
Les triangles cassés, encore et encore

Martin C. Cooper¹Achref El Mouelhi²Cyril Terrioux²¹ IRIT, Université de Toulouse III, 31062 Toulouse, France² Aix-Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296,
13397 Marseille, France

cooper@irit.fr {achref.elmouelhi, cyril.terrioux}@lsis.org

Résumé

Les triangles cassés représentent un concept important, non seulement pour la résolution en temps polynomial des problèmes de satisfaction de contraintes, mais aussi pour l'élimination de variables ou encore la réduction de la taille des domaines par la fusion des valeurs. Plus précisément, pour une variable donnée d'un CSP binaire arc-cohérent, si aucun triangle cassé ne se produit sur aucun couple de valeurs, alors cette variable peut être éliminée tout en préservant la satisfiabilité. Plus récemment, il a été démontré qu'en cas de non applicabilité de cette règle à cause de l'existence d'au moins un triangle cassé, il se peut qu'il existe deux valeurs de cette même variable sur lesquelles aucun triangle cassé ne s'est produit. Dans ce cas, ces deux valeurs peuvent être fusionnées en une seule tout en préservant la satisfiabilité.

Dans ce papier, nous montrons que sous certaines conditions, et même en cas d'existence de quelques triangles cassés sur un couple de valeurs d'une variable donnée, les deux valeurs peuvent être fusionnées sans changer la satisfiabilité de l'instance.

Abstract

Broken triangles constitute an important concept not only for solving constraint satisfaction problems in polytime, but also for variable elimination or domain reduction by merging compatible values. Specifically, for a given variable in a binary arc-consistent CSP, if no broken triangle occurs on any pair of values, then this variable can be eliminated while preserving satisfiability. More recently, it has been shown that even when this rule cannot be applied, it could be possible that for a given pair of values no broken triangle occurs. In this case, we can apply a domain-reduction operation which consists in merging these values while preserving satisfiability.

In this paper, we show that under some conditions, and even if there are some broken triangles on a pair of values of a given variable, these values can be merged without changing the satisfiability of the instance.

1 Introduction

L'étude des classes polynomiales constitue un axe de recherche important en programmation par contraintes. Récemment, une nouvelle classe hybride, appelée BTP (pour *Broken Triangle Property* [4, 5]), a été introduite. Elle possède des propriétés remarquables tant du point de vue théorique (elle capture, par exemple, plusieurs classes polynomiales existantes) que du point de vue pratique (les instances de cette classe pouvant être résolues en temps polynomial par les algorithmes MAC [18] et RFL [17] sans l'aide d'algorithmes spécifiques). Aussi, de nombreuses extensions de ce travail ont défini de nouvelles classes polynomiales [6, 7, 8, 11, 15, 16] généralisant ou étendant la classe BTP. D'autres se sont intéressées à réduire la taille des instances CSP à l'aide de versions locales de la propriété BTP. Plus précisément, étant donnée une variable d'un CSP binaire arc-cohérent [13], si aucun triangle cassé ne se produit sur aucun couple de ses valeurs, alors cette variable peut être éliminée sans modifier la satisfiabilité de l'instance [1]. Dans [2, 3], il a été démontré qu'en cas de non applicabilité de cette règle à cause de l'existence d'au moins un triangle cassé, il se peut qu'il existe deux valeurs de cette même variable sur lesquelles aucun triangle cassé ne se produit. Dans ce cas, ces deux valeurs peuvent être fusionnées en une seule tout en préservant la satisfiabilité. Cette règle de fusion, appelée BTP-fusion, est applicable pour plusieurs benchmarks de la compétition CSP. Mais, malgré cela, certains tests expérimentaux semblent indiquer qu'à l'heure actuelle, elle semble difficilement utilisable comme opération de pré-traitement dans un solveur, du fait d'un temps d'exécution total trop important [2].

Au vu de ces résultats prometteurs, nous allons étudier dans ce papier une version plus légère de la fusion

par BTP tout en autorisant, sous certaines conditions, la présence de quelques triangles cassés sur le couple de valeurs à fusionner.

Dans la section suivante, nous rappelons certaines définitions et notations nécessaires pour la suite de ce travail. Dans la section 3, nous introduisons et généralisons la nouvelle règle, appelée m -WBTP-fusion, qui permet de fusionner deux valeurs d'une variable donnée même en présence de certains triangles cassés. Ensuite, nous prouvons que cette règle de fusion est maximale. Par la suite, dans la section 5, nous commençons par montrer que cette règle de fusion ne permet pas l'élimination de variables puis nous positionnons notre travail par rapport à certaines propriétés comme k -BTP [6] et WBTP [16]. Après, nous évaluons l'intérêt pratique de notre approche. Finalement, nous concluons.

2 Préliminaires

Les *problèmes de satisfaction de contraintes* (CSP, [14]) sont au cœur de nombreuses applications en intelligence artificielle et en recherche opérationnelle. Dans ce papier, nous nous intéressons uniquement aux CSP binaires qui sont formellement définis ainsi :

Définition 1 (instance binaire) *Un CSP binaire est un triplet $I = (X, D, C)$, où $X = \{x_1, \dots, x_n\}$ est un ensemble fini de n variables, $D = \{D(x_1), \dots, D(x_n)\}$ est un ensemble fini de domaines contenant au plus d valeurs, un domaine pour chaque variable et C est un ensemble de contraintes binaires. Chaque contrainte $C_{ij} \in C$ est un couple $(S(C_{ij}), R(C_{ij}))$ avec :*

- $S(C_{ij}) = \{x_i, x_j\} \subseteq X$, la portée de la contrainte,
- $R(C_{ij}) \subseteq D(x_i) \times D(x_j)$, la relation binaire qui définit la compatibilité de valeurs.

Si la contrainte C_{ij} n'est pas définie dans C , alors nous considérons C_{ij} comme une contrainte universelle (c'est-à-dire telle que $R(C_{ij}) = D(x_i) \times D(x_j)$).

L'interaction entre les valeurs de chaque variable à travers les relations associées aux contraintes peut être graphiquement représentée par un graphe de microstructure [10]. Les sommets de ce graphe sont donc les couples variable-valeur (x_i, v_i) ($v_i \in D(x_i)$) et les arêtes sont les tuples autorisés par les contraintes (c'est-à-dire qu'il existe une arête entre les sommets (x_i, v_i) et (x_j, v_j) si $(v_i, v_j) \in R(C_{ij})$). Étant donnée une instance binaire I , décider si I possède une solution (c'est-à-dire une affectation d'une valeur à chaque variable de I ne violant aucune contrainte), est un problème bien connu pour être NP-complet. Mais, en imposant quelques restrictions sur les portées et/ou les

relations des contraintes, nous pouvons avoir des instances résolubles en temps polynomial. Ces instances définissent une classe polynomiale. La classe polynomiale BTP (pour *Broken Triangle Property*), qui s'appuie sur le concept de *triangle cassé*, se situe au cœur des classes polynomiales puisqu'elle généralise et capture certaines classes existantes. La Broken Triangle Property impose l'absence de certains triangles cassés. Formellement, BTP est définie comme suit :

Définition 2 (Broken-Triangle Property [4, 5])

Soient un CSP binaire I et un ordre $<$ sur les variables de I . Une paire de valeurs $v'_k, v''_k \in D(x_k)$ satisfait BTP si pour chaque couple de variables (x_i, x_j) tel que $x_i < x_j < x_k$, si

- $(v_i, v_j) \in R(C_{ij})$,
- $(v_i, v'_k) \in R(C_{ik})$ et
- $(v_j, v''_k) \in R(C_{jk})$,

alors

- soit $(v_i, v''_k) \in R(C_{ik})$,
- soit $(v_j, v'_k) \in R(C_{jk})$.

Une variable x_k satisfait BTP si chaque paire de valeurs de $D(x_k)$ satisfait BTP. I satisfait BTP par rapport à l'ordre $<$ si chacune de ses variables satisfait BTP.

Cette définition peut se représenter graphiquement dans la microstructure du CSP comme indiqué dans la figure 1. Tout au long de ce papier, l'absence d'arête et les arêtes en pointillés représenteront des tuples interdits.

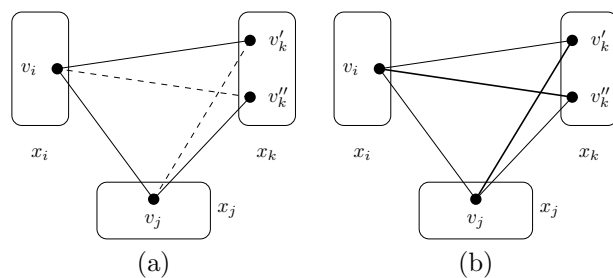


FIGURE 1 – (a) Un triangle cassé (v_i, v_j, v'_k, v''_k) . (b) Un triangle non cassé (v_i, v_j, v'_k, v''_k) et satisfaisant donc la propriété BTP.

Pour la figure 1(a), ce CSP n'est pas BTP par rapport à l'ordre $x_i < x_j < x_k$ car les tuples (v_j, v'_k) et (v_i, v''_k) ne sont pas autorisés. Dans ce cas, (v_i, v_j, v'_k, v''_k) constitue un *triangle cassé* sur les valeurs v'_k et v''_k . L'existence de ce triangle cassé conduit à dire qu'il existe un triangle cassé sur x_k par rapport à x_i et x_j . Par contre, si $(v_i, v''_k) \in R(C_{ik})$ ou

$(v_j, v'_k) \in R(C_{jk})$, alors la propriété est bien vérifiée comme le montre la figure 1(b).

Nous passons maintenant à l'opération de fusion de valeurs puis nous rappelons la condition de fusion sur la base de BTP :

Définition 3 [2] *Fusionner* les valeurs $v'_k, v''_k \in D(x_k)$ d'un CSP binaire consiste à remplacer v'_k, v''_k dans $D(x_k)$ par une nouvelle valeur v_k qui sera compatible avec toutes les valeurs qui sont compatibles avec au moins une des deux valeurs v'_k ou v''_k . Une **condition de fusion de valeurs** v'_k et v''_k d'un CSP binaire est une propriété calculable en temps polynomial de façon que lorsque cette propriété existe, alors le CSP obtenu après fusion des valeurs v'_k et v''_k est satisfiable si et seulement si le CSP de départ est satisfiable aussi.

Le résultat suivant relie la fusion à l'absence de triangles cassés.

Proposition 1 [2] *Étant donné un CSP binaire, fusionner deux valeurs sur lesquelles aucun triangle cassé ne se produit est une condition de fusion de valeurs.*

Par exemple, dans la figure 1(b), les valeurs v'_k et v''_k sont fusionnables.

3 Les triangles légèrement cassés

L'absence de triangles cassés sur un couple de valeurs permet de les fusionner tout en préservant la satisfiabilité. Ici, nous montrons qu'il est possible de tolérer la présence de certains triangles cassés tout en conservant la possibilité de fusionner des valeurs. L'idée est inspirée d'un travail récent de Naanaa [16] sur une nouvelle extension de BTP. La propriété sera appelée *m-wBTP* : le paramètre m définit le nombre de variables soutenant les triangles appelés *légèrement cassés*.

3.1 1-wBTP-fusion

Nous commençons par le cas le plus basique qui s'appuie sur le nouveau concept de *triangle légèrement cassé* qui correspond à un triangle cassé soutenu par une variable.

Définition 4 *Un couple de valeurs $v'_k, v''_k \in D(x_k)$ satisfait 1-wBTP si pour chaque triangle cassé (v_i, v_j, v'_k, v''_k) avec $v_i \in D(x_i)$ et $v_j \in D(x_j)$, il existe au moins une variable $x_\ell \in X \setminus \{x_i, x_j, x_k\}$ tel que $\forall v_\ell \in D(x_\ell)$ si*

- $(v_i, v_\ell) \in R(C_{i\ell})$ et
- $(v_j, v_\ell) \in R(C_{j\ell})$

alors

- $(v'_k, v_\ell) \notin R(C_{k\ell})$ et
- $(v''_k, v_\ell) \notin R(C_{k\ell})$.

Dans ce cas, (v_i, v_j, v'_k, v''_k) sera appelé triangle légèrement cassé soutenu par la variable x_ℓ .

Cette définition devient plus simple quand on la représente avec le graphe de microstructure. Par exemple, dans la figure 2, on a un triangle cassé (v_i, v_j, v'_k, v''_k) . Comme pour toute valeur v_ℓ de la variable x_ℓ , v_ℓ est compatible avec v_i et v_j et on a $(v'_k, v_\ell) \notin R(C_{k\ell})$ et $(v''_k, v_\ell) \notin R(C_{k\ell})$, ce triangle est un triangle légèrement cassé soutenu par x_ℓ .

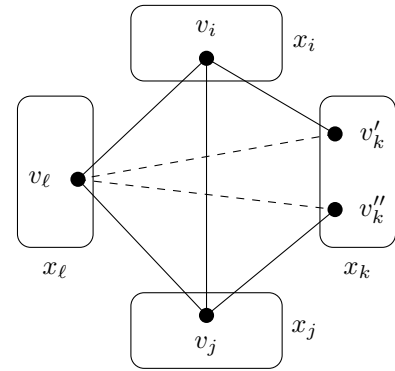


FIGURE 2 – Un triangle légèrement cassé car $(v'_k, v_\ell) \notin R(C_{k\ell})$ et $(v''_k, v_\ell) \notin R(C_{k\ell})$.

Si $(v'_k, v_\ell) \in R(C_{k\ell})$ ou $(v''_k, v_\ell) \in R(C_{k\ell})$, nous disons que $(v_\ell, v_i, v_j, v'_k, v''_k)$ forment un *triangle fortement cassé*.

La distinction de triangle légèrement cassé de triangle fortement cassé nous permet d'avoir un résultat similaire à celui de la proposition 1.

Proposition 2 *Étant donné un CSP binaire, fusionner deux valeurs $v'_k, v''_k \in D(x_k)$ qui satisfont 1-wBTP ne change pas la satisfiabilité de l'instance.*

Preuve : Soit I l'instance originale et I^f la nouvelle instance dans laquelle v'_k, v''_k ont été fusionnées en une nouvelle valeur v_k . Clairement, si I est satisfiable alors il en est de même pour I^f . Donc, il suffit de montrer que si I^f a une solution s qui affecte v_k à x_k , alors I a une solution.

Soient s', s'' deux affectations identiques à s sauf que s' affecte v'_k à x_k et s'' affecte v''_k à x_k . Supposons que ni s' ni s'' sont des solutions de I . Alors, il existe deux variables $x_i, x_j \in X \setminus \{x_k\}$ telles que $(s(x_i), v'_k) \notin R(C_{ik})$ et $(s(x_j), v''_k) \notin R(C_{jk})$. Puisque s est une solution de I^f affectant v_k à x_k , nous devons forcément avoir $(s(x_i), v''_k) \in R(C_{ik})$ et $(s(x_j), v'_k) \in R(C_{jk})$. Évidemment, nous avons $(s(x_i), s(x_j)) \in R(C_{ij})$. Dans ce cas, $(s(x_i), s(x_j), v'_k, v''_k)$ forment un triangle cassé dans I .

Par définition de 1-wBTP, il existe au moins une variable $x_\ell \in X \setminus \{x_i, x_j, x_k\}$ telle que $\forall v_\ell \in D(x_\ell)$ si

- $(s(x_i), v_\ell) \in R(C_{i\ell})$ et
- $(s(x_j), v_\ell) \in R(C_{j\ell})$

alors

- $(v'_k, v_\ell) \notin R(C_{k\ell})$ et
- $(v''_k, v_\ell) \notin R(C_{k\ell})$.

Comme $s(x_\ell)$ est compatible avec $s(x_i)$ et $s(x_j)$, elle ne peut être compatible avec ni v'_k ni v''_k . Par conséquent, $s(x_\ell)$ ne sera pas compatible avec v_k , ce qui implique que s n'est pas une solution de I^f . Or, ceci contredit notre hypothèse de départ. Donc, cette règle de fusion préserve la satisfiabilité. \square

À première vue, l'existence d'un lien entre la cohérence d'arc [13] et cette définition paraît logique. En effet, imposer que les tuples (v'_k, v_ℓ) et (v''_k, v_ℓ) soient interdits peut laisser penser que le but est de rendre les valeurs v'_k et v''_k arc-incohérentes. Mais l'exemple de la figure 3 montre le contraire. En effet, bien que les deux valeurs $v'_k, v''_k \in D(x_k)$ de cette figure satisfont 1-wBTP, l'application de la cohérence d'arc ne supprime aucune valeur (et évidemment aucun tuple) puisqu'elles sont toutes arc-cohérentes. Donc, la cohérence d'arc ne supprimera pas le triangle cassé (v_i, v_j, v'_k, v''_k) .

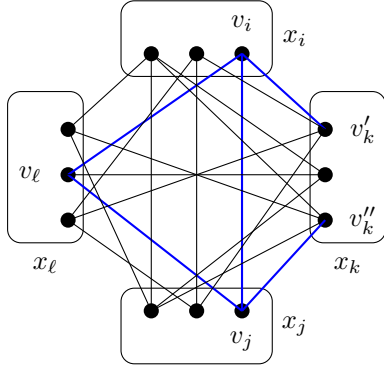


FIGURE 3 – Un CSP dont toutes les valeurs sont arc-cohérentes (en bleu le triangle légèrement cassé).

3.2 m -wBTP-fusion

Grâce à la variable x_ℓ , la fusion de valeurs sur lesquelles il existe seulement des triangles légèrement cassés ne modifie pas la satisfiabilité. En termes de microstructure, la variable x_ℓ empêche l'apparition d'une nouvelle clique de taille n (donc une solution) qui n'existait pas avant la fusion. Ce principe peut évidemment être étendu à m variables ($m \leq n - 3$).

Une affectation $(v_{\ell_1}, \dots, v_{\ell_m}) \in D(x_{\ell_1}) \times \dots \times D(x_{\ell_m})$ est une **solution partielle** si elle satisfait toutes les contraintes C_{ij} telles que $\{x_i, x_j\} \subseteq \{x_{\ell_1}, \dots, x_{\ell_m}\}$.

Définition 5 Un couple de valeurs $v'_k, v''_k \in D(x_k)$ satisfait m -wBTP pour $m \leq n - 3$ si pour chaque triangle cassé (v_i, v_j, v'_k, v''_k) avec $v_i \in D(x_i)$ et $v_j \in D(x_j)$, il existe un ensemble de $r \leq m$ variables $\{x_{\ell_1}, \dots, x_{\ell_r}\} \subseteq X \setminus \{x_i, x_j, x_k\}$ tel que pour tout $(v_{\ell_1}, \dots, v_{\ell_r}) \in D(x_{\ell_1}) \times \dots \times D(x_{\ell_r})$, si $(v_{\ell_1}, \dots, v_{\ell_r}, v_i, v_j)$ est une solution partielle, alors il existe $\alpha \in \{1, \dots, r\}$ tel que $(v_{\ell_\alpha}, v'_k), (v_{\ell_\alpha}, v''_k) \notin R(C_{\ell_\alpha k})$.

La figure 4 montre deux configurations illustrant la définition 5. Dans la première, le couple de valeurs $v'_k, v''_k \in D(x_k)$ satisfait 2-wBTP car pour l'unique solution partielle $(v_{\ell_\sigma}, v_{\ell_\gamma}, v_i, v_j)$ on a $(v_{\ell_\sigma}, v'_k), (v_{\ell_\sigma}, v''_k) \notin R(C_{\ell_\sigma k})$. Dans la deuxième, il n'existe aucune solution partielle pour l'ensemble $\{x_{\ell_\sigma}, x_{\ell_\gamma}, x_i, x_j\}$. Donc $v'_k, v''_k \in D(x_k)$ satisfait trivialement 2-wBTP.

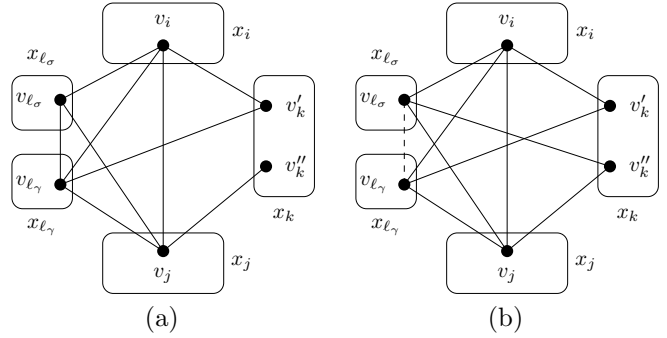


FIGURE 4 – Deux cas différents de deux valeurs v'_k et v''_k qui satisfont la 2-wBTP.

Nous généralisons maintenant le résultat de la proposition 2 sur la fusion de valeurs au cas de m variables.

Proposition 3 *Étant donné un CSP binaire, fusionner deux valeurs $v'_k, v''_k \in D(x_k)$ qui satisfont m -wBTP ne change pas la satisfiabilité de l'instance.*

Preuve : Soit I l'instance originale et I^f la nouvelle instance dans laquelle v'_k, v''_k ont été fusionnées en une nouvelle valeur v_k . Clairement, si I est satisfiable alors il en est de même pour I^f . Donc, il suffit de montrer que si I^f a une solution s qui affecte v_k à x_k , alors I a une solution.

Soient s', s'' deux affectations identiques à s sauf que s' affecte v'_k à x_k et s'' affecte v''_k à x_k . Supposons que ni s' ni s'' sont des solutions de I . Alors, il existe deux variables $x_i, x_j \in X \setminus \{x_k\}$ telles que $(s(x_i), v'_k) \notin R(C_{ik})$

et $(s(x_j), v_k'') \notin R(C_{jk})$. Puisque s est une solution de I^f affectant v_k à x_k , nous devons forcément avoir $(s(x_i), v_k'') \in R(C_{ik})$ et $(s(x_j), v_k') \in R(C_{jk})$. On a aussi $(s(x_i), s(x_j)) \in R(C_{ij})$ puisque s est une solution de I^f . D'où, $(s(x_i), s(x_j), v_k', v_k'')$ forme un triangle cassé dans I .

Les valeurs v_k' et v_k'' satisfont m -wBTP, donc, par définition, il existe un ensemble de $r \leq m$ variables $\{x_{\ell_1}, \dots, x_{\ell_r}\} \subseteq X \setminus \{x_i, x_j, x_k\}$ tel que pour tout $(v_{\ell_1}, \dots, v_{\ell_r}) \in D(x_{\ell_1}) \times \dots \times D(x_{\ell_r})$, si $(v_{\ell_1}, \dots, v_{\ell_r}, v_i, v_j)$ est une solution partielle, alors il existe $\alpha \in \{1, \dots, r\}$ tel que $(v_{\ell_\alpha}, v_k'), (v_{\ell_\alpha}, v_k'') \notin R(C_{\ell_\alpha k})$.

Comme s est une solution pour l'instance I_f , $(s(x_{\ell_1}), \dots, s(x_{\ell_r}), s(x_i), s(x_j))$ est nécessairement une solution partielle, donc il existe $\alpha \in \{1, \dots, r\}$ tel que $(s(x_{\ell_\alpha}), v_k'), (s(x_{\ell_\alpha}), v_k'') \notin R(C_{\ell_\alpha k})$, ce qui implique $(s(x_{\ell_\alpha}), v_k) \notin R(C_{\ell_\alpha k})$, c'est-à-dire une contradiction car s est une solution pour l'instance I_f avec $s(x_k) = v_k$.

Par conséquent, cette règle de fusion préserve la satisfiabilité. \square

La règle BTP-fusion (BTP-merging) de [2] peut être considérée comme 0-wBTP-fusion puisqu'elle s'appuie sur zéro variable de soutien pour la fusion de valeurs. La proposition suivante permet d'établir le lien entre les différentes formes de fusion sur la base de BTP.

Proposition 4 *Étant donné un CSP binaire à n variables, si un couple de valeurs $v_k', v_k'' \in D(x_k)$ satisfait m -wBTP alors il satisfait $(m+1)$ -wBTP (pour $0 \leq m \leq n-4$).*

La règle BTP-fusion généralise la substitution de voisinage [9] et l'interchangeabilité virtuelle [12]. Comme m -wBTP-fusion généralise BTP-fusion pour tout $m \geq 0$, alors ceci nous conduit immédiatement au résultat suivant :

Corollaire 1 *m -wBTP-fusion généralise la substitution de voisinage et l'interchangeabilité virtuelle.*

Par ailleurs, en plus de conserver la satisfiabilité, il est possible de reconstruire en temps polynomial toutes les solutions de I à partir des solutions de toute instance I^f obtenue en appliquant à I une suite de fusions par m -wBTP-fusion. De plus, la reconstruction d'une solution de I à partir d'une solution de I^f peut s'effectuer en temps linéaire dans la taille de l'instance I . Il suffit d'appliquer les mêmes algorithmes que dans le cas de BTP-fusion [2].

4 $(n-3)$ -wBTP-fusion est une condition de fusion maximale

Il est bien connu que tout couple de valeurs qui satisfait BTP peut être fusionné tout en préservant la satisfiabilité [2]. Ici, nous avons montré qu'un couple de valeurs ne satisfaisant pas BTP peut être fusionné tout en conservant la satisfiabilité du moment où il satisfait m -wBTP. En ce sens, BTP n'est pas une condition de fusion maximale. Une condition de fusion est dite *maximale* si la fusion de n'importe quel couple de valeurs ne respectant pas la propriété entraîne nécessairement la modification de la satisfiabilité de l'instance. Par contre, nous pouvons montrer que m -wBTP est une condition de fusion maximale pour $m = n-3$.

Théorème 1 *Étant donné un CSP binaire I à n variables, il n'existe aucun couple de valeurs qui ne satisfait pas m -wBTP pour $m = n-3$ et dont la fusion préserve la satisfiabilité.*

Preuve : Soient un CSP binaire I à n variables et un couple de valeurs $v_k', v_k'' \in D(x_k)$ qui ne satisfait pas m -wBTP pour $m = n-3$. Par définition de la m -wBTP-fusion, il existe un triangle cassé (v_i, v_j, v_k', v_k'') , avec $v_i \in D(x_i)$ et $v_j \in D(x_j)$, tel qu'il existe $(v_{\ell_1}, \dots, v_{\ell_m}) \in D(x_{\ell_1}) \times \dots \times D(x_{\ell_m})$, où $\{x_{\ell_1}, \dots, x_{\ell_m}\} = X \setminus \{x_i, x_j, x_k\}$, tel que $(v_{\ell_1}, \dots, v_{\ell_m}, v_i, v_j)$ est une solution partielle et pour tout $\alpha \in \{1, \dots, m\}$ on a $(v_{\ell_\alpha}, v_k') \in R(C_{\ell_\alpha k})$ ou $(v_{\ell_\alpha}, v_k'') \in R(C_{\ell_\alpha k})$.

Nous avons un triangle cassé, et donc :

- $(v_i, v_k'') \notin R(C_{ik})$
- $(v_j, v_k') \notin R(C_{jk})$
- $(v_i, v_k') \in R(C_{ik})$
- $(v_j, v_k'') \in R(C_{jk})$

Nous avons également, pour tout $\ell \in \{\ell_1, \dots, \ell_m\}$:

- $(v_\ell, v_k') \in R(C_{\ell k})$ ou
- $(v_\ell, v_k'') \in R(C_{\ell k})$.

Après fusion, et par définition de la fusion, la nouvelle valeur v_k satisfait $(v_\ell, v_k) \in R(C_{\ell k})$ pour tout $\ell \in \{\ell_1, \dots, \ell_m\} \cup \{i, j\}$. Nous obtiendrons à la fin une solution formée par $v_{\ell_1}, \dots, v_{\ell_m}, v_i, v_j$ et v_k . Donc nous venons d'introduire une solution qui n'existait pas avant puisque $(v_i, v_k'') \notin R(C_{ik})$ et $(v_j, v_k') \notin R(C_{jk})$. Par conséquent, la fusion de n'importe quel couple de valeurs qui ne satisfait pas m -wBTP ne préserve pas la satisfiabilité. Donc, m -wBTP est une condition de fusion maximale. \square

5 Fusion, élimination de variables et classes polynomiales

BTP permet la fusion de valeurs [2], l'élimination de variables [1] et la définition d'une classe polynomiale [5]. Il existe plusieurs généralisations distinctes de BTP selon la propriété étudiée. La m -wBTP est une généralisation de BTP qui permet de réduire la taille des instances via la fusion de valeurs. La m -wBTP est une condition moins restrictive que BTP et donc permet plus de fusions que BTP. La contrepartie de ce gain en nombre de fusions est le fait que la m -wBTP ne permet pas l'élimination de variables.

Dans [1], il a été démontré que, pour une variable donnée x_k d'un CSP arc-cohérent I , s'il n'existe aucun triangle cassé sur chaque couple de valeurs de $D(x_k)$, alors la variable x_k peut être éliminée de I tout en préservant la satisfiabilité. Ici, nous montrons que ce n'est pas le cas pour la m -wBTP pour $m > 0$.

Proposition 5 *Étant donnée une variable x_k d'un CSP binaire arc-cohérent I , même si chaque paire de valeurs de $D(x_k)$ satisfait m -wBTP, où $m \geq 1$, alors éliminer la variable x_k peut changer la satisfiabilité de I .*

Preuve : Soit I l'instance CSP binaire définie sur quatre variables x_1, \dots, x_4 avec $D(x_i) = \{0, 1, 2\}$ ($i = 1, \dots, 4$) et les contraintes suivantes : $x_1 = x_2$, $x_2 = x_3$, $x_3 = x_1$, $x_1 = (x_4 + 1) \bmod 3$, $x_2 = (x_4 - 1) \bmod 3$, $x_3 = x_4$. Cette instance est arc-cohérente. Il y a trois solutions partielles $(0, 0, 0)$, $(1, 1, 1)$ et $(2, 2, 2)$ sur les variables x_1, x_2, x_3 , mais I n'a pas de solution. Donc, l'élimination de la variable x_4 ne préserve pas la satisfiabilité de l'instance (voir figure 5).

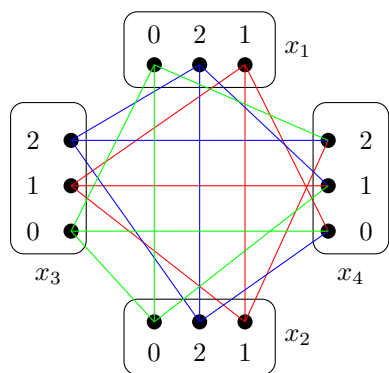


FIGURE 5 – Un CSP incohérent dont chaque couple de valeurs dans $D(x_4)$ satisfait 1-wBTP mais la suppression de x_4 entraîne l'apparition de trois solutions.

Soient x_i, x_j, x_ℓ les variables x_1, x_2, x_3 (dans n'importe quel ordre). Il existe trois triangles cassés (v_i, v_j, v'_4, v''_4) sur les variables x_i, x_j, x_4 (les triangles

légèrement cassés sont représentés par des couleurs différentes dans la figure 5). Dans chacun de ces triangles cassés, nous avons $v_i = v_j$. Pour chacun de ces triangles cassés, il n'y a qu'une seule solution partielle de la forme (v_ℓ, v_i, v_j) sur les variables x_ℓ, x_i, x_j car nous avons forcément $v_\ell = v_i = v_j$. Par le choix des contraintes, les valeurs v_ℓ, v_i, v_j sont compatibles avec trois valeurs différentes dans $D(x_4)$. On peut en déduire que $(v_\ell, v'_4), (v_\ell, v''_4) \notin R(C_{\ell 4})$ car, par la définition d'un triangle cassé, chacune des valeurs v'_4, v''_4 est compatible avec une des valeurs v_i, v_j . Par conséquent, chaque paire de valeurs $v'_4, v''_4 \in D(x_4)$ satisfait 1-wBTP.

Nous avons donné une instance I telle que chaque paire de valeurs de $D(x_4)$ satisfait 1-wBTP, mais éliminer la variable x_4 change la satisfiabilité de I . Pour les valeurs de $m > 1$, il suffit d'ajouter $m - 1$ autres variables sans contrainte à l'instance I . \square

Dans l'instance I de la preuve de la proposition 5, chaque paire de valeurs dans le domaine $D(x_4)$ satisfait 1-wBTP. Cependant, après avoir effectué la fusion de deux valeurs, les deux valeurs qui restent ne satisfont pas 1-wBTP et on ne peut pas les fusionner.

Précédemment dans [6], une version allégée de BTP appelée k -BTP, qui autorise l'existence de certains triangles cassés, a été introduite. Les instances de CSP binaires qui satisfont la k -cohérence forte et k -BTP constituent une classe polynomiale. Elle est définie comme suit :

Définition 6 (k -BTP [6]) *Un CSP binaire P satisfait la propriété k -BTP pour un k donné ($2 \leq k < n$) par rapport à un ordre $<$ sur les variables si et seulement si, pour tout sous-ensemble de variables $x_{i_1}, x_{i_2}, \dots, x_{i_{k+1}}$ tel que $x_{i_1} < x_{i_2} < \dots < x_{i_{k+1}}$, il existe au moins un couple de variables $(x_{i_j}, x_{i_{j'}})$ avec $1 \leq j < j' \leq k$ tel qu'il n'existe pas de triangle cassé sur x_{k+1} par rapport à x_{i_j} et $x_{i_{j'}}$.*

Malheureusement, et contrairement à m -wBTP, la propriété k -BTP ne peut être utilisée pour fusionner des valeurs dès que k est supérieur strictement à 2 (pour rappel 2-BTP = BTP). En effet, les valeurs v'_k et v''_k de la figure 6(a) satisfont la propriété 3-BTP puisque le triplet (x_i, x_ℓ, x_k) n'induit aucun triangle cassé sur x_k par rapport à x_i et x_ℓ . Si on fusionne v'_k et v''_k , ce CSP devient consistant alors qu'il ne l'était pas initialement. Donc, la fusion par k -BTP (pour k strictement supérieur à 2), même si elle autorise la présence de quelques triangles cassés, ne préserve pas la satisfiabilité. De plus, m -wBTP peut autoriser plus de triangles cassés que k -BTP. Par exemple, les valeurs v'_k et v''_k de la figure 6(b) satisfont 1-wBTP mais pas 3-BTP à cause des triangles cassés présents sur la va-

riable x_k quel que soit le triplet de variables incluant x_k .

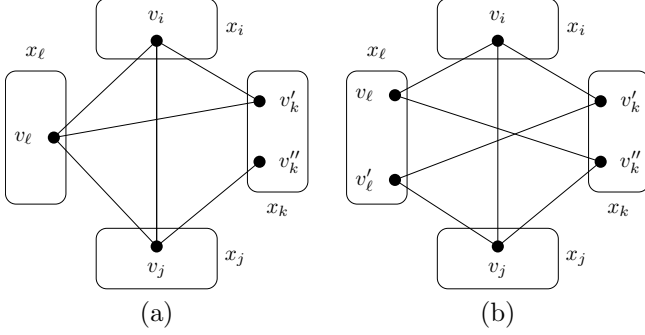


FIGURE 6 – (a) Deux valeurs v'_k et v''_k qui ne satisfont pas la 1-wBTP mais l'instance satisfait 3-BTP. (b) Deux valeurs v'_k et v''_k qui satisfont la 1-wBTP mais l'instance ne satisfait pas 3-BTP.

Naanaa a présenté deux autres généralisations de BTP qui permettent de définir des classes polynomiales [15, 16]. Il a été démontré [6] que la notion de rang directionnel $k - 1$ [15] généralise strictement k -BTP. On peut en déduire que l'exemple de la figure 6(a) satisfait la propriété de rang directionnel 2, ce qui montre que la propriété de rang directionnel k (pour $k \geq 2$) ne peut être utilisée pour fusionner des valeurs (en sachant que le cas $k = 1$ correspond à BTP).

La notion de WBTP [16] a inspiré la notion de 1-wBTP, mais est différente. Nous donnons sa définition avant de démontrer que WBTP peut être vu comme une condition strictement plus forte que 1-wBTP (et donc qui permet moins de fusions).

Définition 7 (WBTP [16]) Un CSP binaire doté d'un ordre $<$ sur ses variables satisfait la propriété WBTP (Weak Broken Triangle Property) si pour tout triplet de variables $x_i < x_j < x_k$ et pour tout $v_i \in D(x_i)$, $v_j \in D(x_j)$ tels que $(v_i, v_j) \in R(C_{ij})$, il existe une variable $x_\ell < x_k$ telle que lorsque $v_\ell \in D(x_\ell)$ est compatible avec v_i et v_j , alors nous avons $\forall v_k \in D(x_k)$,

$$(v_\ell, v_k) \in R(C_{\ell k}) \Rightarrow ((v_i, v_k) \in R(C_{ik}) \wedge (v_j, v_k) \in R(C_{jk}))$$

Proposition 6 Si un CSP binaire doté d'un ordre $<$ sur ses variables satisfait la propriété WBTP, alors elle satisfait la propriété 1-wBTP pour chaque paire de valeurs dans le domaine de la dernière variable (selon l'ordre $<$).

Preuve : Supposons que le CSP binaire I satisfait WBTP pour l'ordre $<$ de ses variables et soit x_k la dernière variable de I selon cet ordre. Supposons pour une

contradiction, que I ne satisfait pas 1-wBTP sur une paire de valeurs $v'_k, v''_k \in D(x_k)$. Alors, par définition de 1-wBTP, il existe un triangle cassé (v_i, v_j, v'_k, v''_k) avec $v_i \in D(x_i)$, $v_j \in D(x_j)$, $v'_k, v''_k \in D(x_k)$ tel qu'il n'existe pas de variable $x_\ell \in X \setminus \{x_i, x_j, x_k\}$ tel que $\forall v_\ell \in D(x_\ell)$ compatible avec v_i et v_j , nous avons $(v_\ell, v'_k), (v_\ell, v''_k) \notin R(C_{\ell k})$.

Mais la propriété WBTP nous garantit qu'il existe une variable $x_\ell < x_k$ telle que $\forall v_\ell \in D(x_\ell)$ compatible avec v_i et v_j , nous avons $\forall v_k \in D(x_k)$,

$$(v_\ell, v_k) \in R(C_{\ell k}) \Rightarrow ((v_i, v_k) \in R(C_{ik}) \wedge (v_j, v_k) \in R(C_{jk}))$$

L'existence du triangle cassé (v_i, v_j, v'_k, v''_k) implique que $x_\ell \notin \{x_i, x_j\}$ et donc $x_\ell \in X \setminus \{x_i, x_j, x_k\}$. D'ailleurs, puisque (v_i, v_j, v'_k, v''_k) est un triangle cassé,

$$(v_i, v_k) \in R(C_{ik}) \wedge (v_j, v_k) \in R(C_{jk})$$

est faux pour $v_k \in \{v'_k, v''_k\}$. On peut en déduire que $(v_\ell, v'_k), (v_\ell, v''_k) \notin R(C_{\ell k})$, c'est-à-dire une contradiction. \square

Imposer la propriété WBTP est strictement plus fort qu'imposer 1-wBTP pour chaque paire de valeurs v'_k, v''_k dans le domaine de la dernière variable x_k . WBTP impose une condition sur toute valeur $v_k \in D(x_k)$ par rapport à la même variable x_ℓ , tandis que 1-wBTP (pour chaque paire de valeurs $v'_k, v''_k \in D(x_k)$) impose une condition équivalente mais pour laquelle la variable x_ℓ peut varier selon les valeurs v'_k, v''_k . L'instance de la figure 7 satisfait 1-wBTP mais ne satisfait pas WBTP car :

- seule la variable x_{ℓ_2} soutient (v_i, v_j, v'_k, v''_k) ,
- seule la variable x_{ℓ_1} soutient (v_i, v_j, v'_k, v'''_k) .

Donc, il n'existe pas une variable qui soutient à la fois les triangles cassés (v_i, v_j, v'_k, v''_k) et (v_i, v_j, v'_k, v'''_k) .

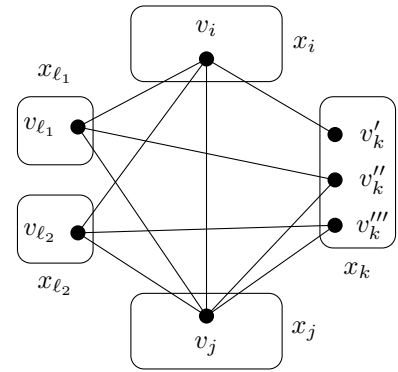


FIGURE 7 – Les couples v'_k, v''_k et v'_k, v'''_k de x_k satisfont 1-wBTP mais pas WBTP.

WBTP définit une classe polynomiale [16]. Une question qui reste ouverte est de savoir s’il est possible d’utiliser 1-wBTP pour définir une classe polynomiale qui est plus générale que celle définie par WBTP.

6 Résultats expérimentaux

Pour tester l’applicabilité de ces règles de fusion et en particulier 1-wBTP-fusion, nous avons réalisé une étude expérimentale sur l’ensemble des benchmarks binaires de la compétition internationale de solveurs CSP de 2008¹ (soit 3 795 instances). Pour cela, l’algorithme de fusion par 1-wBTP-fusion s’inspire de celui présenté dans [3] pour la fusion par BTP-fusion. Plus précisément, étant donnée une variable x_k , nous vérifions pour chaque couple de valeurs $v'_k, v''_k \in D(x_k)$ si les deux valeurs sont fusionnables par rapport à 1-wBTP-fusion. Si un triangle cassé sur v'_k, v''_k a été trouvé, nous cherchons dans les $n - 3$ variables qui restent s’il existe une variable x_ℓ qui soutient ce triangle cassé. Si on la trouve, nous continuons la recherche de triangles cassés. Sinon, le test est fini pour ces deux valeurs. Enfin, s’il n’existe pas de triangles cassés, ou s’il n’existe que des triangles légèrement cassés sur le couple v'_k, v''_k , nous les fusionnons. Nous avons implémenté ces deux algorithmes de fusion en C++ au sein de notre propre bibliothèque CSP. Les expérimentations ont été effectuées sur 8 serveurs-lames Dell PowerEdge M820 dotés de deux processeurs Intel Xeon E5-2609 v2 2,5 GHz et de 32 Go de mémoire et fonctionnant sous Linux Ubuntu 14.04.

Pour chaque benchmark, nous avons lancé BTP-fusion et 1-wBTP-fusion jusqu’à la convergence vers un point fixe avec un délai de traitement d’une heure. Au total, nous avons obtenu les résultats pour 2 535 benchmarks sur 3 795 et nous avons réussi à fusionner au moins une valeur pour 1 001 instances. Dans le tableau 1, la colonne #benchmarks désigne le nombre de benchmarks pour lesquels le test a terminé en une heure. La colonne #valeurs indique la moyenne du nombre total de valeurs de ces benchmarks. Les colonnes BTP-fusion et 1-wBTP-fusion nous renseignent sur le nombre de valeurs supprimées respectivement par BTP-fusion et 1-wBTP-fusion. Quant à la figure 8, elle permet de comparer les pourcentages de valeurs supprimées par BTP-fusion et 1-wBTP-fusion instance par instance. Si, pour une majorité d’instances, les résultats sont comparables, nous pouvons remarquer que pour certaines d’entre elles, 1-wBTP-fusion fusionne significativement plus de valeurs que BTP-fusion. C’est notamment le cas pour les instances des familles `langford*` pour lesquelles 1-wBTP-fusion

fusionne de 25 à 80% des valeurs là où BTP-fusion n’en fusionne aucune.

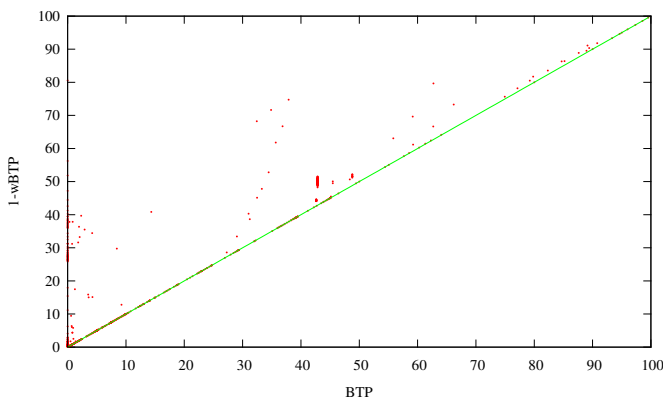


FIGURE 8 – Comparaisons des pourcentages de valeurs fusionnées par BTP et 1-wBTP.

7 Conclusion

Dans ce papier, nous avons étudié la fusion sur la base de BTP et nous avons proposé une famille de définitions qui est basée sur le concept de triangle légèrement cassé, c’est-à-dire de triangle cassé soutenu par une ou plusieurs variables pour préserver la satisfiabilité après fusion.

Il serait intéressant de vérifier si m -wBTP définit une nouvelle classe polynomiale qui généralise BTP et qui conserve ses particularités. Des éventuels liens avec la largeur arborescente devraient aussi être étudiés. Certains autres points pourraient être formellement approfondis tels que l’influence de l’ordre sur le résultat final. Par exemple, la figure 6 nous montre que si on fusionne les valeurs de x_k nous ne pouvons plus fusionner les valeurs de x_ℓ et inversement. Déterminer le meilleur ordre d’opérations de m -wBTP-fusion est NP-difficile même dans le cas $m = 0$ [2].

Références

- [1] David A. Cohen, Martin C. Cooper, Guillaume Escamocher, and Stanislav Zivny. "variable elimination in binary csp via forbidden patterns". In *Proceedings of IJCAI*, 2013.
- [2] Martin C. Cooper, Aymeric Duchain, Achref El Mouelhi, Guillaume Escamocher, Cyril Terrioux, and Bruno Zanuttini. Broken triangles : From value merging to a tractable class of general-arity constraint satisfaction problems. *Artificial Intelligence*, 234 :196 – 218, 2016.

1. Voir <http://www.cril.univ-artois.fr/CPAI08> pour plus de détails.

Famille	#benchmarks	#valeurs	BTP-fusion	1-wBTP-fusion
BH-4-4	10	674	322	348
BH-4-7	20	2 102	883	932
ehi-85	98	2 079	891	1 045
ehi-90	100	2 205	945	1 100
graph-coloring/school	8	4 473	104	104
graph-coloring/sgb/book	26	1 887	534	534
os-taillard-4	30	2 932	1 820	1 978
rlfapScens	1	8 004	341	1 211
rlfapScensMod	6	8 788	2 415	5 169
subs	9	1 479	40	517
langford-2	22	879	0	233
langford-3	20	1 490	0	554
langford-4	16	1 784	0	504
queenAttacking	7	2 196	0	36

TABLE 1 – Résultats expérimentaux sur les benchmarks.

- [3] Martin C. Cooper, Achref El Mouelhi, Cyril Terrioux, and Bruno Zanuttini. On Broken Triangles. In *Proceedings of CP*, pages 9–24, 2014.
- [4] Martin C. Cooper, Peter Jeavons, and Andras Salamon. Hybrid tractable CSPs which generalize tree structure. In *Proceedings of ECAI*, pages 530–534, 2008.
- [5] Martin C. Cooper, Peter Jeavons, and Andras Salamon. Generalizing constraint satisfaction on trees : hybrid tractability and variable elimination. *Artificial Intelligence*, 174 :570–584, 2010.
- [6] Martin C. Cooper, Philippe Jégou, and Cyril Terrioux. A microstructure-based family of tractable classes for csps. In *Proceedings of CP*, pages 74–88, 2015.
- [7] Achref El Mouelhi, Philippe Jégou, and Cyril Terrioux. Hidden Tractable Classes : From Theory to Practice. In *Proceedings of ICTAI*, pages 437–445, 2014.
- [8] Achref El Mouelhi, Philippe Jégou, and Cyril Terrioux. A Hybrid Tractable Class for Non-Binary CSPs. *Constraints*, 20(4) :383–413, 2015.
- [9] Eugene C. Freuder. Eliminating interchangeable values in constraint satisfaction problems. In *Proceedings of AAAI*, pages 227–233, 1991.
- [10] Philippe Jégou. Decomposition of Domains Based on the Micro-Structure of Finite Constraint Satisfaction Problems. In *Proceedings of AAAI*, pages 731–736, 1993.
- [11] Philippe Jégou and Cyril Terrioux. The extendable-triple property : A new CSP tractable class beyond BTP. In *Proceedings of AAAI*, pages 3746–3754, 2015.
- [12] Chavalit Likitvivatanavong and Roland H. C. Yap. Many-to-many interchangeable sets of values in csps. In *Proceedings of SAC*, pages 86–91, 2013.
- [13] Alan K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8 :99–118, 1977.
- [14] Ugo Montanari. Networks of Constraints : Fundamental Properties and Applications to Picture Processing. *Artificial Intelligence*, 7 :95–132, 1974.
- [15] Wady Naanaa. Unifying and extending hybrid tractable classes of csps. *J. Exp. Theor. Artif. Intell.*, 25(4) :407–424, 2013.
- [16] Wady Naanaa. Extending the broken triangle property tractable class of binary csps. In *Proceedings of the 9th Hellenic Conference on Artificial Intelligence (SETN 2016)*, 2016.
- [17] Bernard A. Nadel. *Tree Search and Arc Consistency in Constraint-Satisfaction Algorithms*, pages 287–342. In *Search in Artificial Intelligence*. Springer-Verlag, 1988.
- [18] Daniel Sabin and Eugene C. Freuder. Contradicting Conventional Wisdom in Constraint Satisfaction. In *Proceedings of ECAI*, pages 125–129, 1994.