SOLVING THE THREE-DIMENSIONAL PENTAMINO PUZZLE

Alain Colmerauer **Bruno Gilleta**

Laboratoire d'Informatique de Marseille,
CNRS et Universités de Provence et de la Méditerranée,
163, avenue de Luminy Case 901
13288 Marseille cedex 09, France
colmerauer/gilleta@lim.univ-mrs.fr

Abstract

Pentaminoes are pieces made of 5 connected unit cubes lead on a plane surface. Their 12 different shapes look like the 12 letters F, I, L, P, N, T, U, V, W, X, Y, Z. We are interested in all the different ways of putting these 12 pentaminoes in a box having a volume of 60 unit cubes.

First we express the different pentaminoes configurations as the solutions of a natural set of constraints on unknown integers.

Then we break down these constraints in elementary constraints of six kinds: the membership of an unknown in a known interval, the arithmetic constraints $y = ax$ and $y = x_1 + \cdots + x_n$, the constraint which forces k unknown integers to be different and a constraint on $3k$ unknowns, involving $3k$ parameters, for managing the rotations and the translations of any piece of k cubes.

For each elementary constraint on n unknown, which are forced to be within n intervals, we also establish the formulae for computing the best narrowing of these intervals.

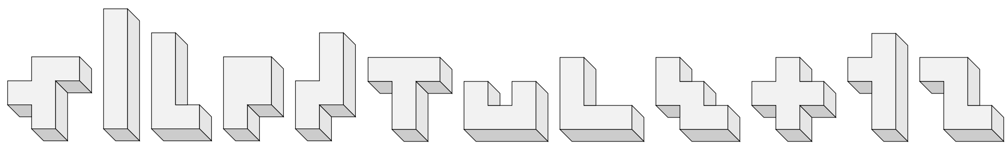
We terminate by presenting an algorithm for solving our constraints by iterations of local narrowing and by splitting intervals in disjoint intervals. Benchmarks show that a classical enumerative algorithm is still the most efficient way to solve our problem. However, our smaller search spaces show that we are working on the right track.

Keywords: Fitting of pieces in space, Three-dimensional pentamino puzzle, Integer constraints, Interval narrowing, Complexity.

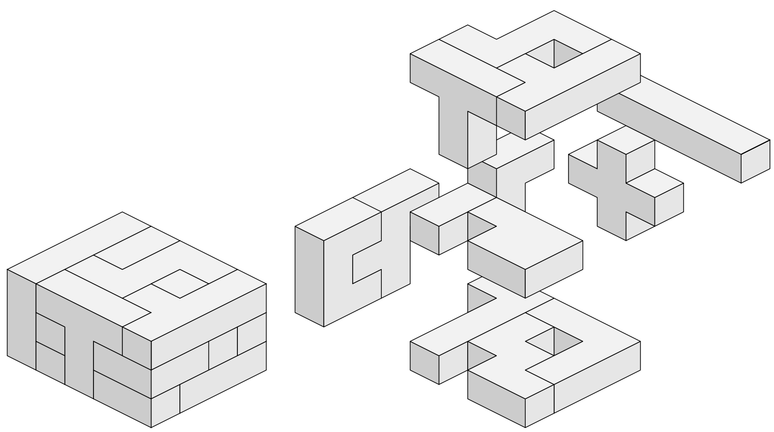
Introduction

A *polyomino* is a simple shape made of n squares 1×1 connected along their edges. The term polyomino was used by S. W. Golomb in 1953 in a presentation he made at the Harvard Mathematics Club. In [1] he presents different ways to tile a chess-board with polyominoes of two (the traditional dominoes), three, four and five squares: *the pentaminoes*. But the first known problem on pentaminoes was published in 1907 by Henry Ernest Dudeny in the Canterbury puzzles.

In [2], M. Gardner gives a thickness to the pentaminoes, thus they are now pieces made of 5 unit cubes connected along their faces on a plane surface. There are 12 different solid pentaminoes and their shapes look like the 12 letters F, I, L, P, N, T, U, V, W, X, Y, Z:



We are interested in fitting in all the possible ways these 12 solid pentaminoes in a right-angled parallelepiped having a volume of 60 unit cubes. For example, here is a placement of the pentaminoes in a block of size $3\times4\times5$:



In this document, we present a program of constraint over integers and its resolution with a classical method of interval narrowing which isolates the set of solutions in Cartesian products of integers that we reduce more and more. This program solves the three-dimensional pentamino puzzle and reveals good constraints over integers to solve problems involving the fitting of pieces in space.

1 Notations and definitions

Intervals and blocks

We denote by $\mathbf{Z}$ the set of integers. If a and b are elements of $\mathbf{Z}$, we call an *interval* the possibly empty set of elements x of $\mathbf{Z}$ such that $a \leq x$ and $x \leq b$. We denote such a set by $[a, b]$ or $a..b$. We denote by $\underline{I}$ (resp. $\overline{I}$) the least (resp. the greatest) element of the interval I, if it exists. If z is a real number, we denote by $\lceil z \rceil$ (resp. $\lfloor z \rfloor$) the least (resp. the greater) integer greater (resp. lower) than z. We call *block* any finite Cartesian product $I_1 \times \cdots \times I_n$ of intervals I_i.

If J is a finite subset of $\mathbf{Z}$ we denote by $\text{hull}(J)$ the least interval (w.r.t. inclusion) which contains J. If r is a finite subset of $\mathbf{Z}^n$, we also denote by $\text{hull}(r)$ the least block (w.r.t. inclusion) which contains r. Among the pleasant properties [3] of map $r \mapsto \text{hull}(r)$ we mention two which are useful to compute out blocks of the form $\text{hull}(r \cap I_1 \times \cdots \times I_n)$:

$$\text{hull}(r_1 \cup \cdots \cup r_n) = \text{hull}(\text{hull}(r_1) \cup \cdots \cup \text{hull}(r_n)) \tag{1}$$

and

$$\text{hull}(r) = \text{hull}(\text{proj}_1(r)) \times \cdots \times \text{hull}(\text{proj}_m(r)), \tag{2}$$

where r and the r_is are subsets of $\mathbf{Z}^n$ and $\text{proj}_i(r)$ is the ith projection of r, that is to say the set of elements b of $\mathbf{Z}$ such that there exists a n-tuple $(a_1, \ldots, a_n)$ in r with $b = a_i$.

Translations and rotations

By a *rotation matrix* M we understand a matrix of elements in $\{-1, 0, 1\}$, which is of the form

$$M = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{32} & \theta_{33} \end{bmatrix} \tag{3}$$

with one and only one non-zero element in each row and column, and with a determinant of value 1.

Let us call a *point* any element of $\mathbf{Z}^3$ and a *configuration* any finite set of points. Let

$$\begin{aligned} f &= \{(a_1, b_1, c_1), \ldots, (a_n, b_n, c_n)\}, \\ f' &= \{(a'_1, b'_1, c'_1), \ldots, (a'_n, b'_n, c'_n)\} \end{aligned}$$

be two configurations of n points. We say that f' is a *translated* configuration of f if there exist elements c, d, e of $\mathbf{Z}$ such that, for each i in $1..n$,

$$\begin{aligned} a'_i &= a_i + c, \\ b'_i &= b_i + d, \\ c'_i &= c_i + e. \end{aligned}$$

We say that f' is a *rotated* configuration of f if there exists a rotation matrix M, of the form (3), such that, for each i in $1..n$,

$$\begin{aligned} a'_i &= \theta_{11} a_i + \theta_{12} b_i + \theta_{13} c_i, \\ b'_i &= \theta_{21} a_i + \theta_{22} b_i + \theta_{23} c_i, \\ c'_i &= \theta_{31} a_i + \theta_{32} b_i + \theta_{33} c_i. \end{aligned}$$

We denote by $\mathit{translated}(f)$ the set of the translated configurations of f and by $\mathit{rotated}(f)$ the set of the rotated configurations of f. We can then introduce the set

$$\mathit{class}(f) = \bigcup_{g \in \mathit{rotated}(f)} \mathit{translated}(g) \tag{4}$$

of all configurations which are equal to f modulo a translatory and a rotating matrix.

Pentaminoes

A *pentamino* is a configuration $\{(a_1, b_1, c_1), \ldots, (a_5, b_5, c_5)\}$ of 5 points which satisfies the following conditions of planarity and connectedness:

- the a_is or the b_is or the c_is are all equal,
- there exists a sequence $i_1, \ldots, i_k$ of elements of 1..5 which involves all the elements of 1..5 and which is such that $(a_{i-1} - a_i)^2 + (b_{i-1} - b_i)^2 + (c_{i-1} - c_i)^2 = 1$, for each i in $2..k$.

We introduce the 12 following basic pentaminoes $\pi_1, \ldots, \pi_{12}$:

$$\begin{array}{lcl}
\pi_1 & = & \{(1,2,0),(2,2,0),(3,2,0),(3,1,0),(2,3,0)\}, \\
\pi_2 & = & \{(1,1,0),(2,1,0),(3,1,0),(4,1,0),(5,1,0)\}, \\
\pi_3 & = & \{(1,1,0),(1,1,0),(2,2,0),(3,3,0),(4,3,0)\}, \\
\pi_4 & = & \{(2,1,0),(1,1,0),(1,2,0),(2,3,0),(3,3,0)\}, \\
\pi_5 & = & \{(1,1,0),(2,1,0),(2,2,0),(3,3,0),(4,3,0)\}, \\
\pi_6 & = & \{(1,1,0),(1,1,0),(1,2,0),(2,3,0),(3,3,0)\}, \\
\pi_7 & = & \{(2,1,0),(1,1,0),(1,2,0),(2,3,0),(2,3,0)\}, \\
\pi_8 & = & \{(1,1,0),(1,1,0),(1,2,0),(2,3,0),(3,3,0)\}, \\
\pi_8 & = & \{(1,1,0),(1,1,0),(2,2,0),(2,3,0),(3,3,0)\}, \\
\pi_9 & = & \{(1,2,0),(2,1,0),(3,2,0),(2,3,0),(2,3,0)\}, \\
\pi_{10} & = & \{(1,1,0),(2,1,0),(3,2,0),(4,3,0),(2,3,0)\}, \\
\pi_{12} & = & \{(1,1,0),(2,1,0),(2,2,0),(2,3,0),(3,3,0)\}.
\end{array} \quad (5)$$

The set of subsets $\{class(\pi_1), \ldots, class(\pi_{12})\}$ is then a partition of the set of pentaminoes in 12 classes. According to their physical shapes, the sets $class(\pi_1)$, ..., $class(\pi_{12})$ are often denoted by F, I, L, P, N, T, U, V, W, X, Y, Z.

2 Statement of the problem

Given three positive integers a, b et c such that $abc = 60$, we want to compute partitions of the set $[0, a-1] \times [0, b-1] \times [0, c-1]$ in 12 pentaminoes $f_1, \ldots, f_n$ belonging to the $class(\pi_1), \ldots, class(\pi_{12})$, respectively. By letting, for each $i \in 1..12$,

$$f_i = \{(x_{5i-4}, y_{5i-4}, z_{5i-4}), \ldots, (x_{5i-0}, y_{5i-0}, z_{5i-0})\}$$

with $((x_{5i-4}, y_{5i-4}, z_{5i-4}), \ldots, (x_{5i-0}, y_{5i-0}, z_{5i-0}))$ increasing w.r.t. the lexical order, the computation of a partition is the same as the computation of a solution in x_1, y_1, z_1, ..., x_{60}, y_{60}, z_{60} of the constraint

$$\left(\begin{array}{ll}
 & x_1 \in [0, a-1] \wedge \ldots \wedge x_{60} \in [0, a-1] \\
\wedge & y_1 \in [0, b-1] \wedge \ldots \wedge y_{60} \in [0, b-1] \\
\wedge & z_1 \in [0, c-1] \wedge \ldots \wedge z_{60} \in [0, c-1] \\
\wedge & (x_1, y_1, z_1, \ldots, x_{60}, y_{60}, z_{60}) \in DistinctPoints \\
\wedge & (x_1, y_1, z_1, \ldots, x_5, y_5, z_5) \in Class(\pi_1) \\
\wedge & \quad \vdots \\
\wedge & (x_{56}, y_{56}, z_{56}, \ldots, x_{60}, y_{60}, z_{60}) \in Class(\pi_{12})
\end{array}\right)$$

where

- *DistinctPoints* is the set of $3n$-tuples $(a_1, b_1, c_1, \ldots, a_n, b_n, c_n)$ of integers which are such that, for each i, j taken in $1..n$, the point (a_i, b_i, c_i) is distinct from the point (a_j, b_j, c_j), if i is distinct from j,

- *Class* (f), with f being a configuration of n points, is the set of $3n$-tuples $(a_1, b_1, c_1, \ldots, a_n, b_n, c_n)$ of integers which are such that

$$\begin{array}{l} \{(a_1, b_1, c_1), \ldots, (a_n, b_n, c_n)\} \in \mathit{class}\,(f) \text{ and} \\ ((a_1, b_1, c_1), \ldots, (a_n, b_n, c_n)) \text{ is increasing w.r.t. the lexical order,} \end{array}$$

- the π_is are the basic pentaminoes defined in (5).

3 Introducing elementary constraints

Linearizing the constraint *to be distinct points*

Having no efficient way for narrowing the constraint "to be n different points (x, y, z)" and taking into account that $(x, y, z) \in [a-1] \times [b-1] \times [c-1]$, we use the mapping $(x, y, z) \mapsto x + ay + abz$ to re-express the constraint by means of the constraint "to be n distinct integers". The constraint

$$(x_1, y_1, z_1, \ldots, x_{60}, y_{60}, z_{60}) \in \mathit{DistinctPoints}$$

becomes then

$$\begin{array}{l} \exists\, m_1 \ldots \exists\, m_{60} \\ \left(\begin{array}{ll} & m_1 \in [0, abc-1] \ \wedge \ldots \wedge \ m_{60} \in [0, abc-1] \\ \wedge & (m_1, x_1, y_1, z_1) \in \mathit{WeightedSum}(1, a, ab) \\ \wedge & \quad \vdots \\ \wedge & (m_{60}, x_{60}, y_{60}, z_{60}) \in \mathit{WeightedSum}(1, a, ab) \\ \wedge & (m_1, \ldots, m_{60}) \in \mathit{DistinctIntegers} \end{array} \right) \end{array}$$

where *DistinctIntegers* is the set of n-tuples of distinct integers and $\mathit{WeightedSum}(c_1, \ldots, c_n)$ the set of $(n+1)$-tuples $(v_0, \ldots, v_n)$ such that $v_0 = c_1 v_1 + \cdots + c_n v_n$.

Narrowing the constraint "weighted sum" with intervals being at least as difficult as solving a knapsack problem, we can eliminate it by using the equivalence:

$$\begin{array}{c} (m, x, y, z) \in \mathit{WeightedSum}(1, a, ab) \\ \Updownarrow \\ \begin{array}{l} \exists\, s\ \exists\, t \\ \left(\begin{array}{ll} & s \in [0, a(b-1)] \\ \wedge & t \in [0, ab(c-1)] \\ \wedge & (s, y) \in \mathit{Product}(a) \\ \wedge & (t, z) \in \mathit{Product}(ab) \\ \wedge & (m, x, s, t) \in \mathit{Sum} \end{array} \right) \end{array} \end{array}$$

where $\mathit{Product}(c)$ is the set of 2-tuples (v_1, v_2) such that $v_1 = c v_2$ and *Sum* is the set of $(n+1)$-tuples $(v_0, \ldots, v_n)$ such that $v_0 = v_1 + \cdots + v_n$.

After all these transformations, our main constraint has the following form:

$$\left(\begin{array}{ll} & x_1 \in [0,a-1] \wedge \ldots \wedge x_{60} \in [0,a-1] \\ \wedge & y_1 \in [0,b-1] \wedge \ldots \wedge y_{60} \in [0,b-1] \\ \wedge & z_1 \in [0,c-1] \wedge \ldots \wedge z_{60} \in [0,c-1] \\ \wedge & (x_1,y_1,z_1,\ldots,x_5,y_5,z_5) \in \mathit{Class}(\pi_1) \\ \wedge & \quad \vdots \\ \wedge & (x_{56},y_{56},z_{56},\ldots,x_{60},y_{60},z_{60}) \in \mathit{Class}(\pi_{12}) \\ \wedge & \exists\, m_1 \ldots \exists\, m_{60} \\ & \left(\begin{array}{ll} & m_1 \in [0,abc-1] \wedge \ldots \wedge m_{60} \in [0,abc-1] \\ \wedge & (m_1,\ldots,m_{60}) \in \mathit{DistinctIntegers} \\ & \exists\, s_1\, \exists\, t_1 \qquad\qquad\qquad\qquad\qquad\qquad \exists\, s_{60}\, \exists\, t_{60} \\ \wedge & \left(\begin{array}{ll} & s_1 \in [0,a(b-1)] \\ \wedge & t_1 \in [0,ab(c-1)] \\ \wedge & (s_1,y_1) \in \mathit{Product}(a) \\ \wedge & (t_1,z_1) \in \mathit{Product}(ab) \\ \wedge & (m_1,x_1,s_1,t_1) \in \mathit{Sum} \end{array}\right) \wedge \ldots \wedge \left(\begin{array}{ll} & s_{60} \in [0,a(b-1)] \\ \wedge & t_{60} \in [0,ab(c-1)] \\ \wedge & (s_{60},y_{60}) \in \mathit{Product}(a) \\ \wedge & (t_{60},z_{60}) \in \mathit{Product}(ab) \\ \wedge & (m_{60},x_{60},s_{60},t_{60}) \in \mathit{Sum} \end{array}\right) \end{array}\right) \end{array}\right)$$

4 Narrowing the integer constraint *Product(a)*

Given an integer a, we remind that $\mathit{Product}(a)$ is the set of ordered integer pairs such that

$$(x,y) \in \mathit{Product}(a) \iff x = ay$$

Proposition: If X, Y, X', Y' are intervals such that $X' \times Y' = \mathrm{hull}(X \times Y \cap \mathit{Product}(a))$, with X, Y non-empty then

$$\begin{pmatrix} X' \\ Y' \end{pmatrix} = \begin{cases} \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix}, \text{ if } \overline{X} < \min(a\underline{Y}, a\overline{Y}) \text{ or } \max(a\underline{Y}, a\overline{Y}) < \underline{X} \\[2ex] \begin{pmatrix} \{0\} \\ Y \end{pmatrix}, \text{ if } \overline{X} \geq \min(a\underline{Y}, a\overline{Y}) \text{ and } \max(a\underline{Y}, a\overline{Y}) \geq \underline{X} \text{ and } a = 0 \\[2ex] \begin{pmatrix} [\max(a\lceil \underline{X}/a \rceil, \min(a\underline{Y}, a\overline{Y})),\ \min(a\lfloor \overline{X}/a \rfloor, \max(a\overline{Y}, a\underline{Y}))] \\ [\max(\min(\lceil \underline{X}/a \rceil, \lceil \overline{X}/a \rceil), \underline{Y}),\ \min(\max(\lfloor \overline{X}/a \rfloor, \lfloor \underline{X}/a \rfloor), \overline{Y})] \end{pmatrix}, \text{ otherwise.} \end{cases}$$

Proof Let us first note that
$\mathit{Product}(a) \cap X \times Y =$
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay\} \cap \{(x,y) \in \mathbf{Z}^2 \mid \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y}\} =$
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y}\}$.
• If $\overline{X} < \min(a\underline{Y}, a\overline{Y})$ or $\max(a\underline{Y}, a\overline{Y}) < \underline{X}$ then
$\mathit{Product}(a) \cap X \times Y =$
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y} \wedge (\overline{X} < \min(a\underline{Y}, a\overline{Y}) \vee \max(a\underline{Y}, a\overline{Y}) < \underline{X})\} =$
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y} \wedge (x \leq \overline{X} < \min(a\underline{Y}, a\overline{Y}) \leq ay \vee ay \leq \max(a\underline{Y}, a\overline{Y}) < \underline{X} \leq x)\} =$
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y} \wedge (y < ay \vee ay < x)\} =$
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge x \neq ay \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y} \wedge\} = \emptyset$.
• If $\overline{X} \geq \min(a\underline{Y}, a\overline{Y})$ and $\max(a\underline{Y}, a\overline{Y}) \geq \underline{X}$ and $a = 0$ then
$\mathit{Product}(a) \cap X \times Y =$

$\{(x,y) \in \mathbf{Z}^2 \mid x = 0 \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y}\}$ =
$\{(0,y) \in \mathbf{Z}^2 \mid \underline{Y} \leq y \leq \overline{Y}\}$ =
$\{0\} \times Y$.

• Let us suppose that $\overline{X} \geq \min(a\underline{Y}, a\overline{Y})$ and $\max(a\underline{Y}, a\overline{Y}) \geq \underline{X}$ and $a \neq 0$. On the one hand we have
$Product(a) \cap X \times Y$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y}\}$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge a\underline{X}/a \leq x \leq a\overline{X}/a \wedge (a\underline{Y} \leq x \leq a\overline{Y} \vee a\overline{Y} \leq x \leq a\underline{Y})\}$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge a\lceil \underline{X}/a \rceil \leq x \leq a\lfloor \overline{X}/a \rfloor \wedge \min(a\underline{Y}, a\overline{Y}) \leq x \leq \max(a\underline{Y}, a\overline{Y})\}$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \max(a\lceil \underline{X}/a \rceil, \min(a\underline{Y}, a\overline{Y})) \leq x \leq \min(a\lfloor \overline{X}/a \rfloor, \max(a\underline{Y}, a\overline{Y}))\}$.
On the other hand we have
$Product(a) \cap X \times Y$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \underline{X} \leq ay \leq \overline{X} \wedge \underline{Y} \leq y \leq \overline{Y}\}$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge (\underline{X}/a \leq ay \leq \overline{X}/a \vee \overline{X}/a \leq ay \leq \underline{X}/a) \wedge \underline{Y} \leq y \leq \overline{Y}\}$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge (\lceil \underline{X}/a \rceil \leq ay \leq \lfloor \overline{X}/a \rfloor \vee \lceil \overline{X}/a \rceil \leq ay \leq \lfloor \underline{X}/a \rfloor) \wedge \underline{Y} \leq y \leq \overline{Y}\}$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \min(\lceil \underline{X}/a \rceil, \lceil \overline{X}/a \rceil) \leq y \leq \max(\lfloor \overline{X}/a \rfloor, \lfloor \underline{X}/a \rfloor) \wedge \underline{Y} \leq y \leq \overline{Y}\}$ =
$\{(x,y) \in \mathbf{Z}^2 \mid x = ay \wedge \max(\min(\lceil \underline{X}/a \rceil, \lceil \overline{X}/a \rceil), \underline{Y}) \leq y \leq \min(\max(\lfloor \overline{X}/a \rfloor, \lfloor \underline{X}/a \rfloor), \overline{Y})\}$.
It follows that

$$\begin{array}{l} Product(a) \cap X \times Y \ = \\ \{x \in \mathbf{Z} \mid \ \max(a\lceil \underline{X}/a \rceil, \min(a\underline{Y}, a\overline{Y})) \leq x \leq \min(a\lfloor \overline{X}/a \rfloor, \max(a\underline{Y}, a\overline{Y}))\} \times \\ \{y \in \mathbf{Z} \mid \ \max(\min(\lceil \underline{X}/a \rceil, \lceil \overline{X}/a \rceil), \underline{Y}) \leq y \leq \min(\max(\lfloor \overline{X}/a \rfloor, \lfloor \underline{X}/a \rfloor), \overline{Y})\}. \end{array}$$

Thus

$$\begin{array}{l} \text{hull}(Product(a) \cap X \times Y) \ = \\ [\max(a\lceil \underline{X}/a \rceil, \min(a\underline{Y}, a\overline{Y})),\ \min(a\lfloor \overline{X}/a \rfloor, \max(a\underline{Y}, a\overline{Y}))] \times \\ [\max(\min(\lceil \underline{X}/a \rceil, \lceil \overline{X}/a \rceil), \underline{Y}),\ \min(\max(\lfloor \overline{X}/a \rfloor, \lfloor \underline{X}/a \rfloor), \overline{Y})]. \end{array}$$

5 Narrowing the integer constraint *Sum*

We remind that *Sum* is the set of tuples of integers such that

$$(x, y_1, \ldots, y_n) \in Sum \ \Leftrightarrow \ x = y_1 + \cdots + y_n$$

Proposition: If $X, Y_1, \ldots, Y_n, X', Y_1', \ldots, Y_n'$ are intervals such that $X' \times Y_1' \times \cdots \times Y_n' = \text{hull}(X \times Y_1 \times \cdots \times Y_n \cap Sum)$ with $X, Y_1, \ldots, Y_n$ non-empty then

$$\begin{pmatrix} X' \\ Y_i' \end{pmatrix} = \begin{cases} \begin{pmatrix} \emptyset \\ \emptyset \end{pmatrix}, \text{ if } \overline{X} < \underline{Y_1} + \cdots + \underline{Y_n} \text{ or } \overline{Y_1} + \cdots + \overline{Y_n} < \underline{X} \\ \begin{pmatrix} [\max(\underline{X}, \underline{Y_1} + \cdots + \underline{Y_n}),\ \min(\overline{X}, \overline{Y_1} + \cdots + \overline{Y_n})] \\ [\max(\underline{Y_i}, \underline{X} - \sum_{j=1, j\neq i}^{n} \overline{X_j}),\ \min(\overline{Y_i}, \overline{X} - \sum_{j=1, j\neq i}^{n} \underline{Y_j})] \end{pmatrix}, \text{ otherwise.} \end{cases}$$

Proof Let us first note that
$Sum \cap X \times Y_1 \times \cdots \times Y_n$ =
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n\}$
$\cap\ \{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid \underline{X} \leq x \leq \overline{X} \wedge \underline{Y_1} \leq y_1 \leq \overline{Y_1} \wedge \ldots \wedge \underline{Y_n} \leq y_n \leq \overline{Y_n}\}$ =
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \underline{X} \leq x \leq \overline{X} \wedge \underline{Y_1} \leq y_1 \leq \overline{Y_1} \wedge \ldots \wedge \underline{Y_n} \leq y_n \leq \overline{Y_n}\}$.

• If $\overline{X} < \underline{Y_1} + \cdots + \underline{Y_n}$ or $\overline{Y_1} + \cdots + \overline{Y_n} < \underline{X}$ then

$Sum \cap X \times Y_1 \times \cdots \times Y_n \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \underline{X} \le x \le \overline{X} \wedge \underline{Y_1} \le y_1 \le \overline{Y_1} \wedge \ldots \wedge \underline{Y_n} \le y_n \le \overline{Y_n}$
$\wedge\, (\overline{X} < \underline{Y_1} + \cdots + \underline{Y_n} \vee \overline{Y_1} + \cdots + \overline{Y_n} < \underline{X})\} \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \underline{X} \le x \le \overline{X} \wedge \underline{Y_1} \le y_1 \le \overline{Y_1} \wedge \ldots \wedge \underline{Y_n} \le y_n \le \overline{Y_n}$
$\wedge\, (x \le \overline{X} < \underline{Y_1} + \cdots + \underline{Y_n} \le y_1 + \cdots + y_n \vee y_1 + \cdots + y_n \le \overline{Y_1} + \cdots + \overline{Y_n} < \underline{X} \le x)\} \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge x \ne y_1 + \cdots + y_n \wedge \underline{X} \le x \le \overline{X}$
$\wedge\, \underline{Y_1} \le y_1 \le \overline{Y_1} \wedge \ldots \wedge \underline{Y_n} \le y_n \le \overline{Y_n}\} \;=\; \emptyset$

• Let us suppose that $\overline{X} \ge \underline{Y_1} + \cdots + \underline{Y_n}$ and $\overline{Y_1} + \cdots + \overline{Y_n} \ge \underline{X}$ hold; on the one hand we have

$Sum \cap X \times Y_1 \times \cdots \times Y_n \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \underline{X} \le x \le \overline{X} \wedge \underline{Y_1} + \cdots + \underline{Y_n} \le y_1 + \cdots + y_n \le \overline{Y_1} + \cdots + \overline{Y_n}\} \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \underline{X} \le x \le \overline{X} \wedge \underline{Y_1} + \cdots + \underline{Y_n} \le x \le \overline{Y_1} + \cdots + \overline{Y_n}\} \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \max(\underline{X}, \underline{Y_1} + \cdots + \underline{Y_n}) \le x \le \min(\overline{X}, \overline{Y_1} + \cdots + \overline{Y_n})\}$

On the other hand for all $i \in 1..n$ we have

$Sum \cap X \times Y_1 \times \cdots \times Y_n \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n$
$\wedge\, \underline{Y_i} \le y_i \le \overline{Y_i} \wedge \underline{X} - \underline{Y_1} - \cdots - \underline{Y_{i-1}} - \underline{Y_{i+1}} - \cdots - \underline{Y_n}$
$\le x - y_1 - \cdots - y_{i-1} - y_{i+1} - \cdots - y_n \le \overline{X} - \overline{Y_1} - \cdots - \overline{Y_{i-1}} - \overline{Y_{i+1}} - \cdots - \overline{Y_n}\} \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \underline{Y_i} \le y_i \le \overline{Y_i} \wedge \underline{X} - \underline{Y_1} - \cdots - \underline{Y_{i-1}} - \underline{Y_{i+1}} - \cdots - \underline{Y_n}$
$\le y_i \le \overline{X} - \overline{Y_1} - \cdots - \overline{Y_{i-1}} - \overline{Y_{i+1}} - \cdots - \overline{Y_n}\} \;=$
$\{(x, y_1, \ldots, y_n) \in \mathbf{Z}^{n+1} \mid x = y_1 + \cdots + y_n \wedge \max(\underline{Y_i}, \underline{X} - \underline{Y_1} - \cdots - \underline{Y_{i-1}} - \underline{Y_{i+1}} - \cdots - \underline{Y_n})$
$\le y_i \le \min(\overline{Y_i}, \overline{X} - \overline{Y_1} - \cdots - \overline{Y_{i-1}} - \overline{Y_{i+1}} - \cdots - \overline{Y_n})\}$

It follows that

$Sum \cap X \times Y_1 \times \cdots \times Y_n \;=$
$\{x \in \mathbf{Z} \mid \max(\underline{X}, \underline{Y_1} + \cdots + \underline{Y_n}) \le x \le \min(\overline{X}, \overline{Y_1} + \cdots + \overline{Y_n})\} \times$
$\{y_1 \in \mathbf{Z} \mid \max(\underline{Y_1}, \underline{X} - \underline{Y_2} - \cdots - \underline{Y_n}) \le y_1 \le \min(\overline{Y_1}, \overline{X} - \overline{Y_2} - \cdots - \overline{Y_n})\} \times \cdots \times$
$\{y_i \in \mathbf{Z} \mid \max(\underline{Y_i}, \underline{X} - \underline{Y_1} - \cdots - \underline{Y_{i-1}} - \underline{Y_{i+1}} - \cdots - \underline{Y_n}) \le y_i \le \min(\overline{Y_i}, \overline{X} - \overline{Y_1} - \cdots - \overline{Y_{i-1}} - \overline{Y_{i+1}} - \cdots - \overline{Y_n})\} \times \cdots \times$
$\{y_n \in \mathbf{Z} \mid \max(\underline{Y_n}, \underline{X} - \underline{Y_1} - \cdots - \underline{Y_{n-1}}) \le y_n \le \min(\overline{Y_n}, \overline{X} - \overline{Y_1} - \cdots - \overline{Y_{n-1}})\}.$

Thus

$\mathrm{hull}(Sum \cap X \times Y_1 \times \cdots \times Y_n) \;=$
$[\max(\underline{X}, \underline{Y_1} + \cdots + \underline{Y_n}), \min(\overline{X}, \overline{Y_1} + \cdots + \overline{Y_n})] \times$
$[\max(\underline{Y_1}, \underline{X} - \underline{Y_2} - \cdots - \underline{Y_n}), \min(\overline{Y_1}, \overline{X} - \overline{Y_2} - \cdots - \overline{Y_n})] \times \cdots \times$
$[\max(\underline{Y_i}, \underline{X} - \underline{Y_1} - \cdots - \underline{Y_{i-1}} - \underline{Y_{i+1}} - \cdots - \underline{Y_n}), \min(\overline{Y_i}, \overline{X} - \overline{Y_1} - \cdots - \overline{Y_{i-1}} - \overline{Y_{i+1}} - \cdots - \overline{Y_n})] \times \cdots \times$
$[\max(\underline{Y_n}, \underline{X} - \underline{Y_1} - \cdots - \underline{Y_{n-1}}), \min(\overline{Y_n}, \overline{X} - \overline{Y_1} - \cdots - \overline{Y_{n-1}})].$

6 Narrowing the *DistinctIntegers* constraint

We remind that *DistinctIntegers* is the set of n-tuples of integers such that

$$(x_1, \ldots, x_n) \in \mathit{DistinctIntegers} \;\Leftrightarrow\; x_1 \ne x_2 \wedge x_1 \ne x_3 \wedge \ldots \wedge x_{n-2} \ne x_n \wedge x_{n-1} \ne x_n$$

For computing the block $X'_1 \times \ldots \times X'_n = \mathrm{hull}(\mathit{DistinctIntegers} \cap X_1 \times \cdots \times X_n)$ where $X_1, \ldots, X_n$ are intervals we actually use the algorithm of complexity $\mathcal{O}(n^2)$ of M. Leconte [4] and J. Zhou [5]. We could also use the more recent algorithm of N. Bleuzen Guernalec and A. Colmerauer [6] which has a complexity of $\mathcal{O}(n \log n)$.

7 Narrowing the *Translated*(f) constraint

For narrowing the constraint *Class*(f) we introduce first the constraint *Translated*(f) and study the way of narrowing it. If f is a configuration of n points then *Translated*(f) is the set of $3n$-tuples of integers such that

$$\begin{gathered}(x_1,y_1,z_1,\dots,x_n,y_n,z_n) \in \mathit{Translated}(f)\\ \Updownarrow \\ \{(x_1,y_1,z_1),\dots,(x_n,y_n,z_n)\} \in \mathit{translated}(f) \text{ and}\\ ((x_1,y_1,z_1),\dots,(x_n,y_n,z_n)) \text{ is in increasing lexical order}\end{gathered}$$

Proposition: Let $X_1, Y_1, Z_1, \dots, X_n, Y_n, Z_n, X_1', Y_1', Z_1', \dots, X_n', Y_n', Z_n'$ be intervals such that $X_1' \times Y_1' \times Z_1' \times \dots \times X_n' \times Y_n' \times Z_n' = \text{hull}(\mathit{Translated}(X_1 \times Y_1 \times Z_1 \times \dots \times X_n \times Y_n \times Z_n)$. If $((a_1,b_1,c_1),\dots,(a_n,b_n,c_n))$ is a n-tuple of points of f in increasing lexical order then, for all $i \in 1..n$

$$\begin{pmatrix}X_i'\\Y_i'\\Z_i'\end{pmatrix} = \begin{cases}\begin{pmatrix}\emptyset\\\emptyset\\\emptyset\end{pmatrix}, \text{ if } \begin{pmatrix}\max_{j=1..n}(\underline{X_j}-a_j) > \min_{j=1..n}(\overline{X_j}-a_j)\\ \text{or } \max_{j=1..n}(\underline{Y_j}-b_j) > \min_{j=1..n}(\overline{Y_j}-b_j)\\ \text{or } \max_{j=1..n}(\underline{Z_j}-c_j) > \min_{j=1..n}(\overline{Z_j}-c_j)\end{pmatrix}\\ \begin{pmatrix}[a_i+\max_{j=1..n}(\underline{X_j}-a_j),\ a_i+\min_{j=1..n}(\overline{X_j}-a_j)]\\ [b_i+\max_{j=1..n}(\underline{Y_j}-b_j),\ b_i+\min_{j=1..n}(\overline{Y_j}-b_j)]\\ [c_i+\max_{j=1..n}(\underline{Z_j}-c_j),\ c_i+\min_{j=1..n}(\overline{Z_j}-c_j)]\end{pmatrix}, \text{ otherwise.}\end{cases}$$

Proof Let us first note that
$\mathit{Translated}(a_1,b_1,c_1,\dots,a_n,b_n,c_n) \cap X_1 \times Y_1 \times Z_1 \times \dots \times X_n \times Y_n \times Z_n =$
$\{(x_1,y_1,z_1,\dots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \dots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w\}$
$\cap\ \{(x_1,y_1,z_1,\dots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid$
$\underline{X_1} \le x_1 \le \overline{X_1} \wedge \underline{Y_1} \le y_1 \le \overline{Y_1} \wedge \underline{Z_1} \le z_1 \le \overline{Z_1} \wedge \dots \wedge \underline{X_n} \le x_n \le \overline{X_n} \wedge \underline{Y_n} \le x_n \le \overline{Y_n} \wedge \underline{Z_n} \le x_n \le \overline{Z_n}\} =$
$\{(x_1,y_1,z_1,\dots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \dots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w$
$\wedge\ \underline{X_1} \le x_1 \le \overline{X_1} \wedge \underline{Y_1} \le y_1 \le \overline{Y_1} \wedge \underline{Z_1} \le z_1 \le \overline{Z_1} \wedge \dots \wedge \underline{X_n} \le x_n \le \overline{X_n} \wedge \underline{Y_n} \le x_n \le \overline{Y_n} \wedge \underline{Z_n} \le x_n \le \overline{Z_n}\}$

- If $\max_{i=1..n}(\underline{X_i} - a_i) > \min_{i=1..n}(\overline{X_i} - a_i)$ or $\max_{i=1..n}(\underline{Y_i} - b_i) > \min_{i=1..n}(\overline{Y_i} - b_i)$ or $\max_{i=1..n}(\underline{Z_i} - c_i) > \min_{i=1..n}(\overline{Z_i} - c_i)$ then

$\mathit{Translated}(a_1,b_1,c_1,\dots,a_n,b_n,c_n) \cap X_1 \times Y_1 \times Z_1 \times \dots \times X_n \times Y_n \times Z_n =$
$\{(x_1,y_1,z_1,\dots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \dots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w$
$\wedge\ \underline{X_1} \le x_1 \le \overline{X_1} \wedge \underline{Y_1} \le y_1 \le \overline{Y_1} \wedge \underline{Z_1} \le z_1 \le \overline{Z_1} \wedge \dots \wedge \underline{X_n} \le x_n \le \overline{X_n} \wedge \underline{Y_n} \le x_n \le \overline{Y_n} \wedge \underline{Z_n} \le x_n \le \overline{Z_n}$
$\wedge\ (\max_{i=1..n}(\underline{X_i}-a_i) > \min_{i=1..n}(\overline{X_i}-a_i) \vee \max_{i=1..n}(\underline{Y_i}-b_i) > \min_{i=1..n}(\overline{Y_i}-b_i)$
$\vee \max_{i=1..n}(\underline{Z_i}-c_i) > \min_{i=1..n}(\overline{Z_i}-c_i))\} =$
$\{(x_1,y_1,z_1,\dots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$\underline{X_1} \le x_1 \le \overline{X_1} \wedge \underline{Y_1} \le y_1 \le \overline{Y_1} \wedge \underline{Z_1} \le z_1 \le \overline{Z_1} \wedge \dots \wedge \underline{X_n} \le x_n \le \overline{X_n} \wedge \underline{Y_n} \le x_n \le \overline{Y_n} \wedge \underline{Z_n} \le x_n \le \overline{Z_n}$
$\wedge\ (\max_{i=1..n}(\underline{X_i}-x_i+u) > \min_{i=1..n}(\overline{X_i}-x_i+u) \vee \max_{i=1..n}(\underline{Y_i}-y_i+v) > \min_{i=1..n}(\overline{Y_i}-y_i+v)$
$\vee \max_{i=1..n}(\underline{Z_i}-z_i+w) > \min_{i=1..n}(\overline{Z_i}-z_i+w))\} =$
$\{(x_1,y_1,z_1,\dots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \dots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w$
$\wedge\ (\max_{i=1..n}(\underline{X_i}-x_i) > \min_{i=1..n}(\overline{X_i}-x_i) \vee \max_{i=1..n}(\underline{Y_i}-y_i) > \min_{i=1..n}(\overline{Y_i}-y_i)$
$\vee \max_{i=1..n}(\underline{Z_i}-z_i) > \min_{i=1..n}(\overline{Z_i}-z_i))\} =$

$\{(x_1,y_1,z_1,\ldots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$(0 \geq \max_{i=1..n}(\underline{X_i}-x_i) > \min_{i=1..n}(\overline{X_i}-x_i) \geq 0 \vee 0 \geq \max_{i=1..n}(\underline{Y_i}-y_i) > \min_{i=1..n}(\overline{Y_i}-y_i) \geq 0$
$\vee\, 0 \geq \max_{i=1..n}(\underline{Z_i}-z_i) > \min_{i=1..n}(\overline{Z_i}-z_i) \geq 0)\} \;=$
$\{(x_1,y_1,z_1,\ldots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid 0 > 0 \vee 0 > 0 \vee 0 > 0\} \;=\; \emptyset.$

• If $\max_{i=1..n}(\underline{X_i} - a_i) \leq \min_{i=1..n}(\overline{X_i} - a_i)$ and $\max_{i=1..n}(\underline{Y_i} - b_i) \leq \min_{i=1..n}(\overline{Y_i} - b_i)$ and $\max_{i=1..n}(\underline{Z_i} - c_i) \leq \min_{i=1..n}(\overline{Z_i} - c_i)$. Then
$\mathit{Translated}(a_1,b_1,c_1,\ldots,a_n,b_n,c_n) \;\cap\; X_1\times Y_1\times Z_1\times\cdots\times X_n\times Y_n\times Z_n \;=$
$\{(x_1,y_1,z_1,\ldots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \ldots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w$
$\wedge\, \underline{X_1} \leq x_1 \leq \overline{X_1} \wedge \underline{Y_1} \leq y_1 \leq \overline{Y_1} \wedge \underline{Z_1} \leq z_1 \leq \overline{Z_1} \wedge \ldots \wedge \underline{X_n} \leq x_n \leq \overline{X_n} \wedge \underline{Y_n} \leq x_n \leq \overline{Y_n} \wedge \underline{Z_n} \leq x_n \leq \overline{Z_n}\} \;=$
$\{(x_1,y_1,z_1,\ldots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \ldots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w$
$\wedge\, \underline{X_1} \leq a_1+u \leq \overline{X_1} \wedge \underline{Y_1} \leq b_1+v \leq \overline{Y_1} \wedge \underline{Z_1} \leq c_1+w \leq \overline{Z_1} \wedge \ldots \wedge \underline{X_n} \leq a_n+u \leq \overline{X_n} \wedge \underline{Y_n} \leq b_n+v \leq$
$\overline{Y_n} \wedge \underline{Z_n} \leq c_n+w \leq \overline{Z_n}\} \;=$
$\{(x_1,y_1,z_1,\ldots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \ldots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w$
$\wedge\, \underline{X_1}-a_1 \leq u \leq \overline{X_1}-a_1 \wedge \underline{Y_1}-b_1 \leq v \leq \overline{Y_1}-b_1 \wedge \underline{Z_1}-c-1 \leq w \leq \overline{Z_1}-c_1$
$\wedge \ldots \wedge \underline{X_n}-a_n \leq u \leq \overline{X_n}-a_n \wedge \underline{Y_n}-b_n \leq v \leq \overline{Y_n}-b_n \wedge \underline{Z_n}-c_n \leq w \leq \overline{Z_n}-c_n\} \;=$
$\{(x_1,y_1,z_1,\ldots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \mid \exists(u,v,w) \in \mathbf{Z}^3$
$x_1 = a_1+u \wedge y_1 = b_1+v \wedge z_1 = c_1+w \wedge \ldots \wedge x_n = a_n+u \wedge y_n = b_n+v \wedge z_n = c_n+w$
$\wedge \max_{i=1..n}(\underline{X_i}-a_i) \leq u \leq \min_{i=1..n}(\overline{X_i}-a_i) \wedge \max_{i=1..n}(\underline{Y_i}-b_i) \leq v \leq \min_{i=1..n}(\overline{Y_i}-b_i)$
$\wedge \max_{i=1..n}(\underline{Z_i}-c_i) \leq w \leq \min_{i=1..n}(\overline{Z_i}-c_i) \wedge \ldots \wedge \max_{i=1..n}(\underline{X_i}-a_i) \leq u \leq \min_{i=1..n}(\overline{X_i}-a_i)$
$\wedge \max_{i=1..n}(\underline{Y_i}-b_i) \leq v \leq \min_{i=1..n}(\overline{Y_i}-b_i) \wedge \max_{i=1..n}(\underline{Z_i}-c_i) \leq w \leq \min_{i=1..n}(\overline{Z_i}-c_i)\} \;=$
$\{(x_1,y_1,z_1,\ldots,x_n,y_n,z_n) \in \mathbf{Z}^{3n} \;\; \exists(u,v,w) \in \mathbf{Z}^3$
$a_1+\max_{i=1..n}(\underline{X_i}-a_i) \leq x_1 \leq a_1+\min_{i=1..n}(\overline{X_i}-a_i)$
$\wedge\, b_1+\max_{i=1..n}(\underline{Y_i}-b_i) \leq y_1 \leq b_1+\min_{i=1..n}(\overline{Y_i}-b_i)$
$\wedge\, c_1+\max_{i=1..n}(\underline{Z_i}-c_i) \leq z_1 \leq c_1+\min_{i=1..n}(\overline{Z_i}-c_i)$
$\wedge \ldots \wedge a_n+\max_{i=1..n}(\underline{X_i}-a_i) \leq x_n \leq a_n+\min_{i=1..n}(\overline{X_i}-a_i)$
$\wedge\, b_n+\max_{i=1..n}(\underline{Y_i}-b_i) \leq y_n \leq b_n+\min_{i=1..n}(\overline{Y_i}-b_i)$
$\wedge\, c_n+\max_{i=1..n}(\underline{Z_i}-c_i) \leq z_n \leq c_n+\min_{i=1..n}(\overline{Z_i}-c_i)\}.$
It follows that

$$\begin{aligned}
&\mathit{Translated}(a_1,b_1,c_1,\ldots,a_n,b_n,c_n) \;\cap\; X_1\times Y_1\times Z_1\times\cdots\times X_n\times Y_n\times Z_n \;=\\
&\{x \in \mathbf{Z} \mid a_1+\max_{i=1..n}(\underline{X_i}-a_i) \leq x \leq a_1+\min_{i=1..n}(\overline{X_i}-a_i)\}\times\\
&\{y \in \mathbf{Z} \mid b_1+\max_{i=1..n}(\underline{Y_i}-b_i) \leq y \leq b_1+\min_{i=1..n}(\overline{Y_i}-b_i)\}\times\\
&\{z \in \mathbf{Z} \mid c_1+\max_{i=1..n}(\underline{Z_i}-c_i) \leq z \leq c_1+\min_{i=1..n}(\overline{Z_i}-c_i)\} \times\cdots\times\\
&\{x \in \mathbf{Z} \mid a_n+\max_{i=1..n}(\underline{X_i}-a_i) \leq x \leq a_n+\min_{i=1..n}(\overline{X_i}-a_i)\}\times\\
&\{y \in \mathbf{Z} \mid b_n+\max_{i=1..n}(\underline{Y_i}-b_i) \leq y \leq b_n+\min_{i=1..n}(\overline{Y_i}-b_i)\}\times\\
&\{z \in \mathbf{Z} \mid c_n+\max_{i=1..n}(\underline{Z_i}-c_i) \leq z \leq c_n+\min_{i=1..n}(\overline{Z_i}-c_i)\}.
\end{aligned}$$

Thus

$$\begin{aligned}
&\mathrm{hull}(\mathit{Translated}(a_1,b_1,c_1,\ldots,a_n,b_n,c_n) \;\cap\; X_1\times Y_1\times Z_1\times\cdots\times X_n\times Y_n\times Z_n) \;=\\
&[a_1+\max_{i=1..n}(\underline{X_i}-a_i), a_1+\min_{i=1..n}(\overline{X_i}-a_i)]\times\\
&[b_1+\max_{i=1..n}(\underline{Y_i}-b_i), b_1+\min_{i=1..n}(\overline{Y_i}-b_i)]\times\\
&[c_1+\max_{i=1..n}(\underline{Z_i}-c_i), c_1+\min_{i=1..n}(\overline{Z_i}-c_i)] \times\cdots\times\\
&[a_n+\max_{i=1..n}(\underline{X_i}-a_i), a_n+\min_{i=1..n}(\overline{X_i}-a_i)]\times\\
&[b_n+\max_{i=1..n}(\underline{Y_i}-b_i), b_n+\min_{i=1..n}(\overline{Y_i}-b_i)]\times\\
&[c_n+\max_{i=1..n}(\underline{Z_i}-c_i), c_n+\min_{i=1..n}(\overline{Z_i}-c_i)].
\end{aligned}$$

8 Narrowing the *Class* (f) constraint

We can now treat the $\mathit{Class}(f)$ constraint. Let f be a configuration of n points, we remind that $\mathit{Class}(f)$ is the set of the $3n$-tuples of integers such that

$$\begin{gathered}(x_1, y_1, z_1, \ldots, x_n, y_n, z_n) \in \mathit{Class}(f) \\ \Updownarrow \\ \{(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)\} \in \mathit{class}(f) \text{ and} \\ ((x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)) \text{ is in ascending lexical order}\end{gathered}$$

Proposition: Let $X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n, X_1', Y_1', Z_1', \ldots, X_n', Y_n', Z_n'$ be intervals such that $X_1, Y_1, Z_1, \ldots, X_n, Y_n, Z_n$, are non-empty and $X_1' \times Y_1' \times Z_1' \times \ldots \times X_n' \times Y_n' \times Z_n' = \mathit{Class}(f) \cap X_1 \times Y_1 \times Z_1 \times \ldots \times X_n \times Y_n \times Z_n$,
let $\{g_1, \ldots, g_{24}\} = \mathit{rotated}(f)$,
let $r = \mathit{Class}(f) \cap X_1 \times Y_1 \times Z_1 \times \ldots \times X_n \times Y_n \times Z_n$,
let $r_j = \mathit{Translated}(g_j) \cap X_1 \times Y_1 \times Z_1 \times \ldots \times X_n \times Y_n \times Z_n$,
let $X_{1j} \times Y_{1j} \times Z_{1j} \times \cdots \times X_{nj} \times Y_{nj} \times Z_{nj} = \mathrm{hull}(r_j)$,
let J be the set of $j \in \{1, \ldots, 24\}$ such that $\mathrm{hull}(r_j) \neq \emptyset$.

Then for all $i \in 1..n$:

$$\begin{pmatrix} X_i' \\ Y_i' \\ Z_i' \end{pmatrix} = \begin{cases} \begin{pmatrix} \emptyset \\ \emptyset \\ \emptyset \end{pmatrix}, & \text{if } J = \emptyset \\ \begin{pmatrix} [\min_{j \in J} \underline{X_{ij}},\ \max_{j \in J} \overline{X_{ij}}] \\ [\min_{j \in J} \underline{Y_{ij}},\ \max_{j \in J} \overline{Y_{ij}}] \\ [\min_{j \in J} \underline{Z_{ij}},\ \max_{j \in J} \overline{Z_{ij}}] \end{pmatrix}, & \text{otherwise,} \end{cases}$$

Proof Follows from the definition of $\mathit{Translated}(f)$, from equality (1), and from equality (4).

9 The general solver

A formula f is *normalized* if it is written in the following form:

$$\exists\, x_{m+1} \ldots \exists\, x_n \begin{pmatrix} & x_1 \in I_1 \wedge \ldots \wedge x_n \in I_n \\ \wedge & p_1 \wedge \ldots \wedge p_k \\ \wedge & q_1 \wedge \ldots \wedge q_l \\ \wedge & \mathit{true} \end{pmatrix}$$

where the x_is are variables, the I_is are intervals of integers, the p_is are constraints of the form $(y_1, \ldots, y_k) \in r$, where r is a set of m-tuples of integers and the y_is are variables, and the q_is are normalized formulae.

Given a normalized formula f, we apply to the formula $f \vee \mathit{false}$ a succession of transformations producing at each step an equivalent formula $f_1 \vee \ldots \vee f_n \vee \mathit{false}$ where the f_is are normalized formulae. After a finite number of transformations, we obtain a formula equivalent to f of the form:

$$\begin{cases} \textit{false}, \text{ if } f \text{ is equivalent to } \textit{false} \\ \textit{true} \vee \ldots \vee \textit{true} \vee \textit{false}, \text{ if } f \text{ is equivalent to } \textit{true} \\ \begin{pmatrix} & x_1 \in \{a_{11}\} \\ \wedge & \vdots \\ \wedge & x_n \in \{a_{1n}\} \\ \wedge & \textit{true} \end{pmatrix} \vee \ldots \vee \begin{pmatrix} & x_1 \in \{a_{p1}\} \\ \wedge & \vdots \\ \wedge & x_n \in \{a_{pn}\} \\ \wedge & \textit{true} \end{pmatrix} \vee \textit{false}, \text{ otherwise.} \end{cases}$$

Thus the solution of f in $x_1, \ldots, x_n$ are the sequences $a_{i1}, \ldots, a_{in}$, for all $i \in 1..p$.

Transformation rules

The six following transformations are applied on a formula $f_1 \vee \ldots \vee f_n \vee \textit{false}$ where the f_is are normalized formulae: when one of its sub-formulae is of the form of the left member of one of these rules (modulo associativity and commutativity of $\wedge$ and $\vee$), we replace it by the right member. The formula $f'_1 \vee \ldots \vee f'_m \vee \textit{false}$ thus obtained is equivalent to $f_1 \vee \ldots \vee f_n \vee \textit{false}$ and the f'_i are normalized formulae:

(1) $p \ \wedge \ \exists x_1 \ldots \exists x_n \ (q \ \wedge \ \textit{true}) \ \Rightarrow \ \exists x_1 \ldots \exists x_n \ (p \ \wedge \ q)$
(2) $(x_1, \ldots, x_n) \in r \ \wedge \ x_1 \in \{a_1\} \ \wedge \ \ldots \ \wedge \ x_n \in \{a_n\} \ \Rightarrow \ x_1 \in \{a_1\} \ \wedge \ \ldots \ \wedge \ x_n \in \{a_n\}$
(3) $\textit{false} \vee \exists x_1 \ldots \exists x_n \ (y \in \emptyset \ \wedge \ p) \ \Rightarrow \ \textit{false}$
(4) $\exists x_1 \ldots \exists x_n \ (x_1 \in I_1 \ \wedge \ p) \ \Rightarrow \ \exists x_2 \ldots \exists x_n \ p$
(5) $(x_1, \ldots, x_n) \in r \ \wedge \ x_1 \in I_1 \ \wedge \ \ldots \ \wedge \ x_n \in I_n \ \Rightarrow \ (x_1, \ldots, x_n) \in r \ \wedge \ x_1 \in I'_1 \ \wedge \ \ldots \ \wedge \ x_n \in I'_n$
(6) $\textit{false} \vee \exists x_1 \ldots \exists x_n \ (x \in I \ \wedge \ p) \ \Rightarrow \ \textit{false} \vee \exists x_1 \ldots \exists x_n \ (x \in J \ \wedge \ p) \vee \exists x_1 \ldots \exists x_n \ (x \in K \ \wedge \ p)$

where p and q are conjunctions of formulae, r is a subset of $\mathbf{Z}^n$, the integers $a_1, \ldots, a_n$ are such that $(a_1, \ldots, a_n) \in r$, the intervals $I_1, \ldots, I_n$ are non-empty and the intervals $I'_1, \ldots, I'_n$ are such that $I'_1 \times \cdots \times I'_n = \text{hull}(I_1 \times \cdots \times I_n \cap r)$ and such that there exists an i with $I'_i \neq I_i$, the intervals I, J, K are non-empty and such that $I = J \cup K$ et $J \cap K = \emptyset$ and the variable x_1 does not occur in p in the fourth rule.

Correctness

We associate to a formula of the form $f_1 \vee \ldots \vee f_n \vee \textit{false}$ where the f_is are normalized formulae the ordered pair of positive integers (n_1, n_2) such that

- $n_1 = |f_1|^2 + \cdots + |f_2|^2$ with $|f_i|$ being the product of the size of the intervals occurring in f_i,
- n_2 is the number of occurrences of $\wedge$ in the formula $f_1 \vee \ldots \vee f_n \vee \textit{false}$

If (n'_1, n'_2) is the pair associated to the formula $f'_1 \vee \ldots \vee f'_m \vee \textit{false}$ obtained by application of the transformation number i then

- $n'_1 < n_1$ for $i \in \{5, 6\}$,
- $n'_1 \leq n_1$ and $n'_2 < n_2$ for $i \in \{1, 2, 3, 4\}$.

Any sequence of transformations applied on the formula $f \vee \textit{false}$ (with f normalized) thus produces a sequence of ordered pairs (n_1, n_2) strictly decreasing lexicographically. Because n_1 and n_2 are positive, such a sequence cannot be infinite.

Solving strategy

Our general algorithm does not specify which interval to split first and how to split it. For our pentamino problem we remark that if the m_is are determined then the other unknowns of the problem are determined. Thus we split first the interval I containing the unknown m_i which have the least element, this interval is split in the two intervals $\{\underline{I}\}$ and $[\underline{I}+1, \overline{I}]$. Thus we simulate the fitting of pentaminoes in empty spaces of lower abscissa, then lower ordinate, then lower height.

We also narrow first the constraints which has the less complex algorithm of narrowing.

10 Experimental results

To have a first idea of the complexity of the pentamino problems, we use a classical enumerative algorithm (described in appendix). For each a, b, c such that $a \times b \times c = 60$, the table which follows gives the total number of solutions found, the time taken by the classical algorithm and the number of nodes of its search tree. We count symmetrical solutions as distinct solutions.

size			*number*	*time*	*number*
a	*b*	*c*	*of solutions*	*taken*	*of nodes*
1	3	20	8	16 s	71,191
1	4	15	1,472	5 m 25 s	1,789,678
1	5	12	4,040	23 m 40 s	8,387,260
1	6	10	9,356	1 h 08 m	25,848,916
2	3	10	96	10 m 46 s	3,543,583
2	5	6	2,112	15 h 21 m	257,639,965
3	4	5	31,520	160 h 16 m	9,279,204,103

Here are the time and the number of interval splittings of our algorithm to find the 8 solutions of the $1 \times 3 \times 20$ case, the 96 solutions of the $2 \times 3 \times 10$ case and the 100 first solutions of the other cases. These figures are compared with the time and the number of search tree nodes of the classical enumerative algorithm

size of the block			*our algorithm*		*classical enumerative algorithm*	
a	*b*	*c*	*time*	*splittings*	*time*	*nodes*
1	3	20	8 m 07 s	72,036	16 s	71,191
1	4	15	13 m 05 s	121,293	15 s	134,982
1	5	12	21 m 39 s	217,212	26 s	247,082
1	6	10	32 m 57 s	351,510	33 s	324,969
2	3	10	4 h 10 m	2,914,667	10 m 46 s	3,543,583
2	5	6	6 h 34 m	4,187,511	16 m 18 s	7,775,773
3	4	5	12 h 05 m	7,588,096	38 m 00 s	19,600,732

These two algorithms are programmed in C with gcc v2.7 and the benchmarks are done on a Pentium running at 90 MHz under the operating system Linux.

11 Conclusion

The ratio ρ between the execution time of our algorithm and the one of the classical enumerative algorithm lays between 20 and 60 and it must be noted that the smallest values of ρ are obtained when we work in three dimensions. However the search space seems be 2.5 times smaller in the $3 \times 4 \times 5$ case which is the most combinatory one.

Without changing the outline of our algorithm we think that it is possible to reduce ρ by an order of magnitude of 10. The elements to support this claim are the following:
(1) the algorithm spends 70% of its time for reducing the *DistinctIntegers* constraint, 20% of its time for reducing the *Class*(f) constraint and 10% of its time for the remaining tasks,
(2) instead of an $\mathcal{O}(n^2)$ algorithm for narrowing *DistinctIntegers*, we can use an $\mathcal{O}(n \log n)$ algorithm,
(3) we can implement incremental versions of the narrowing algorithms for the *DistinctIntegers* and the *Class*(f) constraints,
(4) by a more careful implementation, we can divide the time of the remaining tasks by 5.

But for really cutting down the complexity of our algorithm we must reduce the size of the search space by a better handling of the *DistinctPoints* constraint.

References

[1] S. W. Golomb. Checker boards and polyominoes. *Amer. Math. Monthly*, 61:675–682, 1954.

[2] M. Gardner. Solid polyominoes. *Scientific American*, 1958.

[3] F. Benhamou and W. J. Older. Applying interval arithmetic to real, integer and boolean constraints. *Logic Programming: The ALP Newsletter*, 6(2):13–14, May 1993. (Extended Abstract).

[4] M. Leconte. A bounds-based reduction scheme for constraints of difference. In *Constraint-96, Second International Workshop on Constraint Based Reasoning*, Key West, Florida, 1996.

[5] J. Zhou. A constraint program for solving the job-shop problem. In Eugene C. Freuder, editor, *Lecture Notes in Computer Science, Principle and Practice of Constraint Programming, CP'96*, Cambridge, MA, USA, 1996. Springer.

[6] N. Bleuzen-Guernalec and A. Colmerauer. Narrowing a n-block of sortings in $o(n \log n)$. In G. Smolka, editor, *Lecture Notes in Computer Science, Principle and Practice of Constraint Programming, CP'97*, Austria, 1997. Springer.

Appendix - The enumerative algorithm

The enumerative algorithm fills recursively a block placing at each iteration a pentamino in one of it's possible orientations. Each pentamino is placed in the empty space which has the smallest abscissa, then the smallest ordinate, then the smallest height. The heart of this algorithm can be described by the following Prolog program:

```
tile(Board, []).
tile([FirstTile | Board], Pentaminoes):-
    nonvar(FirstTile), !, tile(Board, Pentaminoes).
tile(Board, AllPentaminoes):-
    withdraw(Pentamino, AllPentaminoes, RemainingPentaminoes),
    inlist(OrientedPentamino, Pentamino), prefix(OrientedPentamino, Board),
    tile(Board, RemainingPentaminoes).
```

where the predicate `withdraw(X, L1, L2)` means that `L2` is the list `L1` without the element `X`, the predicate `inlist(X, L)` means that `X` is an element of the list `L` and the predicate `prefix(L1, L)` means that the list `L1` is a prefix of the list `L`.

Here is an example of use of the `tile` predicate when the volume we want to tile with the pentaminoes if a board of size 6×10:

```
pentaminoes:-
    Board = [ _,_,_,_,_,_,_,_,_,_,o,
              _,_,_,_,_,_,_,_,_,_,o,
              _,_,_,_,_,_,_,_,_,_,o,
              _,_,_,_,_,_,_,_,_,_,o,
              _,_,_,_,_,_,_,_,_,_,o,
              _,_,_,_,_,_,_,_,_,_   ],
    Pentaminoes = [F, I, L, P, N, T, U, V, W, X, Y, Z],
    tile(Board, Pentaminoes).
```

the variables `F`, `I`, `L`, `P`, `N`, `T`, `U`, `V`, `W`, `X`, `Y`, `Z` having beforehand being unified with the lists of all the possible orientations of each pentamino. For example, here is the declaration of the `T`:

```
T = [[ t,_,_,_,_,_,_,_,_,_,_,
       t,t,t,_,_,_,_,_,_,_,_,
       t                     ],
     [     t,_,_,_,_,_,_,_,_,
       t,t,t,_,_,_,_,_,_,_,_,
       _,_,t                 ],
     [ t,t,t,_,_,_,_,_,_,_,_,
       _,t,_,_,_,_,_,_,_,_,_,
       _,t                   ],
     [   t,_,_,_,_,_,_,_,_,_,
       _,t,_,_,_,_,_,_,_,_,_,
       t,t,t                 ]].
```