A polynomial Time Algorithm to Solve The Max-atom Problem

Chams Lahlou and Laurent Truffet
IMT-A
Dpt. Automatique-Productique-Informatique
Nantes, France
email: chams.lahlou@imt-atlantique.fr
email: laurent.truffet@imt-atlantique.fr

June 17, 2021

Abstract

In this paper we consider m ($m \geq 1$) conjunctions of Max-atoms that is atoms of the form $\max(z, y) + r \geq x$, where the offset r is a real constant and x, y, z are variables. We show that the Max-atom problem (MAP) belongs to P. Indeed, we provide an algorithm which solves the MAP in $O(n^6m^2 + n^4m^3 + n^2m^4)$ operations, where n is the number of variables which compose the max-atoms. As a by-product other problems also known to be in $\mathsf{NP} \cap \mathsf{co}\text{-}\mathsf{NP}$ are in P. P1: the problem to know if a tropical cone is trivial or not. P2: problem of tropical rank of a tropical matrix. P3: parity game problem. P4: scheduling problem with AND/OR precedence constraints. P5: problem on hypergraph (shortest path). P6: problem in model checking and μ-calculus.

Keywords. Complexity, polynomial time algorithm.

1 Introduction

In this paper we consider $\mathcal{F}$ a set of m ($m \geq 1$) conjunctions of Max-atoms that is inequalities of the form $\max(z, y) + r \geq x$, where the offset r is a real constant value and x, y, z are variables. And we are looking at non-trivial solutions (ie $\neq -\infty$) of $\mathcal{F}$.

It seems that the Max-atom problem (MAP) has been described for the first time in [2]. According to [3] the Max-atom problem has been introduced as a generalization of Difference Logic (DL), i.e. atoms of the form $y + k \geq x$ with offset $k \in \mathbb{Z}$. DL was used to study delays in circuits using SAT-modulo theory (see e.g. [11]). DL is also known as Difference Bound Matrix (DBM) in model checking of timed automata community. Different extensions of DBM or DL from the one studied in this paper have been proposed for tracking numerical errors in programs (see e.g. [9] and references therein).

The expressiveness of Max-atom over $\mathbb{Z}$ have been stressed in [2], [3]. We recall some results about expressiveness hereafter. Because max is idempotent DL can be expressed as $\max(x, x) + k \geq y$. Strict inequalities $\max(x, y) + k > z$ can be expressed as $\max(x, y) + k - 1 \geq z$. One can also express equalities of the form $\max(x, y) + k = z$, $\max(x, y) + k = \max(x', y') + k'$. And one can also express inequalities of the form $\max(x + k, y + k') \geq z$, with $k \neq k'$.

Hereafter, we list some important problems in diffrent areas of research. All these problems are known to be in $\mathsf{NP} \cap \mathsf{co\text{-}NP}$. After this list we explain their links with the Max-atom problem.

P1 Tropical geometry problem. We consider two $m \times n$-matrices $\boldsymbol{A} = (a_{i,j})$ and $\boldsymbol{B} = (b_{i,j})$ with entries in $\mathbb{R} \cup \{-\infty\}$ and we define the tropical cone $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$ as the following subset of $[-\infty, \infty)^n$:

$$\mathcal{C}(\boldsymbol{A}, \boldsymbol{B}) := \left\{ \boldsymbol{x} = (x_1, \ldots, x_n)' : \forall i \in [m],\ \max_{j \in [n]}(a_{i,j} + x_j) \leq \max_{j \in [n]}(b_{i,j} + x_j) \right\}.$$

Where $[m]$ (resp. $[n]$) denotes the set $\{1, \ldots, m\}$ (resp. $\{1, \ldots, n\}$).
Problem. Does $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B})$ contain a non trivial element $\boldsymbol{x} \neq -\boldsymbol{\infty}$?

P2 Tropical independence and tropical rank of a matrix. We consider a $m \times n$-matrix $\boldsymbol{A} = (a_{i,j})$ with entries in $\mathbb{R} \cup \{-\infty\}$ and $m \geq n$. Are the columns of matrix $\boldsymbol{A}$ tropically linearly dependent ? It means that we have to solve the following problem.
Can we find scalars $\lambda_1, \ldots, \lambda_n$ not all equal to $-\infty$ such that $\forall i \in [m]$, the quantity $z := \max_{j \in [n]}(a_{i,j} + \lambda_j)$ is attained by at least two values of $j \in [n]$?

A $r \times r$-matrix $\boldsymbol{B}$ with entries in $[-\infty, \infty)$ which is a submatrix (minor) of matrix $\boldsymbol{A}$ is tropically singular if its tropical permanent defined as:

$$\max_{s \in S(r)} (b_{1,s(1)} + \ldots + b_{r,s(r)}),$$

where $S(r)$ denotes the set of permutations of set $[r]$, is either attained by at least two permutations or is $-\infty$. The rank of matrix $\boldsymbol{A}$ is defined as the maximum size r such that submatrix $\boldsymbol{B}$ is non-singular.
Problem. For a fixed $k \in [n]$. Does matrix $\boldsymbol{A}$ have a tropical rank $\geq k$?

P3 Two-person games. Let us consider two players $\mathbb{0}$ and $\mathbb{1}$ which are involved in the following path-forming game specified by a digraph $G = (V, A)$ in which every vertex v (i.e. element of V) has > 0 outdegree, $d^+(v)$, and by two disjoint sets $V_{\mathbb{0}}$ and $V_{\mathbb{1}}$ which are subsets of V. The game start at time $k = 0$ at vertex v_0. At any time k the next move on graph G is done as follows. If the vertex attained at epoch k is $v_k \in V_\imath$, $\imath \in \{\mathbb{0}, \mathbb{1}\}$ then player $\imath$ chooses a vertex v_{k+1} such that $(v_k, v_{k+1}) \in A$. Else (i.e. $v_k \in V - (V_{\mathbb{0}} \cup V_{\mathbb{1}})$) then the vertex v_{k+1} is choosen with probability

$\frac{1}{d^+(v_k)}$. When $V = V_{\mathbb{O}} \cup V_{\mathbb{1}}$ the game is said to be determistic. Otherwise it is said to be stochastic. A strategy of a player is the series of moves of the player. These moves are made according to a rule. A strategy is positional if it only depends on the current position of the game and not moves made earlier.

Example of deterministic games are mean-payoff games and parity games. In mean-payoff games the arcs of G are labeled by a function (i.e. a weight) $w : A \to \{-W, \dots, 0, \dots, W\}$, $W \geq 1$. And we are considering the long-run average:

$$J(k) := \frac{1}{k} \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

of the weight of the walk. For mean-payoff games the goal of the player $\mathbb{O}$ (resp. $\mathbb{1}$) is to maximize (resp. minimize) $\liminf J(k)$ (resp. $\limsup J(k)$) as $k \to \infty$.
Problem. In [5], [8] it is proved that such a game has a value x such that $\limsup J(k) \leq x \leq \liminf J(k)$. Does there exists a polynomial time algorithm to compute x ?

Following e.g. [7, Chap. 1, 2, 6 and 7] in parity games each vertex is labeled by an integral priority assignment $p : V \to P := \{0, \dots, \#V - 1\}$, where $\#V$ denotes the number of vertices of V. For an infinite series of moves or path $\pi := (v_0, v_1, \dots) \in V^{\mathbb{N}}$ we define the following subset associated with P and p:

$$I(\pi) := \{l \in P : \forall i \exists j > i, v_j \in \pi \text{ and } p(v_j) = l\}$$

which is the set of priorities occuring infinitely often in π. And we are considering the following quantity:

$$p^{\vee} := \max(I(\pi)).$$

The goal of player $\mathbb{O}$ (resp. $\mathbb{1}$) is to ensure that $p^{\vee}$ is even (resp. odd).

Example of stochastic game is the simple stochastic game [4]. It is a Markovian random walk with an initial state $v_{-\infty} \in V$ and two sink vertices $v_{\infty}^{\mathbb{O}}$ and $v_{\infty}^{\mathbb{1}}$. The transition probabilities matrix $\boldsymbol{P} = (p_{u,v})$ is defined as follows $\forall (u, v) \in A$:

$$p_{u,v} = \begin{cases} 1 & \text{if } u \in V_{\mathbb{O}} \cup V_{\mathbb{1}} \text{ or } u = v = v_{\infty}^{\imath}, \imath \in \{\mathbb{O}, \mathbb{1}\} \\ \frac{1}{d^+(u)} = \frac{1}{2} & \text{if } u \in V - (V_{\mathbb{O}} \cup V_{\mathbb{1}}) \\ 0 & \text{otherwise.} \end{cases}$$

Problem. For the above mentioned games, does there exist a positional winning strategy for player $\mathbb{O}$ and/or $\mathbb{1}$? Can we find the optimal strategy in polynomial time ?

P4 Scheduling problem with AND/OR precedence constraints [10]. Let us consider a DAG $G = (V, A)$. The set of vertices V is called set of jobs. The arcs (i.e. the elements of A) represent the precedence constraints. An AND/OR precedence constraints system is modelled by a set W of pairs $w = (X, j)$ where $j \in V$ and $X \subseteq (V - \{j\})$. The job j can start execution as soon as $\exists i \in X$ such that $(i, j) \in A$ which has been completed. Note that when X is a singleton the constraint system is called AND precedence constraints system.

Problem. One fundamental issue is to compute the vector of earliest jobs start times $\boldsymbol{x} \in \mathbb{Z}^m$ where $m := \#(V \cup W)$. That is find a nonnegative vector $\boldsymbol{x}$ such that $\forall (X, j) \in W$ the following constraint

$$\min_{i \in X}(x_i + k_{iw}) \leq x_j$$

is satisfied. Where $\forall w = (X, j) \in W$ and $\forall i \in X$, $k_{iw} \in \mathbb{Z}$ is a bounded time lag. This system is the dual conterpart of the Max-atom problem. It is based on the following changes:

$$x \leftrightarrow -x, \ \max \leftrightarrow \min, \ \geq \leftrightarrow \leq .$$

P5 Shortest hyperpath problem in weighted directed hypergraph [6]. A weighted directed hypergraph is a tuple $G = (V, A, w)$ where $V = \{v_1, \ldots, v_n\}$ is the set of nodes and A is the set of hyperarcs. An hyperarc is an ordered pair $a = (\mathsf{t}(a), \mathsf{h}(a))$ such that $\mathsf{t}(a), \mathsf{h}(a) \subseteq V$ and $\mathsf{t}(a) \cap \mathsf{h}(a) = \emptyset$. And a map $w : A \to \mathbb{Z}$. A path π_{st} of length q in G is defined as a sequence of nodes and hyperarcs

$$\pi_{st} = (s, a_1, v_2, a_2, \ldots, a_q, t)$$

such that $s \in \mathsf{t}(a_1)$, $t \in \mathsf{h}(a_q)$ and $\forall i \in \{2, \ldots, q\}$ $v_i \in \mathsf{h}(a_{i-1}) \cap \mathsf{t}(a_i)$. There are different notions of hyperpaths. We study the following one when $\#\mathsf{h}(a) = 1$, $\forall a \in A$ (called B-graph in [6]). Let $X \subseteq V$ and $y \in V$. There is a hyperpath from X to y denoted $t_{X,y}$ if either a) $y \in X$ or b) $\exists (Z, y) \in A$ and hyperpaths from X to z, $z \in Z$. The weight of $t_{X,y}$, say $\theta(t_{X,y})$, is recursively defined as follows:

$$\theta(t_{X,y}) := \begin{cases} 0 & \text{if } y \in X \\ w(Z, y) + \max_{\{z \in Z : \exists t_{X,z} \text{ hyperpath}\}}(\theta(X, z)) & \text{if } (Z, y) \in A. \end{cases}$$

Define the $\mathbb{Z} \cup \{-\infty, \infty\}$-valued function

$$\mathsf{d}(X, y) := \min\{\theta(t_{X,y}) : t_{X,y} \text{ is a hyperpath from } X \text{ to } y\}.$$

The function is said to be well-defined if $\max_{y \in V}(\mathsf{d}(X, y)) > -\infty$.

Problem. Decide wether function $\mathsf{d}(X, \cdot)$ is well-defined $\forall \emptyset \neq X \subseteq V$.

P6 Model checking and μ-calculus. Model checking is a method for analysing dynamical systems that can be modeled by state-transition systems. Model checking is widely used for the verification of hardware and software in industry. μ-calculus is a logic describing properties of labelled transition systems. A labelled transition is defined as the tuple $M = (S, \Lambda, \rightarrow)$ where S is a set of states, Λ is a set of labels (actions) and $\rightarrow \subseteq S \times \Lambda \times S$. Propositions and variables are defined as subsets of S. The set of formulas F is defined as follows:

- propositions and variables are formulas
- $\forall f, g$: $f, g \in \mathsf{F} \Rightarrow (f \text{ and } g) \in \mathsf{F}$
- $\forall f \in \mathsf{F}, \forall \alpha \in \Lambda$, $[\alpha] f \in \mathsf{F}$ (after action α necessarily f)
- $\forall f \in \mathsf{F}$, $\forall$ variable z, $\nu z.f \in \mathsf{F}$ (the number of free occurrences of $\neg z$ is even)
- $\forall f \in \mathsf{F}$, $\neg f \in \mathsf{F}$

Problem. Check if a formula $f \in \mathsf{F}$ holds in a state $s \in S$ of the transition system M.

Problem P1 is PTIME equivalent to mean payoff game (described in P3) with the dynamic vectorial operator $(f_1, \ldots, f_n)$ defined by:

$$\forall j \in [n],\ f_j(\boldsymbol{x}) = \min_{i \in [m]} (-a_{i,j} + \max_{k \in [n]} (b_{i,k} + x_k)).$$

This result is proved in e.g. [1].

Problem P2 reduces to mean payoff game (described in P3). This result is also proved in e.g. [1].

In game theory the mean payoff games seem also to play a central role. Indeed it exists a polynomial reduction from MPG to simple stochastic games (see e.g. [12]). Parity games (PG) are PTIME reduced to MPG. Moreover, MPG and PG behave the same way for each pair of positional strategies (see e.g. [7, Lemma 7.5]).

Problem P4 is PTIME equivalent to MPG (see e.g. [10]).

Problem P5 is PTIME equivalent to MAP (see e.g. [2], [3]).

Problem P6 is PTIME equivalent to PG (see e.g. [7, Chap. 10]). Roughly speaking, for a formula f and a state s of a transition system M a graph $G_{M,f}$ is built by induction on the formula f. Then, the parity game associated with $G_{M,f}$ is solved. The player $\mathbb{0}$ (resp. $\mathbb{1}$) wins if and only if f does not hold (resp. holds) in state s.

Concerning the Max-atom problem. It has been proved this problem plays a central role in decision theory. Indeed, the MAP is PTIME equivalent to problem P1, problem P4, problem P5 (see e.g. [2], [3]). Since problems P1 and P4 are PTIME equivalent to MPG, the MAP is also PTIME equivalent to MPG. The known complexity results of MAP are as follows. The MAP is known to be in $\mathsf{NP} \cap \mathsf{co\text{-}NP}$. When the offsets are all in $\mathbb{Z}$ the MAP is weakly polynomial (see [3]). But the algorithm provided in [3] is not polynomial for MAP over $\mathbb{Q}$. Thus, it seems important to be able to solve MAP over $\mathbb{R}$.

The paper is organized as follows. In Section 2 we give the precise formulation of the Max-atom problem. In Section 3 we provide the important simplification rules. These rules are the key points of our algorithm presented in Section 4. And in Section 5 we state that algorithm terminates and we obtain bound on its running time.

2 Problem statement

Let m, n be integers ≥ 1. Let $r_1, \ldots, r_m \in \mathbb{R}$. Let $V := \{x_1, \ldots, x_n\}$, with n integer ≥ 1, be the set of variables. Let $f, s, l : [m] \to [n]$ be three applications. And we consider the following system of inequalities:

$$\forall i \in [m], F_i : \max(x_{f(i)}, x_{s(i)}) + r_i \geq x_{l(i)}. \tag{1}$$

Problem. We are looking for the existence of non-trivial solutions $\boldsymbol{x} = (x_1, \ldots, x_n) \neq -\boldsymbol{\infty}$ of the system (1):

Can we built a non-trivial solution using a strongly polynomial time algorithm ?

3 Simplification rules

In this section we present important lemmas which will be useful to solve the MAP.

Lemma 3.1 *Let us consider the max-atom $F : \max(y, x) + r \geq x$. Then, if $r \geq 0$ F is always true. If $r < 0$ then F is equivalent to $A : y + r \geq x$.*

Proof. If $r \geq 0$ then $\max(y, x) + r \geq x + r \geq x$. Now, if $r < 0$ then

$$\max(y, x) + r \geq x \Leftrightarrow (y + r \geq x) \text{ or } (x + r \geq x),$$

where $(x + r \geq x)$ is false. Hence the result. $\square$.

Lemma 3.2 *Let us consider the following system of max-atoms:*

$$\begin{cases} F : & \max(z, y) + r \geq x \\ F' : & \max(z, y) + r' \geq x. \end{cases}$$

If $r \leq r'$ then $\begin{cases} F \\ F' \end{cases} \Leftrightarrow F$.

Proof. If $r \leq r'$ then $F \Rightarrow F'$ and the result is now obvious. $\square$.

Lemma 3.3 *Let us consider the following system of inequalities:*

$$\left\{\begin{array}{ll} A: & y+r \geq x \\ F': & \max(z,y)+r' \geq x. \end{array}\right.$$

If $r \leq r'$ *then* $\left\{\begin{array}{c} A \\ F' \end{array}\right. \Leftrightarrow A.$

Proof. Assume $r \leq r'$. Then, $\max(z,y)+r' \geq y+r' \geq y+r$ which implies that $A \Rightarrow F'$. And the result is now proved. $\square$.

Lemma 3.4 *Let us consider the following system of inequalities:*

$$\left\{\begin{array}{ll} A: & z+r \geq y \\ F': & \max(z,y)+r' \geq x. \end{array}\right.$$

If $r \leq 0$ *then* $\left\{\begin{array}{c} A \\ F' \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} A: z+r \geq y \\ A': z+r' \geq x \end{array}\right. .$

Proof. If $r \leq 0$ then $z \geq y$ (by A) and $\max(z,y)=z$. Hence the result. $\square$.

Lemma 3.5 *Let us consider the following system of inequalities:*

$$\left\{\begin{array}{ll} A: & y+r \geq x \\ F: & \max(z,x)+r' \geq y. \end{array}\right.$$

If $r+r' < 0$ *then* $\left\{\begin{array}{c} A \\ F' \end{array}\right. \Leftrightarrow \left\{\begin{array}{c} A: y+r \geq x \\ A': z+r' \geq y \end{array}\right.$

Proof. Assume that $r+r' < 0$. We remark that $(x+r' \geq y) \Leftrightarrow (x+r'+r \geq y+r)$. Noticing that $y+r \geq x$ it implies that $(x+r' \geq y)$ is false. Thus, $\max(z,x)+r' \geq y$ is equivalent to $z+r' \geq y$. And the result is proved. $\square$.

4 Algorithm A

Input: a set of variables $V=\{x_1,\ldots,x_n\}$ and a set of m ($m \geq 1$) max-atoms $\mathcal{F}$ where the max-atoms are expressions using only variables in V. And a set of ordered constant values $r_1 \leq \ldots \leq r_m$

Output: A solution $\boldsymbol{x}=(x_1,\ldots,x_n) \neq -\infty$ if exists $-\infty$ otherwise.

Let us introduce the following notation: $F(z,y,x;r) := \max(z,y)+r \geq x$. Note that $F(y,y,x;r)=y+r \geq x$ and $F(z,y,x;r)=F(y,z,x;r)$.

We also need to define the 'sum' of two valued graphs as follows.

Definition 4.1 *Let* $x \overset{w}{\to} y$ *and* $x' \overset{w'}{\to} y'$ *we define the sum* $\hat{+}$ *of these two arcs as follows:*

$$x \overset{w}{\to} y \hat{+} x' \overset{w'}{\to} y' := \begin{cases} \{x \overset{\max(w,w')}{\longrightarrow} y\} & \textit{if } x = x' \textit{ and } y = y' \\ \{x \overset{w}{\to} y, x' \overset{w'}{\to} y'\} & \textit{otherwise.} \end{cases}$$

Definition 4.2 *Let* $(V_1, \mathcal{E}_1)$ *and* $(V_2, \mathcal{E}_2)$ *be two valued graphs. The sum of the graphs* $(V_1, \mathcal{E}_1)$ *and* $(V_2, \mathcal{E}_2)$ *is the graph* $(V, \mathcal{E})$ *denoted* $(V_1, \mathcal{E}_1)\hat{+}(V_2, \mathcal{E}_2)$ *defined by:* $V := V_1 \cup V_2$ *and* $\mathcal{E} := \mathcal{E}_1 \hat{+} \mathcal{E}_2$ *is the Minkowski* $\hat{+}$*-sum of the sets of valued arcs* $\mathcal{E}_1$ *and* $\mathcal{E}_2$*.*

Begin

0 . if $r_1 \geq 0$ then $\boldsymbol{x} := (0, \dots, 0)$ exit() else $\mathcal{E} := \emptyset$

1 $\forall F(x, y, x; r) \in \mathcal{F}$ do (* see Lemma 3.1 *)

  1.1 if $r \geq 0$ then $\mathcal{F} := \mathcal{F} - \{F(x, y, x; r)\}$ else $\mathcal{F} := \mathcal{F} - \{F(x, y, x; r)\} + \{F(y, y, x; r)\}$

2 $\forall (F(z, y, x; r), F(z, y, x; r')) \in \mathcal{F} \times \mathcal{F}$ do (* see Lemma 3.2 *)

  2.1 if $r \geq r'$ then $\mathcal{F} := \mathcal{F} - \{F(z, y, x; r')\}$ else $\mathcal{F} := \mathcal{F} - \{F(z, y, x; r)\}$

3 $\mathcal{F}' := \mathcal{F}$

4 $\forall F(y, y, x; r) \in \mathcal{F}$ do

  4.1 . $\mathcal{E} := \mathcal{E} \hat{+} \{x \overset{-r}{\to} y\}$ (* see definition 4.1 *)

5 . if $(V, \mathcal{E})$ has a > 0 circuit c then

  5.1 . Put all variables of c at the value $-\infty$

  5.2 . Delete the max-atoms of the form $\max(z, y) + r \geq -\infty$ in $\mathcal{F}$

  5.3 . Propagate $-\infty$ in $\mathcal{F}$ using the following rule:

$$-\infty + r \geq x \to x := -\infty$$

6 . if $\boldsymbol{x} = -\boldsymbol{\infty}$ then exit() else

  6.1 . $V := V - \{x \in V : x = -\infty\}$

  6.2 . goto 0.

7 if $\mathcal{E} \neq \emptyset$ then

  7.1 Compute the transitive closure of $(V, \mathcal{E})$: $(V, \mathcal{E}^*)$

7.2 for all $u, v \in V$ compute $r^*_{u,v}$ defined by:

$$r^*_{u,v} = \begin{cases} \max\{\mathsf{weight}(\pi) : \pi \text{ path from } u \text{ to } v\} & \text{if } \exists (u,v) \in \mathcal{E}^* \\ -\infty & \text{otherwise.} \end{cases} \tag{2}$$

7.3 $\mathcal{F} := \mathcal{F} + \sum_{(u,v)\in\mathcal{E}^*} \{F(v, v, u; -r^*_{u,v})\}$

8 $\forall (F(y, y, x; r), F(z, y, x; r')) \in \mathcal{F} \times \mathcal{F}$ do (* see Lemma 3.3 *)

8.1 if $r \leq r'$ then $\mathcal{F} := \mathcal{F} - \{F(z, y, x; r')\}$

9 $\forall (F(z, z, y; r), F(z, y, x; r')) \in \mathcal{F} \times \mathcal{F}$ do (* see Lemma 3.4 *)

9.1 if $r \leq 0$ then $\mathcal{F} := \mathcal{F} - \{F(z, y, x; r'\} + \{F(z, z, x; r'\}$

10 $\forall (F(y, y, x; r), F(z, x, y; r')) \in \mathcal{F} \times \mathcal{F}$ do (* see Lemma 3.5 *)

10.1 if $r + r' < 0$ then $\mathcal{F} := \mathcal{F} - \{F(z, x, y; r')\} + \{F(z, z, y; r')\}$

11 if $\mathcal{F}' \neq \mathcal{F}$ then goto 3

12 . if the set of max-atoms F of $\mathcal{F}$ of the form $\max(z', y') + r \geq x'$ is empty then

12.1 . $\boldsymbol{x} := \mathsf{builtSolution}((V, \mathcal{E}))$

12.2 . exit()

13 $\forall F(z', y', x'; r) \in \mathcal{F}$ do

13.1 Compute $\Phi(z' \geq y', F(z', y', x'; r), \mathcal{F}) \stackrel{\mathsf{not}}{=} \Phi_{z' \geq y'}$

13.2 Compute $\Phi(y' \geq z', F(z', y', x'; r), \mathcal{F}) \stackrel{\mathsf{not}}{=} \Phi_{y' \geq z'}$

13.3 if $\Phi_{z' \geq y'}.status = \Phi_{y' \geq z'}.status$ =FALSE then

13.3.1 . $y', z' := -\infty$

13.3.2 . goto 5.2

13.4 if $\Phi_{z' \geq y'}.status = \Phi_{y' \geq z'}.status$ =TRUE then

13.4.1 . $y' = z'$

13.4.2 $V := V - \{z'\}$

13.4.3 $\mathcal{F} :=$ the set of max-atoms of $\mathcal{F}$ where variable z' is replaced by y'

13.4.4 goto 0.

14 $G := \hat{\sum}_{\{(x,y)\in V\times V : \Phi_{y\geq x}.status = TRUE\}} \Phi_{y \geq x}.G$ (* see definition 4.2 *)

15 $\boldsymbol{x} := \mathsf{builtSolution}(G)$

End.

<u>function Φ</u>:

- Parameters:
  - decision: an inequality of the form $z \geq y$ which is assumed to be a priori true
  - max-atom in which the decision is made (ie of the form $\max(z, y) + r \geq x$)
  - $\mathcal{F}$ the system of max-atoms
- Return:
  - G: synthesis graph
  - status: Boolean which is FALSE if the new synthesis graph G contains > 0 circuits, TRUE otherwise.

Φ.0 $\mathcal{E} := \emptyset$; $\mathcal{F}' := \mathcal{F}$

Φ.1 $\mathcal{F}' := \mathcal{F}' - \{F(z, y, x; r)\} + \{F(z, z, y; 0), F(z, z, x; r)\}$

Φ.2 $\mathcal{F}'' := \mathcal{F}'$

Φ.3 $\forall (F(y, y, x; r), F(z, y, x; r')) \in \mathcal{F}' \times \mathcal{F}'$ do (* see Lemma 3.3 *)

Φ.3.1 if $r \leq r'$ then $\mathcal{F}' := \mathcal{F}' - \{F(z, y, x; r')\}$

Φ.4 $\forall (F(z, z, y; r), F(z, y, x; r')) \in \mathcal{F}' \times \mathcal{F}'$ do (* see Lemma 3.4 *)

Φ.4.1 if $r \leq 0$ then $\mathcal{F}' := \mathcal{F}' - \{F(z, y, x; r'\} + \{F(z, z, x; r'\}$

Φ.5 $\forall (F(y, y, x; r), F(z, x, y; r')) \in \mathcal{F}' \times \mathcal{F}'$ do (* see Lemma 3.5 *)

Φ.5.1 if $r + r' < 0$ then $\mathcal{F}' := \mathcal{F}' - \{F(z, x, y; r')\} + \{F(z, z, y; r')\}$

Φ.6 $\forall F(v, v, u; r) \in \mathcal{F}'$ do

Φ.6.1 . $\mathcal{E} := \mathcal{E} \hat{+} \{u \xrightarrow{-r} v\}$ (* see definition 4.1 *)

Φ.7 if $(V, \mathcal{E})$ has > 0 circuit then

Φ.7.1 $\Phi.status$:=FALSE

Φ.7.2 Return(Φ)

Φ.8 Compute the transitive closure of $(V, \mathcal{E})$: $(V, \mathcal{E}^*)$

Φ.9 for all $u, v \in V$ compute $r^*_{u,v}$ defined by (2)

Φ.10 $\mathcal{F}' := \mathcal{F}' + \sum_{(u,v) \in \mathcal{E}^*} \{F(v, v, u; -r^*_{u,v})\}$

Φ.11 if $\mathcal{F}' \neq \mathcal{F}''$ goto Φ.2

Φ.12 $\Phi.G := (V, \mathcal{E})$

Φ.13 $\Phi.status$:=TRUE

Φ.14 Return(Φ)

function builtSolution:

- Parameters:
  - V: a set of variables $\subseteq \{x_1, \ldots, x_n\}$
  - $\mathcal{E}$: a set of valued arcs $\subseteq V \times V$
- Return:
  - $\boldsymbol{x}$: a $\#V$-dimensional vector with all components $\neq -\infty$

bS.1 add a variable x_0 to V

bS.2 add 0-valued arcs $x_0 \xrightarrow{0} u$, $u \in V$ to set $\mathcal{E}$

bS.3 $\forall u \in V$: $u :=$ longuest path from x_0 to u in $(V, \mathcal{E})$

bS.4 Return($\boldsymbol{x} := (u)_{u \in V}$)

5 Properties of the algorithm A of section 4

In this section we prove that the algorithm A terminates and we compute its worst case time complexity.

5.1 Complexity analysis of the function builtSolution

The main step of the function builtSolution is the step bS.3. Applying e.g. a Bellman-Ford (BF) algorithm we obtain the possible values of vector $\boldsymbol{x}$. The complexity of the BF algorithm is in:

$$O(n^3),$$

which is also the overall complexity of the function builtSolution.

5.2 Complexity analysis of the function Φ

In this paragraph we study the main sub-routine of the algorithm A.

Proposition 5.1 *The function Φ terminates.*

Proof. Let us remark that the number of max-atoms in Φ can only decreases: see steps Φ.1, Φ.3.1, Φ.4.1 and Φ.5.1. These steps can only create atoms (i.e. expressions of the form $y + r \geq x$). Which can modify the set $\mathcal{E}$ (see step Φ.6). If the steps Φ.3.1, Φ.4.1 and Φ.5.1 do not modify $\mathcal{F}'$ the set $\mathcal{E}$ is not modified. Then, there exist two cases. The graph $(V, \mathcal{E})$ has > 0 circuit and Φ stops at step $\Phi.7.2$. Otherwise, we compute the same transitive closure of $\mathcal{E}$ and because the transitive closure of a graph is an idempotent operator the step $\Phi.10$ does not modify the set $\mathcal{F}'$. Thus, Φ terminates. $\square$.

Proposition 5.2 *The number of steps in function Φ is at most:*

$$m \times (3(m+n^2)^2 + m + n^2 + 2n^3).$$

Proof. The loop $\Phi.2$-$\Phi.11$ only modifies max-atoms of the form $\max(z,y)+r \geq x$, with $z \neq y$. Which are at most m. So, the loop $\Phi.2$-$\Phi.11$ is executed at most m times.
The complexity of the loop $\Phi.2$-$\Phi.11$ is as follows. Complexity of $\Phi.3, \Phi.4$ and $\Phi.5$ are in $O((\#\mathcal{F}')^2)$. The complexity of $\Phi.6$ is in $O(\#\mathcal{F}')$. The detection of a > 0 circuit and the closure of a graph with at most n vertices can be done in $O(n^3)$ by using e.g. Bellman-Ford algorithm. So, time complexity of $\Phi.7 - \Phi.9$ is $O(2n^3)$. Now, by $\Phi.10$ we see that the set $\mathcal{F}'$ contains (1) the max-atoms of the form $\max(z,y)+r \geq x$, with $z \neq y$, which number are $\leq m$ and (2) the atoms of the form $y + r \geq x$, $x, y \in V \times V$, which number are $\leq n^2$. So, that

$$\#\mathcal{F}' \leq m + n^2.$$

Hence the result. $\square$.

5.3 Overall complexity analysis

This paragraph is the main paragraph of section 5. We detail the number of operations of the algorithm A.

Proposition 5.3 *The algorithm A terminates.*

Proof. First of all it is clear that if all the offsets r_i are ≥ 0 the solution $\boldsymbol{x} = (0, \ldots, 0)$ is obvious. The steps 0 to 11 tend to decrease the number of max-atoms of the form $\max(z,y) + r \geq x$, with $z \neq y$. Which are at most m. And the number of variables $\neq -\infty$ which are at most n. Moreover, the transitive closure of a graph is idempotent. So, the loops 0-6.2 and 3-11 terminate. The step 12 terminates as the function builtSolution terminates. Because function Φ terminates the steps 13.1 and 13.2 terminate. The loop 5.2-13.3.2 decreases the number of variables $\neq -\infty$ and thus, terminates. And the loop 0-13.4.4 dcreases the number of max-atoms. Thus, it also terminates. $\square$.

Proposition 5.4 *The time complexity of algorithm A is in*

$$O(n^6 m^2 + n^4 m^3 + n^2 m^4).$$

Proof. The time complexity is estimated as follows. Let us denote: ℓ_1 the loop 0-6.2, ℓ_2 the loop 0-13.4.4, ℓ_3 the loop 3-11 and ℓ_4 the loop 5.2-13.3.2. Let us denote $|\ell_i|$ the maximum number of times that loop ℓ_i can be repeated, $i = 1, \ldots, 4$. Then,

- $|\ell_1| = |\ell_2| = |\ell_4| = n$, indeed each loop decreases the number of variables.
- $|\ell_3| = m$, indeed the loop decreases the number of max-atoms the form $\max(z, y) + r \geq x$, with $z \neq y$. Which are at most m.

We have the following time complexity for each step of the algoritm A.

- Step $1 \in \ell_1, \ell_2$: complexity is $O(\#\mathcal{F})$
- Step $2 \in \ell_1, \ell_2$: complexity is $O((\#\mathcal{F})^2)$
- Step $4 \in \ell_1, \ell_2, \ell_3$: complexity is $O(\#\mathcal{F})$
- Step $5 \in \ell_1, \ell_2, \ell_3$: complexity is $O(n^3)$
- Step $5.1 \in \ell_1, \ell_2, \ell_3$: complexity is $O(n)$
- Step $5.2 \in \ell_1, \ell_2, \ell_3, \ell_4$: complexity is $O(\#\mathcal{F})$
- Step $5.3 \in \ell_1, \ell_2, \ell_3, \ell_4$: complexity is $O(\#\mathcal{F})$
- Steps $7.1 - 7.2 \in \ell_2, \ell_3, \ell_4$: complexity is $O(n^3)$
- Steps $8, 9, 10 \in \ell_2, \ell_3, \ell_4$: complexity is $O((\#\mathcal{F})^2)$
- Steps $13.1, 13.2 \in \ell_2, \ell_4$: complexity is m times the time complexity of function Φ
- Step 14: complexity is $O(n^4)$
- Step 15: complexity is the one of the function $\mathsf{builtSolution}$, i.e. $O(n^3)$
- A bound on the number of elements of $\mathcal{F}$ is deduced from 7.3 which is:

$$\#\mathcal{F} \leq m + n^2.$$

The worst case for loops is when they are all considered as nested. Thus, we obtain a worst case complexity in

$$
\begin{aligned}
& f(n,m) = n^2 \times (m + n^2) \\
+\ & n^2 \times (m + n^2)^2 \\
+\ & n^2 m \times (m + n^2) \\
+\ & n^2 m \times O(n^3) \\
+\ & n^2 m \times O(n)
\end{aligned}
$$

$$+\ n^3m \times (m+n^2)$$

$$+\ n^3m \times (m+n^2)$$

$$+\ 2 \times n^2m \times O(n^3)$$

$$+\ 3 \times n^2m \times (m+n^2)^2$$

$$+\ 2 \times n^2m \times [m \times (3(m+n^2)^2 + m + n^2 + 2n^3)]$$

$$+\ O(n^4)$$

$$+\ O(n^3)\ ,$$

which is an $O(n^6m^2 + n^4m^3 + n^2m^4)$ polynomial. Hence, the result. $\square$.

References

[1] M. Akian, S. Gaubert, and A. Guterman. The Correspondence Between Tropical Convexity and Mean Payoff Games. In *Proc. 19th int. Symp. MTNS 2010*, 2010.

[2] M. Bezem, R. Nieuwenhuis, and E. Rodriguez-Carbonell. The Max-Atom Problem and Its Relevance. *LNCS*, 5330, 2008. (47-61).

[3] M. Bezem, R. Nieuwenhuis, and E. Rodriguez-Carbonell. Hard Problems in Max-Algebra, Control Theory, Hypergraphs and Other Areas. *Inform. Process. Letters*, 110(4), 2010. (133-138).

[4] A. Condon. The Complexity of Stochastic Games. *Inform. and Comput.*, 96(2), 1992. (203-224).

[5] A. Ehrenfeucht and J. Mycielski. Positional Strategies For Mean Payoff Games. *Int. Jour. of Game Theory*, 8(2), 1979. (109-113).

[6] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen. Directed Hypergraphs and Applications. *Discr. Appl. Math.*, 42, 1993. (177-201).

[7] E. Gradel, W. Thomas, and T. (Eds) Wilke. *Automata logics, and Infinite Games*. Springer, 2002. LNCS 2500.

[8] V. A. Gurvich, A. V. Karzanov, and L. G. Khachiyan. Cyclic Games and an Algorithm to Find Minimax Cycle Means in Directed Graphs. *USSR Comput. Math. Math. Phys.*, 28, 1988. (85-91).

[9] A. Miné. A New Numerical Abstract Domain Based on Diffrence-Bound Matrices. *LNCS*, 2053, 2001. (155-172).

[10] R. H. Mohring, M. Skutella, and F. Stork. Scheduling with AND/OR Precedence Constraints. *SIAM Journ. on Comp.*, 33(2), 2004. (393-415).

[11] R. Nieuwenhuis, A. Oliveras, and C. Tinelli. Solving SAT and SAT Modulo Theories from an Abstract Davis-Putman-Logemann-Loveland Procedure to DPLL(T). *Journal of the ACM*, 53(6), 2006. (937-977).

[12] U. Zwick and M. Paterson. The Complexity of Mean Payoff Games on Graphs. *Theor. Comp. Science*, 158, 1996. (343-359).