Lecture Notes in Networks and Systems 1068

Kohei Arai *Editor*

Intelligent Systems and Applications

Proceedings of the 2024 Intelligent Systems Conference (IntelliSys)
Volume 4

Springer

Lecture Notes in Networks and Systems 1068

Kohei Arai
Editor

Intelligent Systems and Applications

Proceedings of the 2024 Intelligent Systems Conference (IntelliSys) Volume 4

Springer

Editor
Kohei Arai
Faculty of Science and Engineering
Saga University
Saga, Japan

ISSN 2367-3370 ISSN 2367-3389 (electronic)
Lecture Notes in Networks and Systems
ISBN 978-3-031-66335-2 ISBN 978-3-031-66336-9 (eBook)
https://doi.org/10.1007/978-3-031-66336-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

MathPartner: An Artificial Intelligence Cloud Service

Gennadi Malaschonok[1](✉) and Alexandr Seliverstov[2]

[1] National University of Kyiv-Mohyla Academy, 04070 Kyiv, Ukraine
malaschonok@ukma.edu.ua

[2] Institute for Information Transmission Problems, Russian Academy of Sciences (Kharkevich Institute), Bolshoi Karetny per. 19, 127994 Moscow, Russia
https://www.ukma.edu.ua/eng/

Abstract. In a broad sense, artificial intelligence is a service to find a solution to complex intellectual problems. In this sense, the MathPartner service provides artificial intelligence that allows us to formulate questions and receive answers to questions formulated in a mathematical language. For mathematicians and physicists today, such a language is LaTeX. The MathPartner service uses a dialect of LaTeX, which is called Mathpar. The service is a cloud-based computer algebra system and provides users with the opportunity to solve many mathematical problems. In this publication, we focus only on a small class of extremum problems, which are widely applied in economics, management, logistics, and in many engineering fields. In particular, we consider the shortest path problem and discuss an algorithm that is based on the tropical mathematics. The ability to work with many types of classical and tropical algebras, which are freely available to users, is an important distinguishing feature of this intelligent tool for symbolic-numerical calculations. We also consider the use of the simplex algorithm for solving optimization problems.

Keywords: Artificial intelligence · Tropical mathematics · Symbolic-numerical calculations · Cloud computing · Computer algebra · Cloud service · Mathpar · MathPartner

1 Introduction

Intelligent systems and artificial intelligence should at least have two properties: reflect the real world and interact with a person. If we talk about creating a universal intellectual system in the future, then it must have natural science knowledge, and primarily mathematical knowledge.

While humanity is still far from creating an artificial intelligence that would have mathematical knowledge, but such a task should already be formulated as one of the primary tasks. What is the traditional way of using mathematical knowledge? First, the problem is formulated in a mathematical language, and then the solution of the problem is found using known methods and algorithms.

K. Arai (Ed.): IntelliSys 2024, LNNS 1068, pp. 182–194, 2024.
https://doi.org/10.1007/978-3-031-66336-9_14

Therefore, the first step should be the creation of an intelligent system that is capable of solving traditional mathematical problems, if they are correctly formulated in traditional mathematical language.

Here are some simple examples. A weighted graph is given, two of its vertices are indicated, and it is required to find the shortest distance between these two vertices. A system of linear algebraic equations is given, a linear form is indicated, and it is required to find a solution to the system that maximizes (or minimizes) this linear form. A system of algebraic equations is given and it is required to find all its solutions in a specific numerical region. A system of differential equations is given and it is required to find all its solutions in a certain class of functions.

Of course, a program that can solve such problems is not artificial intelligence. But such a program is the main component without which it is impossible to create artificial intelligence with mathematical knowledge. Today, the most convenient form is the cloud form. The creation of such a cloud component is a very important step towards the creation of artificial intelligence.

We propose to use MathPartner cloud mathematics for this purpose. It is freely available [1]. To get acquainted with its capabilities, you can read the user manual as well as journal articles [2–4]. However, the possibilities of solving extreme problems were very little covered.

Therefore, the purpose of this article is to acquaint the reader with the capabilities that this service provides for solving extreme problems.

To solve extremal problems, you can use both classical and tropical algebras. Note that such algebras are freely available to users and this is one of the important advantages of this service.

Calculations over max-plus as well as min-plus semirings play an important role both in discrete optimization and for the study of systems of equations over the field of complex numbers. These semirings are known as tropical semirings and has been intensively studied in recent decades. In particular, they forms the basis of tropical geometry [5–12]. A new branch of linear algebra was also developed [13–19].

The shortest path problem has many applications in artificial intelligence and operations research. Classical algorithms such as the Dijkstra algorithm are known. This problem is effectively solvable over the min-plus semiring.

For both linear functional and polyhedron defined by linear functions with integer coefficients, the optimization problem can be reformulated over max-plus or min-plus semirings. Therefore, such problems about polyhedra provide a large set of problems related to tropical mathematics. Although in the MathPartner system, the simplex algorithm is implemented over ordinary fields. The average-case complexity of the algorithm is polynomial, but it is high in the worst case [20,21]. The simplex algorithm is widely used, although other methods are more efficient in the worst case [22–24]. Note that the simplex algorithm is sensitive to rounding off at intermediate computation steps [25]. For solving feasibility in linear programs with up to two variables per inequality, n variables, and m inequalities, there is an $O(mn^2 \log m)$ algorithm which is derived from the Fourier–Motzkin elimination method [26,27].

This paper is structured as follows. Section 2 provides basic definitions and theoretical background. Section 3 describes implementation of some tropical algorithms. Section 4 describes implementation of the simplex algorithm. Section 5 concludes the paper.

2 Basic Definitions and Theoretical Background

2.1 The Max-plus Semiring

As usual, tropical algebra refers to the max-plus semiring, i.e., the set $\mathbb{R} \cup \{-\infty\}$ equipped with the addition $x \oplus y = \max(x, y)$ as well as the multiplication $x \odot y = x + y$. For all x, both identities $\max(-\infty, x) = x$ and $-\infty + x = -\infty$ hold. Every $x \neq -\infty$ has the multiplicative inverse $-x$. So, the algebra is known as a semifield [13].

The matrix multiplication over the max-plus semiring is defined as usual. The multiplication symbol between matrices is omitted. For example,

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \max(A_{11} + x_1, A_{12} + x_2) \\ \max(A_{21} + x_1, A_{22} + x_2) \end{pmatrix}.$$

The max() operation acts on matrices of the same size entry-wise, i.e.,

$$\max(A, B)_{jk} = \max(A_{jk}, B_{jk}).$$

Definition 1. Given a $n \times n$ matrix A over the max-plus semiring. Its pseudo-inverse matrix A^- is a $n \times n$ matrix with entries

$$A^-_{jk} = \begin{cases} -A_{kj}, \; A_{kj} \neq -\infty \\ -\infty, \; A_{kj} = -\infty \end{cases}$$

Of course, $(A^-)^- = A$. The pseudo-inverse matrix is defined for every square matrix. It can be calculated easier than the inverse matrix over a field. Nevertheless, it allows to solve tropical equations.

Let $\operatorname{diag}(d_1, \ldots, d_n)$ denote the matrix with entries

$$D_{jk} = \begin{cases} d_k, \; j = k \\ -\infty, \; j \neq k \end{cases}$$

and $I = \operatorname{diag}(0, \ldots, 0)$.

Remark 1. If $A = \operatorname{diag}(d_1, \ldots, d_n)$, where all $d_k \neq -\infty$, then $AA^- = I$, i.e., A^- is the inverse matrix.

Proposition 1. *For a $n \times n$ matrix A and a column b, every column $x \leq (b^- A)^-$ is a particular solution to the inequality $Ax \leq b$. Moreover, the upper bound on the computation complexity equals $O(n^2)$ operations over the max-plus semiring.*

Proof. If $x \le (b^- A)^-$, then for every index k the inequality $x_k \le (\max_j(b_j^- + A_{jk}))^-$ holds. So, $x_k \le \min_j(b_j + A_{jk}^-)$. Next, for all k and j, $x_k \le b_j + A_{jk}^-$ and $A_{jk} + x_k \le b_j$. So, for every index j, $\max_k(A_{jk} + x_k) \le b_j$. Thus, $Ax \le b$.

Proposition 2. *For a $n \times n$ matrix A and a column b, if there exists a solution to the equation $Ax = b$, then a particular solution can be calculated as $x = (b^- A)^-$. Moreover, the upper bound on the computation complexity equals $O(n^2)$ operations over the max-plus semiring.*

Remark 2. For a sparse matrix A, a particular solution to the equation $Ax = b$ can be calculated quicker.

There are many types of equations because the additive operation is not invertible. An equation of the type $Ax = x$ is known as the homogeneous Bellman equation. For a column b, an inhomogeneous equation is one of the type $\max(Ax, b) = x$.

Definition 2. A partial operation $()^\times$ is called the closure when both identities hold:

1. If $\max(X, Y) = Y$ and both $X^\times$ and $Y^\times$ exist, then $\max(X^\times, Y^\times) = Y^\times$.
2. $X^\times = \max(I, XX^\times) = \max(I, X^\times X)$.

Proposition 3. *Given both $n \times n$ matrices A and B over max-plus semiring. If the closure $A^\times$ exists, then the matrix $X = A^\times B$ is a solution to the matrix Bellman equation* $\max(AX, B) = X$.

Proof. $\max(AX, B) = \max(AA^\times B, B) = \max(AA^\times, I)B = A^\times B = X$.

The closure can be calculated as $A^\times = \max(I, A, A^2, \dots)$. For elements of the max-plus semiring, the closure is

$$x^\times = \begin{cases} 0, & x \le 0 \\ \text{undefined}, & x > 0 \end{cases}$$

Moreover, for $n \times n$ matrices, if the closure exists, then it is equal to finite expression $A^\times = \max(I, A, \dots, A^{n-1})$.

The next proposition provides a block-recursive algorithm to calculate the closure of a matrix.

Proposition 4. *Given a block matrix over the max-plus semiring*

$$A = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

If both $E^\times$ and $D^\times$ exist, where $D = \max(H, GE^\times F)$, then the closure $A^\times$ can be found using closures of the blocks:

$$A^\times = \begin{pmatrix} E^\times \max(I, FD^\times GE^\times) & E^\times FD^\times \\ D^\times GE^\times & D^\times \end{pmatrix}.$$

Proof. Let $-\infty$ denotes the matrix whose entries are equal to $-\infty$. First of all, it is easy to check that the block expression for $A^\times$ has the following factorization:

$$\begin{pmatrix} E^\times \max(I, FD^\times GE^\times) & E^\times FD^\times \\ D^\times GE^\times & D^\times \end{pmatrix} =$$

$$\begin{pmatrix} I & E^\times F \\ -\infty & I \end{pmatrix} \begin{pmatrix} I & -\infty \\ -\infty & D^\times \end{pmatrix} \begin{pmatrix} I & -\infty \\ G & I \end{pmatrix} \begin{pmatrix} E^\times & -\infty \\ -\infty & I \end{pmatrix}.$$

Let us prove the identity $A^\times = \max(I, A^\times A)$. Consider the product of the first two matrices and matrix A:

$$\begin{pmatrix} I & -\infty \\ G & I \end{pmatrix} \begin{pmatrix} E^\times & -\infty \\ -\infty & I \end{pmatrix} \begin{pmatrix} E & F \\ G & H \end{pmatrix} = \begin{pmatrix} E^\times E & E^\times F \\ GE^\times & D \end{pmatrix}.$$

It follows from here that

$$A^\times A = \begin{pmatrix} I & E^\times F \\ -\infty & I \end{pmatrix} \begin{pmatrix} I & -\infty \\ -\infty & D^\times \end{pmatrix} \begin{pmatrix} E^\times E & E^\times F \\ GE^\times & D \end{pmatrix}$$

$$= \begin{pmatrix} \max(E^\times E, E^\times FD^\times GE^\times) & E^\times FD^\times \\ D^\times GE^\times & D^\times D \end{pmatrix}.$$

Therefor

$$\max(I, A^\times A) = \max\left(\begin{pmatrix} I & -\infty \\ -\infty & I \end{pmatrix}, \begin{pmatrix} \max(E^\times E, E^\times FD^\times GE^\times) & E^\times FD^\times \\ D^\times GE^\times & D^\times D \end{pmatrix} \right)$$

$$= \begin{pmatrix} E^\times \max(I, FD^\times GE^\times) & E^\times FD^\times \\ D^\times GE^\times & D^\times \end{pmatrix}.$$

The identity $A^\times = \max(I, AA^\times)$ can also be proved similarly.

2.2 The Min-plus Semiring

In the same way one can consider the min-plus semiring $\mathbb{R} \cup \{+\infty\}$ with addition $x \oplus y = \min(x, y)$ as well as multiplication $x \odot y = x + y$. For all x, both identities $\min(+\infty, x) = x$ and $+\infty + x = +\infty$ hold. Every $x \neq +\infty$ has the multiplicative inverse $-x$. Of course, in all definitions max() is replaced by min() as well as $-\infty$ is replaced by $+\infty$. For example, the pseudo-inverse matrix A^- over the min-plus semiring is a matrix with entries

$$A^-_{jk} = \begin{cases} -A_{kj}, & A_{kj} \neq +\infty \\ +\infty, & A_{kj} = +\infty \end{cases}$$

The diagonal matrix $\mathrm{diag}(d_1, \ldots, d_n)$ has entries

$$D_{jk} = \begin{cases} d_k, & j = k \\ +\infty, & j \neq k \end{cases}$$

The closure is defined as $A^{\times} = \min(I, A, A^2, \dots)$.

Note that the order over the min-plus semiring is reversed.

Both max-plus and min-plus semirings are tangled by the sister identities $(\max(A, B))^{-} = \min(A^{-}, B^{-})$ and $(\min(A, B))^{-} = \max(A^{-}, B^{-})$.

The shortest path problem is the problem of finding a path between two vertices in a graph such that the sum of the weights of its constituent edges is minimized. Let the weights be entries of the matrix A. If all weights are nonnegative, then the problem can be reduced to calculating the closure $A^{\times}$ over the min-plus semiring. In fact, it uses only finite sequence of matrices in the equation $A^{\times} = \min(I, A, A^2, \dots)$ because the length of the shortest path is bounded. Of course, if all weights are nonpositive, then one can find the maximum using the max-plus semiring, but this formulation of the problem is not usually used. For nonnegative weights, the travelling salesman problem can also be expressed over the max-plus semiring, but its computational complexity seems to be very high because closely related the Hamiltonian path problem is NP-complete.

2.3 The Algorithm for Calculating the Matrix Closure

We formulate the algorithm for calculating the matrix closure. Let us designate:

- M is an input matrix whose size n is a power of 2;
- `ring` is an algebra type (one of the tropical semirings or classical fields);
- `Split`() is procedure for dividing a matrix into four equal blocks;
- `Join`() is a procedure for joining four blocks into one matrix;
- `Inverse`() is the matrix inversion procedure over a field;
- I is the identity matrix;
- both $+$ and $*$ are basic operations over a (semi)ring.

```
closure(M, ring) {
  if (ring is tropical)
  then
   if(M.size=1) AND (M=[[a]])
   then{b=closure(a);return [[b]]}
   else{
        (E,F,G,H) = Split(M):
        S = closure(E,ring);
        B = G*S;
        R4 = closure(H+B*F, ring);
        R3 = R4+B;
        V = S*F ;
        R2 = V*R4;
        R1 = S+V*R3;
        return Join(R1, R2, R3, R4);
   }
  else return Inverse(I-M);
}
```

In contrast to matrices over a field, there is not known fast algorithm for matrix multiplication over tropical semirings similar to the Strassen algorithm [28,29] because the additive operation is not invertible.

Let γ and β be constants, $3 \geq \beta > 2$, and let $\mathcal{M}(n) = \gamma n^\beta + o(n^\beta)$ be the number of arithmetic operations in one $n \times n$ matrix multiplication. We know today algorithm with $\gamma = 1$ and $\beta = 3$.

Proposition 5. *The closure algorithm has the complexity of matrix multiplication.*

Proof. For calculating the closure of the $n \times n$ matrix we have to compute five matrix multiplications and two matrix closure for the $(n/2) \times (n/2)$ blocks.

So, we get the following recurrent equality for complexity

$$t(n) = 2t(n/2) + 5\mathcal{M}(n/2), t(1) = 1.$$

After summation from $n = 2^k$ to 2^1 we obtain

$$5\gamma \left(2^0 2^{\beta(k-1)} + \cdots + 2^{k-1} 2^{\beta \cdot 0} + 2^k\right) t(1)$$

$$= n + \frac{5\gamma n}{2} \sum_{i=0}^{k-1} 2^{i(\beta-1)} = 5\gamma \frac{n^\beta - n}{2^\beta - 2} + n.$$

Therefore the complexity of the closure algorithm is $\sim \frac{5\gamma n^\beta}{2^\beta - 2}$. For $\gamma = 1$ and $\beta = 3$, we get $\sim \frac{5}{6} n^3$.

3 Tropical Mathematics in MathPartner

3.1 Algebras in MathPartner

Working in the MathPartner computer algebra system involves choosing the algebra over which the calculations are performed. The default algebra is the algebra $R64[x, y, z, t]$ of polynomials in four variables with the monomial order, where inequalities $x < y < z < t$ hold. $R64$ is the set of floating-point 64-bit numbers having 52 bits for mantissa, 11 bits for the exponent and one bit for the sign. Instead of floating-point numbers $R64$, one can use integers Z, rational numbers Q, floating-point numbers of arbitrary precision R, complex floating-point numbers $C64$ and C, modulo prime residues Zp and $Zp32$, as well as idempotent algebras, including *ZMaxPlus*, *ZMinPlus*, *RMaxPlus*, *RMinPlus*, *R64MaxPlus*, and *R64MinPlus*. The algebra can be chosen with a command like `SPACE = ZMaxPlus[]`. Many commands only work in the proper algebra.

The additive operation is always denoted by +, and the multiplicative operation is denoted by $*$.

Example 1. Let us run `SPACE = ZMaxPlus[]; 2 + 3`; the output is $3 = \max(2, 3)$. On the other hand, let us run `SPACE = ZMaxPlus[]; 2 * 3`; the output is 5.

3.2 Simple Equations and Inequalities

For given both matrix A and vector b, to find a particular solution to the system of linear algebraic equations $Ax = b$ over a tropical semiring one can run $\backslash\texttt{solveLAETropic}(A, b)$.

Example 2. Let us consider the matrix

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 0 \end{pmatrix}$$

and the column

$$b = \begin{pmatrix} 5 \\ 7 \end{pmatrix}.$$

Let us run

$$\texttt{SPACE} = \texttt{ZMaxPlus}[\,]; \texttt{A} = [[\texttt{1}, \texttt{2}], [\texttt{3}, \texttt{0}]]; \texttt{b} = [\texttt{5}, \texttt{7}];$$

$$\backslash\texttt{solveLAETropic}(\texttt{A}, \texttt{b});$$

The output is equal to

$$\begin{pmatrix} 4 \\ 3 \end{pmatrix}.$$

In the same way, to solve the system of linear algebraic inequalities one can run $\backslash\texttt{solveLAITropic}(A, b)$.

Example 3. Let us run

$$\texttt{SPACE} = \texttt{ZMaxPlus}[\,]; \texttt{A} = [[\texttt{2}, \texttt{0}], [\texttt{3}, \texttt{1}]]; \texttt{b} = [\texttt{1}, \texttt{1}];$$

$$\backslash\texttt{solveLAITropic}(\texttt{A}, \texttt{b});$$

The output is equal to $[[-\infty, -2], [-\infty, 0]]$.

3.3 The Closure

The closure $A^{\times}$ can be calculated by the $\backslash\texttt{closure}()$ operation. For numbers, if the closure does not exists, then the output is ∞.

Example 4. Let us run $\texttt{SPACE} = \texttt{ZMaxPlus}[\,]; \backslash\texttt{closure}(\texttt{1});$ the output is ∞. On the other hand, let us run $\texttt{SPACE} = \texttt{ZMaxPlus}[\,]; \backslash\texttt{closure}(-\texttt{1});$ the output is 0.

Example 5. Let us calculate the closure of the matrix

$$A = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$$

over the min-plus semiring.

$$\texttt{SPACE} = \texttt{ZMinPlus}[\,]; \texttt{A} = [[\texttt{0}, \texttt{1}], [\texttt{2}, \texttt{0}]]; \backslash\texttt{closure}(\texttt{A});$$

The output is equal to

$$A^{\times} = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$$

Example 6. Let us calculate the closure of the matrix

$$A = \begin{pmatrix} -1 & -2 \\ -3 & -4 \end{pmatrix}$$

over the max-plus semiring.

$$\texttt{SPACE} = \texttt{ZMaxPlus}[]; \texttt{A} = [[-1, -2], [-3, -4]]; \backslash\texttt{closure}(\texttt{A});$$

The output is equal to

$$A^{\times} = \begin{pmatrix} 0 & -2 \\ -3 & 0 \end{pmatrix}$$

3.4 The Bellman Equations and Inequalities

To find a solution to a homogeneous Bellman equation $Ax = x$ one can run \BellmanEquation(A). For a column b, to find a solution to an inhomogeneous Bellman equation one can run \BellmanEquation(A, b).

Two commands \BellmanInequality(A) and \BellmanInequality(A, b) allows to find a solution to Bellman inequalities $Ax \leq x$ and $\max(Ax, b) \leq x$, respectively.

3.5 The Shortest Path

Let A be a matrix with nonnegative or infinite entries, where all entries at the leading diagonal vanish, i.e., for all k, $x_{kk} = 0$. Its entry can be interpreted as the graph edge weight or length. However, the triangle inequality can be violated. If there is no edge between two vertices k and j, then $x_{kj} = \infty$. To find the shortest path from k to j one can run \findTheShortestPath(A, k, j) over the min-plus algebra. To calculate the smallest distances for all vertex pairs, one can run \searchLeastDistances(A).

Example 7. Let us consider the graph with the adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & \infty \\ 1 & 0 & 1 \\ \infty & 1 & 0 \end{pmatrix},$$

i.e., the graph is a path. Let us run

$$\texttt{SPACE} = \texttt{ZMinPlus}[]; \texttt{A} = [[0, 1, \infty], [1, 0, 1], [\infty, 1, 0]];$$

$$\backslash\texttt{searchLeastDistances}(\texttt{A});$$

The output is equal to

$$\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

Next, let us run \findTheShortestPath($A, 2, 1$). The output is equal to $[2, 1]$, that is, the shortest path from 2 to 1 consist of two vertices 2 and 1.

4 Linear Programming in MathPartner

4.1 Systems of Algebraic Inequalities

To calculate the solution to a system of univariate algebraic inequalities one can run `\solve()` over a ring of polynomials.

Example 8. To solve the system of two inequalities $x - 6 > 0$ and $x - 7 < 0$, let us run

```
SPACE = Q[x]; b = \solve([x - 6 > 0, x - 7 < 0]);
```

The output is equal to the interval $(6, 7)$. Note that parentheses denote open interval boundaries, and square brackets denote closed interval boundaries.

4.2 The Simplex Algorithm

To solve a linear programming problem, one can execute either the `\SimplexMax()` command or its counterpart that is the `\SimplexMin()` command. The result is a vector x. Calculations should be carried out over either rational or floating-point numbers.

If the objective function $c^T x$ is to be maximized ($c^T x \to \max$) under the conditions $Ax \leq b$ and $x \geq 0$, then one can run `\SimplexMax`(A, b, c).

Example 9. Let us maximize the $3x_1 + x_2 + 2x_3$ under the conditions $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$, $x_1 + x_2 + 3x_3 \leq 30$, $2x_1 + 2x_2 + 5x_3 \leq 24$, and $4x_1 + x_2 + 2x_3 \leq 36$. Let us run

$$\begin{aligned} &SPACE = R64[];\\ &A = [[1,1,3],[2,2,5],[4,1,2]];\\ &b = [30,24,36];\\ &c = [3,1,2];\\ &x = \texttt{\textbackslash SimplexMax}(A,b,c); \end{aligned}$$

The output is $[8, 4, 0]^T$.

Of course, only symbolic calculations over rational numbers without rounding guarantee the answer according to well-known results [25].

If the objective function $c^T x$ is to be maximized ($c^T x \to \max$) under the conditions $A_1 x \leq b_1$, $A_2 x = b_2$, and $x \geq 0$, then one can run

$$\texttt{\textbackslash SimplexMax}(A_1, A_2, b_1, b_2, c).$$

If the objective function $c^T x$ is to be maximized ($c^T x \to \max$) under the conditions $A_1 x \leq b_1$, $A_2 x = b_2$, $A_3 x \geq b_3$, and $x \geq 0$, then one can run

$$\texttt{\textbackslash SimplexMax}(A_1, A_2, A_3, b_1, b_2, b_3, c).$$

Some matrices can be empty (). If the objective function $c^T x$ is to be minimized ($c^T x \to \min$), then one can run either `\SimplexMin`(A, b, c) or

$$\texttt{\textbackslash SimplexMin}(A_1, A_2, b_1, b_2, c)$$

or

$$\texttt{\textbackslash SimplexMin}(A_1, A_2, A_3, b_1, b_2, b_3, c),$$

respectively.

5 Discussion

We do not give a detailed description of the MathPartner, referring the reader to other sources and manuals, but we dwell in detail on the issue of solving extreme problems.

But still, we note that MathPartner can solve a large class of classical symbolic and numerical problems in mathematical analysis, algebra, graph theory, the theory of differential equations and much more, including the construction of graphics, the depiction of three-dimensional figures, the animation of images, etc.

For example, MathPartner can solve many matrix problems for number fields and polynomial rings. It allows you to factor matrices, find the Bruhat decomposition, LSU, QR, SVD, Cholesky decomposition. The list of matrix functions includes calculation of the inverse and adjoint matrix, calculation of the kernel of a linear operator, Moore-Penrose inverse matrix and pseudo-inverse matrix, calculation of the determinant and echelon form of the matrix, calculation of the characteristic polynomial and others.

For detailed information, we recommend user manual http://mathpar.com/downloads/MathparHandbook_en.pdf.

6 Conclusion

Artificial intelligence, in the form in which it is already available for experimentation today, makes it possible to solve a large class of problems related to text and image processing. But it remains completely helpless when faced with even the simplest problems in the field of mathematics or physics.

We formulate the problem of creating artificial intelligence that can solve mathematical problems and propose to consider the MathPartner service as a possible basis for its creation.

In fact, we can talk about AI, which will have three global levels. The first level is text processing and isolating those fragments that formulate a mathematical problem. The second level is the reformulation of the problem into language Mathpar. And the third level is the solution of the problem using Cloud Mathematics MathPartner. When the solution is obtained at the third level, its text and appearance are already familiar to specialists. It will be enough to return it to the first level and add the necessary text environment.[1]

Acknowledgment. My[1] work was supported by the grants from the Simons Foundation. Awards ID: 1030285 and 1290592. My deepest sympathies go to the family of Jim Simons, a great man and brilliant mathematician, founder of this foundation, who passed away this year on May 10.

[1] Gennadi Malaschonok.

References

1. MathPartner. https://mathpar.ukma.edu.ua/ or https://mathpartner.com
2. Malaschonok, G.I.: MathPartner computer algebra. Programming and Computer Software 43(2), 112–118 (2017). doi: https://doi.org/10.1134/S0361768817020086 (arxiv.org/abs/2204.11549)
3. Malaschonok, G.I., Seliverstov, A.V.: Calculation of integrals in MathPartner. Discrete and Continuous Models and Applied Computational Science 29(4), 337–346 (2021). doi: https://doi.org/10.22363/2658-4670-2021-29-4-337-346 (arxiv.org/abs/2204.11061)
4. Malaschonok, G.I., Seliverstov, A.V.: New features in MathPartner 2021. Computer tools in education, 3, 29-40 (2021). (arxiv.org/abs/2204.11118)
5. Len, Y., Markwig, H.: Lifting tropical bitangents. J. Symb. Comput. **96**, 122–152 (2020). https://doi.org/10.1016/j.jsc.2019.02.015
6. Grigoriev, D.: Decomposing tropical rational functions. J. Symb. Comput. **101**, 61–72 (2020). https://doi.org/10.1016/j.jsc.2019.06.007
7. Grigoriev, D., Podolskii, V.V.: Tropical combinatorial Nullstellensatz and sparse polynomials. Found. Comput. Math. **20**, 753–781 (2020). https://doi.org/10.1007/s10208-019-09431-1
8. Vaccon, T., Verron, T., Yokoyama, K.: On affine tropical F5 algorithms. J. Symb. Comput. **102**, 132–152 (2021). https://doi.org/10.1016/j.jsc.2019.10.012
9. Grigoriev, D.: Entropy of tropical holonomic sequences. J. Symb. Comput. **108**, 91–97 (2022). https://doi.org/10.1016/j.jsc.2021.06.004
10. Grigoriev, D., Radchenko, D.: On a tropical version of the Jacobian conjecture. J. Symb. Comput. **109**, 399–403 (2022). https://doi.org/10.1016/j.jsc.2020.07.012
11. Nishida, Y., Yamada, A., Watanabe, Y.: Characterization of tropical projective quadratic plane curves in terms of the eigenvalue problem. Linear Algebra Appl. **644**, 172–191 (2022). https://doi.org/10.1016/j.laa.2022.03.004
12. Yoshida, R., Cox, S.: Tree topologies along a tropical line segment. Vietnam J. Math. **50**, 395–419 (2022). https://doi.org/10.1007/s10013-021-00526-3
13. Litvinov, G.L., Rodionov, A.Ya., Sergeev, S., Sobolevski, A.N.: Universal algorithms for solving discrete stationary Bellman equations (in Russian). Russian Universities Reports. Mathematics 24(128), 393–431 (2019). doi: https://doi.org/10.20310/2686-9667-2019-24-128-393-431
14. Friedland, S., Gaubert, S.: Spectral inequalities for nonnegative tensors and their tropical analogues. Vietnam J. Math. **48**, 893–928 (2020). https://doi.org/10.1007/s10013-020-00432-0
15. Dolzan, D., Oblak, P.: Cholesky decomposition of matrices over commutative semirings. Linear and Multilinear Algebra **68**(5), 1057–1063 (2020). https://doi.org/10.1080/03081087.2018.1529137
16. Watanabe, S., Fukuda, A., Segawa, E., Sato, I.: A walk on max-plus algebra. Linear Algebra Appl. **598**, 29–48 (2020). https://doi.org/10.1016/j.laa.2020.03.025
17. Jones, D.: Matrix roots in the max-plus algebra. Linear Algebra Appl. **631**, 10–34 (2021). https://doi.org/10.1016/j.laa.2021.08.008
18. Xie, Y.: On 2×2 tropical commuting matrices. Linear Algebra Appl. **620**, 92–108 (2021). https://doi.org/10.1016/j.laa.2021.02.024
19. Yang, L., Yang, S.L.: On generalized inverses of $m \times n$ matrices over a pseudoring. Vietnam J. Math. **50**, 261–274 (2022). https://doi.org/10.1007/s10013-021-00502-x

20. Smale, S.: On the average number of steps of the simplex method of linear programming. Math. Program. **27**(3), 241–262 (1983). https://doi.org/10.1007/BF02591902
21. Vershik, A.M., Sporyshev, P.V.: An estimate of the average number of steps in the simplex method, and problems in asymptotic integral geometry. Sov. Math. Dokl. 28, 195–199 (1983). https://zbmath.org/?q=an:0532.90062
22. Stetsyuk, P.I., Fesiuk, O.V., Khomyak, O.N.: The generalized ellipsoid method. Cybern. Syst. Anal. **54**(4), 576–584 (2018). https://doi.org/10.1007/s10559-018-0058-4
23. Cohen, M.B., Lee, Y.T., Song, Z.: Solving linear programs in the current matrix multiplication time. J. ACM **68**(1), 3 (2021). https://doi.org/10.1145/3424305
24. Chubanov, S.: A scaling algorithm for optimizing arbitrary functions over vertices of polytopes. Math. Program. **190**, 89–102 (2021). https://doi.org/10.1007/s10107-020-01522-0
25. Ogryczak, W.: The simplex method is not always well behaved. Linear Algebra Appl. **109**, 41–57 (1988). https://doi.org/10.1016/0024-3795(88)90197-8
26. Hochbaum, D.S., Naor, J.: Simple and fast algorithms for linear and integer programs with two variables per inequality. SIAM J. Comput. **23**(6), 1179–1192 (1994). https://doi.org/10.1137/S0097539793251876
27. Subramani, K., Wojciechowski, P.: On the parametrized complexity of read-once refutations in UTVPI+ constraint systems. Theoret. Comput. Sci. **883**, 1–18 (2021). https://doi.org/10.1016/j.tcs.2021.05.007
28. Strassen, V.: Gaussian elimination is not optimal. Numer. Math. **13**(4), 354–356 (1969). https://doi.org/10.1007/BF02165411
29. Karstadt, E., Schwartz, O.: Matrix multiplication, a little faster. J. ACM **67**(1), 1 (2020). https://doi.org/10.1145/3364504