The RegularGcc Matrix Constraint

Ronald de Haan

EMCL Student Workshop 2012

Outline

Reminder on CSPs

the RegularGcc Constraint

Example of Practical Use

Complexity of Propagation

Constraint Satisfaction Problem

A constraint satisfaction problem (CSP) is a triple (X, D, C), where:

- $X = (x_1, \ldots, x_n)$ is a sequence of variables
- $D = (D_1, \ldots, D_n)$ is a sequence of domains for these variables
  i.e., $x_i \in D_i$ for all $1 \leq i \leq n$
- $C = \{c_1, \ldots, c_m\}$ is a set of constraints on subsequences of X

A constraint on a sequence $(x_1, \ldots, x_k)$ with domains $(D_1, \ldots, D_k)$ is a subset $C \subseteq D_1 \times \cdots \times D_k$ of the cross-product of the domains.

An instantiation $(d_1, \ldots, d_n) \in D_1 \times \cdots \times D_n$ is a solution for the CSP if for each constraint $c \in C$ on sequence $(x_{i_1}, \ldots, x_{i_k})$ we have that $(d_{i_1}, \ldots, d_{i_k}) \in c$.

Domain Consistency

Many strategies used to solve CSPs use a method called constraint propagation: transforming the CSP to an equivalent one that satisfies some local consistency notions.

A CSP (X, D, C) is domain consistent (DC) for a constraint $c \in C$ on $(x_{i_1}, \ldots, x_{i_k})$ if

- for each x_{i_j} and each $d_{i_j} \in D_{i_j}$,
- there exist $d_{i_1}, \ldots, d_{i_{j-1}}, d_{i_{j+1}}, \ldots, d_{i_k}$ such that:
- $(d_{i_1}, \ldots, d_{i_k}) \in c$.

This $(d_{i_1}, \ldots, d_{i_k})$ is called the support of d_{i_j}.

Domain consistency is also called *generalized arc consistency*.

Outline

Deterministic Finite-State Automata

A deterministic finite-state automaton (DFA) is a quintuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set of states
- Σ is an alphabet
- $\delta : Q \times \Sigma \rightarrow Q$ is a (partial) transition function
- $q_0 \in Q$ is the initial state
- $F \subseteq Q$ is the set of accepting states

A DFA $\mathcal{A}$ accepts a string $w_1 \ldots w_n \in \Sigma^*$ if there are states $q_1, \ldots, q_n \in Q$ such that:

- $q_i = \delta(q_{i-1}, w_i)$ for $1 \leq i \leq n$
- $q_n \in F$

The language a DFA recognizes is the set of strings it accepts. DFAs recognize exactly the class of regular languages.

the Regular Constraint

Given

- a sequence of variables $X = (x_1, \ldots, x_n)$ with domains $(D_1, \ldots, D_n)$, and
- a DFA $\mathcal{A}$,

the constraint $\mathit{Regular}(X, \mathcal{A})$ is the set of those sequences $(d_1, \ldots, d_n) \in D_1 \times \cdots \times D_n$ such that:

- $d_1 \ldots d_n$ is accepted by $\mathcal{A}$.

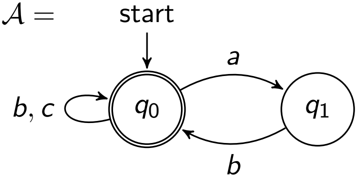
Enforcing DC for the Regular constraint can be done in linear time. [1, 3]

Some suggested literature about the Regular constraint, for those interested: [1, 2, 3, 4].

the Regular Constraint (example)

Let

- $X = (x_1, x_2, x_3, x_4)$,
- $D_1 = D_2 = D_3 = D_4 = \{a, b, c\}$,
- $\mathcal{A} =$



- and $c = Regular(X, \mathcal{A})$.

Then $\{(a, b, c, b), (b, b, b, c)\} \subseteq c$, but $(a, a, b, b) \notin c$ and $(b, b, b, a) \notin c$.

the GlobalCardinality Constraint

Given

- a sequence of variables $X = (x_1, \ldots, x_n)$ with domains $(D_1, \ldots, D_n)$,
- a sequence of values $V = (v_1, \ldots, v_k)$,
- a sequence of lower bounds $L = (l_1, \ldots, l_k) \in \mathbb{N}^k$, and
- a sequence of upper bounds $U = (u_1, \ldots, u_k) \in \mathbb{N}^k$,

the constraint $\mathit{GlobalCardinality}(X, V, L, U)$ is the set of those sequences $(d_1, \ldots, d_n) \in D_1 \times \cdots \times D_n$ such that for each $1 \leq i \leq k$:

- the number of occurrences of value v_i in the sequence $(d_1, \ldots, d_n)$ is at least l_i and at most u_i.

Enforcing DC for the GlobalCardinality constraint can be done in quadratic time. [5]

the GlobalCardinality Constraint (example)

Let

- $X = (x_1, x_2, x_3, x_4)$,
- $D_1 = D_2 = D_3 = D_4 = \{a, b, c\}$,
- $V = (a, b)$,
- $L = (0, 2)$,
- $U = (1, 3)$, and
- $c = GlobalCardinality(X, V, L, U)$.

Then $\{(a, b, c, b), (b, b, b, c)\} \subseteq c$, but $(a, a, b, b) \notin c$ and $(b, b, b, b) \notin c$.

the RegularGcc constraint

Given

- a number of rows $R \in \mathbb{N}$, a number of columns $C \in \mathbb{N}$,
- a $R \times C$ matrix $\mathcal{M}$ of variables $\mathcal{M}_{r,c}$ with domain $D_{r,c}$,
- for each row r a Regular constraint $Regular_r$, and
- for each column c a GlobalCardinality constraint Gcc_c,

the corresponding *RegularGcc* constraint is the set of those instantiations[1] that assign to each $\mathcal{M}_{i,j}$ a value $d_{i,j} \in D_{i,j}$ such that:

- for each row r, $(d_{r,1}, \ldots, d_{r,C}) \in Regular_r$, and
- for each column c, $(d_{1,c}, \ldots, d_{R,c}) \in Gcc_c$.

[1] We implicitly generalize the notion of a *CSP* from sequences to matrices.

Outline

Practical use of RegularGcc

Consider a nurse scheduling problem with n nurses over d days, where each nurse can be assigned one of multiple shifts each day.

Each day there must be a certain number of shifts assigned (capacity requirement).

There are restrictions on the assignment for each nurse (individual requirements). For instance:

- No early morning shift directly after a late night shift.
- At least one off-work period of f days in a row.

Encode this in a RegularGcc constraint on an $n \times d$ matrix.

- possible values $\sim$ different shifts
- capacity requirements $\sim$ column (Gcc) constraints
- individual requirements $\sim$ row (Regular) constraints

Outline

NP-hardness of enforcing DC

We sketch a reduction from 3-SAT. Take a 3-CNF formula $\varphi = \gamma_1 \wedge \cdots \wedge \gamma_C$ on propositional variables $p_1, \ldots, p_R$.

We construct a $R \times C$ matrix $\mathcal{M}$ of variables with domain $\{-1, 0, 1\}$.

Each row r corresponds to a variable p_r and each column c corresponds to a clause γ_c.

We initialize the domains of the variables as follows. For each clause γ_c we set the domain of $\mathcal{M}_{r,c}$ to

- $\{0\}$ if p_r does not occur in γ_c,
- $\{-1, 0\}$ if p_r occurs negatively in γ_c, and
- $\{0, 1\}$ if p_r occurs positively in γ_c.

NP-hardness of enforcing DC

On each column we put the GlobalCardinality constraint that enforces that the value 0 occurs at most $R - 1$ times.

On each row we put the Regular constraint that enforces that the row contains besides 0's either only 1's or only -1's.

(Solution $\Rightarrow$ model)
If a 1 appears in row r, set p_r to $\top$; otherwise to $\bot$.

(Model $\Rightarrow$ solution)
If p_r is assigned $\top$, set all possible 1's in row r, the rest 0's.
If p_r is assigned $\bot$, set all possible -1's in row r, the rest 0's.

NP-hardness of enforcing DC

For instance, take
$\varphi = (p_1 \vee p_2 \vee \neg p_3) \wedge (\neg p_1 \vee \neg p_1 \vee \neg p_2) \wedge (p_3 \vee p_3 \vee p_2)$.

The instantiated matrix looks like this.

	γ_1	γ_2	γ_3
p_1	0,1	-1,0	0
p_2	0,1	-1,0	0,1
p_3	-1,0	0	0,1

NP-hardness of enforcing DC

For instance, take
$\varphi = (p_1 \vee p_2 \vee \neg p_3) \wedge (\neg p_1 \vee \neg p_1 \vee \neg p_2) \wedge (p_3 \vee p_3 \vee p_2)$.

The instantiated matrix looks like this.

	γ_1	γ_2	γ_3
p_1	0,1	-1,0	0
p_2	0,1	-1,0	0,1
p_3	-1,0	0	0,1

The red satisfying instantiation corresponds to the red solution.

More complexity issues I looked at. . .

- It is NP-hard even for more restricted cases (restricted row constraints).
- It is NP-hard even for bounds consistency.
- It is FPT, when parameterized on
  - simultaneously both the number of rows and the (maximal) automaton size.
- We got similar FPT results for slightly more general cases (more general column constraints).
- It is W[2]-hard, when parameterized on
  - just the number of rows.
- We got similar W[2]-hardness results for some more restricted cases.

Some issues I am currently looking at...

- Can the complexity results be extended to cases with symmetry breaking constraints?
  - For lexicographical ordering of rows, it seems so... at least partly...
  - Different symmetry breaking constraints?
- Are there practical restricted cases where propagation is cheaper?

References

Katsirelos, G., Narodytska, N., Quimper, C.G., Walsh, T.: Global matrix constraints. In: Proceedings of the International Workshop on Constraint Modelling and Reformulation. pp. 27–41 (2011)

Katsirelos, G., Maneth, S., Narodytska, N., Walsh, T.: Restricted global grammar constraints. In: Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming (CP'09). vol. 5732, pp. 501–508. Springer (2009)

Pesant, G.: A regular language membership constraint for finite sequences of variables. In: Wallace, M. (ed.) Proceedings of the 10th International Conference on Principles and Practice of Constraint Programming (CP'04). vol. 3258, pp. 482–495. Springer (2004)

Quimper, C.G., Walsh, T.: Decomposing global grammar constraints. In: Bessiere, C. (ed.) Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming (CP'07). vol. 4741, pp. 590–604. Springer (2007)

Regin, J.C.: Generalized arc consistency for global cardinality constraint. In: Proceedings of the 14th National Conference on Artificial intelligence (AAAI'98). pp. 209–215 (1996)