

## Alternatívne programové produkty na výučbu lineárneho programovania

Ivan Brezina<sup>1</sup>, Juraj Pekár<sup>2</sup>, Marian Reiff<sup>3</sup>

### Abstrakt

Lineárne programovanie sa orientuje na modelovanie a optimalizáciu akéhokoľvek výrobného procesu alebo systému, ktorý možno opísať pomocou lineárnych funkcií a lineárnych rovníc, resp. nerovníc. Modely a metódy lineárneho programovania patria medzi najčastejšie používané kvantitatívne nástroje na podporu rozhodovacieho procesu. Jeho cieľom je nájsť extrém lineárnej účelovej funkcie pri stanovených obmedzeniach opísaných systémom lineárnych nerovností, resp. rovností. Modely a metódy lineárneho programovania predstavujú základ vzdelávania v oblasti využitia kvantitatívnych prístupov na ekonomických a technických univerzitách vo svete. Pri výučbe možno použiť rozličné softvérové produkty, ktoré umožňujú pochopiť podstatu lineárneho programovania tak v jeho grafickej, ako aj numerickej podstate. Na Fakulte hospodárskej informatiky Ekonomickej univerzity v Bratislave sa vo vzdelávacom procese používajú rôzne programové produkty, ako MS Excel, POM-QM for Windows a Simplex V4, ktoré sú v príspevku popísané a prezentované sú ich výhody a nevýhody.

### Kľúčové slová

lineárne programovanie, MS Excel, POM-QM for Windows, Simplex V4

### Abstract

Linear programming is focused on modeling and optimization of any manufacturing process or system that can be described by linear functions and linear equations or inequalities. Linear programming models and methods are among the most commonly used quantitative tools to support the decision-making process. It aims to find the extreme of linear objective function within the feasible region defined by constraints, the system of linear inequalities, respectively equalities. Linear programming models and methods belong to fundamental topics in quantitative methods field of education at economic and technical universities in the world. Various software products can be used in the classroom to understand the nature of linear programming and in its graphical and numerical fundamentals. At the Faculty of Economic Informatics of the University of Economics in Bratislava, various software products such as MS Excel, POM-QM for Windows, and Simplex V4 are used in the educational process and are described in this paper with their advantages and disadvantages.

### Key words

Linear Programming, MS Excel, POM-QM for Windows, Simplex V4

### JEL classification

C61, C63

---

<sup>1</sup> Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra operačného výskumu a ekonometrie, Dolnozemska cesta 1/b, 852 35 Bratislava, ivan.brezina@euba.sk.

<sup>2</sup> Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra operačného výskumu a ekonometrie, Dolnozemska cesta 1/b, 852 35 Bratislava, juraj.pekar@euba.sk.

<sup>3</sup> Ekonomická univerzita v Bratislave, Fakulta hospodárskej informatiky, Katedra operačného výskumu a ekonometrie, Dolnozemska cesta 1/b, 852 35 Bratislava, marian.reiff@euba.sk.

## 1 Úvod

Ekonomické operácie a procesy možno transformovať na ekonomicko-matematické modely, ktoré sú založené na vysokej abstrakcii a predstavujú zjednodušený opis reálneho systému, ktorý je založený na najpodstatnejších prvkoch systému a väzbách medzi nimi. Ekonomicko-matematické modely sú veľmi rôznorodé, ale vo všeobecnosti medzi najviac používané patria modely a metódy matematického programovania. Teória matematického programovania bola pritom vypracovaná na riešenie optimalizačných úloh efektívneho využívania ohraničených disponibilných zdrojov, ktoré sa využívajú na dosiahnutie určených cieľov.

Modely a metódy matematického programovania patria medzi najčastejšie používané a najvýkonnejšie nástroje operačného výskumu, môžu byť aplikované na širokú škálu rôznych rozhodovacích problémov. Modely matematického programovania umožňujú pomerne jednoducho transformovať reálne ekonomické procesy do matematických modelov v tvare funkcií, nerovníc, resp. rovníc. Používajú sa predovšetkým v súvislosti s riešením efektívneho využívania ohraničených disponibilných zdrojov (surovín, materiálu, časového fondu strojov a zariadení, plôch, počtu pracovníkov, investičných a finančných prostriedkov a podobne), ktoré sú potrebné na dosiahnutie stanovených cieľov (maximalizácia zisku, minimalizácia nákladov, minimalizácia času a podobne). Ide o relatívne presné kvantitatívne hodnotenie výberu určitého riešenia z veľkého množstva možných riešení, ktoré je z hľadiska matematicky formulovaného cieľa najlepšie.

## 2 Úlohy lineárneho programovania

Každá úloha matematického programovania je konštruovaná s cieľom zobrazit' takú ekonomickú realitu, pre ktorú je charakteristické hľadanie najlepšieho riešenia pri ohraničujúcich predpokladoch. Model matematického programovania zobrazuje nejakú ekonomickú situáciu, v ktorom sa hľadá najlepšie (optimálne) riešenie (prostredníctvom účelovej funkcie) pri určitých ohraničujúcich predpokladoch (ohraničujúce podmienky úlohy). V prípade, ak sú všetky vzťahy v modeli matematického programovania lineárne, ide o model lineárneho programovania.

Model lineárneho programovania možno zapísať vo všeobecnom tvare:  
extremalizovať (maximalizovať, resp. minimalizovať)

$$f(\mathbf{x}) = \sum_{j=1}^n c_j x_j \quad (1)$$

$$\sum_{j=1}^n a_{ij} x_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_i, \quad i = 1, 2, \dots, m \quad (2)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n \quad (3)$$

kde

$c_j$  – koeficienty účelovej funkcie (ÚF),  $j = 1, 2, \dots, n$ ,

$a_{ij}$  – technologické koeficienty sústavy ohraničení,  $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ ,

$b_i$  – koeficienty pravej strany (PS),  $i = 1, 2, \dots, m$ ,

$x_j$  – rozhodovacie premenné,  $j = 1, 2, \dots, n$ .

Rozhodovacie premenné  $x_j$  ( $j = 1, 2, \dots, n$ ) sú neznáme, ktorých hľadané hodnoty vyjadrujú cieľ rozhodovacieho procesu, teda také hodnoty, ktoré predstavujú tzv. *optimálne riešenie*. Koeficienty účelovej funkcie  $c_j$  ( $j = 1, 2, \dots, n$ ) vyjadrujú určité „ocenenie“ rozhodovacích premenných (zvyčajne finančné, časové, vzdialenostné a podobne). Koeficienty pravej strany  $b_i$  ( $i = 1, 2, \dots, m$ ) vyjadrujú obmedzenia, ktorými sú determinované obmedzené zdroje (surovinové, časové, finančné a ďalšie kapacity). Technologické koeficienty sústavy ohraničení  $a_{ij}$  ( $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ ) opisujú „spotrebu“ obmedzených disponibilných zdrojov a sú charakteristické „normami“ spotreby.

Pri riešení zodpovedajúcich úloh lineárneho programovania (ÚLP) sa hľadá taký vektor rozhodovacích premenných  $\mathbf{x}^T = (x_1, x_2, \dots, x_n)$ , pre ktorý je hodnota ÚF (1) maximálna (alebo minimálna) pri splnení predpokladov (2) a (3).

Najčastejšie používané základné pojmy pre riešenie ÚLP (Brezina, Pekár, 2018):

- *Prípustným riešením úlohy* (1) až (3) sa nazýva taký vektor  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , ktorý vyhovuje ohraničujúcim podmienkam (2) a (3).
- *Množinou prípustných riešení úlohy* (1) až (3) sa nazýva množina,  $\Theta = \{ \mathbf{x} \in R^n \mid \mathbf{Ax} \{ \leq, =, \geq \} \mathbf{b}, \mathbf{x} \geq \mathbf{0} \}$ , kde  $R^n$  je  $n$ -rozmerný euklidovský priestor,  $\mathbf{A}$  je matica technologických koeficientov sústavy ohraničení,  $\mathbf{b}$  je vektor koeficientov pravej strany,  $\mathbf{x}$  je vektor rozhodovacích premenných a  $\mathbf{0}$  je nulový vektor.
- Za predpokladu, že hodnosť matice technologických koeficientov  $\mathbf{A}$  je  $m$ , riešenie  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  nazývame *bázickým riešením*, ak vyhovuje nerovniciam (2) upraveným na tvar  $\sum_{j=1}^n a_{ij} x_j = b_i, \quad b_i \geq 0, \quad i = 1, 2, \dots, m$  a obsahuje maximálne  $m$  nenulových zložiek, ktorých prislúchajúca matica štruktúrnych koeficientov má hodnosť  $m$ .
- *Bázickým prípustným riešením úlohy* (1) až (3) sa nazýva také bázické riešenie, ktoré patrí do množiny  $\Theta$ .
- *Optimálnym riešením úlohy* (1) až (3) je také prípustné riešenie, pre ktoré nadobúda účelová funkcia svoj extrém (ak je ÚF maximalizačná, tak svoje maximum, ak je ÚF minimalizačná, tak svoje minimum). To znamená, že vektor  $\mathbf{x}^* \in \Theta$  je optimálnym riešením maximalizačnej úlohy vtedy, keď platí:  $f(\mathbf{x}^*) \geq f(\mathbf{x}), \mathbf{x} \in \Theta$ .

Metódy, ktoré boli vypracované v rámci teórie lineárneho programovania na riešenie následne definovaných úloh lineárneho programovania, sú pomerne jednoduché vo svojej matematickej štruktúre a zároveň použiteľné na riešenie veľkého počtu rôznych úloh. Vo všeobecnosti najpoužívanejšou metódou riešenia ÚLP je simplexová metóda so svojimi modifikáciami, ale na inom princípe sú založené metódy vnútorného bodu (napr. elipsoidná metóda, Karmarkarova metóda vychádzajúca z elipsoidnej metódy, sférické metódy ...).

Časová zložitosť simplexovej metódy je polynomiálna, pričom existuje mnoho numerických experimentov poukazujúcich na skutočnosť, že na riešenie ÚLP s  $m$  ohraničujúcimi podmienkami stačí  $m$  iterácií simplexovej metódy. Svojou podstatou je pritom simplexová metóda ľahko pochopiteľná a tak patrí medzi základné metódy používané na riešenie ÚLP. Elipsoidná metóda je založená na tom, že postupne iteruje k riešeniu tak, že objem elipsoidu sa v každej iterácii znižuje o faktor  $e^{\frac{1}{2(n+1)}}$ , takže výpočtová zložitosť je  $O(n^2)$ , ale v praxi sa tento druh nepoužíva často, pretože je pomalý. Karmarkarov algoritmus má polynomiálnu zložitosť, ale je na rozdiel od elipsoidnej metódy rýchly a je rýchlejší ako simplexová metóda pri ÚLP s veľkým množstvom vstupných údajov.

### 3 Geometrické riešenie úlohy lineárneho programovania

Grafická vizualizácia riešenia akejkoľvek úlohy predstavuje efektívny nástroj na pochopenie a prezentáciu väzieb medzi jednotlivými veličinami. Aj grafické riešenie ÚLP potvrdzuje pri výučbe lineárneho programovania skutočnosť, že jeden obrázok zvyčajne vyjadruje viac ako desiatky slov. Geometrické (grafické) riešenie ÚLP má len ilustračný charakter, zvyčajne sa graficky rieši ÚLP s dvoma premennými, ktorú možno jednoducho zobraziť v dvojrozmernom priestore (graficky možno riešiť aj úlohu s tromi premennými v trojrozmernom priestore, ale pretože reálne ÚLP obsahujú zvyčajne desiatky až stovky premenných, geometrické riešenie ÚLP je len názorné).

Oblasť, ktorá vyhovuje všetkým nerovniciam (štruktúrnym ohraničeniam (2) a podmienkam nezápornosti (3)), predstavuje množinu prípustných riešení úlohy  $\Theta$ . Táto množina vzniká ako prienik všetkých ohraničení a predstavuje konvexnú polyedrálnu množinu, ktorá vzniká ako prienik konečného počtu polpriestorov, definovaných jednotlivými nerovnicami. Prieniky hraničných priamok pritom reprezentujú množinu bázičných riešení.

Počet bodov tvoriacich množinu prípustných riešení  $\Theta$  je nekonečný. Pretože cieľom riešenia ÚLP je nájsť jej extrém, treba nájsť taký bod množiny prípustných riešení  $\Theta$ , v ktorom účelová funkcia nadobúda extrémnu hodnotu. Je pritom zrejmé, účelová funkcia ÚLP (1) nadobúda svoju extrémnu hodnotu, ak existuje, v niektorom krajnom bode konvexnej polyedrálnej množiny, na ktorej je definovaná. Vo všeobecnosti však pri použití grafickej metódy riešenia nemusíme skúmať hodnoty ÚF vo všetkých krajných bodoch, ale pre nájdenie optimálneho riešenia úlohy treba geometricky interpretovať účelovú funkciu, ktorá je reprezentovaná úroveňou priamkou ÚF, ktorá na báze jej gradientu predstavuje smer jej najrýchlejšieho nárastu.

Geometrické riešenie ÚLP poskytuje nielen názornú predstavu o podstate riešeného optimalizačného problému, ale umožňuje aj jednoduchšie pochopenie základnej myšlienky, na ktorej spočíva riešenie ÚLP pomocou simplexovej metódy, ktorá je založená na preskúmaní krajných bodov konvexného polyédru, ktorý reprezentuje sústava ohraničujúcich podmienok (nerovnic).

### 4 Riešenie úlohy lineárneho programovania simplexovou metódou

Simplexová metóda (simplexový algoritmus) je všeobecná metóda na hľadanie optimálneho riešenia ÚLP, pričom neskúma všetky prípustné, ale len bázičné prípustné riešenia. Ako už bolo uvedené, ak má ÚLP optimálne riešenie, tak sa toto optimálne riešenie nachádza v jednom z krajných bodov konvexnej polyedrálnej množiny, na ktorej je definovaná. Táto skutočnosť redukuje problém jeho nájdenia na konečný výpočet, pretože krajných bodov je konečné množstvo. Simplexová metóda je ďalej založená na skutočnosti, že pokiaľ nenadobúda účelová funkcia v krajnom bode extrémnu hodnotu, tak existuje taká hrana prípustného konvexného polyédru, ktorá obsahuje tento bod, po ktorej smerom od daného krajného bodu hodnota účelovej funkcie rastie, resp. klesá (podľa typu extremalizácie). Práve na týchto princípoch je založený postup simplexovej metódy, ktorý spočíva v prechode cez hrany do krajných bodov so stále väčšou, resp. menšou hodnotou účelovej funkcie.

Simplexová metóda predstavuje iteračný postup, ktorý po určitom konečnom počte iterácií umožňuje určiť optimálne riešenie ÚLP alebo poskytuje informáciu o tom, že optimálne riešenie neexistuje. Simplexová metóda je založená na troch základných pojmoch: na *prípustnosti*, *bázičnosti* a *optimálnosti*. V každej iterácii možno získať prípustné bázičné riešenie (ak prípustné bázičné riešenie neexistuje, procedúra končí), ktoré sa testuje, či je optimálne. Ak je riešenie optimálne, výpočet končí, ak nie je, algoritmus pristúpi ku skúmaniu ďalšieho prípustného bázičného riešenia. Prostredníctvom simplexovej metódy možno skúmať postupne len tie prípustné bázičné riešenia (*prípustnému bázičnému riešeniu* vo všeobecnosti

zodpovedá sústava  $m$  lineárne nezávislých vektorov), ktoré nie sú horšie ako predchádzajúce, a po konečnom počte krokov určiť buď optimálne riešenie, alebo zistiť, že optimálne riešenie neexistuje. Na testovanie optimálnosti sa využíva *kritérium optimálnosti* (preskúmanie tzv. čistých efektov účelovej funkcie).

V literatúre sa zvyčajne uvádzajú dva základné algoritmy simplexovej metódy – *primárny (priamy) algoritmus* a *duálny algoritmus* – spolu s niekoľkými modifikáciami ako primárno-duálny algoritmus, primárny revidovaný algoritmus, a ďalšie (Nelder a Mead, 1965, Routh a kol., 1977, Betteridge a kol., 1985 ...).

Aplikácia primárneho simplexového algoritmu spočíva v troch krokoch (základná schéma simplexovej metódy):

1. *preverovanie (testovanie) optimálnosti* na základe *kritéria optimálnosti* (overovanie zlepšenia hodnoty ÚF), ak je vypočítané riešenie optimálne, nemožno už dosiahnuť lepšiu hodnotu ÚF, výpočet sa končí, ak vypočítané riešenie nie je optimálne, výpočet pokračuje;
2. určenie *vstupujúcej premennej*, ktorá prinesie svojim zaradením do bázy riešenia najväčší jednotkový prírastok hodnoty ÚF, teda do bázy sa zaradí vektor zodpovedajúci premennej s najväčším jednotkovým prírastkom ÚF;
3. určenie *vystupujúcej premennej*, teda disponibilného zdroja (vektora zodpovedajúceho príslušnej premennej), ktorý determinuje realizáciu modelovaného procesu, teda vyradenie z bázy vektora bázikkej premennej, ktorej ohraničené množstvo disponibilných zdrojov a normy ich spotreby determinujú použiteľnosť týchto zdrojov.

#### **Primárny algoritmus simplexovej metódy**

Pre testovanie optimálnosti riešenia sa používa *kritérium optimálnosti*  $c_j - z_j$  (resp.  $z_j - c_j$  alebo  $z_j$ ,  $j = 1, 2, \dots, n$ ), ktoré predstavuje hodnoty potenciálneho prírastku ÚF (tzv. redukované ocenenia). Tento prírastok možno dosiahnuť z jednej jednotky príslušnej nebázikkej premennej po vstupe jej zodpovedajúceho vektora do bázy. Aby nájdené riešenie bolo optimálne, musia byť pri maximalizačnej úlohe všetky prvky riadku nekladné, to znamená

$$c_j - z_j = c_j - \mathbf{c}_B^T \cdot \mathbf{B}^{-1} \cdot \mathbf{A}_j \leq 0 \quad j = 1, 2, \dots, n \quad (4)$$

kde

$\mathbf{c}_B$  – jednotkové ceny zodpovedajúce cenovým koeficientom príslušných bázikkých premenných,

$\mathbf{B}^{-1}$  – inverzná matica bázy,

$\mathbf{A}_j$  –  $j$ -ty stĺpcový vektor matice štruktúrnych koeficientov  $\mathbf{A}$ .

Analogicky v minimalizačných úlohách je vypočítané riešenie optimálne práve vtedy, keď platí  $c_j - z_j \geq 0$ .

Pre ďalší výpočet treba určiť:

1. *vstupujúcu premennú*,
2. *vystupujúcu premennú*.

*Vstupujúcu premennú*  $x_k$  možno určiť pomocou vzťahu

$$c_j - z_j = c_j - \mathbf{c}_B^T \cdot \mathbf{B}^{-1} \cdot \mathbf{A}_j \leq 0 \quad j = 1, 2, \dots, n \quad (5)$$

a *vystupujúcu premennú*  $x_r$  pomocou vzťahu

$$x_{rk} = \min_i \left\{ \frac{x_{B_i}}{x_{ik}} \mid x_{ik} > 0 \right\} \quad (6)$$

kde

$x_{B_i}$  – sú modifikované hodnoty bázičných premenných,

$x_{ik}$  – modifikované koeficienty zodpovedajúce rozkladu vedúceho stĺpca podľa bázy,

$x_{rk}$  – vedúci prvok.

Prvok ležiaci na priesečníku vedúceho stĺpca s vedúcim riadkom je *vedúci prvok*. Výpočet zmien, ktoré vznikajú pri hľadaní nového riešenia, sa uskutočňuje pomocou elementárnej zmeny bázy pre všetky hodnoty  $i = 1, 2, \dots, m$  a  $j = 1, 2, \dots, n$  na základe

$$x'_{ij} = x_{ij} - \frac{x_{rj}}{x_{rk}} \cdot x_{ik}, \quad i \neq r \quad (7)$$

kde  $r$  je index vedúceho riadka a  $k$  je index vedúceho stĺpca,  $x'_{ij}$  je nová hodnota príslušného koeficientu rozkladu zodpovedajúceho stĺpcového vektora  $\mathbf{A}_j$  podľa bázy.

Modifikácia pre prípad ÚLP s minimalizačnou ÚF spočíva v tom, že riešenie bude optimálne vtedy, keď pre všetky  $j = 1, 2, \dots, n$  bude  $c_j - z_j \geq 0$ , pričom výber vstupujúcej premennej (kritérium pre vystupujúcu premennú sa nezmení) sa vykoná na základe vzťahu

$$c_k - z_k = \min_j \{c_j - z_j \mid c_j - z_j < 0\}. \quad (8)$$

Pri riešení úloh lineárneho programovania sa často vyskytujú riešenia s určitými špecifikami, *alternatívne riešenie, úloha s neohraničenou množinou prípustných riešení, úloha s prázdnu množinou prípustných riešení, úloha s degenerovaným riešením*. Tieto osobitosti sa tiež dajú dobre interpretovať pomocou grafického riešenia ÚLP.

## 5 Softvérové riešenie úloh lineárneho programovania

Reálne riešenia úloh lineárneho programovania sú vo všeobecnosti časovo náročné. Preto boli vyvinuté programy schopné extremalizovať lineárne obmedzenia dané lineárnymi rovnicami a nerovnosťami tak, aby bolo možné získať optimálnu hodnotu účelovej funkcie. Pre riešenie praktických úloh boli vyvinuté profesionálne softvérové produkty ako sú GAMS, LINDO, LINGO, Python, Gurobi, atď., avšak problematika výučby je praktickejšie vysvetliteľná na nástrojoch, pri ktorých sú študenti schopní analyzovať jednotlivé kroky riešenia, ako aj ich grafickú interpretáciu. Softvér pri výučbe matematického programovania využívajú aj svetoví autori (Anderson, Sweeney, Williams, Camm and Cochran, 2018, Stevenson and Sum, 2014, Taha, 2011 a ďalší).

Podľa [windowsreport.com/linear-programming-software](http://windowsreport.com/linear-programming-software) patria pre windows medzi najlepšie softvéry:

- Gurobi (<https://www.gurobi.com/>)
- CPLEX (<https://www.ibm.com/products/ilog-cplex-optimization-studio>)
- XPRESS (<https://www.fico.com/en/products/optimization>)
- Lingo (<https://www.lindo.com/index.php/products/lingo-and-optimization-modeling>)
- GAMS (General Algebraic Modeling System) (<https://www.gams.com/>)
- Visual Math (<http://www.visualmath.com/>) a ďalšie.

Ale v praxi, aj na akademickej pôde sa používa mnoho ďalších softvérov a softvérových jazykov na riešenie ÚLP (napr. MATLAB, jazyk R, Python a pod.). Ďalej uvedieme spôsob riešenia úlohy lineárneho programovania v najdostupnejšom softvérovom produkte MS Excel, výučbovom programom balíku POM-QM for Windows a novovyvinutom nástroji na výučbu lineárneho programovania Simplex V4, pričom uvedené produkty sú využívané pri výučbe na

univerzitách pri úvodných kurzoch zameraných na matematické programovanie (ako príklad vytvorenia vlastného softvéru na Slovensku možno uviesť, že na UPJŠ v Košiciach bol za účelom riešenia úloh lineárnej optimalizácie vyvinutý program CASSIM (Computer ASsisted SIMplex Method), na SPÚ v Nitre softvér GeoGebra, ktorý je vhodný na vizualizáciu aplikovaných problémov).

### 5.1 Riešenie úloh lineárneho programovania v MS Excel

Pri výučbe lineárneho programovania možno využiť všeobecne dostupný programový produkt MS Excel a to aplikáciou optimalizačného doplnku *Riešiteľ*. Všeobecne možno v programe MS Excel pomocou optimalizačného doplnku *Riešiteľ* riešiť úlohy matematického programovania a teda aj úlohy lineárneho programovania. Na nasledujúcej úlohe lineárneho programovania bude prezentovaný spôsob jej riešenia v uvedenom programe.

Predpokladajme úlohu lineárneho programovania (z dôvodu nasledujúcej grafickej interpretácie výpočtu predpokladáme ÚLP s dvomi rozhodovacími premennými):

$$\begin{aligned}
 f(\mathbf{x}) &= 70x_1 + 80x_2 \rightarrow \max \\
 11x_1 + 8x_2 &\geq 1000 \\
 5x_1 + 10x_2 &\leq 900 \\
 13x_1 + 9x_2 &\leq 1400 \\
 4x_1 + 6x_2 &\geq 450 \\
 x_1, x_2 &\geq 0
 \end{aligned}
 \tag{9}$$

Na riešenie optimalizačnej úlohy (9) v tabuľkovom procesore treba pripraviť vstupné dáta v tvare zápisu ÚLP (1) až (3). Ich usporiadanie môže byť v podstate ľubovoľné, musia však byť dodržané určité pravidlá, ktoré vyžaduje optimalizačný doplnok (*Riešiteľ*). Na obr. 2 je prezentovaná možnosť rozloženia vstupných dát pre príklad (9) v tabuľke MS Excel. Na obr. 1 je uvedené zadanie a optimálne riešenie príkladu (9) po spustení optimalizačného doplnku.

Obr. 1: Optimálne riešenie úlohy (9) v MS Exceli

	A	B	C	D
1	70	80	9282,353	
2	11	8	1205,882	1000
3	5	10	900	900
4	13	9	1400	1400
5	4	6	609,4118	450
6	69,41176	55,29412		
7				

Zdroj: Vlastné spracovanie

V zadanej úlohe lineárneho programovania sú definované dve rozhodovacie premenné. Pre tieto premenné je rezervovaný blok A6:B6. Každé rozhodovacej premennej je priradená počiatočnú hodnotu 0. Na obr. 2 sú prezentované vzorce pre výpočet hodnôt jednotlivých buniek.

Obr. 2: Zápis podmienok úlohy (9) v MS Exceli

	A	B	C	D
1	70	80	=SUMPRODUCT(A1:B1;\$A\$6:\$B\$6)	
2	11	8	=SUMPRODUCT(A2:B2;\$A\$6:\$B\$6)	1000
3	5	10	=SUMPRODUCT(A3:B3;\$A\$6:\$B\$6)	900
4	13	9	=SUMPRODUCT(A4:B4;\$A\$6:\$B\$6)	1400
5	4	6	=SUMPRODUCT(A5:B5;\$A\$6:\$B\$6)	450
6	69,4117	55,29411		

Zdroj: Vlastné spracovanie

Zápis jednotlivých obmedzujúcich podmienok vyžaduje, aby boli vyjadrené ako funkcia ľavej strany podmienky porovnaná s koeficientmi pravej strany. V programe MS Excel je na výpočet skalárneho súčinu k dispozícii funkcia, ktorá sa dá v tejto súvislosti využiť. V slovenskej verzii MS Excel ide o funkciu *SUMPRODUCT(a; b)*, kde *a* a *b* sú bloky obsahujúce vektory (príp. matice), pre ktoré sa má vypočítať skalárny súčin. V nasledujúcej tabuľke 1 (zodpovedá obr. 2) sú uvedené obmedzujúce podmienky podľa zadania a príslušný vzorec ich výpočtu pre ľavú stranu rovnice.

Tab. 1: Zápis obmedzujúcich podmienok a vzorce pre výpočet ľavej strany rovnice

Obmedzujúca podmienka	Bunka	Vzorec
$11x_1 + 8x_2$	C2	=SUMPRODUCT(A2:B2; \$A\$6: \$B\$6)
$5x_1 + 10x_2$	C3	=SUMPRODUCT(A3:B3; \$A\$6: \$B\$6)
$13x_1 + 9x_2$	C4	=SUMPRODUCT(A4:B4; \$A\$6: \$B\$6)
$4x_1 + 6x_2$	C5	=SUMPRODUCT(A5:B5; \$A\$6: \$B\$6)

Zdroj: Vlastné spracovanie

Posledným krokom pri príprave vstupných údajov je definícia pozície optimalizačného kritéria (účelovej funkcie). Toto kritérium musí byť zapísané v tvare vzorca a umiestnené do niektorej bunky. V našom príklade je to bunka C2 (Obr. 2), v ktorej sa optimalizačné kritérium vyjadri ako skalárny súčin vektora očakávaného zisku za 1 kus (blok A1:B1) s vektorom rozhodovacích premenných (blok A6:B6).

Po ukončení prípravy vstupných údajov môže byť aktivovaný vlastný optimalizačný doplnok – *Riešiteľ*. Po jeho spustení (*Údaje – Riešiteľ*) je používateľovi ponúknuté dialógové okno „*Parametre doplnku Riešiteľ*“ určené na zadávanie parametrov riešenej úlohy.

Po definovaní všetkých potrebných údajov v dialógovom okne „*Parametre doplnku Riešiteľ*“ možno spustiť spracovanie pomocou tlačidla „*Riešiť*“. V závislosti od rozsahu riešenej úlohy vlastný výpočet môže trvať rôzne dlho podľa toho, či sú do úlohy zahrnuté podmienky celočíselnosti, a samozrejme, takisto v závislosti od výkonnosti počítača používaného pri spracovaní.

Po skončení výpočtu sa zobrazí dialógové okno, v ktorom je uvedená informácia, či bolo alebo nebolo nájdené riešenie spĺňajúce všetky ohraničujúce podmienky (optimálne riešenie).

Nevýhodou uvedeného riešenia prostredníctvom programu MS Excel je, že nemožno sledovať jednotlivé kroky programu a nie je k dispozícii ani grafické riešenie úlohy. Výhodou je, že je to bežne dostupný program, ktorého profesionálna verzia umožňuje riešiť aj pomerne rozsiahle ÚLP s dostatočne širokou ponukou výstupov (analýza citlivosti, redukované ocenenia, duálne ceny ...).

## 5.2 Riešenie úloh lineárneho programovania v POM-QM for Windows

Zaujímavým a pomerne dobrým nástrojom vhodným pre výučbu riešenia ÚLP je aj voľne dostupný produkt POM-QM for Windows (známy tiež ako POM pre Windows alebo QM pre



Windows). POM-QM for Windows je užívateľsky prívetivý softvérový balík určený na riešenie úloh z oblasti riadenia výroby, riadenia operácií, kvantitatívnych metód, operačného výskumu a z oblasti management science. POM-QM for Windows bol navrhnutý tak, aby pomohol študentom lepšie porozumieť uvedeným oblastiam operačného výskumu. Softvér možno použiť buď priamo na riešenie úloh, alebo na kontrolu odpovedí, ktoré boli riešené ručne. POM-QM for Windows obsahuje pomerne veľké množstvo modulov zodpovedajúcich jednotlivým oblastiam operačného výskumu, ako napríklad modul pre teóriu hier, lineárne programovanie, cieľové programovanie, celočíselné a zmiešané programovanie, zásoby, obsluhu, výpočet rozvrhu pre job shop, sieťovú analýzu, Markovove reťazce, riadenie projektov, atď.

Možnosti riešenia ÚLP v POM-QM for Windows budú taktiež prezentované na príklade (9). Zadanie úlohy v softvéri je veľmi jednoduché, intuitívne a je zobrazené na Obr. 3. Výhodou POM-QM for Windows je skutočnosť, že okrem iného poskytuje grafické riešenie ÚLP, ako aj možnosť sledovať jednotlivé iteračné kroky simplexovej metódy (Obr. 4), analýzu citlivosti, konštrukciu aj riešenie duálnej úlohy a taktiež upozorní na alternatívne riešenie (Obr. 5). Podstatnou nevýhodou však je, že poskytuje výsledky v tvare desatinných čísel a tiež, že numerické riešenie nie je v niektorých špeciálnych prípadoch korektné. Pri grafickom riešení (Obr. 6) neposkytuje možnosť jednoducho vysvetliť základné pojmy lineárneho programovania ako sú bázické riešenie, prípustné bázické riešenie a podobne.

Obr. 3: Zadanie úlohy (9) v POM-QM for Windows

Objective						
<input checked="" type="radio"/> Maximize <input type="radio"/> Minimize						
(untitled)						
	X1	X2		RHS	Equation form	
Maximize	70	80			Max $70X_1 + 80X_2$	
Constraint 1	11	8	$\geq$	1000	$11X_1 + 8X_2 \geq 1000$	
Constraint 2	5	10	$\leq$	900	$5X_1 + 10X_2 \leq 900$	
Constraint 3	13	9	$\leq$	1400	$13X_1 + 9X_2 \leq 1400$	
Constraint 4	4	6	$\geq$	450	$4X_1 + 6X_2 \geq 450$	

Zdroj: Vlastné spracovanie

Obr. 4: Iteračné kroky simplexovej metódy v POM-QM for Windows

(untitled) Solution										
Cj	Basic Variables	Quantity	70 X1	80 X2	0 artfcl 1	0 surplus 1	0 slack 2	0 slack 3	0 artfcl 4	0 surplus 4
1	artfcl 4	86.3636	0	3.0909	-0.3636	0.3636	0	0	1	-1
	zj	86.3636	0	-3.09	2.36	-0.36	0	0	1	1
	cj-zj		0	3.0909	-1.3636	0.3636	0	0	0	-1
Iteration 3										
0	X1	70.5882	1	0	0.1765	-0.1765	0	0	-0.2353	0.2353
0	slack 2	267.64...	0	0	0.2941	-0.2941	1	0	-2.0588	2.0588
0	slack 3	230.88...	0	0	-1.2353	1.2353	0	1	0.1471	-0.1471
0	X2	27.9412	0	1	-0.1176	0.1176	0	0	0.3235	-0.3235
	zj	0	0	0	2	0	0	0	2	0
	cj-zj		0	0	-1.0	0	0	0	-1.0	0
Phase 2										
70	X1	70.5882	1	0	0.1765	-0.1765	0	0	-0.2353	0.2353
0	slack 2	267.64...	0	0	0.2941	-0.2941	1	0	-2.0588	2.0588
0	slack 3	230.88...	0	0	-1.2353	1.2353	0	1	0.1471	-0.1471
80	X2	27.9412	0	1	-0.1176	0.1176	0	0	0.3235	-0.3235
	zj	7 176....	70	80	2.94	-2.94	0	0	9.41	-9.41
	cj-zj		0	0	-2.9412	2.9412	0	0	-9.4118	9.4118
Iteration 5										
70	X1	40.0	1	0	0.1429	-0.1429	-0.1143	0	0	0
0	surplus 4	130.0	0	0	0.1429	-0.1429	0.4857	0	-1	1
0	slack 3	250.0	0	0	-1.2143	1.2143	0.0714	1	0	0
80	X2	70.0	0	1	-0.0714	0.0714	0.1571	0	0	0
	zj	8 400	70	80	4.29	-4.29	4.57	0	0	0
	cj-zj		0	0	-4.2857	4.2857	-4.5714	0	0	0
Iteration 6										
70	X1	69.4118	1	0	0	0	-0.1059	0.1176	0	0
0	surplus 4	159.41...	0	0	0	0	0.4941	0.1176	-1	1
0	surplus 1	205.88...	0	0	-1	1	0.0588	0.8235	0	0
80	X2	55.2941	0	1	0	0	0.1529	-0.0588	0	0
	zj	9 282....	70	80	0	0	4.82	3.53	0	0
	cj-zj		0	0	0	0	-4.8235	-3.5294	0	0

Zdroj: Vlastné spracovanie

Obr 5: Riešenie úlohy (9) v POM-QM for Windows

Objective:  Maximize  Minimize

Note: Multiple optimal solutions exist

	X1	X2		RHS	Dual
Maximize	70	80			
Constraint 1	11	8	>=	1000	0
Constraint 2	5	10	<=	900	4.82
Constraint 3	13	9	<=	1400	3.53
Constraint 4	4	6	>=	450	0
Solution	69.41	55.29		9282.35	

Variable	Value	Reduced C.	Original Val	Lower Bound	Upper Bound
X1	69.41	0	70	40	115.56
X2	55.29	0	80	48.46	140
Dual Value		Slack/Surplus	Original Val	Lower Bound	Upper Bound
Constraint 1	0	205.88	1000	-Infinity	1205.88
Constraint 2	4.82	0	900	577.38	1555.56
Constraint 3	3.53	0	1400	1150	2340
Constraint 4	0	159.41	450	-Infinity	609.41

Original Problem	X1	X2			
Maximize					
Constraint 1	11	8	>=	1000	
Constraint 2	5	10	<=	900	
Constraint 3	13	9	<=	1400	
Constraint 4	4	6	>=	450	

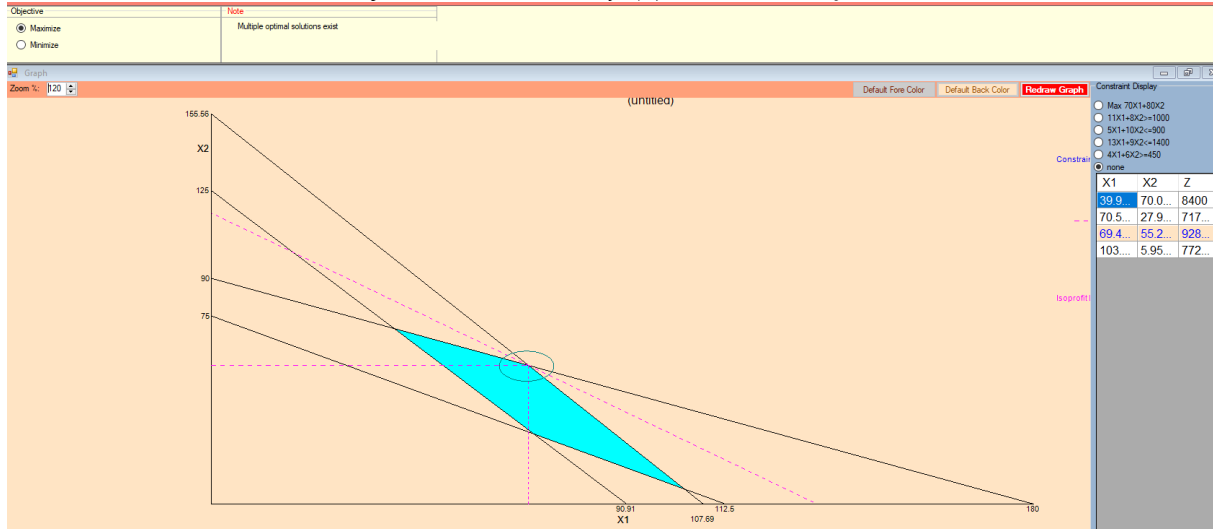
  

Dual Problem	Constr...	Constr...	Constr...	Constr...	
Minimize	-1000	900	1400	-450	
X1	-11	5	13	-4	>= 70
X2	-8	10	9	-6	>= 80

Variable	Status	Value
X1	Basic	69.41
X2	Basic	55.29
surplus 1	Basic	205.88
slack 2	NONBasic	0
slack 3	NONBasic	0
surplus 4	Basic	159.41
Optimal Value (Z)		9282.35

Zdroj: Vlastné spracovanie

Obr. 6: Grafické riešenie úlohy (9) v POM-QM for Windows



Zdroj: Vlastné spracovanie

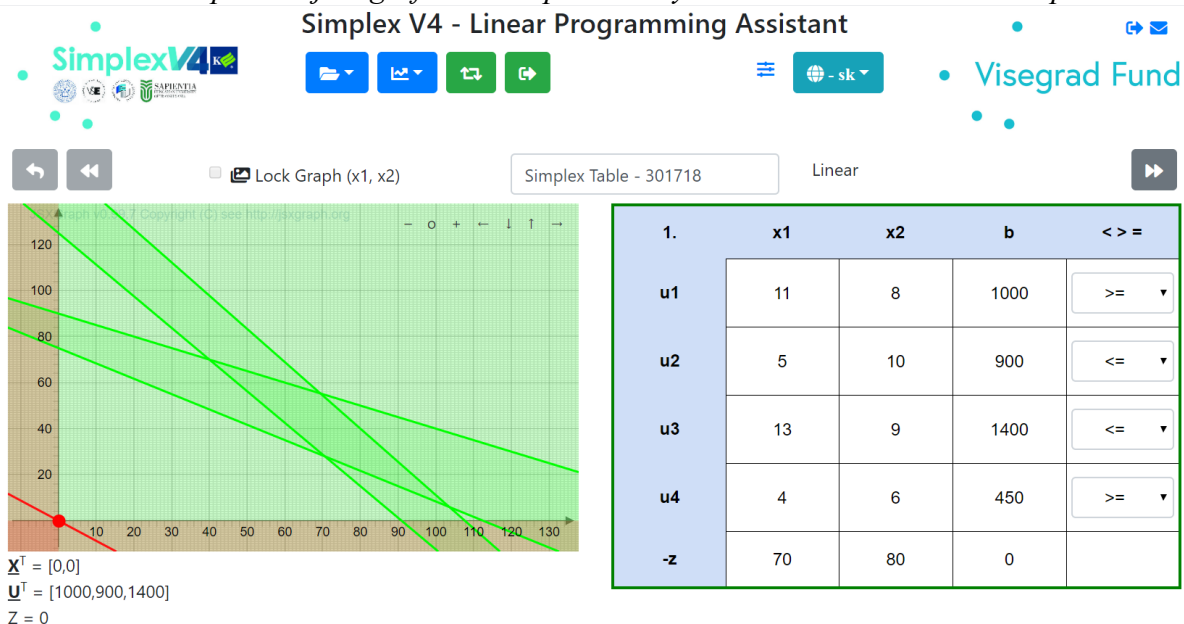
### 5.3 Riešenie úloh lineárneho programovania v Simplex V4

Pre potreby vyučovacieho procesu v oblasti lineárneho programovania bol v rámci projektu ID: 21830031, „SimplexV4 common free Tool development and Day“ vyvinutý program Simplex V4, ktorý je zameraný na prezentáciu základných operácií elementárnej zmeny bázy v zlomkovom tvare. Tento program tiež umožňuje sledovať grafické riešenie ÚLP a predovšetkým interpretovať základné pojmy lineárneho programovania, ako sú bázičné riešenie, bázičné prípustné riešenie, výber vedúceho prvku.

Zápis vstupných údajov je opäť veľmi intuitívny a umožňuje zápis všetkých druhov možných ohraničení. Kritérium optimálnosti je v Simplex V4 v riadku  $-z$  v dolnej časti tabuľky (Obr. 7).

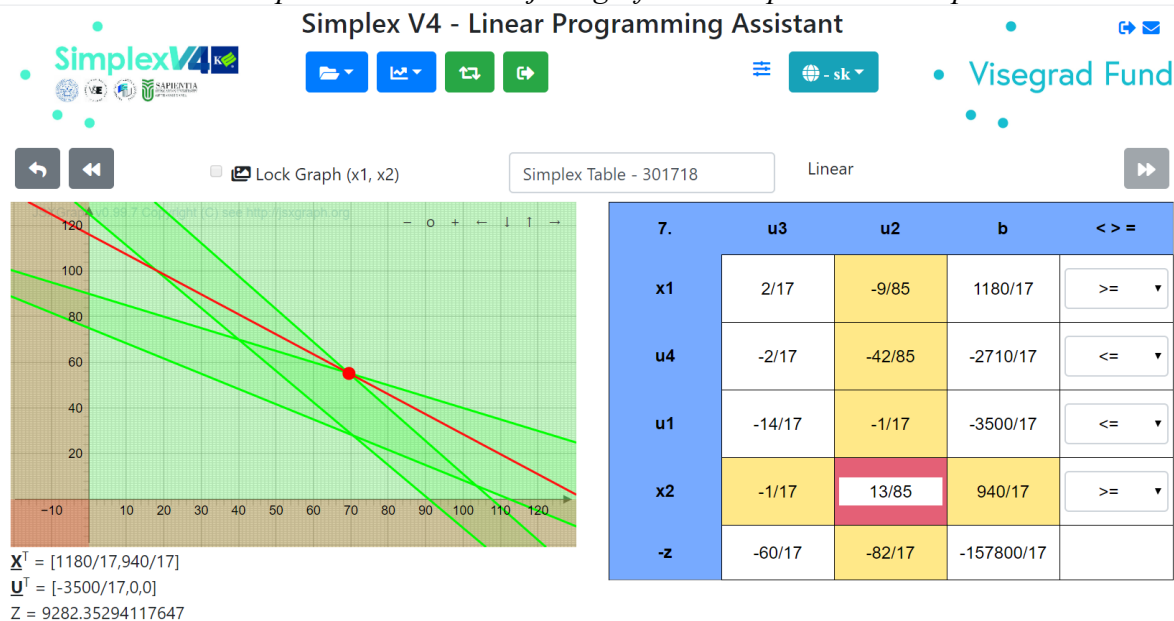
Po realizácii elementárnej zmeny bázy možno vždy získať nové riešenie tak v tabuľkovej, ako aj v grafickej podobe a po príslušnom počte krokov dospieť k optimálnemu riešeniu (Obr. 8).

Obr. 7: Vstupné údaje a grafická interpretácia východiskového riešenia v Simplex V4



Zdroj: Vlastné spracovanie

Obr. 8: Optimálne riešenie a jeho grafická interpretácia v Simplex V4



Zdroj: Vlastné spracovanie

Uvedený program Simplex V4 umožňuje interpretovať popísaný postup grafického riešenia ÚLP tak, ako bol opísaný v časti 2 príspevku – Geometrické riešenie úlohy lineárneho programovania. Oblasť, ktorá vyhovuje všetkým nerovniciam (štruktúrnym ohraničeniam (2) a podmienkam nezápornosti (3)), predstavuje množinu prípustných riešení úlohy  $\Theta$  a na obrázku predstavuje prienik všetkých šiestich polrovín (4 štruktúrne ohraničenia, 2 podmienky nezápornosti). Táto množina vzniká ako prienik konečného počtu polpriestorov (v uvedenom príklade šiestich) definovaných jednotlivými nerovnicami. Prieniky hraničných priamok pritom reprezentujú množinu bázických riešení.

Počet bodov tvoriacich túto množinu prípustných riešení  $\Theta$  je nekonečný, ale pretože cieľom riešenia ÚLP je nájsť jej extrém, treba nájsť taký bod množiny prípustných riešení  $\Theta$ , v ktorom účelová funkcia nadobúda extrémnu hodnotu. Účelová funkcia ÚLP (1) nadobúda svoju

extrémnu hodnotu v niektorom krajnom bode konvexnej polyedrálnej množiny, na ktorej je definovaná a pre jeho nájdenie treba geometricky interpretovať účelovú funkciu, ktorá je reprezentovaná úroveňou priamkou ÚF, ktorá na báze jej gradientu predstavuje smer jej najrýchlejšieho nárastu. Na Obr. 6 a Obr. 7 je reprezentovaná červenou priamkou.

Program Simplex V4 ako nástroj na podporu výpočtov realizovaných študentami vyžaduje od študentov na báze vzťahu (5) určiť vedúci stĺpec (vstupujúcu premennú) a na báze vzťahu (6) vedúci riadok (vystupujúcu premennú) a po ich výbere realizovať elementárnu zmenu bázy na základe vzťahu (7).

## 6 Záver

Príspevok vznikol ako výsledok spolupráce krajín V4 v rámci projektu ID: 21830031, „SimplexV4 common free Tool development and Day“ ako programový produkt určený na výučbu najrozšírenejšej disciplíny operačného výskumu, výučbu lineárneho programovania. Výučba lineárneho programovania je v súčasnosti založená na pochopení základnej metódy riešenia úloh lineárneho programovania, na aplikácii simplexovej metódy. Klasický výpočet pomocou simplexovej metódy je pomerne časovo náročný a vyžaduje niektoré základné matematické vedomosti.

V príspevku sú prezentované 3 softvérové produkty, ktoré sa v súčasnosti používajú pri výučbe lineárneho programovania na Ekonomickej univerzite v Bratislave. Samozrejme, ide o prezentáciu softvéru na výučbu, na riešenie výskumných úloh je používaný profesionálny softvér, predovšetkým GAMS, MATLAB atď. Výhody jednotlivých prezentovaných softvérov možno analyzovať na základe cieľov sledovaných pri výučbe.

Pri použití doplnku *Riešiteľ* pod *MS Excelom* je hlavnou výhodou jeho dostupnosť, ale nie je vhodný ako pedagogická pomôcka pri vysvetľovaní podstaty simplexovej metódy. Preto treba pri vysvetľovaní postupu výpočtu pomocou simplexovej metódy vysvetliť jednotlivé kroky riešenia osobitne. V rámci tohto doplnku MS Excelu tiež nie je dostupná zodpovedajúca grafická interpretácia.

Programový balík *POM-QM for Windows* na druhej strane zobrazuje celý postup riešenia, spolu s grafickou interpretáciou, čo je výhodou pri kontrole postupu riešenia študentmi. Jeho nevýhodou je výpočet s desatinnými číslami, čo znižuje jeho numerickú presnosť, problémy pri interpretácii niektorých špeciálnych prípadov riešenia ÚLP (predovšetkým alternatívnosť a degenerovanosť riešenia) a takisto neoposkytuje možnosť jednoducho vysvetliť základné pojmy lineárneho programovania ako sú bázičné riešenie, prípustné bázičné riešenie a podobne.

Ako posledný bol charakterizovaný výsledok spolupráce univerzít v rámci V4, softvér *Simplex V4*. Tento softvér bol vytvorený ako doplnok k existujúcim softvérom s cieľom grafickej aj numerickej interpretácie jednotlivých krokov riešenia ÚLP pomocou simplexovej metódy. Umožňuje samostatnú prácu študentov so znalosťami postupu riešenia bez zložitých numerických výpočtov. Ako prednosť tohto softvéru sa nám javia jeho výsledky v zlomkovom tvare bez zaokrúhľovania, čo zodpovedá výpočtom na báze elementárnej zmeny bázy. Výhodou je aj možnosť grafického zobrazenia jednotlivých riešení, čo umožňuje jednoduché pochopenie základných pojmov lineárneho programovania ako sú bázičné riešenie, prípustné bázičné riešenie a elementárna zmena bázy. Zatiaľ je jeho hlavnou nevýhodou, že automaticky neobsahuje možnosť zobrazenia celého postupu riešenia, teda automatický výber vstupujúcej a vystupujúcej premennej a následnú elementárnu zmenu bázy.

Výber každého výučbového softvéru treba realizovať na základe cieľov výučby. K úspešnému pochopeniu podstaty metód riešenia úloh lineárneho programovania, grafickej metódy a simplexovej metódy, sa javí ako veľmi vhodný softvérový nástroj *Simplex V4*, ktorý núti študentov implementovať teoretické vedomosti získané v priebehu výučby k dosiahnutiu optimálneho riešenia úlohy lineárneho programovania.

**Príspevok bol spracovaný v rámci riešenia grantovej úlohy Visegrad fund ID: 21830031, "SimplexV4 Common Free Tool Development and Day".**

### Literatúra

- [1] Anderson, D. R., Sweeney, D. J., Williams, T. A., Camm, J. D. & Cochran, J. J. (2018). *An introduction to management science: quantitative approach*. Cengage learning.
- [2] Betteridge, D. Wade, A.P. & Howard, A.G. (1985). *Reflections on the modified simplex I, II*. *Talanta*, 32(8), 709-734.
- [3] Brezina, I. & Pekár, J. (2019). *Úvod do operačného výskumu I*. Bratislava: LetraEdu.
- [4] CPLEX (<https://www.ibm.com/products/ilog-cplex-optimization-studio>).
- [5] Drábeková, J. (2019). *Geogebra a grafické riešenie úloh lineárneho programovania*. Nitra: Slovenská poľnohospodárska univerzita v Nitre, Fakulta ekonomiky a manažmentu, Katedra matematiky.
- [6] Džama, J. (2015): *Podpora pre riešenie úlohy lineárneho programovania*. Košice: Univerzita Pavla Jozefa Šafárika v Košiciach, Prírodovedecká fakulta.
- [7] GAMS (General Algebraic Modeling System) (<https://www.gams.com/>).
- [8] Gurobi (<https://www.gurobi.com/>).
- [9] Jablonský, J. (2002). *Operační výzkum*. Praha: Professional Publishing.
- [10] Lingo (<https://www.lindo.com/index.php/products/lingo-and-optimization-modeling>).
- [11] Nelder, J.A. & Mead, R. (1965). *A simplex method for function minimization*. *Comput. J.*, 7, pp. 308–313.
- [12] Routh, M.W., Swartz, P.A. & Dentón, M.B. (1977). *Performance of the super modified simplex*. *Analyt. Chem.* 49, pp. 1422-1428.
- [13] Stevenson, W. J., & Sum, C. C. (2014). *Operations management* (Vol. 10). New York, NY: McGraw-Hill/Irwin.
- [14] Taha, H. A. (2011). *Operations research: an introduction* (Vol. 790). Upper Saddle River, NJ, USA: Pearson/Prentice Hall.
- [15] Visual Math (<http://www.visualmath.com/>)
- [16] Weiss, H. J. (2009). *POM-QM for Windows: including POM for Windows and QM for Windows*.
- [17] [windowsreport.com/linear-programming-software](http://windowsreport.com/linear-programming-software).