Representative Explanations for Over-Constrained Problems

Barry O'Sullivan and **Alexandre Papadopoulos**
Cork Constraint Computation Centre
University College Cork, Ireland
{b.osullivan|a.papadopoulos}@4c.ucc.ie

Boi Faltings and **Pearl Pu**
Ecole Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
{boi.faltings|pearl.pu}@epfl.ch

Abstract

In many interactive decision making scenarios there is often no solution that satisfies all of the user's preferences. The decision process can be helped by providing *explanations*. Relaxations show sets of consistent preferences and, thus, indicate which preferences can be enforced, while exclusion sets show which preferences can be relaxed to obtain a solution. We propose a new approach to explanation based on the notion of a *representative* set of explanations. The size of the set of explanations we compute is exponentially more compact than that found using common approaches from the literature based on finding all minimal conflicts.

Introduction

We consider a configuration tool where a user can specify preferences for options. These preferences are expressed as constraints. When preferences conflict, we want to help the user find which preferences to relax. In an iterative process, the user might relax constraints until at least one consistent solution is found. Alternatively, the user might prefer to select a solution from a list of solutions that partially satisfy the user's constraints. It would be good to categorise solutions according to which constraints are satisfied/violated, and the benefits of that have been shown in earlier work (Pu, Faltings, & Torrens 2004). However, this requires that they are in some way representative.

Most current approaches to explanation generation in constraint-based settings are based on the notion of a (set-wise) minimal set of unsatisfiable constraints, also known as a minimal conflict set of constraints. However, a minimal conflict does not necessarily give an intuitive explanation, in that many users will want to be shown which subsets of their constraints they can satisfy and which they cannot satisfy. Furthermore, we argue that users need more than one explanation in order to avoid drawing false conclusions. It has also been shown that minimal conflict-based explanations can also be spurious and misleading (Friedrich 2004).

Example 1. *Consider a simple car configuration problem, based on an example presented in (Junker 2004), with the following set of options; note that the Boolean variable* $x_i \in \{0, 1\}$ *indicates whether constraint* c_i *is in the current set of active constraints or not:*

	Option	*Selector*	*Cost*
c_1	*Budget*	$x_1 = 1$	$\sum_{i \in \{2,\ldots,5\}} (k_i \cdot x_i) \leq 3000$
c_2	*Roof Rack*	$x_2 = 1$	$k_2 = 500$
c_3	*Convertible*	$x_3 = 1$	$k_3 = 500$
c_4	*CD Player*	$x_4 = 1$	$k_4 = 500$
c_5	*Leather Seats*	$x_5 = 1$	$k_5 = 2600$

Assume that the technical constraints of the configuration problem forbid convertible cars having roof racks, therefore, constraints c_2 *and* c_3 *form a conflict. Note that, given the budget constraint, if the user selects option* c_5*, it is not possible to have any of the options* c_2, c_3, c_4*.* ▲

The set of all minimal conflicts for this example are: $\{c_2, c_3\}, \{c_1, c_2, c_5\}, \{c_1, c_3, c_5\}, \{c_1, c_4, c_5\}$. As explanations, these conflicts are sufficient to explain, using a subset of the user's constraints, why all constraints cannot be satisfied simultaneously. However, it is not necessarily sufficient to remove one of the constraints in a minimal conflict, thus eliminating the conflict, to regain consistency.

Consider, for example, what happens if we present $\{c_1, c_2, c_5\}$ as an explanation for why the set of constraints $\{c_1, \ldots, c_5\}$ is not satisfiable. The user would be mistaken in thinking that simply eliminating this conflict by removing, say, constraint c_2 is enough to recover consistency. It is not, because $\{c_1, c_3, c_5\}$ is also a conflict. Similarly, relaxing c_5 from $\{c_1, c_2, c_5\}$ would not have been enough because $\{c_2, c_3\}$ is a conflict. Minimal conflicts only explain why a set of constraints is inconsistent. In order to recover consistency all minimal conflicts must be eliminated by relaxing a set of constraints that form a hitting set of the conflicts.

Instead, we propose offering as an explanation a set of maximal consistent subsets of the user's constraints and their corresponding sets of constraints that must be excluded. We define a notion of representativeness that ensures the set of explanations we compute is representative of all possible maximal relaxations and exclusion sets. Our approach ensures that the worst-case size of our set of representative explanations is linear in the number of user posted constraints.

The Approach

We present a sequence of examples, based on Example 1, demonstrating the approach we propose in this paper. Table 1 presents the set of all explanations, each showing how the user can satisfy at least some of his constraints. Each

explanation comprises a (set-wise) maximal consistent relaxation, i.e. the set of constraints that can be satisfied, and the corresponding minimal exclusion set of constraints that must be excluded. For example, consider Explanation I: we can simultaneously satisfy the constraints in $\{c_3, c_4, c_5\}$, but we must exclude c_1 and c_2.

Table 1: The set of relaxations and exclusion sets for the over-constrained problem presented in Example 1. We show both the subset of the constraints in the relaxation (marked with a ✓) and those that are in the exclusion set, i.e. those that must be removed (marked with a ×).

	Constraints						
Exp.	c_1	c_2	c_3	c_4	c_5	Relaxation	Exclusion Set
I	×	×	✓	✓	✓	$\{c_3, c_4, c_5\}$	$\{c_1, c_2\}$
II	×	✓	×	✓	✓	$\{c_2, c_4, c_5\}$	$\{c_1, c_3\}$
III	✓	×	✓	✓	×	$\{c_1, c_3, c_4\}$	$\{c_2, c_5\}$
IV	✓	✓	×	✓	×	$\{c_1, c_2, c_4\}$	$\{c_3, c_5\}$
V	✓	×	×	×	✓	$\{c_1, c_5\}$	$\{c_2, c_3, c_4\}$

However, we cannot hope to be able to present the user with every possible relaxation/exclusion set for his set of preferences, because the number of maximal relaxations is exponential in the number of user constraints in the worst case.

Theorem 1 (Worst-case Number of Maximal Relaxations)**.** *Given a inconsistent set of n constraints, the worst-case number of set-wise maximal relaxations is* $\binom{n}{\lfloor n/2 \rfloor}$.

Proof. Immediate from Sperner's Theorem. □

The best we can hope for is to, therefore, present a subset of all possible relaxations/exclusion sets of the user's constraints. For example, we might require that every constraint that appears in a relaxation appears at least once in a relaxation in our chosen subset. This is the scenario presented in Table 2. However, this approach has the potential to mislead the user. In our example, the user might believe that it is never useful to exclude c_4, and might miss that it is possible to get option c_5 while still satisfying the budget constraint c_1, drawing the wrong conclusion.

Table 2: A set of representative relaxations.

	Constraints				
Explanation	c_1	c_2	c_3	c_4	c_5
I	×	×	✓	✓	✓
II	×	✓	×	✓	✓
III	✓	×	✓	✓	×

A similar problem arises if we present a set of explanations that ensures that every constraint that must be relaxed once, appears in at least one exclusion set as shown in Table 3. Here, the user might be lead to believe that constraint c_2 must always be excluded.

Instead, we propose that the subset of explanations that are presented to the user should be both representative of the relaxations and exclusion sets of the problem. Specifically, a set of explanations should be presented that contains at least one maximal relaxation containing each constraint that can be satisfied, and at least one minimal exclusion set containing each constraint that must be excluded at least once. The set of solutions must satisfy the property that a constraint should only appear in all relaxations iff it is always satisfied, or that one should only appear in all exclusion sets iff it never participates in a maximal relaxation. A set of explanations that satisfies these properties is presented in Table 4.

Table 3: A set of representative exclusion sets.

	Constraints				
Explanation	c_1	c_2	c_3	c_4	c_5
I	×	×	✓	✓	✓
III	✓	×	✓	✓	×
V	✓	×	×	×	✓

Table 4: A set of representative explanations.

	Constraints				
Explanation	c_1	c_2	c_3	c_4	c_5
I	×	×	✓	✓	✓
IV	✓	✓	×	✓	×
V	✓	×	×	×	✓

Representative Explanations

We focus on constraint satisfaction problems in this paper, but the results hold for many other settings in which consistency is monotonic. In other words, the set of solutions to a set of constraints $\mathcal{C}$ is a subset of the solutions to any set of constraints that are a subset of $\mathcal{C}$.

In addition, we focus on constraint satisfaction problems that are solved in an interactive manner, e.g. product configuration problems. It is useful to distinguish between a background set of constraints, $\mathcal{B}$ that cannot be relaxed, and a set of constraints, $\mathcal{U}$, that are added by the user as he finds a preferred solution to $\mathcal{B}$ by finding a solution to $\mathcal{B} \cup \mathcal{U}$, the constraint problem we denote as $\mathcal{P} \stackrel{\text{def}}{=} \langle \mathcal{B}, \mathcal{U} \rangle$.

We say that a set of constraints is consistent if it admits a solution. We will assume that the set of background constraints, $\mathcal{B}$, admits at least one solution. If a set of constraints does not admit a solution, at least one constraint must be excluded in order to recover consistency. Specifically, we are interested in finding *maximal relaxations* of $\mathcal{P}$.

Definition 1 (Maximal Relaxation)**.** *Given a constraint problem $\mathcal{P} \stackrel{\text{def}}{=} \langle \mathcal{B}, \mathcal{U} \rangle$ that is inconsistent, a subset R of $\mathcal{U}$ is a* relaxation *of $\mathcal{P}$ if $\mathcal{B} \cup R$ admits a solution. The relaxation R is a* maximal relaxation *if $\forall R' \supset R$, $\mathcal{B} \cup R'$ is inconsistent.*

Based on the notion of maximal relaxation, we can also define the complementary notion of minimal exclusion set.

Definition 2 (Minimal Exclusion Set of Constraints)**.** *Given a constraint problem $\mathcal{P} \stackrel{\text{def}}{=} \langle \mathcal{B}, \mathcal{U} \rangle$ that is inconsistent, and a*

maximal relaxation $R \subseteq \mathcal{U}$ of $\mathcal{P}$, we define $E \stackrel{\text{def}}{=} \mathcal{U} \setminus R$ to be a minimal exclusion set.

Let $\mathcal{R}$ and $\mathcal{E}$ be the set of all maximal relaxations and minimal exclusion sets of $\mathcal{P}$, respectively. Note that every pair of maximal relaxations are set-wise incomparable, i.e. for all $R_i, R_j \in \mathcal{R} : i \neq j, R_i \not\subseteq R_j$ and $R_i \not\supseteq R_j$. Similarly, all pairs of exclusion sets in $\mathcal{E}$ are set-wise incomparable.

A maximal relaxation defines a maximal set of constraints that the user can satisfy, while the corresponding minimal exclusion set tells the user which constraints he must remove. However, the user might not be satisfied with an arbitrary explanation. Since the size of $\mathcal{R}$ can be exponential in the number of constraints in $\mathcal{U}$, presenting all explanations is not practical in general. We are, therefore, interested in selecting a subset of the explanations that are "representative" of all possibilities.

Definition 3 (Representative Set of Explanations)**.** *Given a constraint problem $\mathcal{P} \stackrel{\text{def}}{=} \langle \mathcal{B}, \mathcal{U} \rangle$ that is inconsistent, $\mathcal{R}' \subseteq \mathcal{R}$ a set of maximal relaxations of $\mathcal{P}$, and $\mathcal{E}' = \{\mathcal{U} \setminus R | R \in R'\}$ the corresponding set of minimal exclusion sets, we say that $\mathcal{X} \stackrel{\text{def}}{=} \{(R, \mathcal{U} \setminus R) | R \in \mathcal{R}'\}$ is a representative set of explanations iff $\forall c \in \bigcup_{R \in \mathcal{R}} R$ there exists a relaxation in $\mathcal{R}'$ containing c, and $\forall c \in \bigcup_{E \in \mathcal{E}} E$ there exists an exclusion set in $\mathcal{E}'$ containing c.*

Clearly, the example set presented in Table 4 is a representative set of explanations since it contains a relaxation (an exclusion set) containing each constraint that appears in a relaxation (respectively, an exclusion set).

Of course, we wish to restrict the size of our representative sets to an optimal size. It is more computationally efficient to focus on set-wise minimal sets of explanations. We therefore, define the notion of minimal representative set.

Definition 4 (Minimal Representative Set of Explanations)**.** *Given a constraint problem $\mathcal{P} \stackrel{\text{def}}{=} \langle \mathcal{B}, \mathcal{U} \rangle$ that is inconsistent, we say that a set of representative set of explanations is minimal if any strict subset of it is not representative.*

Theorem 2 (Number of Explanations)**.** *The size of any minimal representative set of explanations is at most $|\mathcal{U}|$, and this bound is tight.*

Proof. Let $(\mathcal{R}, \mathcal{E})$ be a minimal set of explanations, and let $\mathcal{R} = R_1, ..., R_m$ and $\mathcal{E} = E_1, ..., E_m$. R_2 is incomparable to R_1 so it must contain a constraint that is not in R_1, and E_2 is incomparable to E_1 so it must contain a constraint not in E_1. Let $|R_1| = k$ and thus $|E_1| = |\mathcal{U}| - k$. Since $(\mathcal{R}, \mathcal{E})$ is minimal, for every explanation (R_i, E_i) either R_i has to cover a constraint not covered by any other R_j or E_i has to cover a constraint not covered by any other E_j. There are only $|\mathcal{U}| - k - 1$ constraints not covered by R_1 or R_2, and $k - 1$ constraints not covered by E_1 or E_2, so there are no more than $|\mathcal{U}| - 2$ such constraints to be covered by the remaining $m - 2$ explanations. Thus, $m \leq |\mathcal{U}|$. To show that the bound is tight, consider an example where the only minimal conflict is the set $\mathcal{U}$. Then there are $|\mathcal{U}|$ minimal exclusion sets, each containing one of the constraints. □

We can generalise all the above definitions to richer forms of representativeness. For example, we might want to insist that all pairs, triples, etc. of constraints appearing in a relaxation or exclusion set also appear as such in our representative explanations. However, we will not consider these generalisations further in this paper due to limitations on space.

Complexity

We assume a polynomial consistency checker Π. This assumption holds for consistency algorithms such as arc consistency on tree-structured problems. However, in general the assumption serves as a basis for establishing the baseline complexity classes of the basic decision problems we are interested in here.

To ensure that we have found a representative set of explanations, we must at least find a minimal exclusion set for each constraint, provided that each constraint appears in at least one minimal exclusion set. Unfortunately, this decision problem is *NP*-Complete.

Theorem 3 (Complexity)**.** *Given an inconsistent constraint problem $\mathcal{P} \stackrel{\text{def}}{=} \langle \mathcal{B}, \mathcal{U} \rangle$, a procedure Π for testing consistency of a set of constraints in polynomial time, and a constraint $c \in \mathcal{U}$, deciding whether there is a relaxation R such that $c \notin R$ is NP-Complete.*

Proof. Given a relaxation R, we can test in polynomial time whether it is maximal by running the consistency checker Π on $R \cup \{c'\}$ for every $c' \notin R$. We can easily test that $c \notin R$. Thus, the problem is in *NP*. To show completeness, we will use a reduction from 3SAT. Let $x_1, \ldots, x_n$ be the variables and f the CNF formula of an instance of 3SAT. We build an instance of the problem of deciding whether there exists a maximal relaxation of an inconsistent problem that does not contain a given constraint. For each variable x_i, let c_i (resp. c_i') be the constraint that enforces $x_i = 1$ (resp. $x_i = 0$) and $\perp$ the constraint that holds iff the variables form an instantiation that violates one of the clauses of f. We define $\mathcal{P}$ with $\mathcal{U} = (\cup_{i=1}^{n} \{c_i, c_i'\}) \cup \{\perp\}$ and no background constraint. Clearly, $\mathcal{P}$ is inconsistent. There are 2^n maximal relaxations, each corresponding to a different value assignment to each of the n variables. Let R be a relaxation corresponding to an assignment of the n variables. This assignment satisfies f iff $\perp$ cannot be added to R, i.e. R is a maximal relaxation. Therefore, f has a solution iff there is a maximal relaxation not containing $\perp$. □

Corollary. *Finding a minimal representative set of explanations is NP-Hard.*

The Algorithm

The complexity results show us that we cannot expect to have a polynomial algorithm for computing representative explanations, unless *P = NP*. The algorithm we propose is based on a modification of an existing algorithm for computing all minimal exclusion sets. As a post-processing step, we can reduce the set of minimal exclusion sets to obtain a minimal representative set of explanations. This post-processing, which we call *minimisation*, involves greedily removing explanations not required to ensure representativeness.

Amongst the best algorithms for computing all minimal conflict sets of constraints are (Bailey & Stuckey 2005)

and (Gregoire, Mazure, & Piette 2007). The latter is an algorithm tailored for SAT problems, while the former has the advantage here in that it is for general constraints. Therefore, we based our algorithm on (Bailey & Stuckey 2005).

Algorithm 1: REPRESENTATIVEXPLAIN

```
   Data: P =def (B,U), an inconsistent problem, B consistent.
   Result: X a minimal representative set of explanations.
1  X ← ∅
2  E ← RepresentativeDA(P)
3  foreach c ∈ U do
4      if sat(B ∪ {c}) ∧ (∀E ∈ E, c ∈ E) then
5          R ← grow({c}, U \ {c})
6          X ← X ∪ {(U \ R, E)}
7  minimise(X)

8  function RepresentativeDA(P)
9      C ← ∅
10     E ← ∅
11     H ← {∅}
12     R ← ∅
13     U ← U
14     repeat
15         R ← grow(R, U \ U)
16         R ← grow(R, U)
17         if (U \ R) ∩ U ≠ ∅ then
18             U ← U \ (U \ R)
19             E ← E ∪ {U \ R}
20         H ← Min(H ⊗ {{c}, c ∈ U \ R})
21         R ← ∅
22         for H ∈ H \ C do
23             if sat(B ∪ H) then
24                 R ← H
25                 break
26             else C ← C ∪ {H}
27     until (U = ∅ ∨ R = ∅)
28     return E

29 function grow(S, M)
30     foreach c ∈ M \ S do
31         if sat(B ∪ S ∪ {c}) then S ← S ∪ {c}
32     return S

33 function Min(H)
34     if H = ∅ then return ∅
35     Let H ∈ H s.t. ∀H' ∈ H, |H| ≤ |H'|
36     return {H} ∪ Min({H' ∈ H / H ⊈ H'})
```

The key property that is exploited in the algorithm is the relationship between exclusion sets and conflicts: if $\mathcal{E}$ is the set of all minimal exclusions and $\mathcal{C}$ is the set of all minimal conflicts, then $\mathcal{E}$ is the set of all minimal hitting sets of $\mathcal{C}$, and vice versa. Intuitively, if we remove all the constraints in an exclusion set, we break all the minimal conflicts, and therefore restore consistency.

The `grow()` function (Lines 29-32) takes a consistent subset as a seed and grows it to a maximal relaxation by greedily adding from the remaining constraints in M those that do not create inconsistency, as detected by our propagator, called by the `sat()` function.

At each point of the iteration of the **repeat** loop in function REPRESENTATIVEDA (Line 14), $\mathcal{H}$ contains all the minimal hitting sets of the current $\mathcal{E}$. The important property used in this function is that when $\mathcal{E}$ contains some but not all of the minimal exclusions, there must be at least one consistent minimal hitting set in $\mathcal{H}$. This is because if $\mathcal{R}$ is the set of maximal relaxations corresponding to the minimal exclusions in $\mathcal{E}$, $\mathcal{H}$ is precisely the set of all minimal subsets incomparable with any element of $\mathcal{R}$. Because there is at least one maximal relaxation not yet in $\mathcal{R}$, and all maximal relaxations are incomparable, then this new relaxation must be a super-set of an element in $\mathcal{H}$. Therefore, the condition $R = \emptyset$ is satisfied iff all elements in $\mathcal{H}$ are inconsistent iff $\mathcal{E}$ contains all the minimal exclusions. The hitting sets are incrementally computed in Line 20 by computing the cross product of two sets, where $\mathcal{H}_1 \otimes \mathcal{H}_2 = \{H_1 \cup H_2 / H_1 \in \mathcal{H}_1 \wedge H_2 \in \mathcal{H}_2\}$, and then minimising the result, where `Min()` keeps all set-wise minimal elements of the given set.

The algorithm used in REPRESENTATIVEDA is an adaptation of Bailey and Stuckey's (2005) Dualize and Advance algorithm. The essentials of the modification are that we change the termination condition at Line 27 to either (a) having covered all the user constraints, or (b) having found all minimal exclusions, whichever occurs first. Clearly, we want that all user constraints be covered as quickly as possible, and so try to first add constraints not in U so that as many as possible in U will be in the exclusion set (Line 15).

Of course, when not all the user constraints belong to at least one minimal exclusion, only condition (b) is met, and we cannot expect to do much better than generating all minimal exclusion sets. However, even in this case, our algorithm can still find a set of explanations quickly, while the rest of the effort is spent proving representativeness.

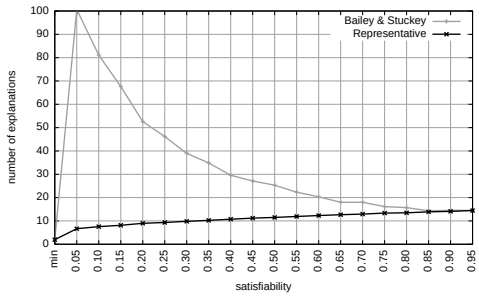
Experiments

We studied the performance of our algorithm on both random and real-world problems[1]. The objective of the experiment was to study the reduction in the number of explanations one achieves by considering a minimal set of representative explanations, as computed using REPRESENTATIVEXPLAIN, rather than all minimal exclusions, as computed using (Bailey & Stuckey 2005), which we will refer to as the *baseline algorithm*.

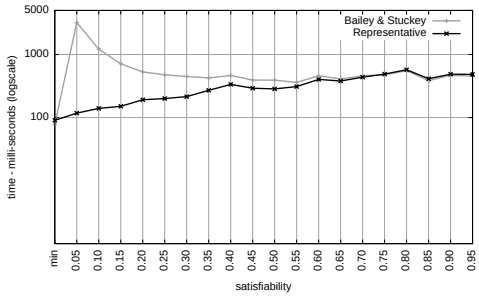
The random problems consisted of fifteen boolean variables with one 15-ary background constraint defined on those variables. We varied the proportion of all possible assignments that were consistent with this background constraint, i.e. its satisfiability, taking 20 settings in all. Since we have assumed that the background constraints are always consistent, the least satisfiable background constraint accepted only one assignment.

For a given satisfiability, we generated 10 randomly generated background constraints. For every background constraint, we generated 10 inconsistent queries for which each algorithm generated a set of explanations. Each query was

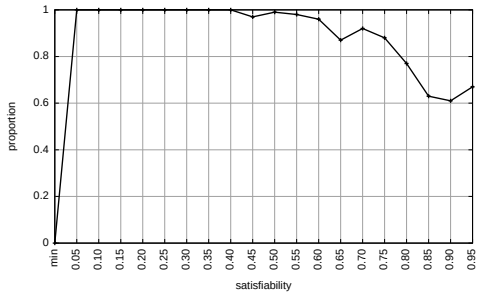
[1]Experiments were implemented in Java and run on an iMac (2.33GHz Intel Core 2 Duo), 3GB RAM, running Mac OSX 10.4.8.



(a) Cardinality of the sets of explanations.



(b) Times required to generate sets of explanations.



(c) Proportion of queries per instances in which all constraints were involved in at least one exclusion set.

Figure 1: Results for a series of random problems.

defined in terms of a set of user constraints that assigned a random value to each variable such that the whole set of constraints was inconsistent. We plot the average results over 100 queries at each satisfiability setting in Figure 1.

Consider the size of the representative set of explanations (Figure 1(a)). For most settings of satisfiability we observe a significant gap between the total number of exclusions, as found by the Bailey and Stuckey algorithm, and the number of representative explanations found by our algorithm. We noted that in the vast majority of cases, the set of explanations was already almost always (set-wise) minimal, and were already representative.

From Figure 1(b) we can see that the difference between algorithms in terms of running time mimics the difference in the size of the sets of explanations they generate. Note that REPRESENTATIVEXPLAIN can avoid enumerating all relaxations if all user constraints are involved in at least one exclusion. We refer to instances in which this occurs as "true" instances. As we highlighted before, we can hope for a potentially large decrease in the execution time on these "true" instances. Figure 1(c) confirms this, as we see that the difference in running times tends to decrease as the proportion of "true" instances decreases.

To analyse this behaviour more deeply on pure "false" instances, we ran a second kind of experiment. We used exactly the same process for generating random instances before, and just added a trivially satisfied constraint to each instance, so that it belongs to all maximal relaxations. The results are presented in Figure 2. On these instances we cannot hope to be much faster than a full enumeration of all maximal relaxations. We measured two different times: the time when the last relaxation has been found by REPRESENTATIVEXPLAIN and the time when it terminates, i.e. the time to find a representative set of explanations and the time to also prove representativeness, respectively. Here again, the results are positive. We observe that we actually find a representative set much faster than it takes to find all relaxations. However, unexpectedly, REPRESENTATIVEXPLAIN can terminate a little quicker than the baseline algoritm.

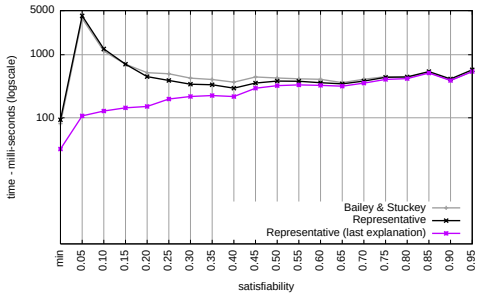


Figure 2: Average times for finding all relaxations, a representative set of explanations and the last explanation.

We also ran experiments on a real-world problem, the Renault Megane car configuration problem (Amilhastre, Fargier, & Marquis 2002). This problem is defined by 99 variables and has 2.8×10^{12} solutions. We extracted four problem instances of this problem by restricting it in the following way. We ordered the variables by increasing domain size. Then, by a dichotomic search, we instantiated the variables with the largest domain sizes in order to reduce the number of solutions to a more reasonable level for an interactive application, while still honouring the real world structure of the problem. The restricted instances of the problem provided four possible sets of background constraints, reducing the number of solutions by a factor of $10^6, 10^7, 10^8$ and 10^9 in each case. We compiled each instance into an automaton, similar to that presented in (Amilhastre, Fargier, & Marquis 2002). The user's set of constraints was generated by randomly assigning 30 of the remaining uninstantiated problem variables. The results of this experiment are presented in Table 5; the instances are labelled by the reduction factor in the number of solutions as compared with the original Renault problem. For each instance of the background constraint we computed explanations for 15 incon-

Table 5: The results for the Renault problem.

	Baseline		REPRESENTATIVEXPLAIN		
Instance	time	#exps	time last	time all	#exp
renault 10^6	474.76	17	318.87	618.76	3
renault 10^7	263.95	11	125.51	324.71	3
renault 10^8	205.82	8	97.98	232.32	3
renault 10^9	293.00	12	139.67	350.51	3

sistent queries, presenting the medians (due to the size of the instances) in this table.

These results confirm those on the random problems, except on one point. All of the instances are "false" instances, which was expected, and the total running time of REPRESENTATIVEXPLAIN is a little higher than the time for the baseline algorithm. This can be explained by the fact that REPRESENTATIVEXPLAIN performs more set operations (intersection and filtering on U) which have not been optimised for this bigger instance. However, the important results that the set of representative explanations is much more compact than the set of all explanations, and the last explanation can be found faster, are very encouraging.

Related Work

There have been many technical papers about explanation (Amilhastre, Fargier, & Marquis 2002; Friedrich 2004; Junker 2004; Bowen 1997; Freuder *et al.* 2003; O'Callaghan, O'Sullivan, & Freuder 2005; Sqalli & Freuder 1996; Pu, Faltings, & Torrens 2004). The dominant approach to explanation in configuration is based on computing minimal conflicting sets of constraints. However, as discussed in the introduction, minimal conflicts have disadvantages. Our approach addresses these disadvantages by showing the user a representative set of constraints that can be satisfied provided some other constraints are excluded.

Approaches have been proposed that attempt to be more "helpful" by presenting users with partial consistent solutions (Pu, Faltings, & Torrens 2004), or advise on how to relax constraints in order to achieve consistency (O'Callaghan, O'Sullivan, & Freuder 2005). Our approach is complementary to these by providing a basis for presenting a representative subset of choices facing the user.

Recent work has focused on finding minimal unsatisfiable subproblems in temporal problems (Liffiton *et al.* 2005), satisfiability (Liffiton & Sakallah 2005; Gregoire, Mazure, & Piette 2007) and type error debugging (Bailey & Stuckey 2005). These techniques find all minimal unsatisfiable sets of constraints, which can be exponential in the number of constraints. Our work is complementary to these approaches: we propose a principled basis for compacting the set of all explanations into an interesting set of explanations that has size linear in the number of user constraints.

Conclusions and Future Work

We have proposed the notion of a representative set of explanations. A representative set of explanations in this context means that every constraint that can be satisfied is shown in a relaxation and every constraint that must be excluded is shown in an exclusion set. We presented an algorithm for computing a minimal representative set of explanations, and demonstrated its performance on a variety of random and real-world problem instances.

Our future work will focus on developing suitable user interfaces for presenting representative explanations to users, informed by in-depth user studies. Also, we will study how to inform our choice of representative explanations by considering the user's preferences over constraints. Another direction of work will be to study how to generate explanations in the context of optimisation problems.

Acknowledgements. This work was supported by Science Foundation Ireland (Grant No. 05/IN/I886) and the Swiss National Science Foundation (Grant No. 200020-111888).

References

Amilhastre, J.; Fargier, H.; and Marquis, P. 2002. Consistency restoration and explanations in dynamic CSPs application to configuration. *Artif. Intell.* 135(1-2):199–234.

Bailey, J., and Stuckey, P. J. 2005. Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In *Proceedings of PADL*, 174–186.

Bowen, J. 1997. Using dependency records to generate design coordination advice in a constraint-based approach to concurrent engineering. *Computers in Industry* 22(1):191–199.

Freuder, E. C.; Likitvivatanavong, C.; Moretti, M.; Rossi, F.; and Wallace, R. J. 2003. Computing explanations and implications in preference-based configurators. In *Recent Advances in Constraints*, LNAI 2627, 76–92.

Friedrich, G. 2004. Elimination of spurious explanations. In *Proceedings of ECAI*, 813–817.

Gregoire, E.; Mazure, B.; and Piette, C. 2007. Boosting a complete technique to find mss and mus thanks for a local search oracle. In *Proceedings of IJCAI*, 2300–2305.

Junker, U. 2004. QuickXplain: preferred explanations and relaxations for over-constrained problems. In *Proceedings of AAAI*, 167–172.

Liffiton, M. H., and Sakallah, K. A. 2005. On finding all minimally unsatisfiable subformulas. In *Proceedings of SAT*, 173–186.

Liffiton, M. H.; Moffitt, M. D.; Pollack, M. E.; and Sakallah, K. A. 2005. Identifying conflicts in overconstrained temporal problems. In *Proceedings of IJCAI*, 205–211.

O'Callaghan, B.; O'Sullivan, B.; and Freuder, E. C. 2005. Generating corrective explanations for interactive constraint satisfaction. In *Proceedings of CP*, 445–459.

Pu, P.; Faltings, B.; and Torrens, M. 2004. Effective interaction principles for online product search environments. In *Web Intelligence*, 724–727.

Sqalli, M. H., and Freuder, E. C. 1996. Inference-based constraint satisfaction supports explanation. In *Proceedings of AAAI*, 318–325.