Using Gröbner Basis Theory to Compute Constraint Networks in Globally Solved Form

M.R.C. van Dongen (dongen@cs.ucc.ie)
Department of Computer Science,
National University of Ireland, Cork

August 11, 1999

Abstract

In this paper a new technique is presented which allows for the transformation of any extensional constraint network to an equivalent network which is in globally solved form with respect to a certain given variable ordering. This guarantees that all solutions can be found with backtracking without encountering "dead-ends." The resulting network corresponds to a reduced Gröbner basis with respect to a lexicographical term order. The number of solutions of the constraint network can be computed by inspecting the reduced Gröbner basis.

1 Introduction

This paper presents a new technique to transform any constraint network with extensional constraints to an equivalent network which is in globally solved form w.r.t. (with respect to) a certain variable ordering. This guarantees a backtrack-free search and guarantees that all solutions can be found without encountering "dead-ends."

The process of transforming the network consists of three steps. First the constraint network is transformed to a generating basis of a polynomial ideal. Next the basis is transformed to its reduced Gröbner basis w.r.t. some lexicographical term order. Finally, the reduced Gröbner basis is transformed to a constraint network.

The number of solutions of the constraint network can be counted by inspecting the Gröbner basis which was computed during the transformation process.

It has been tried to keep this work as self-contained as possible. Therefor the paper starts by discussing mathematical background. Next salient aspects of the constraint satisfaction paradigm are recalled. It proceeds by formally explaining the technique. Next the technique is demonstrated with an example. Finally, conclusions are presented.

2 Related Mathematical Theory

The theory of *Gröbner bases* allows for the algorithmic solution of many problems in multivariate polynomial ideal theory [2, 1, 3]. This section is an introduction to Gröbner basis theory.

Let $k[x_1, \ldots, x_n]$ denote the polynomial ring with variables $\{x_1, \ldots, x_n\}$ and coefficients in k. Let $F \subseteq k[x_1, \ldots, x_n]$. The *ideal generated by* F, denoted $\mathbf{I}(F)$, is given by $\{\sum_{f \in F} \phi(f) \times f | \phi \in k[x_1, \ldots, x_n]^F\}$, where $k[x_1, \ldots, x_n]^F$ is the set containing all functions from F to $k[x_1, \ldots, x_n]$.

The *variety* of $F \subseteq k[x_1, \ldots, x_n]$, denoted $\mathbf{V}(F)$, is the set of common zeroes of the members of F, i.e. $\mathbf{V}(F) = \{v \in k^n | (\forall f \in F)(\epsilon_v(f) = 0)\}$, where $\epsilon_v(f)$ is the evaluation of f at v.

Let $W \subseteq k^n$ be a variety. The *ideal* of W, denoted $\mathbf{I}(W)$, is the set of all polynomials in $k[x_1, \ldots, x_n]$ which vanish at W, i.e. $\mathbf{I}(W) = \{f \in k[x_1, \ldots, x_n] | (\forall v \in W)(\epsilon_v(f) = 0)\}$.

A *term order* $\prec$ is a total order on terms of polynomials with the property that 1 is its smallest member and for any terms u_1 and u_2 it holds that $u_1 \prec u_2 \Rightarrow (\forall u)(u_1 u \prec u_2 u)$. *Lexicographical* term orders compare exponents of terms lexicographically.

Let f be a non-zero polynomial, I an ideal and $\prec$ a term order. The *initial monomial* $in_\prec(f)$ of f w.r.t. $\prec$ is the greatest monomial of f w.r.t. $\prec$. The *initial ideal* of I w.r.t. $\prec$, denoted $in_\prec(I)$, is the ideal $\mathbf{I}(\{in_\prec(f) | f \in I\})$. The *standard monomials* of I w.r.t. $\prec$ are the monomials in $k[x_1, \ldots, x_n]$ which have coefficients 1 and are not in $in_\prec(I)$.

A field is *algebraically closed* if every non-constant polynomial has a root. From now on let k be algebraically closed, $I \subseteq k[x_1, \ldots, x_n]$ an ideal and $\prec$ a term order.

A set $\mathcal{G}_\prec(I) \subseteq I \setminus \{0\}$ is called a *Gröbner basis* of I w.r.t. $\prec$ if $(\forall f \in I \setminus \{0\})(\exists g \in \mathcal{G}_\prec(I))(in_\prec(g) \mid in_\prec(f))$, where $g \mid f$ iff g properly divides f. A Gröbner basis $\mathcal{G}^r_\prec(I)$ is called *reduced* if for every $g \in \mathcal{G}^r_\prec(I)$, $in_\prec(g)$ does not properly divide any monomial of any polynomial in $\mathcal{G}^r_\prec(I) \setminus \{g\}$.

The *Buchberger Algorithm* allows for the computation of (reduced) Gröbner bases [2, 1, 3].

Theorem 1 (Triangular Form Theorem) *Let $\prec$ be any term order and $I \subset k[x_1, \ldots, x_n]$ a proper ideal. I is zero-dimensional*[1] *iff $\mathcal{G}^r_\prec(I)$ contains a non-constant polynomial with initial monomial $x_i^{\alpha_i}$, for $1 \leq i \leq n$. (Theorem 6.54 (i) and (iv) of [1].)*

The l-th *elimination ideal* of I, denoted $\mathbf{I}_l(I)$, is the ideal $I \cap k[x_{l+1}, \ldots, x_n]$. The ideal $\mathbf{I}_l(I)$ corresponds to eliminating of the l most significant variables from I. The following theorem provides an algorithm for computing elimination ideals.

Theorem 2 (Elimination Theorem) *Let $\prec$ be the lexicographical term order where $x_n \prec \cdots \prec x_1$ and $1 \leq i \leq n$, then $\mathbf{I}_l(I) = \mathbf{I}(\mathcal{G}^r_\prec(I) \cap k[x_{l+1}, \ldots, x_n])$. (See [3] pp. 113,114,118).*

One application of elimination ideals is in solving systems of polynomial equations. The idea is to eliminate variables to create a system involving less variables. Next the smaller system is solved and the (partial) solutions of the smaller system are extended. The following theorem predicts when partial solutions can be extended.

[1] An ideal I is called zero-dimensional if it is proper and the cardinality of its variety is finite.

Theorem 3 (Extension Theorem) *Let $F = \{f_1, \ldots, f_m\} \subset k[x_1, \ldots, x_n]$, and $I = \mathbf{I}(F)$. Furthermore, for each $1 \leq i \leq m$ let $f_i = g_i x_1^{N_i} + r_i$, such that g_i has no terms in x_1, and r_i has only terms in x_1 of degree $\leq N_i - 1$, where $N_i \geq 0$. Suppose $(a_2, \ldots, a_n) \in \mathbf{V}(\mathbf{I}_1(I))$. If $(a_2, \ldots, a_n) \notin \mathbf{V}(\{g_1, \ldots, g_m\})$ then there exists $a_1 \in k$ such that $(a_1, \ldots, a_n) \in \mathbf{V}(I)$. (See [3]).*

The following theorem provides an algorithm for computing the cardinality of a variety corresponding to a zero-dimensional ideal.

Theorem 4 (Counting Theorem) *Let k be a perfect field, W a variety in k^n, and $I = \mathbf{I}(W) \subset k[x_1, \ldots, x_n]$ a zero-dimensional ideal. The number of standard monomials of I equals the number of common zeroes of I. (See Theorem 8.32 of [1]).*

3 Constraint Satisfaction

This section reviews salient aspects of constraint satisfaction. A *Constraint Satisfaction Problem* (CSP) involves finding "value-assignments" for variables while satisfying a set of constraints [9]. Without loss of generality it is assumed that domains of variables are subsets of $k = \mathbb{C}$. This allows for constraints to be viewed as subsets of affine spaces, thus facilitating links to geometry and algebra.

CSPs are tuples (X, C), where $X = \{x_1, \ldots, x_n\}$ is a set containing the variables of the CSP and C is a set containing the constraints of the CSP. The constraints restrict the values the variables can take simultaneously. A value assignment is allowed if it "satisfies" each constraint in C. Traditionally CSPs also contain the domains of the variables. However, domains can be regarded as constraints [9]. A constraint network is a collection of variables and constraints on the variables. The arity of a constraint is the number of variables involved in that constraint.

Let $P = (X, C)$ be a CSP. Solutions of P are members of k^n. A solution of the form $x_1 = v_{x_1}, \ldots, x_n = v_{x_n}$ corresponds to $(v_{x_1}, \ldots, v_{x_n}) \in k^n$. Each member $(v_{x_1}, \ldots, v_{x_n}) \in k^n$ can also be written as $\sum_{x_i \in X} v_{x_i} e_{x_i}$, where $\cup_{x_i \in X} \{e_{x_i}\}$ is the canonical basis of k^n. Let $S \subseteq T \subseteq X$ and C_S be a constraint. $\sum_{x_i \in T} v_{x_i} e_{x_i} \in k^n$ *satisfies* C_S if $\sum_{x_i \in S} v_{x_i} e_{x_i} \in C_S$.

A constraint is called *unary* if its arity equals one. It is called *binary* if its arity equals two. A CSP is *binary* if all its constraints are of arity two or less.

Let (X, C) be a CSP, where $X = \{x_1, \ldots, x_n\}$. Let $\prec$ be an ordering such that $x_1 \prec \cdots \prec x_n$. The CSP is in *globally solved form w.r.t.* $\prec$ if either (1) the CSP is unsatisfiable and $\emptyset = C_{\{x_1\}} \in C$; or (2) the CSP is satisfiable and for all integers i, $1 < i \leq n$, if $(v_{x_1}, \ldots, v_{x_{i-1}})$ satisfies the CSP then there exists an member $v_{x_i} \in D_{x_i}$ such that $(v_{x_1}, \ldots, v_{x_i})$ satisfies the CSP as well.

A CSP in globally solved form can be solved efficiently. All its solutions can be found without encountering dead-ends by extending partial solutions. If it is unsatisfiable this can be found out easily by considering the unary constraint on the smallest variable (node-consistency).

The purpose of a backtrack-free search is to compute a solution without encountering a dead-end. [7] provides sufficient conditions and an algorithm for the case where the constraint-graph of the network is a tree. [6] generalises this for arbitrary binary constraint networks by relating the width of the constraint-graph to the level of (i,j)-consistency of the network.

As observed by [4] weaker properties may also ensure backtrack-free search. Directional consistency methods are proposed for binary constraint networks.

How to compute globally solved networks has been answered in [5]. The resulting networks are created by repeatedly adding constraints and restricting constraints by removing those partial solutions which cannot be extended.

4 Computing Elimination Networks

This section describes a transformation technique from any constraint network with extensional constraints to an equivalent constraint network which is in globally solved form w.r.t. some ordering on the variables. The resulting network corresponds to a reduced Gröbner basis w.r.t. a lexicographical term order, and is called an elimination network. Properties and possible improvements for the algorithm are discussed at the end of this section.

In the following let (X,C) be a CSP, where $X=\{x_1,\ldots,x_n\}$. Let D_{x_i} denote the domain of x_i for $1\leq i\leq n$, let $\prec$ be the lexicographical term order s.t. $x_1\prec\cdots\prec x_n$, and let $\mathcal{S}$ denote the solution set of the CSP. Note that $\mathcal{S}=\mathbf{V}(\sum_{C_S\in C}\mathbf{I}(C_S))$. For each step in the transformation process its postcondition is enclosed within the brackets [and]. The transformation is given by:

1. (a) For each $C_S\in C$ compute basis $B_S\subset k[x_1,\ldots,x_n]$ for $\mathbf{I}(C_S)$. [$\mathcal{S}=\mathbf{V}(\sum_{C_S}\mathbf{I}(B_S))$]
   (b) Set $B_X=\cup_{C_S\in C}B_S$. [$\mathcal{S}=\mathbf{V}(\mathbf{I}(B_X))$]
2. (a) Compute the reduced Gröbner basis $\mathcal{G}^r_\prec(\mathbf{I}(B_X))$. [$\mathcal{S}=\mathbf{V}(\mathcal{G}^r_\prec(\mathbf{I}(B_X)))$]
3. (a) For each polynomial g occurring in $\mathcal{G}^r_\prec(\mathbf{I}(B_X))$, let S_g denote its variables. Compute the maximal (w.r.t. inclusion) subset B'_{S_g} of polynomials in $\mathcal{G}^r_\prec(\mathbf{I}(B_X))$ the variables of which equal S_g. [$\mathcal{S}=\mathbf{V}(\mathbf{I}(\cup_{g\in\mathcal{G}^r_\prec(\mathbf{I}(B_X))}B'_{S_g}))=\mathbf{V}(\sum_{g\in\mathcal{G}^r_\prec(\mathbf{I}(B_X))}\mathbf{I}(B'_{S_g}))$]
   (b) If $\mathcal{G}^r_\prec(\mathbf{I}(B_X))=\{1\}$ then set $C'_{\{x_1\}}=\emptyset$ and C' to $\{C'_{\{x_1\}}\}$. Otherwise, for each B'_{S_g} computed in the previous step: compute $C'_{S_g}=\mathbf{V}(B'_{S_g})\cap\mathbf{X}_{x_i\in S_g}D_{x_i}$, where $\mathbf{X}$ denotes the Cartesian product operator. Set $C'=\{C'_{S_g}|g\in\mathcal{G}^r_\prec(\mathbf{I}(B_X))\}$.
   [$\mathcal{S}=\mathbf{V}(\sum_{C'_{S_g}\in C'}\mathbf{I}(C'_{S_g}))$.]

The resulting network corresponds to C'. The bases B_S in step 1 (a) can be computed with the `INTERSECTION` algorithm from [1].

If W is a variety with a finite cardinality, then $I=\mathbf{I}(W)$ is zero-dimensional. The Extension Theorem and the Triangular Form Theorem guarantee that extending non-empty partial solutions of elimination ideals of I must succeed. Similarly the existence of a unary constraint for the smallest variable is guaranteed. Therefor the network is in globally solved form w.r.t. $\prec$.

Gröbner bases are difficult to compute. Given a set F of generators of a zero-dimensional ideal it requires (worst case) $d^{O(n)}$ time to compute the Gröbner basis of the ideal generated by

F, where d is the maximum total degree of a monomial in F and n the number of variables. However, the following observations may be made:

- Computing elimination networks corresponds to finding all solutions of a CSP which is a difficult problem as well;
- It may be possible to exploit the fact that factorizations are known of the polynomials in the bases B_S in step 1 (b). The factorizations are known because these polynomials can be constructed by multiplying ideals of varieties which are known (each variety corresponds to a member of a constraint);
- The operations in the Buchberger algorithm correspond to addition, subtraction, multiplication and division of polynomials. Polynomials may be viewed as representations of ideals/varieties. It should be possible to use different representations for these ideals/varieties and define the equivalents of these operations for these different representations. Of course the notion of leading term w.r.t. a certain ordering should be preserved. This may improve the algorithm and may possibly eliminate the need of the transformations to and from the polynomial ring. E.g. multiplying two ideals corresponds to the union of their varieties, adding two ideals corresponds to intersecting their varieties, and so on;
- Certain problems have to be solved many times. To these problems the algorithm presented above can be applied as a preprocessing step.

5 An Example Application

In this section a constraint network is transformed to an elimination network using the technique described in the previous section.

The following constraints model a set of German traffic lights and are based on [8].

$$C_{\{v_i,p_i,v_{i+1 \bmod 4},p_{i+1 \bmod 4}\}} = \{(r,r,g,g),(r_y,r,y,r),(g,g,r,r),(y,r,r_y,r)\},$$
$$C_{\{v_i\}} = \{r,g,r_y,y\}, \qquad C_{\{p_i\}} = \{r,g\}, \qquad \text{for } i \in \{0,1,2,3\}.$$

The macro-structure of the network is depicted in Figure 1. The eight variables are given by

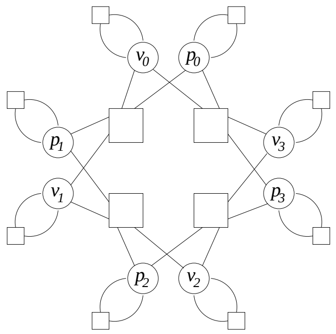

Figure 1: Original Network

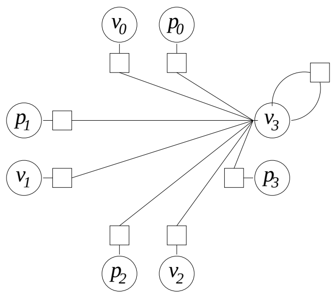

Figure 2: Elimination Network

p_0, p_1, p_2, p_3, v_0, v_1, v_2 and v_3. There are four 4-ary constraints $C_{\{v_0,p_0,v_1,p_1\}}$, $C_{\{v_1,p_1,v_2,p_2\}}$, $C_{\{v_2,p_2,v_3,p_3\}}$, $C_{\{v_3,p_3,v_0,p_0\}}$, and eight unary constraints corresponding to a domain of each of the variables. Assume $g = 0, r_y = 1, y = 2, r = 3$. The generating bases for the constraints are:

$$\begin{aligned}
&B_{\{v_i,p_i,v_{i+1 \bmod 4},p_{i+1 \bmod 4}\}} \\
&\quad = \{ \quad v_{i+1 \bmod 4}^4 - 6v_{i+1 \bmod 4}^3 + 11v_{i+1 \bmod 4}^2 - 6v_{i+1 \bmod 4}, v_i + v_{i+1 \bmod 4} - 3, \\
&\qquad\quad 2p_{i+1 \bmod 4} - v_{i+1 \bmod 4}^3 + 6v_{i+1 \bmod 4}^2 - 11v_{i+1 \bmod 4}, \\
&\qquad\quad 2p_i + v_{i+1 \bmod 4}^3 - 3v_{i+1 \bmod 4}^2 + 2v_{i+1 \bmod 4} - 6 \quad \} \\
&B_{\{v_i\}} = \{ \quad v_i^4 - 6v_i^3 + 11v_i^2 - 6v_i \quad \} \\
&B_{\{p_i\}} = \{ \quad p_i^2 - 3p_i \quad \} \qquad \text{for } i \in \{0, 1, 2, 3\}.
\end{aligned}$$

The union B_X of the bases consists of 12 binomials and 8 monomials. Let $\prec$ be the lexicographical term order given by $v_3 \prec v_2 \prec v_1 \prec v_0 \prec p_3 \prec p_2 \prec p_1 \prec p_0$. Then $\mathcal{G}_{\prec}^r(\mathbf{I}(B_X))$ equals:

$$\begin{aligned}
\{ \quad & v_3^4 - 6v_3^3 + 11v_3^2 - 6v_3, v_2 + v_3 - 3, v_1 - v_3, v_0 + v_3 - 3, 2p_3 - v_3^3 + 6v_3^2 - 11v_3, \\
& 2p_2 + v_3^3 - 3v_3^2 + 2v_3 - 6, 2p_1 - v_3^3 + 6v_3^2 - 11v_3, 2p_0 + v_3^3 - 3v_3^2 + 2v_3 - 6 \quad \}.
\end{aligned}$$

The reduced Gröbner basis has revealed a structure which was implicit in the original constraint network. Since the basis does not equal $\{1\}$, the network is satisfiable. The standard monomials of $\mathbf{I}(\mathcal{G}_{\prec}^r(\mathbf{I}(B_X)))$ are given by $\{1, v_3, v_3^2, v_3^3\}$. Since there are four of them the Counting Theorem guarantees that there are exactly four solutions to the constraint network.

The zeroes of each polynomial in the basis correspond to one of the following constraints of the resulting elimination network the macro-structure of which is depicted in Figure 2:

$$\begin{array}{llllll}
C'_{\{p_0,v_3\}} & = & \{(r,g),(r,r_y),(r,y),(g,r)\}, & C'_{\{p_1,v_3\}} & = & \{(g,g),(r,r_y),(r,y),(r,r)\}, \\
C'_{\{p_2,v_3\}} & = & \{(r,g),(r,r_y),(r,y),(g,r)\}, & C'_{\{p_3,v_3\}} & = & \{(g,g),(r,r_y),(r,y),(r,r)\}, \\
C'_{\{v_0,v_3\}} & = & \{(r,g),(y,r_y),(r_y,y),(g,r)\}, & C'_{\{v_1,v_3\}} & = & \{(g,g),(r_y,r_y),(y,y),(r,r)\}, \\
C'_{\{v_2,v_3\}} & = & \{(r,g),(y,r_y),(r_y,y),(g,r)\}, & C'_{\{v_3\}} & = & \{g,r_y,y,r\}.
\end{array}$$

6 Conclusions

In this paper, a new technique has been presented to transform extensional constraint networks to equivalent networks in globally solved form w.r.t. a certain variable ordering. The resulting networks correspond to reduced Gröbner basis for lexicographical term orders, and are called elimination networks.

If the elimination network is satisfiable, a backtrack-free search for the first solution exists and all solutions can be found without encountering dead-ends. The number of solutions of the network can be computed by counting the standard monomials of its reduced Gröbner basis.

Most of the time needed in the transformation process is spent on the computation of a Gröbner basis of a zero-dimensional ideal. The general problem of computing Gröbner bases is very difficult. Computing such bases for zero-dimensional ideals is much easier in practice. Suggestions have been presented on how to improve the algorithm.

7 Acknowledgements

The work reported here was funded by the European Commission under ESPRIT project number 20501, with acronym CEDAS. The author would like to thank Jim Bowen and Patrick Fitzpatrick for comments on an early draft of the paper.

References

[1] T. Becker and V. Weispfenning. *Gröbner Bases* A Computational Approach to Commutative Algebra. Graduate Texts in Mathematics. Springer, 1993.

[2] B. Buchberger. Gröbner bases: An algorithmic method in polynomial ideal theory. In R. Bose, editor, *Multidimensional Systems Theory*, Mathematics and Its Applications, chapter 6, pages 184–232. D.Reidel Publishing Company, 1985.

[3] D. Cox, J. Little, and J. O'Shea. *Ideals, Varieties, and Algorithms* An Introduction to Computational Algebraic Geometry and Commutative Algebra. Springer, 1996.

[4] R. Dechter and J. Pearl. Network-based heuristics for constraint-satisfaction problems. *Artificial Intelligence*, 34(1):1–38, 1988.

[5] R. Dechter and P. van Beek. Local and global relational consistency—summary of recent results. In U. Montanari and F. Rossi, editors, *PPCP*, volume 976 of *LNCS*, pages 240–257. Springer, 1995.

[6] E.C. Freuder. A sufficient condition for backtrack-bounded search. *Journal of the ACM*, 32(4):755–761, 1985.

[7] Eugene C. Freuder. A sufficient condition for backtrack-free search. *Journal of the ACM*, 29(1):24–32, 1982.

[8] W. Hower. Constraint satisfaction — algorithms and complexity analysis. *Information Processing Letters*, 55(3):171–178, 1995.

[9] A.K. Mackworth. Consistency in networks of relations. *Artificial Intelligence*, 8:99–118, 1977.