Supporting education in algorithms of computational mathematics by dynamic visualizations using computer algebra system

Włodzimierz Wojas[1] [0000−0002−1194−7342] and Jan Krupa(✉)[1] [0000−0001−5967−6417]

Warsaw University of Life Sciences (SGGW), Department of Applied Mathematics, ul. Nowoursynowska 159, 02-776 Warsaw, Poland {wlodzimierz_wojas,jan_krupa}@sggw.pl

Abstract. Computer algebra systems (CAS) are often used programs in universities to support calculations and visualization in teaching mathematical subjects. In this paper we present some examples of dynamic visualizations which we prepared for students of Warsaw University of Life Sciences using Mathematica. Visualizations for simplex algorithm and Karush-Kuhn-Tucker algorithm are presented. We also describe a didactic experiment for students of the Production Engineering Faculty of Warsaw University of Life Sciences using dynamic visualization of the network flow problem.

Keywords: computational mathematics · symbolic calculations · mathematical didactics · linear programming · nonlinear programming · network flows · CAS
Subject Classifications:97B40, 97R20; 90C30; 97I60

1 Introduction

The development of computational mathematics techniques, technologies and tools in recent decades has created new educational challenges in teaching mathematics and IT subjects in higher education. On the other hand, the development of computational mathematics technology has created new educational perspectives based on the possibility of using new computational and visualization tools in education. CAS such as wxMaxima, Mathematica, Maple, Sage, and others are often used to support calculations and visualization in teaching mathematical subjects [6,7,10,11,9]. In teaching mathematical algorithms in the field of mathematical analysis, mathematical programming or graph theory, the possibility of symbolic calculations and visualizing the algorithm steps seems useful from an educational point of view. This allows a better and deeper understanding of the topic. In this paper we present two examples of dynamic visualizations which we prepared for students of the Faculty of Informatics and Econometric of Warsaw University of Life Sciences using Mathematica [8,12] within the course of Mathematical Programming. We present dynamic visualizations for simplex algorithm for linear programming (LP) problem [2,5] and for nonlinear programming (NLP) problem which we solve using Karush-Kuhn-Tucker (KKT) algorithm [3,5]. A didactic experiment for students of the Production Engineering Faculty of Warsaw University of Life Sciences using dynamic visualization of the network flow problem will also be discussed.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_47

2 Visualization of primal simplex algorithm steps – 3D example

In this example the authors present some new complex approach to visualization of simplex method steps.

The authors propose to use for visualization expanded Simplex Tableau which contains for each simplex step: current simplex table for this step, graph of feasible region for canonical form of LP problem with current corner point and „simplex path", level sets of goal function (hyperplanes : lines in $2D$, planes in $3D$), axis with current value of objective function for this step.

Let us solve the following LP problem:

Example 1.

$$
\begin{array}{ll}
\text{Maximize} & z = 4x_1 + x_2 + 6x_3 \\
\text{Subject to} & \begin{array}{rrrcr}
x_1 & & & \leq & 9 \\
 & x_2 & & \leq & 5 \\
 & -\ x_2 & +5x_3 & \leq & 15 \\
-2x_1 & -\ 3x_2 & +3x_3 & \leq & 3 \\
5x_1 & -\ x_2 & +5x_3 & \leq & 45
\end{array} \\
 & x_i \geq 0 \text{ for } i = 1,2,3.
\end{array}
$$

Corresponding to it canonical form is:

$$
\begin{array}{ll}
\text{Maximize} & z = 3x_1 + 2x_2 \\
\text{Subject to} & \begin{array}{rrrrrrrrcr}
x_1 & & & +x_4 & & & & & = & 9 \\
 & x_2 & & & +x_5 & & & & = & 5 \\
 & -\ x_2 & +5x_3 & & & +x_6 & & & = & 15 \\
-2x_1 & -3x_2 & +3x_3 & & & & +x_7 & & = & 16 \\
5x_1 & -\ x_2 & +5x_3 & & & & & +x_8 & = & 45
\end{array} \\
 & x_i \geq 0 \text{ for } i = 1,2,\ldots,8
\end{array}
$$

Let S be feasible region for this LP problem (in standard form). S is presented in each Figure 1–5. It is convex polyhedral set with vertices at: $v_1 = (0,0,0), v_2 = (9,0,0), v_3 = (9,5,0), v_4 = (0,5,0), v_5 = (0,0,1), v_6 = (3,0,3), v_7 = (6,0,3), v_8 = (9,5,1), v_9 = (6,5,4), v_{10} = (0,5,4), v_{11} = (0,5/2,7/2)$. In each Figs. 1–5 we present expanded Simplex Tableaus for subsequent vertices of simplex path. The current vertex of simplex path and level set are in red and the previous ones are in blue. The dynamic versions of the Figs. 1–5 can be found in the Electronic supplementary material:
https://drive.google.com/open?id=1vgBC1ij7Z9mNL8_nmhVgrwi3qFRzzhYN

Fig. 1: First expanded Simplex Tableau.

Fig. 2: Second expanded Simplex Tableau.

Fig. 3: Third expanded Simplex Tableau.

Fig. 4: Forth expanded Simplex Tableau.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_47

Fig. 5: Fifth expanded Simplex Tableau.

Finally we get that $z_{\max} = z(6,5,4) = 53$.

The above example and also another $2D$ and $3D$ examples, were presented students of Informatics and Econometrics Faculty of Warsaw University of Life Science in the framework of the course of Mathematical Programming. These were introductory examples illustrating the simplex algorithm steps. Using the expanded Simplex Tableau allows students to trace the steps of simplex algorithm in quite comprehensive manner - taking into account both computational and geometric aspects of the algorithm. From Figures 2–4 we can see the the planes: $4x_1 + x_2 + 6x_6 = 6$, $4x_1 + x_2 + 6x_6 = 30$, $4x_1 + x_2 + 6x_6 = 42$ are not supporting planes of the feasible region for this LP problem S, but from Figure 5 we can see that the plane $4x_1 + x_2 + 6x_6 = 53$ is supporting plane of S (this can be better seen in the the dynamic versions of the Figs. 1–5 in the Electronic supplementary material). So we may conclude independently that $z_{\max} = z(6,5,4) = 53$. We cannot find analogical approach to present simplex algorithm steps in available literature.

3 Karush-Kuhn-Tucker necessary conditions

We will consider NLP problems in the following form:

$$\begin{array}{ll} \underset{(x_1,x_2,\ldots x_n)}{\text{minimize/maximize}} & f(x_1,x_2,\ldots,x_n) \\ \text{subject to:} & g_i(x_1,x_2,\ldots,x_n) \geq 0,\ i = 1,2,\ldots m, \\ & (x_1,x_2,\ldots,x_n) \in X, \end{array} \qquad \text{(P)}$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_47

where n and m are positive integers, X is a subset of $\mathbb{R}^n$ and f, g_i are real-valued functions on X with at least one function of f, g_i $(i = 1, 2, \ldots, m)$ being nonlinear. A feasible region of the NLP problem is defined as a set of all possible points that satisfy all the problem's constraints.

We start with the following theorem ([3]):

***Theorem* (Karush-Kuhn-Tucker Necessary Conditions).** *Let X be a nonempty open set in $\mathbb{R}^n$, and let $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $g_i : \mathbb{R}^n \mapsto \mathbb{R}$ for $i = 1, \ldots, m$. Consider the Problem (P) to minimize $f(x)$ subject to $g_i(x) \geq 0$ for $i = 1, \ldots, m$. Let $\bar{x}$ be a feasible solution, and denote $I = \{i : g_i(\bar{x}) = 0\}$. Suppose that f and g_i, for $i = 1, 2, \ldots m$ are differentiable at $\bar{x}$. Furthermore, suppose that $\nabla g_i(\bar{x})$ for $i \in I$ are linearly independent. If $\bar{x}$ solves Problem (P) locally, there exist scalars λ_i, for $i = 1, 2, \ldots m$ such that:*

$$\nabla f(\bar{x}) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\bar{x}) = 0 \tag{1}$$

$$\lambda_i g_i(\bar{x}) = 0 \quad \textit{for } i = 1, 2, \ldots m \tag{2}$$

$$\lambda_i \geq 0 \quad \textit{for } i = 1, 2, \ldots m. \tag{3}$$

Example 2. Using KKT necessary conditions we determine global minimum and maximum function $f(x,y) = y^2 - x$ subject to: $5 - x - y \geq 0, xy - 4 \geq 0, x - 1 \geq 0$ and next we visualise the solution graphically using dynamic plots.

Formally, KKT necessary conditions require an independent solution of two tasks - one for minimum and one for maximum. We first solve the NLP problem for minimum and next the NLP problem for maximum.

a) Consider the following problem:

$$\begin{array}{lll} \underset{(x,y)\in\mathbb{R}^2}{\text{minimize}} & y^2 - x & \\ \text{subject to:} & 5 - x - y & \geq 0, \\ & xy - 4 & \geq 0, \\ & x - 1 & \geq 0. \end{array} \tag{4}$$

Let $g_1(x,y) = 5 - x - y$, $g_2(x,y) = xy - 4$ and $g_3(x,y) = x - 1$.
$\nabla f(x,y) = \lambda_1 \nabla g_1(x,y) + \lambda_2 \nabla g_2(x,y) + \lambda_3 \nabla g_3(x,y)$ is equivalent to:
$[-1, 2y] = \lambda_1[-1,-1] + \lambda_2[y,x] + \lambda_3[1,0]$. So we must solve the following system:

$$\begin{cases} -1 = -\lambda_1 + \lambda_2 y + \lambda_3 \\ 2y = -\lambda_1 + \lambda_2 x \\ 5 - x - y \geq 0 \\ xy - 4 \geq 0 \\ x - 1 \geq 0 \\ \lambda_1(5 - x - y) = 0 \\ \lambda_2(xy - 4) = 0 \\ \lambda_3(x - 1) = 0 \end{cases} \tag{5}$$

where $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, $\lambda_3 \geq 0$.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_47

Case 1. Let $g_1(x,y) = 0$, $g_2(x,y) = 0$ and $g_3(x,y) = 0$. Hence $5 - x - y = 0$, $xy - 4 = 0$ and $x = 1$. So $x = 1$ and $y = 4$ hence $-1 = -\lambda_1 + 4\lambda_2 + \lambda_3$ and $8 = -\lambda_1 + \lambda_2$, hence $-9 = 3\lambda_2 + \lambda_3$ contradicts nonnegativity of λ_2, λ_3.

Case 2. Let $g_1(x,y) = 0$, $g_2(x,y) = 0$ and $g_3(x,y) > 0$. Hence $5 - x - y = 0$ and $xy - 4 = 0$ and $\lambda_3 = 0$. So $x = 4$ and $y = 1$ hence $-1 = -\lambda_1 + \lambda_2$ and $2 = -\lambda_1 + 4\lambda_2$ hence $\lambda_2 = 1$ and $\lambda_1 = 2$. $\nabla g_1(4,1) = [-1,-1]$, $\nabla g_2(4,1) = [1,4]$ are linearly independent.

Case 3. Let $g_1(x,y) = 0$ and $g_2(x,y) > 0$ and $g_3(x,y) = 0$. Hence $\lambda_2 = 0$, $x = 1$ and $y = 4$, but $g_2(1,4) = 0$ contradicts assumption in current **Case 3** $g_2(x,y) > 0$.

Case 4. Let $g_1(x,y) = 0$ and $g_2(x,y) > 0$ and $g_3(x,y) > 0$. Hence $\lambda_2 = \lambda_3 = 0$ hence

$$\begin{cases} -1 = -\lambda_1 \\ 2y = -\lambda_1 \\ 5 - x - y = 0. \end{cases} \tag{6}$$

It follows that $\lambda_1 = 1$. Hence $y = -1/2$ and $x = 11/2$. $g_2(11/2, -1/2) = -11/4 - 4 < 0$ contradicts assumption in current **Case 4** $g_2(x,y) > 0$.

Case 5. Let $g_1(x,y) > 0$ and $g_2(x,y) = 0$ and $g_3(x,y) = 0$. Hence $\lambda_1 = 0$ and $x = 1$ and $y = 4$, hence $-1 = 4\lambda_2 + \lambda_3$ and $8 = \lambda_2$. So $\lambda_3 = -33 < 0$ contradicts nonnegativity of λ_3.

Case 6. Let $g_1(x,y) > 0$ and $g_2(x,y) = 0$ and $g_3(x,y) > 0$. Hence $\lambda_1 = \lambda_3 = 0$ hence $-1 = \lambda_2 y$ and $2y = \lambda_2 x$ hence $\lambda_2^2 = -\frac{1}{x} < 0$ contradiction because $x > 1$.

Case 7 and 8. Let $g_1(x,y) > 0$ and $g_2(x,y) > 0$ and $g_2(x,y) \geq 0$. Hence $\lambda_1 = \lambda_2 = 0$, hence $-1 = \lambda_3$ a contradiction.

From **Cases 1,2,3,...8** it follows that the only point which satisfies KKT conditions is $(4,1)$, $\lambda_1 = 2$, $\lambda_2 = 1$ and $\lambda_3 = 0$.

We solve this problem also using Mathematica procedures. We present the solution below:

Listing 1.1: Mathematica code for point **a)**:

```
In[1]:= (* min *)
 {f = y^2 - x, g2 = xy - 4, g1 = 5 - y - x, g3 = x - 1, F = f - λ1 * g1 - λ2 * g2 - λ3 * g3}
Out[1]= {-x + y^2, -4 + xy, 5 - x - y, -1 + x, -λ3(-1 + x) - x - λ1(5 - x - y) + y^2
    -λ2(-4 + xy)}
In[2]:= {F1 = D[F,x], F2 = D[F,y]}
Out[2]= {-1 + λ1 - λ3 - λ2y, λ1 - λ2x + 2y}
In[3]:= r1 =
 Solve[{F1 == 0, F2 == 0, λ1 * g1 == 0, λ2 * g2 == 0, λ3 * g3 == 0, λ1 >= 0, λ2 >= 0,
    λ3 >= 0, g1 >= 0, g2 >= 0, g3 >= 0}, {x, y, λ1, λ2, λ3}, Reals]
Out[3]= {{x-> 4, y-> 1, λ1-> 2, λ2-> 1, λ3-> 0}}
In[4]:= f/.r1
Out[4]= {-3}
In[5]:= Minimize[{y^2 - x, g1 >= 0, g2 >= 0, g3 >= 0}, {x, y}]
Out[5]= {-3, {x-> 4, y-> 1}}
```

b) Consider the following problem:

$$\begin{array}{lll} \underset{(x,y)\in\mathbb{R}^2}{\text{maximize}} & y^2 - x & \\ \text{subject to:} & 5-x-y & \geq 0, \\ & xy-4 & \geq 0, \\ & x-1 & \geq 0. \end{array} \tag{7}$$

Let $g_1(x,y) = -x^2 - 4y^2 + 1$, $g_2(x,y) = -1 + x + 2y$ and $g_3(x,y) = x - 1$. $-\nabla f(x,y) = \lambda_1 \nabla g_1(x,y) + \lambda_2 \nabla g_2(x,y) + \lambda_3 g_3(x,y)$ is equivalent to: $[1, -2y] = \lambda_1[-1,-1] + \lambda_2[y,x] + \lambda_3[1,0]$. So we must solve the following system:

$$\begin{cases} 1 = -\lambda_1 + \lambda_2 y + \lambda_3 \\ -2y = -\lambda_1 + \lambda_2 x \\ 5-x-y \geq 0 \\ xy-4 \geq 0 \\ x-1 \geq 0 \\ \lambda_1(5-x-y) = 0 \\ \lambda_2(xy-4) = 0 \\ \lambda_3(x-1) = 0 \end{cases} \tag{8}$$

where $\lambda_1 \geq 0$, $\lambda_2 \geq 0$ and $\lambda_3 \geq 0$.

Case 1. Let $g_1(x,y) = 0$ and $g_2(x,y) = 0$ and $g_3(x,y) = 0$. Hence $5-x-y = 0$ and $xy-4 = 0$ and $x = 1$. So $x = 1$ and $y = 4$ hence $1 = -\lambda_1 + 4\lambda_2 + \lambda_3$ and $-8 = -\lambda_1 + \lambda_2$. Let $\lambda_2 = \lambda_1 - 8$ hence $\lambda_3 = 1 + \lambda_1 - 4\lambda_1 + 32 = 33 - 3\lambda_1$ hence $8 \leq \lambda_1 \leq 11$. We can choose $\lambda_1 = 8$ than $\lambda_2 = 0$ and $\lambda_3 = 9$. $\nabla g_1(1,4) = [-1,-1]$, $\nabla g_3(1,4) = [1,0]$, are linearly independent.

Case 2. Let $g_1(x,y) = 0$ and $g_2(x,y) = 0$ and $g_3(x,y) > 0$. Hence $5-x-y = 0$ and $xy-4 = 0$ and $x > 1$ and $\lambda_3 = 0$. So $x = 4$ and $y = 1$ hence $1 = -\lambda_1 + \lambda_2$ and $-2 = -\lambda_1 + 4\lambda_2$ hence $\lambda_2 = -1 < 0$ contradicts nonnegativity of λ_2.

Case 3. Let $g_1(x,y) = 0$ and $g_2(x,y) > 0$ and $g_3(x,y) = 0$. Hence $\lambda_2 = 0$, and $x = 1$ hence

$$\begin{cases} 1 = -\lambda_1 + \lambda_3 \\ -2y = -\lambda_1 \\ 5-x-y = 0. \end{cases} \tag{9}$$

Hence $y = 4$ and $\lambda_1 = 8$ and hence $\lambda_3 = 9$ (**Case 1**).

Case 4. Let $g_1(x,y) > 0$ and $g_2(x,y) = 0$ and $g_3(x,y) = 0$. Hence $\lambda_1 = 0$ and $x = 1$ and $y = 4$, hence $1 = 4\lambda_2 + \lambda_3$ and $-8 = \lambda_2$- a contradiction.

Case 5. Let $g_1(x,y) = 0$ and $g_2(x,y) > 0$ and $g_3(x,y) > 0$. Hence $\lambda_2 = \lambda_3 = 0$ hence $\lambda_1 = -1$- a contradiction.

Case 6. Let $g_1(x,y) > 0$ and $g_2(x,y) = 0$ and $g_3(x,y) > 0$. Hence $\lambda_1 = \lambda_3 = 0$ hence $1 = \lambda_2 x$ and $-2y = \lambda_2 x$. λ_2 must be nonzero, so $y = \frac{1}{\lambda_2}$ hence $\lambda_2^2 = -\frac{2}{x} < 0$ because $x > 1$- a contradiction.

Case 7 and 8. Let $g_1(x,y) > 0$ and $g_2(x,y) > 0$ and $g_3(x,y) \geq 0$. Hence $\lambda_1 = \lambda_2 = 0$, hence $1 = \lambda_3$ and $y = 0$ hence $g_2(x,0) = -4 < 0$ contradicts assumption in current **Case 7 and 8** $g_2(x,y) > 0$.

From **Cases 1,2,..., 8** it follows that the only point which satisfies KKT conditions is $(1,4)$, $\lambda_1 = 8$, $\lambda_2 = 0$ and $\lambda_3 = 9$.

$f(1,4) = 15$, $f(4,1) = -3$.

The set D is compact and f is continuous on it so finally we get: the greatest value 15 and smallest value -3 of f on D are attained corresponding at points $(1,4)$ and $(4,1)$ respectively.

Fig. 6: Graphical method for **Example 2** using dynamic plots

In Figure 6 we present the solution of the above NLP problem graphically using dynamic plots. We solve the problem from **Example 2** using graphical method. In the Figure 6 we present Mathematica dynamic plot for the **Example 2**. The level sets is a family of parabolas $y^2 - x = c$.

The dynamic versions of the Fig. 6 can be found in the Electronic supplementary material:
https://drive.google.com/open?id=1vgBC1ij7Z9mNL8_nmhVgrwi3qFRzzhYN

Global maximum of the function $f(x,y) = y^2 - x$ at point $(1,4)$ we can determine graphically from the level set: $y^2 - x = 15$. Similarly, global minimum of the function f at point $(4,1)$ we can determine graphically from the level set: $y^2 - x = -3$.

This example and also another $2D$ and $3D$ examples, were dedicated students of Informatics and Econometrics Faculty of Warsaw University of Life Science within the course of Mathematical Programming. These were introductory examples illustrating the KKT method. Even relatively simple NLP problems with three or four constraints can lead to the need to consider many subcases, what is rather not suitable for hand calculations in class. The presented example shows that calculations are possible to do in class with support of computer programs such as Mathematica. Visualization presents graphical way of reaching the optimal solution of the NLP problem (for min and for max respectively) with graphical interpretation of KKT necessary conditions for **Example** 2 **a)**. In the Fig. 6 we see that: vector $\nabla f(4,1)$ is a linear combination of the vectors $\nabla g_1(4,1) = [-1,-1]$, $\nabla g_2(4,1) = [1,4]$ with coefficients $\lambda_1 = 2, \lambda_2 = 1$ ($\lambda_3 = 0$ and we see in Fig. 6 that the constraint $g_3 \geq 0$ is not active) which means that $\nabla f(4,1) = [-1,2] = 2[-1,-1] + 1[1,4]$. For graphical interpretation of KKT necessary conditions for **Example** 2 **b)** see Fig. 6 in Electronic supplementary material:
https://drive.google.com/open?id=1vgBC1ij7Z9mNL8_nmhVgrwi3qFRzzhYN.

4 The didactic experiment with dynamic visualization of network flows

This experiment was carried out in the form of a mathematics test on a group of 41 first-year students of the Production Engineering Faculty of Warsaw University of Life Sciences within the course of Higher Mathematics II. Students were after a semester course of Higher Mathematics I which included differential and integral calculus of functions of one variable. These students have not encountered the topic of network flows before. The experiment consisted of five parts. In the first part, students were introduced to the basic definitions of networks and flows in networks [1,4]). The following definitions were presented:

Definition 1. *A directed graph is a pair (N,A) where N is a finite set and A is a set of ordered pairs (v,w) such that $v,w \in N$. Elements of the set N we call nodes, and elements of the set A we call directed arcs.*

Most often directed graph is presented as a set of points representing nodes connected by arcs with arrows.

Definition 2. *A directed path in directed graph (N,A) from node v_1 to node v_2 is a sequence of nodes and arcs: $v_1 - k_1 - v_2 - k_2 - \ldots - v_{n-1} - k_{n-1} - v_n$ without any repetition of nodes where $k_i = (v_i, v_{i+1}) \in A$ for $i = 1,2,\ldots,n-1$ and $v_i \in A$ for $i = 1,2,\ldots,n$.*

Definition 3. *A pure network is a triple (N,A,u) such that (N,A) is a directed graph and $u : A \to [0,\infty)$ is a function of an upper bound of capacity.*

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_47

We define sets: $A(v) = \{(v,w) : w \in N, (v,w) \in A\}$, $B(v) = \{(w,v) : w \in N, (w,v) \in A\}$. For a pair of functions (F, f) such that: $F, f : A \to \mathbb{R}$, we define a divergence of node v by: $\text{div}_{(F,f)}(v) = \sum_{k\in A}(v)F(k) - \sum_{k\in B}(v)f(k)$. In the case of $F = f$ we denote: $\text{div}_f(v) = \sum_{k\in A(v)} f(k) - \sum_{k\in B(v)} f(k)$.

Definition 4. *A flow in a pure network (N,A,u) from source $s \in A$ to the sink $t \in A$ is a function $f : A \to [0,\infty)$ which satisfies the following conditions:*
1) $0 \leq f(k) \leq u(k)$ *for all* $k \in A$,
2) $\text{div}_f(s) \geq 0$,
3) $\text{div}_f(v) = 0$ *for all* $v \in N \setminus \{s,t\}$.

A number $V = \text{div}_f(s)$ we call the value of a flow f. A flow in pure network is also called static flow.

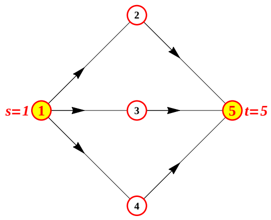

Fig. 7

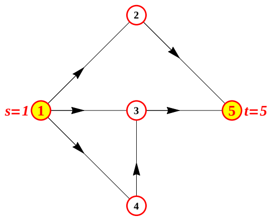

Fig. 8

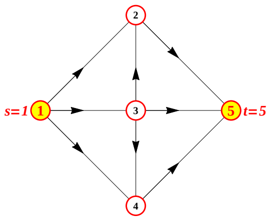

Fig. 9

The definitions were discussed by the lecturer on the example of a simple network (Figure 7) drawn on a table with simple examples of flows in this network. The numerical values of flows on the arcs were recorded above the arcs. The first part lasted about 25 minutes. In the second part, students were given the first task to solve. It was based on giving an example of flow in the network (Figure 8) from source s to the sink t with the value of flow $V = 5$. Students received 10 minutes of time to solve this task. In the third part, two dynamic visualizations of flows in network were presented to the students: **Example 3** and **Example 4** presented in the subsection 4.1. These visualizations were discussed during the presentation by the lecturer. The presentation of the visualization together with the discussion lasted about 10 minutes. In the fourth part, students were again given a task to solve. It consisted, as before, of giving an example of flow in the network (Figure 9) from source s to outlet t with the value of flow $V = 4$. Students were given 10 minutes of time to solve this task. In the fifth part, students answered the question - to what extent dynamic flow visualizations were useful in solving the task in the fourth part. They chose one of four options: a) they were not helpful, b) they were a bit helpful, c) they were helpful, d) they were very helpful. We received the following experiment results: only 24% of students presented the correct solution to the task in the second part of the experiment, 66% of the students presented the correct solution to the task in the fourth part of the experiment (after seeing dynamic visualizations), 5% of the students chose the answer a), 19% answer b), 54% answer c) and 22% answer d).

4.1 Dynamic visualization of static flow

For directed graph (N,A) presented below we define: $s = 1, t = 7, V = 5, u(k) = 3$ for all $k \in A$. We present two different static flows: static flows 1, static flows 2 in the pure network (N,A,u).

Example 3. (static flow 1)

Fig. 10: Dynamic visualizations of static flow 1 in pure network (N,A,u)

Example 4. (static flow 2)

Fig. 11: Dynamic visualizations of static flow 2 in pure network (N, A, u)

The dynamic versions of the Figs. 10 and 11 can be found in the Electronic supplementary material:
https://drive.google.com/open?id=1vgBC1ij7Z9mNL8_nmhVgrwi3qFRzzhYN

A standard method of graphical presentation flows in networks presented in many academic books, is to note the flows values near the arcs of directed graph using one or several figures. This method can be called a static visualization of network flow (for example [1, page 79, Figure 3.9a]; [4, pages 142–143, Figures 9.2–9.4]). In the **Example** 3 we present another - dynamic approach to visualize flows in networks where flows in network have a form of dynamic animations. Full form of animation for **Example** 3 is presented in electronic supplement. In Figures 10 and 11 we present only two single frames of the animation.

5 Summary and conclusions

This article presents visualizations for two algorithms: simplex algorithm for LP problem and KKT algorithm for NLP problem which we have prepared for students of Informatics and Econometrics Faculty of Warsaw University of Life Sciences. The visualizations were prepared using Mathematica. In **Example 1** we demonstrate some new didactic proposition - expanded simplex tableau. The expanded simplex tableau contains for each algorithm step: current simplex table in traditional form for this step, graph of feasible region for canonical form of LP problem with current corner point and current simplex path, level sets of goal function (hyperplanes : lines in $2D$, planes in $3D$), axis with current value of objective function for this step. In our opinion, using such visualization allows students to better understand the steps of the simplex algorithm and its geometric interpretation. In **Example 2** we solve the NLP problem using KKT method.

We demonstrate dynamic visualization of graphic method of obtaining the optimal solution in this example. In general, computer support ((e.g. by CAS) in solving tasks by KKT method seems very useful due to the need to consider sometimes a large number of subcases difficult to consider by hand calculations. The visualizations presented in this example, seem very useful to us in a better understanding of solving NLP problems graphically. This article also presents the result of the didactic experiment carried out on a group of 41 first-year students of the Production Engineering Faculty of Warsaw University of Life Sciences. In this experiment dynamic visualizations for two examples of network flows were presented to these students. This had a significant impact on the result of the test consisting in the construction of flow in the given network. The number of correct solutions increased more than 2.5 times compared to the number of correct solutions of the analogous task before presenting this visualizations. Over 75% of students said that these visualizations were helpful or very helpful in solving the second task (after seeing dynamic visualizations).

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: theory, algorithms, and applications. Prentice Hall, 2nd edn. (1993)
2. Bazaraa, S.M., Shetty, C.M.: Linear Programming and Network Flows. John Wiley and Sons, 2nd edn. (1979)
3. Bazaraa, S.M., Shetty, C.M.: Nonlinear programming. Theory and algorithms. John Wiley and Sons, 2nd edn. (1979)
4. Ford, L., Fulkerson, D.: Network Flows. John Wiley and Sons, 2nd edn. (1979)
5. Griva, I., Nash, S., Sofer, A.: Linear and nonlinear optimization. SIAM (2009)
6. Guyer, T.: Computer algebra systems as the mathematics teaching tool. World Applied Sciences Journal **3**(1), 132–139 (2008)
7. Kramarski, B., Hirsch, C.: Using computer algebra systems in mathematical classrooms. Journal of Computer Assisted Learning **19**, 35–45 (2003)
8. Ruskeepaa, H.: Mathematica Navigator: Graphics and Methods of applied Mathematics. Academic Press, Boston (2005)
9. Wojas, W., Krupa, J.: Familiarizing Students with Definition of Lebesgue Integral: Examples of Calculation Directly from Its Definition Using Mathematica. Mathematics in Computer Science **11**, 363–381 (2017), http://doi.org/10.1007/s11786-017-0321-5
10. Wojas, W., Krupa, J.: Some remarks on Taylor's polynomials visualization using Mathematica in context of function approximation. Springer Proceedings in Mathematics and Statistics **198**, 487–498 (2017)
11. Wojas, W., Krupa, J.: Teaching Students Nonlinear Programming with Computer Algebra System. Mathematics in Computer Science **13**(1-2), 297–309 (2019)
12. Wolfram, S.: The Mathematica Book. Wolfram Media Cambridge University Press (1996)