N}^+ \cup \{0\}$	max, 0	$+$, 0	∞	∞	No	Yes
7	$\mathbb{R}^+ \cup \{0\}$	$+$, 0	$\times$, 1	0	∞	Yes	Yes
8	$\mathbb{R}^+ \cup \{0\}$	max, 0	$\times$, 1	0	∞	Yes	Yes
9	$\mathbb{N}^+ \cup \{0\}$	min, ∞	$+$, 0	∞	0	Yes	Yes
10	$\mathbb{N}^+ \cup \{0\}$	min, ∞	$\times$, 1	∞	0	Yes	Yes

Table 1. Properties of Various Commutative Semirings

- *$\oplus$ is idempotent;*
- *$a \oplus b = b$.*

Given a partial order $\leq_{\mathbf{S}}$ of semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$, we know that the additive identity element $\mathbf{0}$ is the minimum element of the ordering. In other words, $\mathbf{0} \leq_{\mathbf{S}} a, \forall a \in \mathbf{A}$. If the additive absorbing $\alpha_\oplus$ exists, it will be the maximum element of the ordering according to the partial order definition. Also note that the two conditions are only sufficient conditions for the existence of a partial order. For example, the commutative semiring $\mathbf{S_{prob}} = \langle \mathbb{R}^+ \cup \{0\}, +, \times \rangle$ has a partial order while does not satisfy the two conditions.

Finally, we define two more important properties for some commutative semirings. The two properties are the foundation of applying local consistency techniques to general CBI problems.

Definition 6 (A Commutative Semiring is Eliminative). *A commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ is eliminative if:*

- *There exists the multiplicative absorbing element $\alpha_\otimes \in \mathbf{A}$;*
- *$\alpha_\otimes = \mathbf{0}$, in other words, the multiplicative absorbing element is equal to the additive identity element.*

Definition 7 (A Commutative Semiring is Monotonic). *A commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ is monotonic if:*

- *There exists a total order $\leq_{\mathbf{S}}$ on $\mathbf{A}$;*
- *The additive identity element $\mathbf{0}$ is the minimum element w.r.t. $\leq_{\mathbf{S}}$. In other words, $\mathbf{0} \leq_{\mathbf{S}} a$, $\forall a \in \mathbf{A}$;*
- *Additive Monotonic: $a \leq_{\mathbf{S}} b$ implies $a \oplus c \leq_{\mathbf{S}} b \oplus c$, $\forall a, b, c \in \mathbf{A}$;*
- *Multiplicative Monotonic: $a \leq_{\mathbf{S}} b$ implies $a \otimes c \leq_{\mathbf{S}} b \otimes c$, $\forall a, b, c \in \mathbf{A}$.*

Table 1 displays some commutative semirings with their identity and absorbing elements and properties.

In the following sections, we use bold letters to denote sets of elements and regular letters to denote individual elements. Given a set of elements $\mathbf{X}$ and an element $Z \in \mathbf{X}$, $\mathbf{X}_{-Z}$ denotes the set of elements $\mathbf{X} \setminus \{Z\}$. Given a value assignment $\mathbf{x}$ of variable subset $\mathbf{X}$ and $\mathbf{Y} \subseteq \mathbf{X}$, $\mathbf{x}_{\downarrow \mathbf{Y}}$ denotes the value assignment projection of $\mathbf{x}$ to the variable subset $\mathbf{Y}$.

3 A Semiring-Based Unifying Framework for CBI Problems

Constraint-Based Inference (CBI) is an umbrella term for various superficially different problems. It concerns the automatic discovery of new constraints from a set of given constraints over individual entities. New constraints reveal undiscovered properties about a set of entities or about the world of all entities. A constraint here is seen as a function that maps possible value assignments to a specific value domain. Many practical problems from different fields can be seen as constraint-based inference problems. These problems cover a wide range of topics in computer science research, which include probabilistic inferences, decision-making under uncertainty, constraint satisfaction problems (CSP), propositional satisfiability problems (SAT), decoding problems, and possibility inferences.

A CBI problem is defined in terms of a set of variables with values in finite domains and a set of constraints on these variables. We use commutative semirings to unify the representation of constraint-based inference problems from various disciplines into a single formal framework [9], based on the synthesis of the existing generalized representation frameworks [4, 10] and algorithmic frameworks [11, 12, 8] from different fields. Formally:

Definition 8 (Constraint-Based Inference (CBI) Problem). *A constraint-based inference (CBI) problem* $\mathbf{P}$ *is a tuple* $(\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ *where:*

- $\mathbf{X} = \{X_1, \cdots, X_n\}$ *is a set of variables;*
- $\mathbf{D} = \{\mathbf{D_1}, \cdots, \mathbf{D_n}\}$ *is a collection of finite domains, one for each variable;*
- $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ *is a commutative semiring;*
- $\mathbf{F} = \{f_1, \cdots, f_r\}$ *is a set of constraints. Each constraint is a function that maps value assignments of a subset of variables to values in* $\mathbf{A}$

More specifically, we use $Scope(f)$ to denote the subset of variables that is in the scope of the constraint f. We use $\mathbf{D}_X$ to denote the value domain of a variable X. Given a variable $X \in Scope(f)$, $Scope(f)_{-X}$ denotes the variable subset $Scope(f) \setminus \{X\}$. Then we define the two basic constraint operators as follows.

Definition 9 (The Combination of Two Constraints). *The combination of two constraints* f_1 *and* f_2 *is a new constraint* $g = f_1 \otimes f_2$*, where* $Scope(g) = Scope(f_1) \cup Scope(f_2)$ *and* $g(\mathbf{w}) = f_1(\mathbf{w}_{\downarrow Scope(f_1)}) \otimes f_2(\mathbf{w}_{\downarrow Scope(f_2)})$ *for every value assignment* $\mathbf{w}$ *of variables in the scope of the constraint* g*.*

Definition 10 (The Marginalization of a Constraint). *The marginalization of* X *from a constraint* f*, where* $X \in Scope(f)$*, is a new constraint* $g = \bigoplus_X f$*, where* $Scope(g) = Scope(f)_{-X}$ *and* $g(\mathbf{w}) = \bigoplus_{x_i \in \mathbf{D}_X} f(x_i, \mathbf{w})$ *for every value assignment* $\mathbf{w}$ *of variables in the scope of the constraint* g*.*

According to the definitions of the CBI problem and the basic constraint operators, we can define the abstract inference and allocation tasks for a CBI problem.

Definition 11 (The Inference Task for a CBI Problem). *Given a subset of interested variables* $\mathbf{Z} = \{Z_1, \cdots, Z_t\} \subseteq \mathbf{X}$*, let* $\mathbf{Y} = \mathbf{X} \setminus \mathbf{Z}$*, the inference task for a CBI problem* $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ *is defined as computing:*

$$g_{CBI}(\mathbf{Z}) = \bigoplus_{\mathbf{Y}} \bigotimes_{f \in \mathbf{F}} f \tag{1}$$

Given a CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$, if $\oplus$ is idempotent, we can define the allocation task for a CBI problem.

Definition 12 (The Allocation Task for a CBI Problem). *Given a subset of variables* $\mathbf{Z} = \{Z_1, \cdots, Z_t\} \subseteq \mathbf{X}$*, let* $\mathbf{Y} = \mathbf{X} \setminus \mathbf{Z}$*, the allocation task for a CBI problem* $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ *is to find a value assignment for the marginalized variables* $\mathbf{Y}$*, which leads to the result of the corresponding inference task* $g_{CBI}(\mathbf{Z})$*. Formally, we compute:*

$$\mathbf{y} = \arg \bigoplus_{\mathbf{Y}} \bigotimes_{f \in \mathbf{F}} f \tag{2}$$

where arg *is a prefix of operator* $\oplus$*. In other words,* $\arg \oplus$ *is an operator that returns arguments of the* $\oplus$ *operator.*

In general, $\otimes$ is a combination operator in CBI problems that combines a set of constraints into a constraint with a larger scope; $\oplus_{\mathbf{Y}} = \oplus_{\mathbf{X} \setminus \mathbf{Z}}$ is a marginalization operator that projects a constraint over the scope $\mathbf{X}$ into its subset $\mathbf{Z}$, through enumerating all possible value assignments of $\mathbf{Y} = \mathbf{X} \setminus \mathbf{Z}$.

Many CBI problems from different disciplines can be embedded into our semiring-based unifying framework [9]. These problems include the decision task and allocation task of CSP and SAT, Max SAT and Max CSP, Fuzzy CSP, Weighted CSP, probability assessment, most probable explanation (MPE), dynamic Bayesian networks (DBN), possibility inference with various t-norms, and maximum likelihood decoding. In [9], we also generalized various systematic inference approaches, including exact and approximate variable elimination, exact and approximate junction tree and variants, and loopy message propagation, into this semiring-based unifying framework.

4 Applying Arc Consistency to CBI Problems

4.1 Arc Consistency and Eliminative Property

Here, we are particularly interested in CBI problems defined on a commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ with the eliminative property. More specifically, we propose in this paper that local consistency techniques in constraint processing can be extended to handle general CBI problems like probability inference

Input: A CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$
Output: A generalized arc consistency CBI problem $\mathbf{P}' = (\mathbf{X}, \mathbf{D}', \mathbf{S}, \mathbf{F}')$
Let $\mathbf{Q}$ be a queue of all the variable-constraint pairs (X, f)
repeat
    Pop the first variable-constraint pair $(X, f) \in \mathbf{Q}$
    if REVISE(X, f) **then**
        for each $g \in \mathbf{F}$ with $X \in \mathit{Scope}(g)$ **do**
            Remove all tuples in g with the value that is removed from X
            for each $Z \in \mathit{Scope}(g)$ and $X \neq Z$ **do**
                if Pair $(Z, g) \notin \mathbf{Q}$ **then**
                    $\mathbf{Q} := \mathbf{Q} \cup \{(Z, g)\}$
                end if
            end for
        end for
    end if
until $\mathbf{Q}$ is empty
Return $\mathbf{P}' := \mathbf{P}$

Fig. 1. Generalization of Generalized Arc Consistency Algorithm GGAC($\mathbf{P}$)

and maximum likelihood decoding, if the corresponding commutative semiring of the problem representation is eliminative. Formally, we define the generalized arc consistency of a CBI problem as follows:

Definition 13 (A CBI Problem is GGAC). *A CBI Problem* $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ *with an eliminative commutative semiring* $\mathbf{S}$ *is generalized arc consistent (GGAC) if:* $\forall f \in \mathbf{F}$, $\forall X \in \mathit{Scope}(f)$, $\forall x \in \mathbf{D}_X$, $\exists \mathbf{w}$, *a value assignment of variables* $\mathit{Scope}(f)_{-X}$, *s.t.* $f(x, \mathbf{w}) \neq \alpha_{\otimes}$

Figure 1 shows a generalized version of generalized arc consistency (GGAC) enforcing algorithm for a CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ with an eliminative commutative semiring $\mathbf{S}$. The procedure REVISE of GGAC is shown in Figure 2.

Theorem 1 (GGAC Enforces Generalized Arc Consistency). *Applying GGAC algorithm to a CBI problem* $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ *with an eliminative commutative semiring* $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ *leads to a generalized arc consistent CBI problem* $\mathbf{P}' = (\mathbf{X}, \mathbf{D}', \mathbf{S}, \mathbf{F}')$.

Proof: Assume there exists a constraint $f' \in \mathbf{F}'$ and a variable $X \in \mathit{Scope} f'$ that lead to generalized arc inconsistency in $\mathbf{P}'$. We know the pair (X, f') must be popped from the queue sometime since X and f' are in $\mathbf{P}'$. However, the REVISE procedure ensures that every pair popped from the queue is generalized arc consistent, which contradicts the assumption. □

The equivalency of a CBI problem with an eliminative commutative semiring and the generalized arc consistency CBI problem yielded by GGAC algorithm in Figure 1 w.r.t. the results of their inference tasks is proved by Theorem 2.

Input: A variable $X \in \mathbf{X}$ and a constraint $f \in \mathbf{F}$
Output: TRUE if a value is removed from the domain of X else FALSE

$flag := TRUE$
for each $x \in \mathbf{D}_X$ **do**
  for each value assignment $\mathbf{w}$ of $Scope(f)_{-X}$ **do**
    if $f(x, \mathbf{w}) \neq \alpha_{\otimes}$ **then**
      $flag := FALSE$
      Break loop
    end if
  end for
  if $flag$ **then**
    Remove x from $\mathbf{D}_X$
    Return TRUE
  end if
end for
Return FALSE

Fig. 2. Procedure REVISE(X, f) for Eliminating a Domain Value from a Variable According to the Local Constraint

Theorem 2 (Closure of GGAC). *Let $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ be a CBI problem and the commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ is eliminative. Let $\mathbf{P}' = (\mathbf{X}, \mathbf{D}', \mathbf{S}, \mathbf{F}')$ be the CBI problem yielded by GGAC algorithm. For any subset of interested variables $\mathbf{Z} \subseteq \mathbf{X}$, the inference tasks for $\mathbf{P}$ and $\mathbf{P}'$ are equivalent.*

Proof: Let (X, f) be a pair that is revised by the procedure REVISE, where $x \in \mathbf{D}_X$ is removed because of generalized arc inconsistency. Consider the global constraint g of the combination of all the constraints in $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$. We have $g(\mathbf{X}) = f(\mathbf{X}_{\downarrow Scope(f)}) \otimes \bigotimes_{h \in \mathbf{F}, h \neq f} h(\mathbf{X}_{\downarrow Scope(h)})$. More specifically, for any value assignment $\mathbf{u}$ of variables $\mathbf{X}_{-X}$, we have $g(x, \mathbf{u}) = f(x, \mathbf{u}_{\downarrow Scope(f)}) \otimes \bigotimes_{h \in \mathbf{F}, h \neq f} h(\mathbf{u}_{\downarrow Scope(h)}) = \alpha_{\otimes}$, since $f(x, \mathbf{u}_{\downarrow Scope(f)}) = \alpha_{\otimes}$ is the absorbing element of the operator $\otimes$. Given $g(X = x, \mathbf{u}) = \alpha_{\otimes}$ is also the identity element of the operator $\oplus$, the inference task of $\mathbf{P}$ (Equation 1) is to compute:

$$\begin{aligned} g_{CBI}(\mathbf{Z}) &= \bigoplus_{\mathbf{Y}} g(X, \mathbf{X}_{-X}) \\ &= \bigoplus_{\mathbf{Y}} \left(g(X = x, \mathbf{X}_{-X}) \oplus g(X \neq x, \mathbf{X}_{-X}) \right) \\ &= \bigoplus_{\mathbf{Y}} g(X \neq x, \mathbf{X}_{-X}) \end{aligned} \tag{3}$$

On the other hand, let us consider the global constraint g' of $\mathbf{P}' = (\mathbf{X}, \mathbf{D}', \mathbf{S}, \mathbf{F}')$. We have: $g'(\mathbf{X}) = \bigotimes_{f' \in \mathbf{F}'} f' = g(X \neq x, \mathbf{X}_{-X})$ according to the GGAC algorithm in Figure 1. Then it is straightforward to get $g'_{CBI}(\mathbf{Z}) = \bigoplus_{\mathbf{Y}} g'(\mathbf{X}) = g_{CBI}(\mathbf{Z})$. $\square$

In other words, Theorem 2 shows that when we detect that there exists a constraint of a given CBI problem of an eliminative semiring structure that maps all its value assignments with a specific value to the multiplicative absorbing element, the value can be safely removed from the variable's domain. All value assignments with this value can be safely removed from any constraint with this variable in its scope, without modifying the computational result of the inference task.

Theorems 1 and 2 together imply the correctness of the GGAC algorithm.

Theorem 3 (Time Complexity of GGAC). *The worst case time complexity of the GGAC algorithm in Figure 1 is $O(r \cdot d^{k+1})$, where r is the number of constraints, d is the maximum domain size, and k is the maximum scope size of constraints.*

Proof: Basically the GGAC algorithm is a straightforward revision of the generalized arc consistency enforcing algorithm for classic non-binary CSPs [2]. For each constraint with at most k variables in its scope we need d^k checks. Each variable-constraint pair enters the queue at most d times, so the total number of checks is $O(r \cdot d^{k+1})$. □

We extend the application of Theorem 2 by introducing another equivalency statement of the inference task for a CBI problem.

Theorem 4. *Solving the inference task of a CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ is equivalent to solving $\mathbf{P'} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F'})$, where $\mathbf{F'} = \mathbf{F_p} \cup \{f\}$, $\mathbf{F_p} \subset \mathbf{F}$ and $f = \bigotimes_{h \in \mathbf{F} \setminus \mathbf{F_p}} h$.*

Proof: Easy to prove given the definition of the inference task of a CBI problem in Equation 1. □

Combining Theorem 2 and Theorem 4 lead to the local consistency property of general CBI problems. We do not have to focus on the original constraints in $\mathbf{F}$ of the CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$. We can combine some original constraints to a local constraint, then apply the conclusion of Theorem 2 to refine the representation of the original CBI problem.

If the allocation task can be defined on a CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$, in other words, $\oplus$ is idempotent, we have the similar result, as shown in Theorem 5.

Theorem 5 (Closure of GGAC for Allocation Task). *Let $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ be a CBI problem and the commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ is eliminative. $\oplus$ is idempotent. Let $\mathbf{P'} = (\mathbf{X}, \mathbf{D'}, \mathbf{S}, \mathbf{F'})$ be the CBI problem yielded by GGAC algorithm. For any subset of interested variables $\mathbf{Z} \subseteq \mathbf{X}$, we have the alloction tasks for $\mathbf{P}$ and $\mathbf{P'}$ are equivalent.*

Proof: Similar to the proof of Theorem 2. It is easy to show that the value $x \in \mathbf{D}_X$ cannot appear in any value assignment that leads to the inference task's result $g_{CBI}(\mathbf{Z})$. □

Through shrinking the domain of a variable as well as deleting possible value assignments of constraints, the size of the original CBI problem is reduced by factor $(|\mathbf{D}_X| - 1)/|\mathbf{D}_X|$. Repeatedly applying the GGAC algorithm, we will get

Input: A CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ and a variable subset $\mathbf{Z}$ of interest
Output: $g_{CBI}(\mathbf{Z}) = \bigoplus_{\mathbf{X} \setminus \mathbf{Z}} \bigotimes_{f \in \mathbf{F}} f$

```
P := GGAC(P)
Let Y = X \ Z
Choose an elimination ordering σ =< Y_1, ···, Y_k > of Y
for i = k to 1 do
    F' := ∅
    for each f ∈ F do
        if Y_i ∈ Scope(f) then
            F' := F' ∪ {f}
            F := F \ {f}
        end if
    end for
    f' := ⊕_{Y_i} ⊗_{f∈F'} f
    F := F ∪ {f'}
    for each X ∈ Scope(f') do
        if REVISE(X, f') then
            P := GGAC(P)
            Break loop
        end if
    end for
end for
Return g_CBI(Z) := ⊗_{f∈F} f
```

Fig. 3. Generalization of Variable Elimination with Arc Consistency Algorithm GVE-AC($\mathbf{P}, \mathbf{Z}$)

a series of equivalent smaller CBI problems. The generalized arc consistency enforcement provides opportunities for performing inference more efficiently. We may either preprocess the CBI problem then apply regular systematic or stochastic inference approaches or simplify the problem during the running of inference approaches. For example, it is straightforward to incorporate generalized arc consistency enforcing into generalized variable elimination algorithm [9], as shown in Figure 3, if a CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ has an eliminative commutative semiring $\mathbf{S}$.

4.2 Approximate Local Consistency and Monotonic Property

Given a CBI problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$, if the commutative semiring $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$ is both eliminative and monotonic, we can propose a scheme to enforce local consistency approximately for this CBI problem. In other words, for an eliminative and monotonic commutative semiring, we use an element $\epsilon \in \mathbf{A}$ to approximate the multiplicative absorbing element $\alpha_{\otimes}$ that is equal to the additive identity element $\mathbf{0}$ for an eliminative commutative semiring.

Formally, we define the generalized ϵ arc consistency of a CBI problem as follows:

Input: A variable $X \in \mathbf{X}$, a constraint $f \in \mathbf{F}$, an element $\epsilon \in \mathbf{A}$
Output: TRUE if a value is removed from the domain of X; FALSE if else

```
flag := TRUE
for each x ∈ D_X do
  for each value assignment w of Scope(f)_{-X} do
    if ε ≤_S f(x, w) then
      flag := FALSE
      Break loop
    end if
  end for
  if flag then
    Remove x from D_X
    Return TRUE
  end if
end for
Return FALSE
```

Fig. 4. Procedure ϵ-REVISE(X, f, ϵ) for Eliminating a Domain Value from a Variable According to the Approximation of a Local Constraint.

Definition 14 (A CBI Problem is ϵ-GGAC). *A CBI Problem* $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ *with an eliminative commutative semiring* $\mathbf{S}$ *is* ϵ *generalized arc consistent (*ϵ*-GGAC) if:* $\forall f \in \mathbf{F}$, $\forall X \in Scope(f)$, $\forall x \in \mathbf{D}_X$, $\exists \mathbf{w}$, *a value assignment of variables* $Scope(f)_{-X}$, *s.t.* $f(x, \mathbf{w}) \geq_{\mathbf{S}} \epsilon$

An ϵ-GGAC algorithm can be achieved by modifying the REVISE procedure in Figure 2 to the procedure ϵ-REVISE(X, f, ϵ) in Figure 4. It is straightforward to show that the GGAC algorithm in Figure 1 with the procedure ϵ-REVISE leads to a ϵ-GGAC CBI problem of the given CBI Problem $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ with an eliminative and monotonic commutative semiring $\mathbf{S}$.

Theorem 6 show that ϵ-GGAC algorithm can be used to simplify the original CBI problem with a controlled threshold. The simplified CBI problem gives a lower bound of the estimation of the inference task.

Theorem 6 (Lower Bound Estimation of ϵ-GGAC). *Given a CBI problem* $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$ *with an eliminative and monotonic commutative semiring* $\mathbf{S} = \langle \mathbf{A}, \oplus, \otimes \rangle$*, the* ϵ*-GGAC algorithm yields a CBI problem* $\mathbf{P}' = (\mathbf{X}, \mathbf{D}', \mathbf{S}, \mathbf{F}')$ *that is an approximation of* $\mathbf{P}$ *w.r.t. the results of their inference tasks. For any value assignment* $\mathbf{z}$ *of interested variables* $\mathbf{Z}$*, the inference task of* $\mathbf{P}'$, $g'_{CBI}(\mathbf{z})$*, is a lower bound of* $g_{CBI}(\mathbf{z})$*, w.r.t. the partial order* $\leq_{\mathbf{S}}$ *of the monotonic commutative semiring* $\mathbf{S}$.

Proof: Let (X, f) be a pair that is revised by the procedure ϵ-REVISE, where $x \in \mathbf{D}_X$ is removed because of ϵ generalized arc inconsistency. Consider the global constraint g of the combination of all the constraints in $\mathbf{P} = (\mathbf{X}, \mathbf{D}, \mathbf{S}, \mathbf{F})$. We have $g(\mathbf{X}) = f(\mathbf{X}_{\downarrow Scope(f)}) \otimes \bigotimes_{h \in \mathbf{F}, h \neq f} h(\mathbf{X}_{\downarrow Scope(h)})$. More specifically, for any value assignment $\mathbf{u}$ of variables $\mathbf{X}_{-X}$, we have $g(x, \mathbf{u}) = f(x, \mathbf{u}_{\downarrow Scope(f)}) \otimes$

$\bigotimes_{h\in\mathbf{F},h\neq f} h(\mathbf{u}_{\downarrow Scope(h)})$. Since $\alpha_\otimes \leq_\mathbf{S} f(x,\mathbf{u}_{\downarrow Scope(f)}) \leq_\mathbf{S} \epsilon$ and $\otimes$ is monotonic, we have $\alpha_\otimes \leq_\mathbf{S} g(x,\mathbf{u})$.

Given that $\alpha_\otimes$ is also the identity element of the operator $\oplus$ ($\mathbf{S}$ is eliminative) and $\oplus$ is monotonic, the inference task of $\mathbf{P}$ (Equation 1) is to compute:

$$\begin{aligned} g_{CBI}(\mathbf{Z}) &= \bigoplus_{\mathbf{Y}} g(X,\mathbf{X}_{-X}) \\ &= \bigoplus_{\mathbf{Y}} (g(X=x,\mathbf{X}_{-X}) \oplus g(X\neq x,\mathbf{X}_{-X})) \\ &\geq_\mathbf{S} \bigoplus_{\mathbf{Y}} g(X\neq x,\mathbf{X}_{-X}) \end{aligned} \tag{4}$$

On the other hand, let us consider the global constraint g' of $\mathbf{P'} = (\mathbf{X},\mathbf{D'},\mathbf{S},\mathbf{F'})$. We have: $g'(\mathbf{X}) = \bigotimes_{f'\in\mathbf{F'}} f' = g(X\neq x,\mathbf{X}_{-X})$ according to the ϵ-GGAC algorithm. Then it is straightforward to get $g'_{CBI}(\mathbf{Z}) = \bigoplus_\mathbf{Y} g'(\mathbf{X}) \leq_\mathbf{S} g_{CBI}(\mathbf{Z})$ for every value assignment of interested variable subset $\mathbf{Z}$. □

Theorem 7 (Time Complexity of ϵ-GGAC). *The worst case time complexity of the ϵ-GGAC algorithm is $O(r\cdot d^{k+1})$, where r is the number of constraints, d is the maximum domain size, and k is the maximum scope size of constraints.*

Proof: The worst case time complexity of the ϵ-GGAC algorithm is the same as the GGAC algorithm, which is $O(r\cdot d^{k+1})$. □

5 Arc Consistency in Probability Assessment: An Example

Probability inference problems can be seen as constraint-based inference by treating conditional probability distributions (CPDs) as soft constraints over variables. A Bayesian network (BN) [13] is a graphical representation for probability inference under conditions of uncertainty. BN is defined as a directed acyclic graph (DAG) where vertices $\mathbf{X} = \{X_1,\cdots,X_n\}$ denote n random variables and directed edges denote causal influences between variables. $\mathbf{D} = \{D_1,\cdots,D_n\}$ is a collection of finite domains for the variables. A set of conditional probability distributions $\mathbf{F} = \{f_1,\cdots,f_n\}$, where $f_i = P(X_i|Parents(X_i))$ is attached to each variable (vertex) X_i. Then the probability distribution over $\mathbf{X}$ is given by $P(\mathbf{X}) = \prod_{i=1}^n f_i$.

As a fundamental problem of probability inference, the probability assessment problem in Bayesian networks computes the posterior marginal probability of a subset of variables, given values for some variables as known evidence. We show in [9] that the probability assessment problem can be represented as a CBI problem using the commutative semiring $\mathbf{S_{prob}} = \langle \mathbb{R}^+ \cup \{0\}, +, \times\rangle$. As discussed in Section 1, soft arc consistency enforcing techniques in the Semiring CSP and Valued CSP frameworks cannot handle this problem. We show in this section that our GGAC and ϵ-GGAC enforcing algorithms can preprocess the

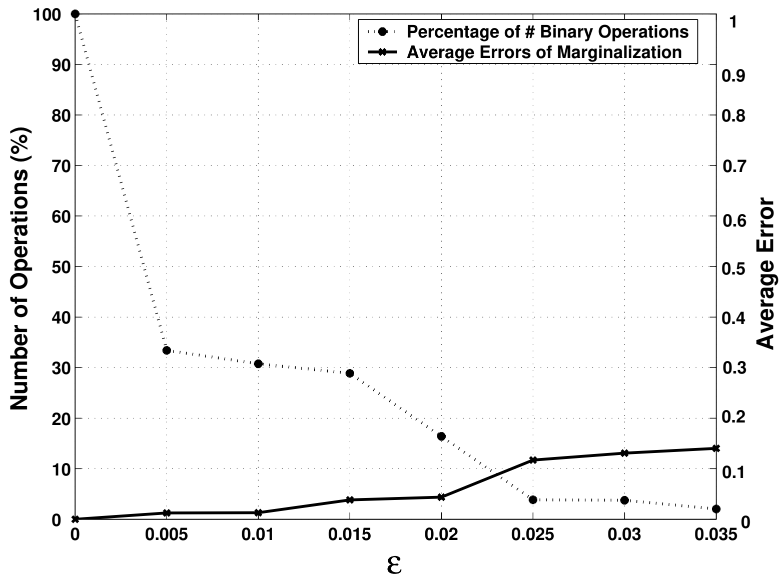

Fig. 5. The number of binary operations required for probability assessment after using the ϵ-GGAC algorithm (shown as a fraction of the number required without ϵ-GGAC) and the resultant error of the marginal probability for the Insurance network as a function of ϵ.

probability assessment problem efficiently. It is easy to show that $\alpha_{\otimes} = \mathbf{0} = 0$ and $\mathbf{S_{prob}}$ is monotonic.

The Bayesian network used here is the Insurance network from the Bayesian network Repository [14]. The network has 27 variables and 27 non-binary constraints (CPDs). In our experiments, we randomly choose two variables as observed. The ϵ-GGAC algorithm is used to preprocess the problem. The junction tree algorithm in Lauritzen-Spiegelhalter architecture [15] is used to infer the marginal probability of every unobserved variable. We compare the number of binary operations required for probability assessment after using the ϵ-GGAC algorithm (shown as a fraction of the number required without ϵ-GGAC) and the resultant error of the marginal probability for the Insurance network as a function of ϵ in Figure 5. At each value of ϵ, we collect data of 5 runs. Results of our experiments is shown in Figure 5. It is clear that ϵ controls the tradeoff of the precision and the speed of the inference.

6 Conclusion

As the most important local consistency techniques in constraint programming, arc consistency [1] and its non-binary version, generalized arc consistency [2, 3], are widely studied. The soft arc consistency algorithms [6, 7] in the Semiring CSP [4] and Valued CSP [5] frameworks extend successfully the notion of arc consistency to the soft constraint processing. However, the idempotent constraint combination operators or the fair valuation structure in these frameworks restrict the application of arc consistency enforcing algorithms to other problems with analogous representations to hard and soft CSPs. As the first result of this paper, we propose a weaker condition of applying generalized arc consistency enforcing techniques to a broader coverage of constraint-based inference problems, based on a semiring-based unified framework for CBI problems [9]. The weaker condition proposed here depends only on the existence and property of the combination absorbing element and does not depend on other semiring properties. We also present a concept of ϵ-GGAC that simplifies the representation of a CBI problem approximately. We show in this paper that the approximate inference task is a lower bound of the exact one w.r.t the total ordering of values in the commutative semiring structures. We also presented several generalized arc consistency enforcing algorithms in this paper. The worst time complexity of our generalization of generalized arc consistency enforcing algorithm is $O(r \cdot d^{k+1})$, where r is the number of constraints, d is the maximum domain size, and k is the maximum scope size of constraints. Our generalization of generalized arc consistency provides opportunities to researchers in the constraint programming community to extend their knowledge of local consistency enforcing approaches to other constraint-based inference problems such as probability inference and decoding problems.

References

1. Mackworth., A.K.: Consistency in networks of relations. Artificial Intelligence **8** (1977) 99–118
2. Mackworth, A.K.: On reading sketch maps. In: IJCAI77. (1977) 598–606
3. Mohr, R., Masini, G.: Good old discrete relaxation. In: European Conference on Artificial Intelligence. (1988) 651–656
4. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint satisfaction and optimization. J. ACM **44** (1997) 201–236
5. Schiex, T., Fargier, H., Verfaillie, G.: Valued constraint satisfaction problems: Hard and easy problems. In: IJCAI95, Montreal (1995) 631–637
6. Cooper, M., Schiex, T.: Arc consistency for soft constraints. Artificial Intelligence **154** (2004) 199–227
7. Bistarelli, S.: Semirings for Soft Constraint Solving and Programming. Springer-Verlag (2004)
8. Aji, S.M., McEliece, R.J.: The generalized distributive law. IEEE Transactions on Information Theory **46** (2000) 325–343
9. Chang, L.: Generalized constraint-based inference. Master's thesis, Department of Computer Science, University of British Columbia (2005)
10. Kohlas, J., Shenoy, P.: Computation in valuation algebras. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning. Kluwer, Dordrecht (2000) 5–40
11. Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. In: 12th Conf. on Uncertainty in Artificial Intelligence. (1996) 211–219

12. Kask, K., Dechter, R., Larrosa, J.: Unifying cluster-tree decompositions for automated reasoning. Submitted to the AIJ (2003)
13. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc. (1988)
14. Friedman, N., Goldszmidt, M., Heckerman, D., Russell, S.: (Bayesian network repository, http://www.cs.huji.ac.il/labs/compbio/repository/)
15. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, Series B **50** (1988) 157–224