A Linear Constraint Satisfaction Approach to Cyclicity

Eugene Santos Jr.

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-92-03
January 1992

A Linear Constraint Satisfaction Approach to Cyclicity[1]

Eugene Santos Jr.
Department of Computer Science
Box 1910, Brown University
Providence, RI 02912

January 22, 1992

Abstract

Abductive reasoning (or explanation) is basically a backward-chaining process on a collection of causal rules. When given an observed event, we attempt to determine the set of causes which brought about this event. *Cost-based abduction* is a model for abductive reasoning which provides a concrete formulation of the explanation process. However, it restricts itself to acyclic causal knowledge bases. The existence of cyclicity results in anamolous behavior by the model. For example, assume in our knowledge base that A can cause B and B can cause A. When faced with having to explain the occurrence of A, we could postulate B. Now, since A already exists, we can use it the explain B. Our backward-chaining process to find an explanation can certainly fall into this trap. In this paper, we present a new model called *generalized cost-based abduction* for general causal knowledge bases. Furthermore, we provide an approach for solving this model by using *linear constraint satisfaction.*

[1]This work has been supported by the National Science Foundation under grant IRI-8911122 and by the Office of Naval Research, under contract N00014-88-K-0589. Special thanks to Eugene Charniak for important pointers and critical review of this paper.

1 Introduction

Explaining events such as the failure of our television set to turn on involves reasoning out the causes from the effects (the effects being the events we encountered). We can see this more clearly if we consider our causal knowledge base to consist of propositional rules of the form $A \Longrightarrow B$ where A is a cause and B is an effect. Given an event B, a possible explanation for B would be the cause A. Thus, explanation is performed through a backward-chaining process on the causal knowledge base.

Explanation (abduction) violates the logical rule of *modus ponens* since $A \Longrightarrow B$ and B does not necessarily imply A which certainly creates various problems. [1, 4, 15, 16, 17, 5, 2, 9, 8]. Abduction has been formally defined as the process of searching for a set of hypotheses which can "best" prove the given observation/events. Obviously, an enormous number of such hypotheses sets may be available as proofs. We choose only one such set which we "conclude" to be our "best" explanation. Thus, some ordering must have been imposed on these sets to make such a selection. Various models of abduction provide different approaches to the ordering problem.

In particular, we consider the *cost-based abduction* model [1], a minor variant of *weighted abduction* [4, 17]. We are given a causal knowledge base of propositional rules. A proposition which does not occur as the antecedent of any rule is called a *hypothesis*. Costs are then attached to these hypotheses to reflect the likelihood of assuming them true. A hypotheses set which can prove a given observation (evidence) is called an *explanation* for the observation. The cost of each hypotheses set is simply the sum of the costs of the hypotheses used. The best explanation is then taken to be the hypotheses set with least cost.

Cost-based abduction is restricted to those knowledge bases which are *acyclic* in nature. That is, there cannot be two propositions A and B in the knowledge base where both A can be used in a proof for B and B can be used in a proof for A. In the most degenerate case where $A \Longrightarrow B$ and $B \Longrightarrow A$ are both in the knowledge base, if we had as evidence that B is true, then A can be assigned true to prove B. Furthermore, since B is already true, we can use it now to prove A. Thus, no other propositions need to be assigned true to explain B! Clearly, this explanation is counter-intuitive and provides little information. Also, since none of the hypotheses are used, no cost is incurred which can make this explanation, the best explanation.

A more sophisticated example involving cyclicity arises when we are modeling faulty electrical outlets. Suppose that our television set and radio are both plugged into such an outlet. Being faulty, when the fuse is blown in one of the components, the accompanying surge causes the other fuse to also blow.

Yet another source which leads to cyclicity occurs when you include the logical rules:

$$(\mathsf{foo}\ a) \wedge (=\ a\ b\ ) \Longrightarrow (\mathsf{foo}\ b)$$
$$(\mathsf{foo}\ b) \wedge (=\ a\ b\ ) \Longrightarrow (\mathsf{foo}\ a)$$

Thus for the most part, knowledge bases for various domains are highly cyclic necessitating the need to correct the limitation in cost-based abduction.

Since abduction is a backward chaining process on the causal rules, the search for the best explanation in cost-based abduction can be performed as a graph searching problem. Starting from the evidence, we proceed backwards to the hypotheses through the causal implications. In this way, we build many partial proofs to use as guides for determining the least cost proof. Obviously, the construction of partial proofs serve as a heuristic in our search. However, it was shown that the search for a best explanation is NP-hard [1]. Introducing cyclicity further exacerbates the problem.

Recently, Santos [11, 13, 12] showed that cost-based abduction can be reduced to linear constraint satisfaction. The search for the best explanation could now be done with the highly efficient tools and techniques long developed and understood in Operations Research. Methods such as the *Simplex method* and *Karmarkar's projective scaling algorithm* could be used in our search [6, 7, 14]. Experimental results indicated that the linear constraint satisfaction approach was superior to existing graph search techniques [11, 13, 12] and actually exhibited an expected-case polynomial run-time.

In this paper, we present an approach to the problem of cyclicity in cost-based abduction. We arrived at our solution by studying cyclicity under linear constraint satisfaction. The solution itself represents a natural extension of our constrained optimization approach and remains a linear constraint satisfaction formulation.

The next section provides a brief overview of cost-based abduction and its reduction to linear constraint satisfaction. Complete details can be found in [11, 13, 12]. Next, Section 3 discusses the problems of cyclicity and presents a new model we call *generalized cost-based abduction*. In Section 4, we present our linear constraint satisfaction formulation of the new abduction model completing our constrained optimization solution to cyclicity.

2 Constraint Systems

The keystone of *cost-based abduction* [1] is the *weighted* AND/ OR *directed acyclic graph* (WAODAG) which models the causal relationships between objects and/or

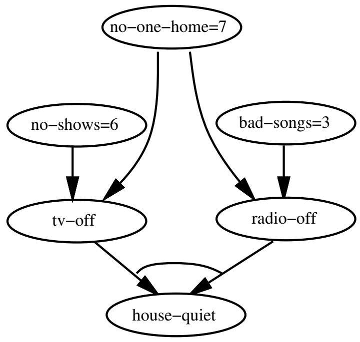

FIG. 2.1. A simpler WAODAG. The AND-node house-quiet is the observation. The nodes no-shows, blackout and bad-songs are the hypotheses with associated costs 6, 7 and 3, respectively.

concepts in the world. Each node in the dag alone embodies some object or concept while each edge represents direct causal relationships between nodes incident to the edge.

Since we are reasoning via abduction as opposed to deduction, our goal is to attempt to find the *best* set of hypotheses which can *prove* the given observation. A proof simply consists of a set of hypotheses plus some set of rules. In cost-based abduction, the *cost* of a proof is a measure on the set of hypotheses it uses. Each hypothesis is assigned a *cost* and the cost of a proof is the sum of the costs of all the hypotheses used. The best proof is then the one with minimal attached cost.

We now formalize the minimum cost-based abduction problem:
NOTATION. $\Re$ denotes the set of real numbers.

DEFINITION 2.1. *A* WAODAG *is a 4-tuple* (G, c, r, S), *where:*

1. *G is a directed acyclic graph,* $G = (V, E)$.
2. *c is a function from* $V \times \{\mathsf{true}, \mathsf{false}\}$ *to* $\Re$, *called the* cost function.
3. *r is a function from V to* $\{\text{AND}, \text{OR}\}$, *called the* label. *A node labeled* AND *is called an* AND-node, *etc.*
4. *S is a subset of nodes in V called the* evidence nodes.[2]

Clearly, an explanation is the same as a proof for cost-based abduction. We formally define a proof for a set of observations as an assignment of truth values to the different objects in the knowledge base such that the observations are true and

[2]S represents the set of observations to be explained. Without loss of generality, we only consider positive evidence.

the assignments are consistent with respect to the rule-based causal relationships between the objects.

In Figure 2.1, assume that house-quiet $=$ true is the observation to be explained. One possible proof would be the following assignment of truth values: {house-quiet, radio-off, bad-songs, tv-off, no-shows} are assigned true and {no-one-home} is assigned false. A second possibility is to assign all of them to true. And as a third possibility, we can assign {house-quiet, radio-off, tv-off, no-one-home} to true and {bad-songs, no-shows} to false. As we can easily see, all three assignments are internally consistent with the associated relationships and assign house-quiet to true.

More formally, we define this as follows:

DEFINITION 2.2. *A* truth assignment *for a* WAODAG $W = (G, c, r, S)$ *where* $G = (V, E)$ *is a function* e *from* V *to* {true, false}. *We say that such a function is* valid *iff (if and only if) the following conditions hold:*

1. *For all* AND*-nodes* q*,* $e(q) =$ true *iff for all nodes* p *such that* (p, q) *is an edge in* E*,* $e(p) =$ true*.*
2. *For all* OR*-nodes* q*,* $e(q) =$ true *iff there exists a node* p *such that* (p, q) *is an edge in* E *and* $e(p) =$ true*.*

Furthermore, we say that e *is an* explanation *iff* e *is valid and for each node* q *in* S*,* $e(q) =$ true*.*

As we mentioned earlier, the cost of a proof is simply the sum of the individual costs of the hypotheses assumed.[3]

DEFINITION 2.3. *We define the* cost of an explanation e *for* $W = (G, c, r, S)$ *where* $G = (V, E)$ *as*

$$C(e) = \sum_{q \in V} c(q, e(q)) \tag{1}$$

An explanation e *which minimizes* C *is called a* best explanation *for* W*.*

From (1), we find that our three proofs above have costs 9, 16 and 7, respectively. Of the three our best proof is the third one with the cost of 7. This proof also turns out to be the best explanation (minimal-cost proof) for Figure 2.1.

As we mentioned earlier, finding the minimal-cost proof can be a difficult task. Using brute force enumeration techniques is fine for small problems but grows exponentially in complexity (Charniak and Shimony [1] have proven that this problem is NP-complete.) This problem can be straightforwardly transformed into

[3]Although we have only discussed having costs associated with hypotheses, our approach permits costs to be associated with any node.

a graph search problem on AND/OR dags. However, efficient admissible heuristics seem to be difficult to find.

Basically, cost-based abduction is ultimately an *optimization* problem. So, instead of treating it as a traditional graph search problem, we consider the problem in terms of *constraint satisfaction*. In what follows, we will show how cost-based abduction can be directly transformed into a minimization problem on a collection of linear constraints. Central to this transformation is the formulation of *constraint systems* which can be shown to correctly solve the minimum cost-based abduction by determining the minimal-cost proof.

DEFINITION 2.4. *A* constraint system *is a 3-tuple* (Γ, I, ψ) *where* Γ *is a finite set of variables, I is a finite set of linear inequalities based on* Γ, *and* ψ *is a function from* $\Gamma \times \{\mathsf{true}, \mathsf{false}\}$ *to* $\Re$.

NOTATION. For each node q in V, let $D_q = \{p | (p, q)$ *is an edge in* $E\}$, the parents of q. $|D_q|$ is the cardinality of D_q.

From Definition 2.2, for a truth assignment to be a possible explanation, we must guarantee the internal consistency of the assignments required in the definition.

We formalize our construction as follows:

DEFINITION 2.5. *Given a* WAODAG $W = (G, c, r, S)$ *where* $G = (V, E)$, *we can construct a constraint system* $L(W) = (\Gamma, I, \psi)$ *where:*

1. Γ *is a set of variables indexed by* V, *i.e.,* $\Gamma = \{x_q | q \in V\}$,
2. $\psi(x_q, X) = c(q, X)$ *for all* $q \in V$ *and* $X \in \{\mathsf{true}, \mathsf{false}\}$,
3. *I is the collection of all inequalities of the forms given below:*

$$x_q \leq x_p \in I \text{ for each } p \in D_q \text{ if } r(q) = \text{AND} \tag{2}$$

$$\sum_{p \in D_q} x_p - |D_q| + 1 \leq x_q \in I \text{ if } r(q) = \text{AND} \tag{3}$$

$$\sum_{p \in D_q} x_p \geq x_q \in I \text{ if } r(q) = \text{OR} \tag{4}$$

$$x_q \geq x_p \in I \text{ for each } p \in D_q \text{ if } r(q) = \text{OR} \tag{5}$$

We say that L(W) is induced *by W. Furthermore, by including the additional constraints:*

$$x_q = 1 \text{ if } q \in S \tag{6}$$

we say that the resulting constraint system is induced evidentially *by W and is denoted by* $L_E(W)$.

DEFINITION 2.6. *A* variable assignment *for a constraint system* $L = (\Gamma, I, \psi)$ *is a function s from* Γ *to* $\Re$. *Furthermore,*

1. *If the range of s is $\{0,1\}$, then s is a* 0-1 assignment.
2. *If s satisfies all the constraints in I, then s is a* solution *for L.*
3. *If s is a solution for L and is a 0-1 assignment, then s is a* 0-1 solution *for L.*

With our formulation of constraint systems and variable assignments, we can now prove that 0-1 solutions are equivalent to explanations. Given a 0-1 assignment s for $L(W)$, we can construct a truth assignment e for W as follows:

1. For all q in V, $s(x_q) = 1$ iff $e(q) = \mathsf{true}$.
2. For all q in V, $s(x_q) = 0$ iff $e(q) = \mathsf{false}$.

Conversely, given a truth assignment e for W, we can construct a 0-1 assignment s for $L(W)$.

NOTATION. $e[s]$ and $s[e]$ denote, respectively, a truth assignment e constructed from a 0-1 assignment s, and a 0-1 assignment s constructed from a truth assignment e.

THEOREM 2.1. *e is an explanation for W iff $s[e]$ is a 0-1 solution for $L_E(W)$.*

To complete the formulation, we must also be able to compute the costs associated with each proof. We can do this as follows:

DEFINITION 2.7. *Given a constraint system $L(W) = (\Gamma, I, \psi)$ induced (evidentially) by a* WAODAG *W, we construct a function Θ_L from variable assignments to $\Re$ as follows:*

$$\Theta_L(s) = \sum_{x_q \in \Gamma} \{s(x_q)\psi(x_q, \mathsf{true}) + (1 - s(x_q))\psi(x_q, \mathsf{false})\}. \tag{7}$$

Θ_L is called the objective function *of $L(W)$.*

DEFINITION 2.8. *An* optimal 0-1 solution *for a constraint system $L(W) = (\Gamma, I, \psi)$ induced (evidentially) by a* WAODAG *W is a 0-1 solution which minimizes Θ_L.*

As we can clearly see, (7) is identical to (1). From Theorem 2.1 and the relationship between node assignments and variable assignments, an optimal 0-1 solution in $L_E(W)$ is a best explanation for W and vice versa.

Taking the set of linear inequalities I and the objective function Θ_L, we observe that we have the elements known in operations research as a *linear program* [14, 7, 6]. Highly efficient methods such as the *simplex method*[4] and *Karmarkar's projective scaling algorithm* are used to solve linear programs [14, 7, 6].

Experiments performed in [11, 13, 12] comparing our approach against existing heuristic search techniques has shown ours to be the superior technique. Also, the

[4]For a quick overview of the simplex method, see [3].

studies strongly suggested that our constraint satisfaction approach is practical and efficient for solving cost-based abduction problems.

3 Generalized Cost-Based Abduction

We now return to address the issue of cyclicity and present a generalization to cost-based abduction. Consider the following set of rules:

$$\begin{array}{l} \text{tv-off} \wedge \text{radio-off} \Longrightarrow \text{house-quiet} \\ \text{tv-fuse-blown} \Longrightarrow \text{tv-off} \\ \text{radio-fuse-blown} \Longrightarrow \text{radio-off} \\ \text{tv-fuse-blown} \wedge \text{Socket-4} \Longrightarrow \text{radio-fuse-blown} \\ \text{radio-fuse-blown} \wedge \text{Socket-4} \Longrightarrow \text{tv-fuse-blown} \\ \text{tube-blown} \Longrightarrow \text{tv-fuse-blown} \\ \text{wire-short} \Longrightarrow \text{radio-fuse-blown} \end{array}$$

where tv-fuse-blown, radio-fuse-blown represents the fuses inside them, tube-blown represents the picture tube being blown, wire-short represents a shorted connection in the radio, and Socket-4 represents that both the television and radio are plugged into the same electrical outlet. Basically, this rule set is suppose to model the case where we have a faulty outlet and that when one of the components attached fails, it also causes the other to fail. We can easily see that this set of rules is cyclic.

Again, observe that if Socket-4 is true, then tv-fuse-blown could be explained by radio-fuse-blown and vice versa. Invariably, this is an explanation under cost-based abduction and will most likely be the best one.

Intuitively, to avoid this type of "self-supporting" anomaly, a proper explanation must guarantee that some "outside" agent be present, such as either tube-blown to explain tv-fuse-blown or wire-short to explain radio-fuse-blown, when tv-fuse-blown, radio-fuse-blown and Socket-4 are all true. For this fairly simple case, we can easily enumerate the desired behavior as follows:

- When tv-fuse-blown, radio-fuse-blown and Socket-4 are all true, then either tube-blown or wire-short must be true.
- When tv-fuse-blown = true and Socket-4 = $false$, then tube-blown must be true.
- When radio-fuse-blown = true and Socket-4 = $false$, then wire-short must be true.
- When tv-fuse-blown = radio-fuse-blown = false and Socket-4 is either true or false, then nothing special needs to be done.
- The remaining states are inconsistent and must be prevented from occurring.

From the above behavior list, we can make the following very important observation: Causal reasoning is an inherently time-dependent process since obviously causes must temporally precede effects. Thus, when attempting to explain the occurrence of some given event, we are temporally ordering all the other events which lead up to it. Acyclicity simply gives us a single unique partial ordering of the events where as cyclicity actually provides multiple partial orderings. In our above example, consider the first behavioral item and pick wire-short to be true. What we have effectively done is choose the following sequence of events:

1. A short circuit occurred inside the radio.
2. The short circuit caused the fuse to blow.
3. Since television is plugged into the same socket as the radio which is a very old electrical outlet, the television fuse blew also.

When we consider the problem in terms of our cost-based abduction graphs, we find that our WAODAGs clearly reflect unique partial orderings. Any explanations constructed under cost-based abduction properly determined the sequence of causal events. By adding cyclicity, these new more general graphs can be viewed as collections of WAODAGs. Our goal now is to be able to form explanations which properly determine the sequence of causal events.

We now present our new model of cost-based abduction called *generalized cost-based abduction.*

DEFINITION 3.1. *A* cost-based graph *is a 4-tuple* (G, c, r, S)*, where*

1. G *is a directed graph,* $G = (V, E)$.
2. c *is a function from* $V \times \{\mathsf{true}, \mathsf{false}\}$ *to* $\Re$*, called the* cost function.
3. r *is a function from* V *to* $\{\text{AND}, \text{OR}\}$*, called the* label.
4. S *is a subset of* $V \times \{\mathsf{true}, \mathsf{false}\}$ *called the* evidence.

Clearly, a cost-based graph is a generalization of our original WAODAGs.

DEFINITION 3.2. *Given a cost-based graph* $Z = (G, c, r, S)$ *where* $G = (V, E)$*, a* solution graph $G_s = (V_s, E_s)$ *for* Z *is a subgraph of* G *which satisfies the following conditions:*

1. G_s *is acyclic.*
2. *If* $q \in V_s$ *and* $r(q) = \text{AND}$*, then* $D_q \subseteq V_s$ *and* $(p, q) \in E_s$ *for all* $p \in D_q$.
3. *If* $D_q \subseteq V_s$ *and* $r(q) = \text{AND}$*, then* $q \in V_s$ *and* $(p, q) \in E_s$ *for all* $p \in D_q$.
4. *If* $q \in V_s$ *and* $r(q) = \text{OR}$*, then there exists a node* p *in* D_q *such that* $p \in V_s$ *and* $(p, q) \in E_s$.
5. *If* $D_q \cap V_s \neq \phi$ *and* $r(q) = \text{OR}$*, then* $q \in V_s$ *and there exists a node* $p \in D_q \cap V_s$ *such that* $(p, q) \in E_s$.
6. *If* $(q, \mathsf{true}) \in S$*, then* $q \in V_s$.
7. *If* $(q, \mathsf{false}) \in S$*, then* q *is not in* V_s.

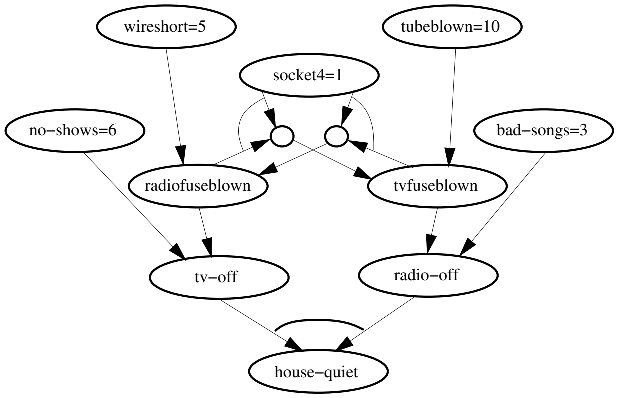

FIG. 3.1. A cost-based graph for our fuses example.

Basically, solution graphs correspond to explanations in that they represent all the nodes which are to be assigned true. Nodes not included in a solution graph are assumed to be set to false. The condition of acyclicity will guarantee that we have a proper sequencing of causal events.

DEFINITION 3.3. *We define the* cost *of a solution graph* $G_s = (V_s, E_S)$ *for* $Z = (G, c, r, S)$ *where* $G = (V, E)$ *as*

$$C(G_s) = \sum_{q \in V_s} c(q, \mathsf{true}) + \sum_{q \in V - V_s} c(q, \mathsf{false}). \tag{8}$$

A solution graph G_s *which minimizes* C *is called a* best explanation *for* Z*.*

We have now completely defined our *generalized cost-based abduction* model. We can also easily see that our original cost-based abduction model is a special case of our new model.

Let's return to our fuse example above and consider the associated cost-based graph (see Figure 3.1). Our "self-supporting" anomaly arises when we have the following truth assignment to our nodes: house-quiet = true, tv-off = radio-off = *true*, radio-fuse-blown = tv-fuse-blown = true, no-shows = bad-songs = false, Socket-4 = true and wire-short = tube-blown = false. We can easily see that no subgraph of our cost-based graph containing the nodes which are assigned true in our anomaly is a solution graph. They all end up violating either the edge inclusion requirements or the acyclicity constraint.

4 Constraints Formulation

We now consider how we might reduce generalized cost-based abduction to linear constraint satisfaction. As we mentioned at the beginning of this paper, our work on cyclicity was motivated by our constrained optimization view on cost-based abduction. We were trying to see how we might manipulate linear constraints in order to achieve our desired behaviors. We found that we could properly satisfy the behavior list in the previous section by modifying the original set of constraints generated by Definition 2.5.

We first eliminate any constraints involving tv-fuse-blown and radio-fuse-blown. We then introduce the following new constraints:

$$\begin{aligned}
\textsf{tv-fuse-blown} + \textsf{Socket-4} - 1 &\leq \textsf{radio-fuse-blown} \\
\textsf{radio-fuse-blown} + \textsf{Socket-4} - 1 &\leq \textsf{tv-fuse-blown} \\
\textsf{tv-fuse-blown} + \textsf{radio-fuse-blown} + \textsf{Socket-4} - 2 &\leq \textsf{tube-blown} + \textsf{wire-short} \\
\textsf{tv-fuse-blown} - \textsf{Socket-4} &\leq \textsf{tube-blown} \\
\textsf{radio-fuse-blown} - \textsf{Socket-4} &\leq \textsf{wire-short} \\
\textsf{tv-fuse-blown} &\geq \textsf{tube-blown} \\
\textsf{radio-fuse-blown} &\geq \textsf{wire-short}
\end{aligned}$$

Clearly, we can associate these constraints directly to the items in the list. We can readily show that this new set of constraints will result in the behaviors we dictated. Thus, our solution demonstrates that our linear constraint satisfaction approach can be extended to handle at least the cyclicity problem in the previous section. However, when we proceed to more general cyclicity problems which may involve interactions between large numbers of propositions, using this sort of brute force enumeration of relationships is certainly infeasible. Even with moderate problems, this approach grows exponentially in the number of constraints we would need to introduce. Just consider the introduction of a fish bowl on top of the television set: If the tube blows, the explosion knocks the fish bowl over. If the fish bowl falls over, then the water pours into the set causing the tube to blow.

Obviously, we need a general solution to cyclicity in linear constraint satisfaction which could compactly capture the essence of our problem. We now present our solution.

We begin by first taking a cost-based graph $Z = (G, c, r, S)$ where $G = (V, E)$ and generating all the constraints according to Definition 2.5 with appropriate modifications for the evidence information. Designate this set of constraints by $L(Z)$. Now, we identify all the directed cycles in G. Let's call this collection of directed cycles, X. Next, for each cycle y in X, we perform the following cumulative modifications to $L(Z)$: Assume $y = \{a_1, a_2, \ldots, a_n\}$ which represents all the nodes in the directed path and a_n will also point to a_1 to complete the cycle. For each pair of nodes (a_i, a_{i+1}),

1. If $r(a_{i+1}) = \text{OR}$, then construct a new 0-1 integer variable x_i and
   - Mark x_i.
   - Take the equation generated by (4) for a_{i+1} and replace the real variable associated with a_i with the new variable x_i.
   - Add the new constraint $x_i \leq x_{a_i}$ where x_{a_i} is the real variable associated with a_i.
2. If $r(a_{i+1}) = \text{AND}$, then mark the real variable associated with a_{i+1}.

Now, let $b_1, b_2, \ldots, b_n$ be the real variables marked for y. We introduce the new constraint $\sum_{i=1}^{n} b_i \leq n - 1$.

DEFINITION 4.1. *$L(Z)$ constructed above is the constraint system induced by Z.*

Disallowing the presence of causal cycles in our solution graphs is analogous to removing the associated 0-1 assignments from our 0-1 solution space. Intuitively, the new variables introduced in our above construction serve to indicate whether or not the nodes are causally linked, that is, whether one node is directly used to prove the other. For an AND-node, when it is true, then all its parents must be used. However, for an OR-node, only some of its parents need be used. A situation can arise where although a parent is true, it is not used as a proof for the child. This must certainly occur in the presence of cycles. By taking these new variables plus the AND-nodes of a cycle, we can determine whether a causal chain is legitimate using the last constraint constructed above.

We now go about and prove that our induced constraint system will indeed determine the best explanation in our cost-based graph. By making the appropriate translations of 0-1 solutions to explanations and solution graphs to explanations, we can prove the following.

THEOREM 4.1. *$G_s = (V_s, E_s)$ is a solution graph for $Z = (G, c, r, S)$ where $G = (V, E)$ iff s is a 0-1 solution for $L(Z)$.*

THEOREM 4.2. *G_s is a best explanation iff s is an optimal 0-1 solution.*

The above theorems show that we can solve generalized cost-based abduction by reducing it to linear constraint satisfaction.

Let's return to our earlier cyclicity example in Figure 3.1. In our new set of constraints, we have added the following to deal with the cyclicity:

$$\begin{aligned} \text{radio-fuse-blown} &\leq \text{wire-short} + m_2 \\ m_2 &\leq \text{AND}_2 \end{aligned}$$

$$\begin{aligned}
\textsf{tv-fuse-blown} &\leq \textsf{tube-blown} + m_1 \\
m_1 &\leq \textsc{and}_1 \\
m_1 + \textsc{and}_1 + m_2 + \textsc{and}_2 &\leq 3
\end{aligned}$$

where $\textsc{and}_1$ and $\textsc{and}_2$ are the AND-nodes and m_1, m_2 are the special marker variables. We can easily see that the anomalous truth assignment causes $\textsc{and}_1 = \textsc{and}_2 = m_1 = m_2 = 1$ which will violate the last constraint above, thus demonstrating that our linear constraints properly eliminates the "self-supporting" assignment.

The only requirement in our linear constraint satisfaction approach to solving generalized cost-based abduction with cyclicity is the generation of all the cycles in the cost-based graph. However, we note that the problem of generating the cycles can be done in $O((|V| + |E|)(n+1))$ where n is the number of cycles generated. Basically, a depth-first search is used whereby through an elegant approach to adding edges will generate each cycle. Details and analysis of the algorithm can be found in [10].

5 Conclusions

We have considered the problem of cyclicity with regards to cost-based abduction and presented generalized cost-based abduction to appropriately incorporate the phenomenon. This was accomplished by studying cyclicity from the standpoint of linear constraint satisfaction which allowed us to concretely identify the problem. Our goal was to determine whether we could manipulate the constraints from the original cost-based abduction problem to handle cyclicity. Indeed, we demonstrated that the problem could be solved through a natural extension of our linear constraint satisfaction approach. Furthermore, the extensions required seemed quite compact in size demonstrating the power inherent in our approach.

References

[1] Eugene Charniak and Solomon E. Shimony. Probabilistic semantics for cost based abduction. In *Proceedings of the AAAI Conference*, pages 106–111, 1990.

[2] Michael R. Genesereth. The use of design descriptions in automated diagnosis. *Artificial Intelligence*, pages 411–436, 1984.

[3] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. Holden-Day, Inc., 1967.

[4] Jerry R. Hobbs, Mark Stickel, Paul Martin, and Douglas Edwards. Interpretation as abduction. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, 1988.

[5] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *Proceedings of the AAAI Conference*, 1986.

[6] C. McMillan. *Mathematical Programming*. John-Wiley & Sons, Inc., 1975.

[7] G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors. *Optimization: Handbooks in Operations Research and Management Science Volume 1*, volume 1. North Holland, 1989.

[8] Judea Pearls. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.

[9] Y. Peng and J. A. Reggia. *Abductive Inference Models for Diagnostic Problem-Solving*. Springer-Verlag, 1990.

[10] Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo. *Combinatorial Algorithms: Theory and Practice*. Prentice-Hall, Inc., 1977.

[11] Eugene Santos, Jr. Cost-based abduction and linear constraint satisfaction. Technical Report CS-91-13, Department of Computer Science, Brown University, 1991.

[12] Eugene Santos, Jr. Cost-based abduction, linear constraint satisfaction, and alternative explanations. In *Proceedings of the AAAI Workshop on Abduction*, 1991.

[13] Eugene Santos, Jr. A linear constraint satisfaction approach to cost-based abduction. Submitted to Artificial Intelligence Journal, 1991.

[14] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons Ltd., 1986.

[15] Bart Selman and Hector J. Levesque. Abductive and default reasoning: A computational core. In *Proceedings of the AAAI Conference*, pages 343–348, 1990.

[16] Murray Shanahan. Prediction is deduction but explanation is abduction. In *Proceeding of IJCAI*, 1989.

[17] Mark E. Stickel. A prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Technical Report Technical Note 451, SRI International, 1988.

A Proofs

THEOREM 4.1. *$G_s = (V_s, E_s)$ is a solution graph for $Z = (G, c, r, S)$ where $G = (V, E)$ iff s is a 0-1 solution for $L(Z)$.*

Proof. Formally, we must first prove that given any solution graph $G_s = (V_s, E_s)$ for $Z = (G, c, r, S)$ where $G = (V, E)$ we can construct a 0-1 solution s for $L(Z) = (\Gamma, I, \psi)$ such that $p \in V_s$ if and only if $s(x_p) = 1$.

We begin by satisfying the condition that $p \in V_s$ if and only if $s(x_p) = 1$. Next, consider the special marker variables used in our construction for $L(Z)$ to deal with cyclicity. For example, let $\{p_1, p_2, \ldots, p_n\}$ be a cycle in G. We denote $m_{p_i p_{i+1}}$ to be the special real variable introduced between p_i and p_{i+1} when $r(p_{i+1}) = \text{OR}$. Although, p_i and p_{i+1} may adjacently appear in other cycles, without loss of generality, we assume that we have only created one such $m_{p_i p_{i+1}}$ to be repeatedly used in all the cycles. Now, we continue our construction of s as follows:

- If $(p, q) \in E_s$ and there exists a corresponding $m_{pq} \in \Gamma$, then $s(m_{pq}) = 1$.
- If (p, q) is not in E_s and there exists a corresponding $m_{pq} \in \Gamma$, then $s(m_{pq}) = 0$.

s is now a 0-1 assignment for $L(Z)$.

Assume s is not a 0-1 solution for $L(Z)$. This implies at least one of the following constraint forms have been violated by s:

1. $x = b$ where $x \in \Gamma$ and $b \in \{0, 1\}$.
2. $x \leq y$ where $x, y \in \Gamma$.
3. $x \leq y_1 + y_2 + \ldots + y_n$ where $x, y_1, \ldots, y_n \in \Gamma$.
4. $x \geq y_1 + y_2 + \ldots + y_n - n + 1$ where $x, y_1, \ldots, y_n \in \Gamma$.
5. $x_1 + x_2 + \ldots + x_n \leq n - 1$ where $x_1, \ldots, x_n \in \Gamma$.

- CASE 1: $x = b$ has been violated. Without loss of generality, assume that $b = 1$. This implies that $s(x) = 0$. Since this constraint is only generated the evidence in S, this implies that there is some node p in V such that $x \equiv x_p$ where x_p is the real variable associated with p in $L(Z)$. Now, $\{x_p = 1\} \in I \Rightarrow \{p, \mathsf{true}\} \in S \Rightarrow p \in V_s \Rightarrow s(x_p) = 1$. However, $x \equiv x_p$. Contradiction.
- CASE 2: $x \leq y$ has been violated. This implies that $s(x) = 1$ and $s(y) = 0$. This constraint can be generated from one of the following sources:
  - From equation (2), we see that x and y are the real variables associated with some nodes q and p in V, respectively. Furthermore, $r(q) = \text{AND}$ and $(p, q) \in E$. Now, $s(x) = 1$ and $s(y) = 0$ imply $q = \mathsf{true}$ and $p = \mathsf{false}$ which further implies that $q \in V_s$ but p is not in V_s. However,

this violates Definition 3.2 thus implying that G_s is not a solution graph. Contradiction.

  – From equation (5), we see that x and y are the real variables associated with some nodes p and q in V, respectively. Furthermore, $r(q) = \text{OR}$ and $(p, q) \in E$. Now, $s(x) = 1$ and $s(y) = 0$ imply $p = \mathsf{true}$ and $q = \mathsf{false}$ which further implies that $p \in V_s$ but q is not in V_s. However, this violates Definition 3.2 thus implying that G_s is not a solution graph. Contradiction.
  – From our construction for dealing with cyclicity, we find that y is the real variable associated with some node p_i in V and x is a special marker variable created between node p_i and some node p_{i+1} in V and $x \equiv m_{p_i p_{i+1}}$. Now, $s(x) = 1$ and $s(y) = 0$ imply that $(p_i, p_{i+1}) \in E_s$ and $p_i = \mathsf{false}$. However, $(p_i, p_{i+1}) \in E_s$ implies that $p_i \in V_s$. Contradiction.

- CASE 3: $x \leq y_1 + \ldots + y_n$ has been violated. This implies that $s(x) = 1$ and $s(y_i) = 0$ for $i = 1, \ldots, n$. We can easily see that this type of constraint is generated either by the equation (4) or the modification(s) introduced in our cyclicity construction. Regardless of either techniques, we know that x is a real variable associated with some node p in V and the $r(p) = \text{OR}$. Furthermore, $s(x) = 1 \Rightarrow p \in V_s$.
  If this constraint had been generated directly from (4), we know that each y_i is a real variable associated with some node q_i in V. Furthermore, $s(y_i) = 0$ implies that q_i is not in V_s which violates Definition 3.2. Thus, G_s is not a solution graph. Contradiction.
  If generated by our cyclicity constraints, then some of the y_i's is associated to some special marker variable $m_{q_i p}$. Now, $s(y_i) = 0 \Rightarrow s(m_{q_i p}) = 0$ which further implies that (q_i, p) is not in E_s. For the remaining y_i's which are associated to some node q_i in V, $s(y_i) = 0$ implies that q_i is not in V_s. This also implies that (q_i, p) is not in E_s. However, this violates Definition 3.2. Thus, G_s is not a solution graph. Contradiction.
- CASE 4: $x \geq y_1 + \ldots + y_n - n + 1$ has been violated. This implies that $s(x) = 0$ and $s(y_i) = 1$ for all $i = 1, \ldots, n$. Since this constraint can only be generated by equation (3), x is the real variable associated with some node p in V, $r(p) = \text{AND}$ and y_i is associated with some q_i in V for all i. This implies that $q_i \in V_s$ for all i but p is not in V_s which violates Definition 3.2. Thus, G_s is not a solution graph. Contradiction.
- CASE 5: $x_1 + \ldots + x_n \leq n - 1$ has been violated. This implies that $s(x_i) = 1$ for all $i = 1, \ldots, n$. Finally, this constraint is created to handle cyclicity. It is directly associated with some cycle $\{p_1, \ldots, p_{n-1}\}$ in G. Now, for $i = 1, \ldots, n-1$ each x_i either represents the AND-node p_{i+1} or the special marker

variable $m_{p_ip_{i+1}}$. If x_i is associated to the AND-node p_{i+1}, then $p_{i+1} \in V_s$ and according to Definition 3.2, $(p_i, p_{i+1}) \in E_s$. If x_i is associated to $m_{p_ip_{i+1}}$, then $s(x_i) = 1$ implies that $(p_i, p_{i+1}) \in E_s$. Finally, x_n is a special case joining p_{n-1} and p_1 which will imply that $(p_{n-1}, p_1) \in E_s$. However, this implies that G_s has a cycle. Contradiction.

All the cases have now been covered. Thus s is a 0-1 solution for $L(Z)$.

Now, we must finish our proof by showing that for each 0-1 solution s for $L(Z)$, we can construct a solution graph G_s for Z such that $s(x_p) = 1$ iff $p \in V_s$.

We begin our construction of G_s by satisfying the condition that $s(x_p) = 1$ iff $p \in V_s$. Next, for each node $q \in V$, if $r(q) = \text{AND}$ and $s(x_q) = 1$, then for all $p \in D_q$ such that $s(x_p) = 1$, $(p, q) \in E_s$. For each pair of nodes p, q in V, if $r(q) = \text{OR}$, $s(x_p) = 1$, $s(x_q) = 1$ and $p \in D_q$,

- If there does not exist a special marker variable m_{pq} in Γ, then $(p, q) \in E_s$.
- If there exists a special marker variable m_{pq} in Γ such that $m_{pq} = 1$, then $(p, q) \in E_s$.

We finish the construction by stating that any edge in E not added to E_s by one of the above conditions will not be in E_s.

Assume that $G_s = (V_s, E_s)$ is not a solution graph for Z. This implies that one or more of the following conditions have been violated:

1. If $(q, \mathsf{true}) \in S$, then $q \in V_s$.
2. If $(q, \mathsf{false}) \in S$, then q is not in V_s.
3. If $q \in V_s$ and $r(q) = \text{AND}$, then $D_q \subseteq V_s$ and $(p, q) \in E_s$ for all $p \in D_q$.
4. If $D_q \subseteq V_s$ and $r(q) = \text{AND}$, then $q \in V_s$ and $(p, q) \in E_s$ for all $p \in D_q$.
5. If $q \in V_s$ and $r(q) = \text{OR}$, then there exists a node p in D_q such that $p \in V_s$ and $(p, q) \in E_s$.
6. If $D_q \cap V_s \neq \phi$ and $r(q) = \text{OR}$, then $q \in V_s$ and there exists a node $p \in D_q \cap V_s$ such that $(p, q) \in E_s$.
7. G_s is acyclic.

- CASE 1: If $(q, \mathsf{true}) \in S$, then $q \in V_s$.
  q not in V_s implies that $s(x_q) = 0$. However, according to our construction of $L(Z)$, if $(q, \mathsf{true}) \in S$, then $s(x_q) = 1$. Contradiction.
- CASE 2: If $(q, \mathsf{false}) \in S$, then q is not in V_s.
  $q \in V_s$ implies that $s(x_q) = 1$. However, according to our construction of $L(Z)$, if $(q, \mathsf{false}) \in S$, then $s(x_q) = 0$. Contradiction.
- CASE 3: If $q \in V_s$ and $r(q) = \text{AND}$, then $D_q \subseteq V_s$ and $(p, q) \in E_s$ for all $p \in D_q$.
  This implies that either $D_q - V_s \neq \phi$ or there exists some $p \in D_q$ such that (p, q) is not in E_s. If $D_q - V_s \neq \phi$, then let $p \in D_q - V_s$. Now, $s(x_p) = 0$

and $s(x_q) = 1$. However, since $r(q) = \text{AND}$, we have the constraint $x_q \leq x_p$ in $L(Z)$. Contradiction.
D_q must be a subset of V_s. This implies that from our construction of G_s, for all $p \in D_q$, $(p,q) \in E_s$. Contradiction.

- CASE 4: If $D_q \subseteq V_s$ and $r(q) = \text{AND}$, then $q \in V_s$ and $(p,q) \in E_s$ for all $p \in D_q$.
  This implies that either q is not in V_s or there exists some $p \in D_q$ such that (p,q) is not in E_s. If q is not in V_s, then $s(x_q) = 0$. Also, $s(x_p) = 1$ for all $p \in D_q$. However, since $r(q) = \text{AND}$, we have the constraint $x_q \geq \sum_{p \in D_q} x_p - |D_q| + 1$. Contradiction.
- CASE 5: If $q \in V_s$ and $r(q) = \text{OR}$, then there exists a node p in D_q such that $p \in V_s$ and $(p,q) \in E_s$.
  This implies that for all $p \in D_q$, either p is not in V_s or (p,q) is not in E_s. We know that $s(x_q) = 1$ and $r(q) = \text{OR}$. This implies that we have a constraint of the form $x_q \leq y_1 + \ldots + y_n$ where each y_i represents one of the following:
  - y_i is the real variable associated with the node p_i in V.
  - y_i is the special marker variable associated with the nodes p_i and q in V and is called $m_{p_i q}$.

  Note, $D_q = \{p_1, \ldots, p_n\}$ where p_i is related to y_i.
  According to our constraint, there exists some $s(y_i) = 1$. If $y_i \equiv x_{p_i}$, then $p_i \in V_s$ and according to our construction of G_s, $(p_i, q) \in E_s$, since there is no special marker associated to p_i and q. Contradiction. If $y_i \equiv m_{p_i q}$, then we also have the constraint that $m_{p_i q} \leq x_{p_i}$ in $L(Z)$. This implies that $s(x_{p_i}) = 1$ and according to our construction of G_s, $(p_i, q) \in E_s$. Contradiction.
- CASE 6: If $D_q \cap V_s \neq \phi$ and $r(q) = \text{OR}$, then $q \in V_s$ and there exists a node $p \in D_q \cap V_s$ such that $(p,q) \in E_s$.
  This implies that either q is not in V_s or for all $p \in D_q \cap V_s$, (p,q) is not in E_s. We know that for all $p \in D_q$, $s(x_p) = 1$ and $r(q) = \text{OR}$. From our construction of $L(Z)$, we have the constraints $x_p \leq x_q$ for all $p \in D_q$. Since $D_q \cap V_s \neq \phi$, $s(x_q) = 1$ which implies that $q \in V_s$. Thus, this leaves us with for all $p \in D_q \cap V_s$, (p,q) is not in E_s. Now, for each $p \in D_q \cap V_s$, if there is no m_{pq}, then $(p,q) \in E_s$. If there is an m_{pq}, then m_{pq} must be 0. Looking at the constraint for OR-node q, $x_q \leq y_1 + \ldots + y_n$ where y_i is either associated with a node in D_q or a special marker variable, we find that all the y_i's must be set to 0. Contradiction since $s(x_q) = 1$.
- CASE 7: G_s is acyclic.
  This implies that there exists a cycle $\{p_1, \ldots, p_n\}$ of nodes in G_s. Without loss of generality, assume that all the nodes are OR-nodes. A cycle in G_s is

of course a cycle in G. By our cyclicity construction, we have special marker variables $m_{p_i p_{i+1}}$ between the adjacent nodes in the cycle. The cycle implies that $(p_i, p_{i+1}) \in E_s$ for all i including the edge (p_n, p_1). By our construction of G_s, $s(m_{p_i p_{i+1}}) = 1$ for all i. However, we have the constraint

$$m_{p_1 p_2} + \ldots + m_{p_{n-1} p_n} + m_{p_n p_1} \leq n - 1.$$

Contradiction.

Contradiction. G_s is a solution graph for Z. □

THEOREM 4.2. *G_s is a best explanation iff s is an optimal 0-1 solution.*

Proof. Follows from Theorem 4.1 and the fact that our costs computations are identical. □