Pairwise decomposition for combinatorial optimization in graphical models

Aurélie Favier[*], **Simon de Givry**[*], **Andrés Legarra**[†], **Thomas Schiex**[*]
[*] Unité de Biométrie et d'Intelligence Artificielle UR0875, INRA, F-31320 Castanet Tolosan, France
[†] Station d'Amélioration Génétique des Animaux UR0631, INRA, F-31320 Castanet Tolosan, France
{afavier,degivry,alegarra,tschiex}@toulouse.inra.fr

Abstract

We propose a new additive decomposition of probability tables that preserves equivalence of the joint distribution while reducing the size of potentials, without extra variables. We formulate the Most Probable Explanation (MPE) problem in belief networks as a Weighted Constraint Satisfaction Problem (WCSP). Our pairwise decomposition allows to replace a cost function with smaller-arity functions. The resulting pairwise decomposed WCSP is then easier to solve using state-of-the-art WCSP techniques. Although testing pairwise decomposition is equivalent to testing pairwise independence in the original belief network, we show how to efficiently test and enforce it, even in the presence of hard constraints. Furthermore, we infer additional information from the resulting nonbinary cost functions by projecting&subtracting them on binary functions. We observed huge improvements by preprocessing with pairwise decomposition and project&subtract compared to the current state-of-the-art solvers on two difficult sets of benchmark.

1 Introduction

Probabilistic Graphical Models (PGM) provides a general approach for automated reasoning under uncertainty [Koller and Friedman, 2009]. PGMs cover Bayesian networks, Markov networks, but also deterministic frameworks such as constraint networks. This paper focuses on discrete PGMs and the MPE optimization problem which consists in maximizing a product of nonnegative functions over a set of discrete variables. In Section 2, we show how, using log-space transformation, one can reformulate the MPE problem as a WCSP, a general constraint optimization framework which consists in minimizing a sum of nonnegative *cost functions* over a set of discrete variables [Meseguer *et al.*, 2006].

Exact algorithms for MPE (or WCSP) are based either on Depth-First Branch and Bound (DFBB) search algorithm or on inference methods, including variable elimination, jointree, and compilation. Inference methods generally run out of memory on large and complex networks (with large treewidth). The current state-of-the-art MPE solvers combine DFBB and memory-bounded inference methods (see for example [Sánchez *et al.*, 2008; Marinescu and Dechter, 2009]). DFBB is a complete tree search method using linear memory space. During search, for minimization problems, the algorithm maintains an upper bound UB on the minimal cost solution, such as the cost of the best solution found so far. Moreover, each node in the search tree is associated with a lower bound LB, an underestimation of a minimum cost solution to the subproblem below the node. If $LB \geq UB$, DFBB prunes the search space below the current node. Among the different techniques used to provide strong lower bound, soft local consistencies have been introduced in WCSPs. Since their worst-case time complexities are exponential in the maximum arity (number of variables) of the cost functions [Cooper *et al.*, 2010], they are usually enforced on small arity (2 or 3) cost functions only. It is therefore desirable to decompose cost functions into smaller arity cost functions. Note that variable elimination may also benefit from this decomposition (as the number of neighboring variables may decrease).

It is well-known that *conditional independences* (CI) allows to factorize probability distributions. However neither Bayesian networks nor Markov networks can explicitly represent all CI perfectly [Koller and Friedman, 2009][1]. There may also be finer-grain *context-specific* independences, such as $(X \perp\!\!\!\perp Y \mid Z = z)$, which is not explicit in the network and which can be exploited after entering evidence $Z = z$ or performing variable elimination or value pruning by soft local consistency. In Markov networks, several factorizations may exist in the case of nonpositive distributions. In Section 3, we show how to exploit efficiently pairwise independence in WCSPs in the context of DFBB enhanced with soft local consistency and variable elimination.

2 Preliminaries

A *Probabilistic Graphical Model* (PGM) (or *Gibbs distribution*) is defined by a product of nonnegative functions $\mathcal{F}$, over a set of discrete variables $\mathcal{X}$, conveying probabilistic or deter-

This work has been partially funded by the french "Agence nationale de la Recherche", reference ANR-10-BLA-0214.

[1] A famous example is the *segregation network* [Fishelson *et al.*, 2005], where $\mathbb{P}(G_{i,jp} \mid G_{a,jp}, G_{a,jm}, S_{i,jp}) = 2\mathbb{P}(G_{i,jp} \mid G_{a,jp}, S_{i,jp})\mathbb{P}(G_{i,jp} \mid G_{a,jm}, S_{i,jp})$ because $(G_{a,jp} \perp\!\!\!\perp G_{a,jm} \mid G_{i,jp}, S_{i,jp})$, but this cannot be represented in a BN formulation.

ministic information [Koller and Friedman, 2009]. Subsets of $\mathcal{X}$ will be denoted using bold letters such as $\mathbf{S}$.

Definition 1 (PGM). *A PGM is a triplet $(\mathcal{X}, \mathcal{D}, \mathcal{F})$ with $\mathcal{X} = \{X_1, \ldots, X_n\}$, a set of variables, $\mathcal{D} = \{D_{X_1}, \ldots, D_{X_n}\}$, a set of finite domains of maximum size $d = \max_{i=1}^{n} d_{X_i}$ ($d_{X_i} = |D_{X_i}|$), and $\mathcal{F} = \{f_1, \ldots, f_e\}$, a set of nonnegative real valued functions, each defined over a subset of variables $\mathbf{S}_i \subseteq \mathcal{X}$ (i.e., the scope). It defines a joint distribution:*

$$\mathbb{P}(\mathcal{X}) = \frac{\prod_{i=1}^{e} f_i(\mathbf{S}_i)}{\sum_{\mathcal{X}} \prod_{i=1}^{e} f_i(\mathbf{S}_i)}$$

Bayesian networks represent a specific case using one conditional probability per variable ($e = n$) and a unit normalizing constant. The *Most Probable Explanation* (MPE) problem is to find the most likely assignment to all variables in $\mathcal{X}$ maximizing $\mathbb{P}(\mathcal{X})$.

A *Constraint Satisfaction Problem* (CSP) is a triplet $(\mathcal{X}, \mathcal{D}, \mathcal{C})$ with $\mathcal{X}, \mathcal{D}$ defined as for PGM, and $\mathcal{C}$, a set of constraints. Each constraint is defined as a boolean function over a subset of variables $\mathbf{S}_i \subset \mathcal{X}$ indicating which tuples in $D_{\mathbf{S}_i} = \prod_{X \in \mathbf{S}_i} D_X$ are authorized or not. The problem is to find a *feasible assignment* of $\mathcal{X}$, satisfying all the constraints. A *Weighted Constraint Satisfaction Problem* (WCSP) is an extension of CSP targeted towards optimization.

Definition 2 (WCSP). *A WCSP is a triplet $(\mathcal{X}, \mathcal{D}, \mathcal{W})$ with $\mathcal{X}, \mathcal{D}$ defined as for PGM, and $\mathcal{W} = \{w_1, \ldots, w_e\}$, a set of* cost functions. *Each cost function is defined over a subset of variables and takes nonnegative integer values in $E^+ = \mathbb{N} \cup \{\top\}$ ($\top = +\infty$ being associated to forbidden assignments[2]). The goal is to find a feasible assignment of $\mathcal{X}$ minimizing $\sum_{i=1}^{e} w_i(\mathbf{S}_i)$.*

A PGM $(\mathcal{X}, \mathcal{D}, \mathcal{F})$ can be translated into a WCSP $(\mathcal{X}, \mathcal{D}, \mathcal{W})$ with $\forall i \in [1, e], w_i(\mathbf{S}_i) = \lceil -M \log(f_i(\mathbf{S}_i)) + C \rceil$ (M, C being two positive constants for real-to-integer precision and nonnegativity of w_i) which preserves the set of optimal solutions if a sufficiently large M value is used.

In the rest of the paper, we use f to denote cost functions. Given an assignment $t \in D_{\mathcal{X}}$, $t[\mathbf{S}]$ denotes the subassignment of t to the variables in $\mathbf{S}$. Let $f = f_1 + f_2$ denote the *sum (join) of two cost functions* defined as $f(t) = f_1(t[\mathbf{S}_1]) + f_2(t[\mathbf{S}_2]), \forall t \in D_{\mathbf{S}_1 \cup \mathbf{S}_2}$. Let $f = f_1 - f_2$ denote the *subtraction of two cost functions* such that $\mathbf{S}_2 \subseteq \mathbf{S}_1$ and $f(t) = f_1(t) - f_2(t[\mathbf{S}_2]), \forall t \in D_{\mathbf{S}_1}$ with $\top - \top = \top$. Let $f[\mathbf{S}']$ denote the *projection of a cost function* over a subset $\mathbf{S}'$ of its variables $\mathbf{S}$ such that $\mathbf{S}' \subseteq \mathbf{S}$ and $\forall t' \in D_{\mathbf{S}'} f[\mathbf{S}'](t') = \min_{t \in D_{\mathbf{S}} \text{ s.t. } t[\mathbf{S}'] = t'} f(t)$. A cost function over two (resp. one) variable(s) is called a *binary* (resp. *unary*) cost function. An *empty cost function* has all its costs equal to zero and is removed from the WCSP.

Depth-First Branch and Bound (DFBB) is a complete tree search method to solve WCSPs. Soft local consistency methods produce strong lower bounds for DFBB by applying *Equivalence-Preserving Transformations* (EPT) [Cooper *et al.*, 2010]. An *arc EPT* adds the projection $f[X]$ of a cost function $f(\mathbf{S})$ to a unary cost function $f_1(X), X \in \mathbf{S}$ and subtracts $f[X]$ from f (*i.e.* replaces f by $f - f[X]$) in order to get an equivalent problem. Enforcing *soft arc consistency* consists in applying arc EPTs for all cost functions and all directions ($\forall f(\mathbf{S}) \in \mathcal{W}, \forall X \in \mathbf{S}$) until all the projections are empty. A *directional arc EPT* adds the projection $(f + f_2)[X]$ from binary and unary cost functions $f(X, Y)$ and $f_2(Y)$ to $f_1(X)$ and subtracts the result of this projection from f. Enforcing *soft directional arc consistency* (DAC) consists in applying directional arc EPTs on all cost functions in one direction only ($X < Y$), as indicated by a total order on $\mathcal{X}$, thereby ensuring termination. DFBB can be further improved using on-the-fly *variable elimination* [Larrosa, 2000]. Let $F = \{f_i \in \mathcal{W} \text{ s.t. } X \in \mathbf{S}_i\}$ and $\mathbf{S} = \bigcup_{f_i \in F} \mathbf{S}_i$, the elimination of variable X consists in replacing X and F by the projection $(\sum_{f \in F} f)[\mathbf{S} \setminus \{X\}]$ in the current WCSP. The elimination of a variable is exponential in the size of $\mathbf{S}$ and is cheap if $|\mathbf{S}|$ is small but also if X is assigned or if it is connected to the rest of the problem either through a single cost function or through at least one bijective (equality) constraint. DFBB enhanced with on-the-fly i-bounded elimination (all variables with $|\mathbf{S}| \leq i$ are eliminated) and soft local consistency (EDAC for binary&ternary cost functions [Sánchez *et al.*, 2008]) is denoted as DFBB-VE(i) in the experiments.

W	X	Y	U	f
1	1	1	1	5
1	1	1	2	$\top$
1	1	2	1	$\top$
1	1	2	2	3
1	2	1	1	6
1	2	1	2	5
1	2	2	1	$\top$
1	2	2	2	$\top$
2	1	1	1	4
2	1	1	2	$\top$
2	1	2	1	$\top$
2	1	2	2	2
2	2	1	1	2
2	2	1	2	1
2	2	2	1	3
2	2	2	2	1

=

W	X	Y	f_1
1	1	1	5
1	1	2	3
1	2	1	5
1	2	2	$\top$
2	1	1	4
2	1	2	2
2	2	1	1
2	2	2	1

+

X	Y	U	f_2
1	1	1	0
1	1	2	$\top$
1	2	1	$\top$
1	2	2	0
2	1	1	1
2	1	2	0
2	2	1	2
2	2	2	0

Figure 1: $f(W, X, Y, U)$ applied on four Boolean variables is decomposable w.r.t. W, U as $f_1(W, X, Y) + f_2(X, Y, U)$.

3 Pairwise cost function decomposition

We first define the notion of *pairwise decomposition*.

Definition 3. *A* pairwise decomposition *of a cost function $f(\mathbf{S})$ with respect to two variables $X, Y \in \mathbf{S}$ ($X \neq Y$), is a rewriting of f into a sum of two (nonnegative) cost functions $f_1(\mathbf{S} \setminus \{Y\})$ and $f_2(\mathbf{S} \setminus \{X\})$ such that:*

$$f(\mathbf{S}) = f_1(\mathbf{S} \setminus \{Y\}) + f_2(\mathbf{S} \setminus \{X\})$$

An example of decomposable function is given in Figure 1. A pairwise decomposition replaces a cost function of arity $r = |\mathbf{S}|$ with smaller-arity $(r - 1)$ cost functions. This decomposition process can be recursively repeated on the resulting cost functions f_1, f_2 until no pairwise decomposition can be found for each remaining cost function. A cost function $f(\mathbf{S})$ that cannot be pairwise decomposed for any choice of $X, Y \in \mathbf{S}$ is said to be *non decomposable*. A WCSP is *pairwise decomposed* if all its cost functions are non decomposable. This property is enforced by applying the previously described iterative process to all cost functions.

[2] For simplicity, we use infinite $\top$, but our results are still valid for finite $\top$.

The following theorem shows that testing if a cost function can be pairwise decomposed w.r.t X, Y is equivalent to testing pairwise independence between X and Y in an appropriate probability distribution.

Theorem 1 (Equivalence of pairwise independence and decomposition)**.** *Let $\mathbf{S} = \{X, Y\} \cup \mathbf{Z}$ a set of random variables, $f(\mathbf{S})$ a cost function on $\mathbf{S}$ with at least one feasible assignment, and a distribution $\mathbb{P}_f = \frac{1}{\sum_{\mathbf{S}} \exp(-f(\mathbf{S}))} \exp(-f(\mathbf{S}))$ (a* Gibbs distribution *parameterized by f). $(X \perp\!\!\!\perp Y \mid \mathbf{Z})$ denotes that X and Y are* pairwise independent *given all other variables in $\mathbb{P}_f$. Then,*

$$(X \perp\!\!\!\perp Y \mid \mathbf{Z}) \iff f(X, \mathbf{Z}, Y) = f_1(X, \mathbf{Z}) + f_2(\mathbf{Z}, Y)$$

Instead of testing pairwise independence in $\mathbb{P}_f$, which involves summations and multiplications of real numbers, we propose a simpler test for pairwise decomposability based on equalities of cost differences. For a decomposable cost function $f(X, \mathbf{Z}, Y) = f_1(X, \mathbf{Z}) + f_2(\mathbf{Z}, Y)$ that takes only finite costs, and for any tuple $z \in D_{\mathbf{Z}}$, if we consider any fixed pair of values k, l for Y then the difference $f(x, z, k) - f(x, z, l) = (f_1(x, z) + f_2(z, k)) - (f_1(x, z) + f_2(z, l)) = (f_2(z, k) - f_2(z, l))$ does not depend on x. When $f(X, \mathbf{Z}, Y)$ is not limited to finite costs however, subtracting an infinite cost from another cost is ill-defined. For example, if $f_2(z, k)$ and $f_2(z, l)$ are both equal to $\top$ then $f(x, z, k) = f(x, z, l) = \top$ for all x. Therefore, $f_1(x, z)$ can take *any* value and the decomposition will still hold. Infinite costs offer additional freedom in decomposition and must be specifically handled.

To design a test that can exploit infinite costs, we introduce an extended cost valuation structure $E = \mathbb{Z} \cup \{-\top, \top, \Omega\}$ including negative costs and a special absorbing element Ω which captures the freedom generated by subtraction of infinite costs. Let $a \boxminus b$ denote the difference of two elements $a, b \in E$: $a \boxminus b = a - b$, except for $\top \boxminus \top = -\top \boxminus -\top = a \boxminus \Omega = \Omega \boxminus b = \Omega$, $\top \boxminus -\top = \top \boxminus c = c \boxminus -\top = \top$, $-\top \boxminus \top = -\top \boxminus c = c \boxminus \top = -\top$ with $c \in \mathbb{Z}$. The comparison of two elements $a, b \in E$ denoted by $a \stackrel{\circ}{=} b$ is true if and only if (1) $a, b \in \mathbb{Z} \cup \{\top, -\top\}$ and $a = b$, or (2) $a = \Omega$ or $b = \Omega$. We then have:

Theorem 2. *A cost function $f(X, \mathbf{Z}, Y)$ is pairwise decomposable w.r.t. X, Y iff $\forall z \in D_{\mathbf{Z}}, \forall k, l \in D_Y (k < l)$:*

$$\stackrel{\circ}{=}_{x \in D_X} f(x, z, k) \boxminus f(x, z, l)$$

Example 1. *In Figure 1, we have $f(W, X, Y, U)$ being pairwise decomposable w.r.t. W and U due to the fact that:*

$f(1111) \boxminus f(1112) \stackrel{\circ}{=} f(2111) \boxminus f(2112)$ *:* $5 \boxminus \top \stackrel{\circ}{=} 4 \boxminus \top$
$f(1121) \boxminus f(1122) \stackrel{\circ}{=} f(2121) \boxminus f(2122)$ *:* $\top \boxminus 3 \stackrel{\circ}{=} \top \boxminus 2$
$f(1211) \boxminus f(1212) \stackrel{\circ}{=} f(2211) \boxminus f(2212)$ *:* $6 \boxminus 5 \stackrel{\circ}{=} 2 \boxminus 1$
$f(1221) \boxminus f(1222) \stackrel{\circ}{=} f(2221) \boxminus f(2222)$ *:* $\top \boxminus \top \stackrel{\circ}{=} 3 \boxminus 1$

Next, we show how to construct f_1, f_2 of a pairwise decomposable cost function using projections and subtractions of cost functions (see Figure 1 for its application).

Theorem 3. *Let $f(X, \mathbf{Z}, Y)$ be pairwise decomposable w.r.t. X, Y. Then $f_1(X, \mathbf{Z}) = f[X, \mathbf{Z}]$ and $f_2(\mathbf{Z}, Y) = (f - f_1)[\mathbf{Z}, Y]$ is a valid decomposition of f, i.e. $f = f_1 + f_2$.*

Enforcing pairwise decomposition on a cost function $f(\mathbf{S})$ with $r = |\mathbf{S}|$ may require $\frac{r(r-1)}{2}$ tests, *i.e.* checking for all the pairs of variables in $\mathbf{S}$. And it may produce two (non empty) $(r-1)$-ary cost functions. Repeating pairwise decomposition on the resulting cost functions will produce at most $\min(\binom{r-p}{r}, 2^p)$ cost functions of arity $(r-p)$ with a different scope. Each test is linear in the size of the $(r-p)$-ary cost functions, in $O(d^{(r-p)+1})$. So the overall worst-case complexity of enforcing pairwise decomposition for e original cost functions with maximum arity r is $O(e \sum_{p=0}^{r-1} 2^p (r-p)^2 d^{(r-p)+1}) = O(e \sum_{p=0}^{r-1} (r-p)^2 d^{r+1}) = O(e \frac{r(r+1)(2r+1)}{6} d^{r+1}) = O(e r^3 d^{r+1})$ in time and $O(e \max_{p=0}^{r-1} 2^p d^{(r-p)}) = O(e d^r)$ in space.

At each step, the choice of the variables used for decomposition and the way costs are spread across f_1 and f_2 may influence the decomposability of f_1 and f_2 for next iteration. Under certain conditions, it is possible to find a sequence of pairwise decompositions (which pair of variables to select in f and how to spread f across f_1 and f_2) which will lead to an optimal decomposition (with minimal arity).

Property 1 (Unique minimal decomposition for finite cost functions [Hammersley and Clifford, 1971])**.** *A cost function with only finite costs admits a* unique minimal *decomposition.*

Sketch of proof. If a cost function $f(\mathbf{S})$ has only finite costs, then its associated probability distribution $\mathbb{P}_f$ (see Theorem 1) is positive. The *Hammersley & Clifford* theorem [Hammersley and Clifford, 1971][3] says that a positive distribution factorizes over a unique minimal Markov network $\mathcal{H}$. The network is minimal in the sense that removing an edge will create a new conditional independence not satisfied by $\mathbb{P}_f$. By taking the maximal cliques in $\mathcal{H}$, each clique gives the scope of a factor (a nonnegative function) and $\mathbb{P}_f$ is then equal to the product of these factors divided by a normalizing constant. This factorization is equivalent to a sum of cost functions by taking $-\log(\mathbb{P}_f)$ (see the proof of Theo 1). From this resulting decomposition, it is easy to construct a reverse sequence of valid (nonnegative) cost function decompositions until the original cost function f is reached. □

However, as soon as there are infinite costs, the previous theorem fails to apply. In this case, we propose a heuristic approach whose goal is to accumulate costs on the first variables in the DAC variable ordering. This should provide stronger DAC-based lower bounds. We therefore test pairs of variables in $f(\mathbf{S})$ in a reverse DAC order[4] and project cost functions (Theorem 3) on the scope containing the first variables in the DAC order first. In Figure 1, assuming a lexicographic order, this corresponds to project first on $\{W, X, Y\}$. We show that our heuristic approach is able to find an optimal decomposition for some particular functions.

Property 2 (Tree structure identification for hard constraints [Meiri *et al.*, 1990])**.** *A cost function with only infinite costs that admits a decomposition into a tree of binary constraints is identifiable by our decomposition strategy.*

[3] See also Theorem 4.5 page 121 in [Koller and Friedman, 2009].

[4] We test Y, Z before W, X if $\max(Y, Z) > \max(W, X) \vee (\max(Y, Z) = \max(W, X) \wedge \min(Y, Z) > \min(W, X))$.

This result follows from [Meiri *et al.*, 1990] and the fact that a pairwise decomposition w.r.t. X, Y will remove a redundant edge $\{X, Y\}$ in the *minimal network*[5] without affecting the set of solutions. For instance, a global equality constraint on $\mathbf{S}$ will be decomposed into a tree of binary equality constraints: $f(\mathbf{S}) = \sum_{Y \in \mathbf{S} \setminus \{X\}} f_Y(X, Y)$ with $f_Y(X, Y) \equiv (X = Y)$.

Similarly, a linear cost function $f(X_1, \ldots, X_r) = \sum_{i=1}^{r} a_i X_i$ will be decomposed into a sum of r unary cost functions[6].

Related Works

Related works have considered the factorization of specific probability functions such as the *noisy-or* and its generalizations [Heckerman and Breese, 1996], or more general factorizations dedicated to probabilistic inference [Savicky and Vomlel, 2007]. Compared to our approach, both works add extra variables. Note that another possible approach to decrease the arity of cost functions is to switch to the dual representation [Meseguer *et al.*, 2006]. This has been tested in [de Givry *et al.*, 2003] for Max-3SAT and was apparently ineffective. This should likely worsen for larger domains.

4 Project&Subtract on binary cost functions

When a WCSP is pairwise decomposed, it is still possible to infer valuable information from the resulting nonbinary cost functions by projecting & subtracting them on smaller arity cost functions. We choose to project each nonbinary cost function on all the possible binary cost functions inside its scope[5] by following an order of projections&subtractions compatible with the DAC variable ordering (*i.e.* project first on the pair of variables with the smallest DAC positions). Each projection $f_i(X, Y) = f[X, Y]$ is followed by a subtraction $f = f - f_i$ in order to preserve the problem equivalence. Note that f may be empty after these subtractions and is then removed from the problem. Note also that the same binary cost function may receive projected costs from several nonbinary cost functions having overlapping scopes, resulting in stronger inference. The worst-case complexity of *project&subtract* applied on e cost functions with maximum arity r is $O(er^2d^r)$ in time and $O(er^2d^2)$ in space.

Example 2. *In the example of Figure 1,* $f_1(W, X, Y) = b_1(W, X) + b_2(X, Y) + f_3(W, X, Y)$ *and* $f_2(X, Y, U) = b_3(X, U) + b_4(Y, U) + f_4(X, Y, U)$. *Finally, a lower bound equal to 1 is deduced by soft arc consistency applied to* b_1.

W	X	b_1
1	1	3
1	2	5
2	1	2
2	2	1

X	Y	b_2
1	1	2
1	2	0
2	1	0
2	2	0

X	U	b_3
1	1	0
1	2	0
2	1	1
2	2	0

Y	U	b_4
1	1	0
1	2	0
2	1	1
2	2	0

[5] In the CSP framework, the binary CSP, called the *minimal network*, that best approximates a nonbinary relation, is defined by the projections of the relation on all pairs of variables [Meiri *et al.*, 1990].

[6] Note that soft arc consistency will do the same if the resulting empty cost function is removed after the projections.

W	X	Y	f_3
1	1	1	0
1	1	2	0
1	2	1	0
1	2	2	$\top$
2	1	1	0
2	1	2	0
2	2	1	0
2	2	2	0

X	Y	U	f_4
1	1	1	0
1	1	2	$\top$
1	2	1	$\top$
1	2	2	0
2	1	1	0
2	1	2	0
2	2	1	0
2	2	2	0

5 Experimental results

DFBB-VE(i)[7] maintaining EDAC and using a dynamic *conflict-based* variable ordering [Lecoutre *et al.*, 2009] obtained the best result in the UAI'08[8] (and UAI'10[9] for the 20-minutes track) MPE Evaluation contest, except for two hard benchmarks: linkage analysis and grid networks.

The genetic linkage analysis networks. The problem instances have 334 to 1289 variables. The maximal domain size d is between 3 and 7 and the maximal arity is 5. The benchmark family consists of 22 problems.

The grid networks. Each problem is an $l \times l$ grid and each 3-ary CPT is generated uniformly randomly. 50, 75 or 90% of the CPTs are deterministic and all variables are Boolean. The instances have l ranging from 12 to 50 (*i.e.* $n = 2500$ variables). The benchmark family has 32 problems.

Experimentations[7] were performed on a 2.6 GHz Intel Xeon computer with 4GB running Linux 2.6. Total CPU-times are in seconds and limited to 1 hour for each instance ("-" means time out). No initial upper bound is given. We tested enforcing pairwise decomposition (DFBB-VE(i)+dec), project&subtract of n-ary cost functions (DFBB-VE(i)+ps), and the combination of both techniques (DFBB-VE(i)+dec+ps). Pairwise decomposition, project&subtract, and variable elimination of degree $|\mathbf{S}| \leq i$ are performed in preprocessing only. On-the-fly variable elimination for degree at most 2 is used during search. All methods use *MinFill* variable elimination ordering heuristic and the reverse order is used for DAC. The following table summarizes the number of problems solved by the various versions of DFBB-VE(i). Pairwise decomposition and project&subtract solved more problems than DFBB-VE(i) alone, and their combination obtained the best result, except for Linkage with $i = 2, 3$. Remarkably, all grids instances were solved with large i-bounded variable elimination, showing the effect of pairwise decomposition, especially when there is sufficient determinism. A similar factorization was observed in [Sánchez *et al.*, 2004] for CSPs.

	i=2	i=3	i=4	i=5	i=6	i=7
Linkage (22)						
DFBB-VE(i)	14	14	15	13	12	10
DFBB-VE(i)+dec	16	**19**	17	13	14	13
DFBB-VE(i)+ps	**17**	16	17	18	16	16
DFBB-VE(i)+dec+ps	16	18	**19**	**19**	**19**	**17**
Grids (32)						
DFBB-VE(i)	28	28	25	24	17	18
DFBB-VE(i)+dec	29	29	29	28	28	31
DFBB-VE(i)+ps	28	28	28	23	25	24
DFBB-VE(i)+dec+ps	**31**	**30**	**31**	**31**	**32**	**32**

[7] `mulcyber.toulouse.inra.fr/projects/toulbar2` version 0.9.4.

[8] `graphmod.ics.uci.edu/uai08/Evaluation`

[9] `www.cs.huji.ac.il/project/UAI10`

Problem				DFBB-VE(i)			DFBB-VE(i) +dec+ps			AOBB-C+SMB(j)		+VE(i)		+VE(i)+dec+ps	
	n	d	w	time (s)	i	n'	time (s)	i	n'	j	i	time (s)	n'	time (s)	n'
Linkage															
ped1	334	4	16	0.12	3	159	**0.07**	3	142	10	3	0.1	159	0.08	142
ped7	1068	4	38	4.04	2	375	**1.18**	4	213	20	4	1915.56	274	131.14	213
ped9	1118	4	31	-			**3.36**	6	147	20	6	-	179	104.62	147
ped18	1184	5	24	149.71	4	365	**3.19**	5	213	20	5	54.67	255	18.82	213
ped19	793	5	32	-			-			20	6	-	337	-	304
ped20	437	5	23	3.46	4	95	**0.39**	6	42	16	6	407.6	68	66.53	42
ped23	402	5	26	0.09	4	79	**0.05**	3	98	12	3	6.05	114	1.52	98
ped25	1289	5	29	1207.97	4	269	**0.65**	6	102	20	6	25.91	158	32.51	102
ped30	1289	5	24	543.20	2	633	**5.44**	5	213	20	5	28.39	272	10.10	231
ped33	798	4	30	0.84	3	272	**0.54**	6	151	18	6	21.33	175	8.35	151
ped34	1160	5	36	1.13	2	326	**0.36**	5	167	20	5	-	196	24.56	167
ped37	1032	4	22	0.21	5	103	**0.11**	4	120	10	4	87.18	141	9.58	120
ped38	724	5	18	0.24	5	122	**0.18**	5	97	12	5	137.60	122	124.70	97
ped39	1272	5	22	16.68	4	183	**0.24**	5	107	18	5	6.87	148	2.60	107
ped41	1062	5	35	-			**302.05**	4	308	20	4	-	372	1271.31	308
ped42	448	5	24	1.94	4	140	**0.34**	6	81	16	6	240.94	95	155.39	81
ped44	811	4	31	-			505.46	5	215	20	5	2631.71	248	**333.89**	215
ped50	514	6	18	0.90	4	133	**0.18**	4	118	12	4	316.92	133	521.92	118
Grids															
90-24-1	576	2	35	0.07	3	566	**0.04**	3	291	18	3	2323.01	566	0.63	291
90-26-1	676	2	41	0.29	3	668	**0.07**	3	500	16	3	2821.66	668	20.16	500
90-30-1	900	2	49	5.50	2	766	**0.26**	4	382	18	4	-	766	3.57	382
90-34-1	1156	2	63	328.42	2	1052	**0.48**	5	518	20	5	-	1037	14.41	518
90-38-1	1444	2	64	-			1.96	6	249	20	6	-	983	**0.64**	249

Table 1: Comparison with/w. out the preprocessing dec+ps for DFBB-VE(i) and AOBB-C+SMB(j)

We then compared the best method with AND/OR Branch and Bound with caching and static mini-bucket j-bound (AOBB-C+SMB(j)) which got the best results for Linkage and Grids at UAI'08 Evaluation. This method is implemented in `aolibWCSP`[10]. The value j is chosen as in [Marinescu and Dechter, 2009]. Table 1 gives problem sizes (n, d), treewidth (w), and summarizes the total CPU-time used by the various methods with the corresponding values i (choosing $i \in [2, 7]$ which provides the best result) and j. The preprocessing "dec+ps" globally improves the results of DFBB-VE(i) and AOBB-C+SMB(j). The preprocessing time was always less than a second in our experiments (n', number of variables after preprocessing). DFBB-VE(i)+dec+ps gave the best results for all problems except ped44 and 90-38-1. Note that the range of variation of i for DFBB-VE(i)+dec+ps is smaller than the range of j for AOBB-C+SMB(j)[11].

6 Conclusion

Pairwise decomposition combined with projections on binary cost functions and variable elimination is a powerful technique inside DFBB for MPE. Further work should be done on approximate pairwise decomposition and experiments performed on other problems, including probabilistic inference.

[10] `graphmod.ics.uci.edu/group/aolibWCSP`

[11] A good way of tuning i is to take the value corresponding to the maximum of the degree distribution for the considered graphical model (*i.e.* $i = 5$ for Linkage and $i = 6$ for Grids).

A Proof of Theorem 1

In the following, we use $\mathbb{P}$ as a shorthand for $\mathbb{P}_f$ and $C = \sum_{\mathbf{S}} \exp(-f(\mathbf{S})) > 0$, a normalizing constant. By definition of conditional independence [Lauritzen, 1996], we have: $(X \perp\!\!\!\perp Y \mid \mathbf{Z}) \iff \mathbb{P}(X, Y, \mathbf{Z}) = \mathbb{P}(X \mid \mathbf{Z})\mathbb{P}(Y \mid \mathbf{Z})\mathbb{P}(\mathbf{Z})$ $\iff \mathbb{P}(X, Y, \mathbf{Z}) = \frac{\mathbb{P}(X,\mathbf{Z})}{\mathbb{P}(\mathbf{Z})}\mathbb{P}(Y, \mathbf{Z})$, with $\mathbb{P}(\mathbf{Z}) > 0$.

$\boxed{\Longrightarrow}$ Assume $\mathbb{P}(X, Y, \mathbf{Z}) = \mathbb{P}(X \mid \mathbf{Z})\mathbb{P}(Y \mid \mathbf{Z})\mathbb{P}(\mathbf{Z})$. We have $f(X, \mathbf{Z}, Y) = -\log(C\mathbb{P}(X, Y, \mathbf{Z})) = -\log(C\mathbb{P}(X \mid \mathbf{Z})\mathbb{P}(Y \mid \mathbf{Z})\mathbb{P}(\mathbf{Z}))$. Let define $f'_1(X, \mathbf{Z}) = -\log(\mathbb{P}(X \mid \mathbf{Z}))$ and $f'_2(\mathbf{Z}, Y) = -\log(\mathbb{P}(Y \mid \mathbf{Z})\mathbb{P}(\mathbf{Z}))$, two non-negative cost functions. Thus $f(X, \mathbf{Z}, Y) = f'_1(X, \mathbf{Z}) + f'_2(\mathbf{Z}, Y) - \log(C)$. Because f is nonnegative, we have $\forall x \in D_X, \forall y \in D_Y, \forall z \in D_{\mathbf{Z}}, f'_1(x, z) + f'_2(z, y) - \log(C) \geq 0$. If $\log(C)$ is negative, we just add $-\log(C)$ to f'_1 or f'_2 in order to obtain f_1, f_2. Otherwise, for each $z \in D_{\mathbf{Z}}$, we decompose $\log(C) = c_z^1 + c_z^2$ into two positive numbers. Let $\hat{x}_z = \operatorname{argmin}_{x \in D_X} f'_1(x, z)$ and $\hat{y}_z = \operatorname{argmin}_{y \in D_Y} f'_2(z, y)$. Moreover, we have $f'_1(\hat{x}_z, z) + f'_2(z, \hat{y}_z) - \log(C) = f(\hat{x}_z, z, \hat{y}_z) \geq 0$. A feasible solution is $c_z^1 = \min(f'_1(\hat{x}_z, z), \log(C))$ and $c_z^2 = \log(C) - c_z^1$. Either $c_z^1 = \log(C) \leq f'_1(\hat{x}_z, z)$ and $c_z^2 = 0$, or $c_z^1 = f'_1(\hat{x}_z, z)$ and $c_z^2 = \log(C) - f'_1(\hat{x}_z, z)$. In this case, $f'_2(z, \hat{y}_z) - c_z^2 = f'_2(z, \hat{y}_z) + f'_1(\hat{x}_z, z) - \log(C) \geq 0$, thus $\forall y \in D_Y, f'_2(z, y) - c_z^2 \geq 0$. We define $\forall x, y, z$, $f_1(x, z) = f'_1(x, z) - c_z^1$ and $f_2(z, y) = f'_2(z, y) - c_z^2$, which are nonnegative numbers. Finally, we deduce $f(X, \mathbf{Z}, Y) = f_1(X, \mathbf{Z}) + f_2(\mathbf{Z}, Y)$.

$\boxed{\Longleftarrow}$ Assume we have three cost functions $f(X,\mathbf{Z},Y), f_1(X,\mathbf{Z}), f_2(\mathbf{Z},Y)$ such that $f(X,\mathbf{Z},Y) = f_1(X,\mathbf{Z}) + f_2(\mathbf{Z},Y)$. We have $\exp(-f(X,\mathbf{Z},Y)) = \exp(-f_1(X,\mathbf{Z}))\exp(-f_2(\mathbf{Z},Y))$. By marginalization, we have $\mathbb{P}(X,\mathbf{Z}) = \sum_Y \mathbb{P}(X,Y,\mathbf{Z}) = \frac{1}{C}\sum_Y \exp(-f(X,\mathbf{Z},Y)) = \frac{1}{C}\exp(-f_1(X,\mathbf{Z})(\sum_Y \exp(-f_2(\mathbf{Z},Y)))$, $\mathbb{P}(Y,\mathbf{Z}) = \sum_X \mathbb{P}(X,Y,\mathbf{Z}) = \frac{1}{C}\exp(-f_2(\mathbf{Z},Y))(\sum_X \exp(-f_1(X,\mathbf{Z}))$ and $\mathbb{P}(\mathbf{Z}) = \sum_{X,Y} \mathbb{P}(X,Y,\mathbf{Z}) = \frac{1}{C}(\sum_X \exp(-f_1(X,\mathbf{Z})))(\sum_Y \exp(-f_2(\mathbf{Z},Y)))$. Finally, we deduce $\frac{\mathbb{P}(X,\mathbf{Z})}{\mathbb{P}(\mathbf{Z})}\mathbb{P}(Y,\mathbf{Z}) = \frac{1}{C}\exp(-f_1(X,\mathbf{Z}))\exp(-f_2(\mathbf{Z},Y)) = \mathbb{P}(X,Y,\mathbf{Z})$.

B Proof of Theorem 2

It is always possible to decompose a cost function f into three (nonnegative) cost functions $f(X,\mathbf{Z},Y) = f_1(X,\mathbf{Z}) + f_2(\mathbf{Z},Y) + \Delta(X,\mathbf{Z},Y)$ (f_1, f_2 possibly being equal to zero constant functions). A nonzero cost function is called *reducible* if f_1 or f_2 are nonzero cost functions. Otherwise it is called *irreducible*. We give a specific condition for testing *reducibility*.

Property 3. *A nonzero cost function $f(X,\mathbf{Z},Y)$ is* reducible *if $\exists x \in D_X, \exists z \in D_\mathbf{Z}$ s.t. $\min_{y\in D_Y} f(x,z,y) > 0$ (i.e. $f_1 = f[X,\mathbf{Z}] \neq 0$) or $\exists y \in D_Y, \exists z \in D_\mathbf{Z}$ s.t. $\min_{x\in D_X} f(x,z,y) > 0$ (i.e. $f_2 = f[\mathbf{Z},Y] \neq 0$).*

Property 4. *If $f_1(X,\mathbf{Z})$, $f_2(\mathbf{Z},Y)$, $\Delta(X,\mathbf{Z},Y)$, three (nonnegative) cost functions, is a maximal reducing of $f(X,\mathbf{Z},Y)$ then $\Delta(X,\mathbf{Z},Y)$ is irreducible.*

Using Property 3, we prove the following lemma.

Lemma 1. *If $f(X,\mathbf{Z},Y)$ is a nonzero* irreducible *cost function, then $\exists z \in D_\mathbf{Z}, \exists k,l \in D_Y (k \neq l), \exists u,v \in D_X (u \neq v)$ s.t. $f(u,z,k) \boxminus f(u,z,l) \not\triangleq f(v,z,k) \boxminus f(v,z,l)$.*

Proof by contradiction. Assume $\forall z \in D_\mathbf{Z}, \forall k,l \in D_Y (k \neq l), \forall u,v \in D_X (u \neq v)$ s.t. $f(u,z,k) \boxminus f(u,z,l) \triangleq f(v,z,k) \boxminus f(v,z,l)$. Remember that cost functions always have costs in $E^+ = \mathbb{N} \cup \{\top\}$ and not in E. We have either:
First case: $\exists u,z,k,l$ **s.t.** $f(u,z,k) \boxminus f(u,z,l) \not\triangleq 0$

$\boxed{f(u,z,k) \boxminus f(u,z,l) > 0 \text{ or } f(u,z,k) \boxminus f(u,z,l) = \top}$

So $0 \leqslant f(u,z,l) < f(u,z,k)$. Moreover $\forall v \in D_X$, $f(u,z,k) \boxminus f(u,z,l) \triangleq f(v,z,k) \boxminus f(v,z,l)$. Thus $\forall v \in D_X$, $f(v,z,k) \boxminus f(v,z,l) > 0$, or $f(v,z,k) \boxminus f(v,z,l) = \top$, or $f(v,z,k) \boxminus f(v,z,l) = \Omega$. Consequently $0 \leqslant f(v,z,l) < f(v,z,k)$ or $f(v,z,k) = f(v,z,l) = \top$. So $\min_{v\in D_X} f(v,z,k) > 0$.

$\boxed{f(u,z,k) \boxminus f(u,z,l) < 0 \text{ or } f(u,z,k) \boxminus f(u,z,l) = -\top}$

So $0 \leqslant f(u,z,k) < f(u,z,l)$. Moreover $\forall v \in D_X$, $f(u,z,k) \boxminus f(u,z,l) \triangleq f(v,z,k) \boxminus f(v,z,l)$. Thus $\forall v \in D_X$, $f(v,z,k) \boxminus f(v,z,l) < 0$, or $f(v,z,k) \boxminus f(v,z,l) = -\top$, or $f(v,z,k) \boxminus f(v,z,l) = \Omega$. Consequently $0 \leqslant f(v,z,k) < f(v,z,l)$ or $f(v,z,k) = f(v,z,l) = \top$. So $\min_{v\in D_X} f(v,z,l) > 0$.
Second case: $\forall u,z,k,l$ **s.t.** $f(u,z,k) \boxminus f(u,z,l) \triangleq 0$

$\boxed{f(u,z,k) \boxminus f(u,z,l) = \Omega}$

So $f(u,z,k) = f(u,z,l) = \top$. Moreover $\forall p \in D_Y$, $f(u,z,p) \boxminus f(u,z,l) = 0$ or $f(u,z,p) \boxminus f(u,z,l) = f(u,z,p) \boxminus \top = \Omega$. Thus $f(u,z,p) = \top$. So $f(u,z,l) = f(u,z,k) = f(u,z,p) = \top$, thus $\min_{p\in D_Y} f(u,z,p) = \top > 0$.

$\boxed{f(u,z,k) = f(u,z,l) > 0}$

So $f(u,z,k) \boxminus f(u,z,l) = 0$ and by assumption, $\forall p \in D_Y$, $f(u,z,p) \boxminus f(u,z,l) = 0$. Thus $f(u,z,k) \boxminus f(u,z,l) = f(u,z,p) \boxminus f(u,z,l)$ and $0 < f(u,z,l) = f(u,z,k) = f(u,z,p)$. So $\min_{p\in D_Y} f(u,z,p) > 0$.

$\boxed{f(u,z,k) = f(u,z,l) = 0}$

By assumption, $\forall p \in D_Y$, $f(u,z,p) \boxminus f(u,z,l) \triangleq 0$. Then $f(u,z,p) \boxminus 0 \triangleq 0$. Thus $f(u,z,p) \boxminus f(u,z,l) \neq \Omega$ and so $f(u,z,k) \boxminus f(u,z,l) = f(u,z,p) \boxminus f(u,z,l)$. Thus $\forall p \in D_Y, f(u,z,p) = 0$. From all these cases, we conclude using Property 3 that $f(X,\mathbf{Z},Y)$ is reducible or equal to the zero function (if always being in the last sub-case). □

Let $a \boxplus b$ denote the addition of two elements $a,b \in E$: $a \boxplus b = a + b$ except for $\top \boxplus \top = \top \boxplus c = c \boxplus \top = \top$, $-\top \boxplus -\top = -\top \boxplus c = c \boxplus -\top = -\top$, $\top \boxplus -\top = -\top \boxplus \top = a \boxplus \Omega = \Omega \boxplus b = \Omega$ with $c \in \mathbb{Z}$.
Now, we are able to prove Theorem 2.
$\boxed{\Longrightarrow}$ Assume $f(X,\mathbf{Z},Y) = f_1(X,\mathbf{Z}) + f_2(\mathbf{Z},Y)$ and $f_1(X,\mathbf{Z})$, $f_2(\mathbf{Z},Y)$ nonnegative functions ($0 \leqslant f_1(X,\mathbf{Z}) \leqslant f(X,\mathbf{Z},Y)$). Then the following systems $\mathcal{S}_z$ of linear equations (using $\boxplus$ and $\triangleq$ operators) must have a solution.

$$(\mathcal{S}_z) \begin{cases} f_1(1z) \boxplus f_2(z1) \triangleq f(1z1) \\ f_1(uz) \boxplus f_2(zk) \triangleq f(uzk) \quad \forall u,k \\ f_1(d_X z) \boxplus f_2(z d_Y) \triangleq f(d_X z d_Y) \end{cases}$$

$\forall z \in D_\mathbf{Z}, \forall k,l \in D_Y, \forall u \in D_X$, from $\mathcal{S}_z$, we deduce
$f(uzk) \boxminus f(uzl) \triangleq (f_1(uz) \boxplus f_2(zk)) \boxminus (f_1(uz) \boxplus f_2(zl))$
$\triangleq (f_1(uz) \boxminus f_1(uz)) \boxplus (f_2(zk) \boxminus f_2(zl))$
1^{st} **case:** $f_1(uz) \in \mathbb{N}$
$f(uzk) \boxminus f(uzl) \triangleq (f_1(uz) \boxminus f_1(uz)) \boxplus (f_2(zk) \boxminus f_2(zl)) \triangleq 0 \boxplus (f_2(zk) \boxminus f_2(zl)) \triangleq f_2(zk) \boxminus f_2(zl)$
2^{nd} **case:** $f_1(uz) = \top$
$f(uzk) \boxminus f(uzl) \triangleq (f_1(uz) \boxminus f_1(uz)) \boxplus (f_2(zk) \boxminus f_2(zl)) \triangleq \Omega \boxplus (f_2(zk) \boxminus f_2(zl)) \triangleq \Omega$

We conclude that $\forall z \in D_\mathbf{Z}, \forall k,l \in D_Y (k \neq l) \triangleq_{u\in D_X} f(uzk) \boxminus f(uzl)$
$\boxed{\Longleftarrow}$ Assume $\forall z \in D_\mathbf{Z}, \forall k,l \in D_Y (k < l) : \triangleq_{u\in D_X} f(uzk) \boxminus f(uzl)$.
Moreover, we have $f(X,\mathbf{Z},Y) = f_1(X,\mathbf{Z}) + f_2(\mathbf{Z},Y) + \Delta(X,\mathbf{Z},Y)$ $f_1(X,\mathbf{Z})$, $f_2(\mathbf{Z},Y)$ and $\Delta(X,\mathbf{Z},Y)$ nonnegative functions which defines the following linear systems:

$$(\mathcal{S}'_z) \begin{cases} f_1(1z) \boxplus f_2(z1) \boxplus \Delta(1z1) \triangleq f(1z1) \\ f_1(uz) \boxplus f_2(zk) \boxplus \Delta(uzk) \triangleq f(uzk) \quad \forall u,k \\ f_1(d_X z) \boxplus f_2(z d_Y) \boxplus \Delta(d_X z d_Y) \triangleq f(d_X z d_Y) \end{cases}$$

$f(uzk) \boxminus f(uzl)$
$\triangleq (f_1(uz) \boxplus f_2(zk) \boxplus \Delta(uzk)) \boxminus (f_1(uz) \boxplus f_2(zl) \boxplus \Delta(uzl))$
$\triangleq f_1(uz) \boxplus f_2(zk) \boxplus \Delta(uzk) \boxplus (-f_1(uz)) \boxplus (-f_2(zl)) \boxplus (-\Delta(uzl))$
$\triangleq (f_1(uz) \boxminus f_1(uz)) \boxplus (f_2(zk) \boxminus f_2(zl)) \boxplus (\Delta(uzk) \boxminus \Delta(uzl))$

Assume $f(X,\mathbf{Z},Y)$ does not decompose in $\{f_1(X,\mathbf{Z}),f_2(\mathbf{Z},Y)\}$ and with the property 4 we can find $\Delta(X,\mathbf{Z},Y)$ such that $\Delta(X,\mathbf{Z},Y)$ is a nonzero irreducible cost function. With the theorem 1 we have $\exists z \in D_{\mathbf{Z}}$, $k,l \in D_Y, k < l$ and $u,v \in D_X, u < v$ s.t. $\Delta(uzk) \boxminus \Delta(uzl) \not\trianglerighteq \Delta(vzk) \boxminus \Delta(vzl)$.

We deduce that $\Delta(uzk) \boxminus \Delta(uzl) \neq \Omega$ and $\Delta(vzk) \boxminus \Delta(vzl) \neq \Omega$ with the definition of $\trianglerighteq$. Also, we have $f(uzk) \boxminus f(uzl) \neq \Omega \neq f(vzk) \boxminus f(vzl)$. Thus $f(uzk) \boxminus f(uzl) \trianglerighteq f(vzk) \boxminus f(vzl) \trianglerighteq \varphi$ s.t. $\varphi \in \mathbb{Z} \cup \{\top,-\top\}$ so $f_1(uz) \boxminus f_1(uz) = 0 = f_1(vz) \boxminus f_1(vz)$ et $f_2(zk) \boxminus f_2(zl) \neq \Omega$

We use these properties below:

$\Delta(uzk) \boxminus \Delta(uzl) \neq \Delta(vzk) \boxminus \Delta(vzl)$
$f_2(zk) \boxminus f_2(zl) \boxplus \Delta(uzk) \boxminus \Delta(uzl)$
$\quad \neq f_2(zk) \boxminus f_2(zl) \boxplus \Delta(vzk) \boxminus \Delta(vzl)$
$f_1(uz) \boxminus f_1(uz) \boxplus f_2(zk) \boxminus f_2(zl) \boxplus \Delta(uzk) \boxminus \Delta(uzl)$
$\quad \neq f_1(vz) \boxminus f_1(vz) \boxplus f_2(zk) \boxminus f_2(zl) \boxplus \Delta(vzk) \boxminus \Delta(vzl)$
$f_1(uz) \boxplus f_2(zk) \boxplus \Delta(uzk) \boxminus (f_1(uz) \boxplus f_2(zl) \boxplus \Delta(uzl))$
$\quad \neq f_1(vz) \boxplus f_2(zk) \boxplus \Delta(vzk) \boxminus (f_1(vz) \boxplus f_2(zl) \boxplus \Delta(vzl))$
$f(uzk) \boxminus f(uzl) \neq f(vzk) \boxminus f(vzl)$

This is in contradiction with the first hypothesis : $\forall\, k,l \in D_Y$, $\forall\, z \in D_{\mathbf{Z}}\ f(uzk) \boxminus f(uzl) \trianglerighteq f(vzk) \boxminus f(vzl)$.

C Proof of Theorem 3

Lemma 2. *Let a cost function $f(X,\mathbf{Z},Y)$ be a pairwise decomposition w.r.t. X, Y. If $f_1(x,z) = \min_{y\in D_Y} f(x,z,y)$ then $\forall z \in D_{\mathbf{Z}}\ \exists k \in D_Y$ such that $\forall x \in D_X,\ f_1(x,z) = f(x,z,k)$.*

Proof. Let $z \in D_{\mathbf{Z}}$.

$\boxed{\exists x \in D_x, \exists y,\, f(x,z,y) \neq \top}$ Let $D_K = \{k_1,k_2,\ldots,k_{d_Y}\}$ such that $\forall i < j,\ f(x,z,k_i) \leqslant f(x,z,k_j)$. We deduce $\forall i \in [1,d_Y],\ f(x,z,k_1) \leqslant f(x,z,k_i)$ and $f_1(x,z) = \min_{y\in D_Y} f(x,z,y) = f(x,z,k_1)$.
Using Theorem 2, we find $\forall k_p \in D_K\ \forall u \in D_X$, $0 \leqslant f(x,z,k_p) \boxminus f(x,z,k_1) \trianglerighteq f(u,z,k_p) \boxminus f(u,z,k_1)$. So $f(u,z,k_1) \leqslant f(u,z,k_p)$ or $f(u,z,k_1) = f(u,z,k_p) = \top$, thus $\forall u \in D_X, f_1(x,z) = \min_{y\in D_Y} f(u,z,y) = f(u,z,k_1)$.

$\boxed{\forall x \in D_x, \forall y,\, f(x,z,y) = \top}$ $\forall x \in D_X$, $f_1(x,z) = \min_{y\in D_Y} f(x,z,y) = \top$, so let $k \in D_y$, $\forall x \in D_X$, $f_1(x,z) = f(x,z,k) = \top$ □

Now, we are able to prove Theorem 3. We have $f_1(x,z) = \min_{y\in D_Y} f(x,z,y)$ and $f_2(z,y) = \min_{x\in D_X}[f(x,z,y) - f_1(x,z)]$ so $f_2(z,y) \leqslant f(x,z,y) - f_1(x,z)$
If $f_2(z,y) = f(x,z,y) - f_1(x,z)$, $f(x,z,y) = f_1(x,z) + f_2(z,y)$.
If $f_2(z,y) < f(x,z,y) - f_1(x,z)$,

$\boxed{f(x,z,y) = f_1(x,z) = \top}$

We have $f_1(x,z) + \min_{x\in D_X}[f(x,z,y) - f_1(x,z)] = \top = f(x,z,y)$ so $f(x,z,y) = f_1(x,z) + f_2(z,y)$.

$\boxed{f(x,z,y) \neq \top \textbf{ or } f_1(x,z) \neq \top}$

We have $\min_{u\in D_X}[f(u,z,y) - f_1(u,z)] < f(x,z,y) - f_1(x,z)$ so $f(u,z,y) - f_1(u,z) < f(x,z,y) - f_1(x,z)$ with $u = \operatorname{argmin}_{u\in D_X}[f(u,z,y) - f_1(u,z)]$. Using Lemma 2, we have $l = \operatorname{argmin}_{k\in D_Y} f(x,z,k) = \operatorname{argmin}_{k\in D_Y} f(u,z,k)$, thus $f(u,z,l) - f(u,z,l) < f(x,z,y) - f(x,z,l)$, but also $f(u,z,y) \boxminus f(u,z,l) \trianglerighteq f(x,z,y) \boxminus f(x,z,l)$ because f is pairwise decomposable w.r.t. X,Y. Moreover $f(x,z,y) \neq \top$ or $f(x,z,l) \neq \top$, thus $f(u,z,y) - f(u,z,l) = f(x,z,y) - f(x,z,l)$. Finally $f_2(z,y) = f(x,z,y) - f_1(x,z)$, thus $f(x,z,y) = f_1(x,z) + f_2(z,y)$.

References

[Cooper *et al.*, 2010] M. Cooper, S. de Givry, M. Sanchez, T. Schiex, M. Zytnicki, and T. Werner. Soft arc consistency revisited. *Artificial Intelligence*, 174(7–8):449–478, 2010.

[de Givry *et al.*, 2003] S. de Givry, J. Larrosa, P. Meseguer, and T. Schiex. Solving max-sat as weighted csp. In *Proc. of CP-03*, pages 363–376, Kinsale, Ireland, 2003.

[Fishelson *et al.*, 2005] M Fishelson, N Dovgolevsky, and D Geiger. Maximum likelihood haplotyping for general pedigrees. *Human Heredity*, 59:41–60, 2005.

[Hammersley and Clifford, 1971] J Hammersley and P Clifford. Markov fields on finite graphs and lattices, 1971.

[Heckerman and Breese, 1996] D. Heckerman and J. Breese. Causal independence for probability assessment and inference using bayesian networks. *IEEE Systems, Man, and Cyber.*, 26(6):826–831, 1996.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic Graphical Models*. The MIT Press, 2009.

[Larrosa, 2000] J. Larrosa. Boosting search with variable elimination. In *CP*, pages 291–305, Singapore, 2000.

[Lauritzen, 1996] S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[Lecoutre *et al.*, 2009] C. Lecoutre, L Saïs, S. Tabary, and V. Vidal. Reasoning from last conflict(s) in constraint programming. *Artificial Intelligence*, 173:1592,1614, 2009.

[Marinescu and Dechter, 2009] R. Marinescu and R. Dechter. Memory intensive and/or search for combinatorial optimization in graphical models. *Artificial Intelligence*, 173(16-17):1492–1524, 2009.

[Meiri *et al.*, 1990] I. Meiri, R. Dechter, and J. Pearl. Tree decomposition with applications to constraint processing. In *Proc. of AAAI'90*, pages 10–16, Boston, MA, 1990.

[Meseguer *et al.*, 2006] P. Meseguer, F. Rossi, and T. Schiex. Soft constraints processing. In *Handbook of Constraint Programming*, chapter 9. Elsevier, 2006.

[Sánchez *et al.*, 2004] M. Sánchez, P. Meseguer, and J. Larrosa. Using constraints with memory to implement variable elimination. In *ECAI*, pages 216–220, Spain, 2004.

[Sánchez *et al.*, 2008] M. Sánchez, S. de Givry, and T. Schiex. Mendelian error detection in complex pedigrees using weighted constraint satisfaction techniques. *Constraints*, 13(1):130–154, 2008.

[Savicky and Vomlel, 2007] P. Savicky and J. Vomlel. Exploiting tensor rank-one decomposition in probabilistic inference. *Kybernetika*, 43(5):747–764, 2007.