EKONOMICKÁ UNIVERZITA V BRATISLAVE

FAKULTA HOSPODÁRSKEJ INFORMATIKY

Evidenčné číslo: 103003/B/2023/36124048426170116

Riešenie úloh lineárneho programovania s využitím softvérových nástrojov

Bakalárska práca

2023 **Alžbeta Dubovská**

EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY

Riešenie úloh lineárneho programovania s využitím softvérových nástrojov

Bakalárska práca

Študijný program: Data science v ekonómii
Študijný odbor: Ekonómia a manažment
Školiace pracovisko: Katedra operačného výskumu a ekonometrie
Vedúci záverečnej práce: doc. Ing. Michaela Chocholatá, PhD.

Bratislava 2023 **Alžbeta Dubovská**

ABSTRAKT

DUBOVSKÁ, Alžbeta: *Riešenie úloh lineárneho programovania s využitím softvérových nástrojov*. – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra operačného výskumu a ekonometrie. – Vedúci záverečnej práce: doc. Ing. Michaela Chocholatá, PhD. Bratislava: FHI, 2023, 52 strán.

Cieľom bakalárskej práce je priblížiť oblasť lineárneho programovania, predstaviť vybrané typy úloh z tejto oblasti a poukázať na možnosti ich riešenia s využitím vybraných softvérových nástrojov. Bakalárska práca je rozdelená do piatich kapitol. Obsahuje 7 tabuliek a 12 obrázkov. Prvá kapitola je venovaná operačnému výskumu, matematickému programovaniu, lineárnemu programovaniu a metódam riešenia úloh lineárneho programovania. Predmetom druhej kapitoly je formulácia cieľa práce, tretia kapitola poskytuje prehľad o metodike práce a použitých metódach skúmania – predstavená je formulácia úloh matematického programovania s osobitným dôrazom na úlohy lineárneho programovania, pozornosť je venovaná tiež vybraným typom úloh lineárneho programovania vrátane vybraných možností ich softvérového riešenia. V štvrtej kapitole sú prezentované výsledky riešenia konkrétnych úloh lineárneho programovania vrátane popisu ich softvérového riešenia. Piata kapitola je venovaná diskusii. Bakalársku prácu uzatvára časť s názvom Záver.

Kľúčové slová: operačný výskum, matematické programovanie, lineárne programovanie

ABSTRACT

DUBOVSKÁ, Alžbeta: *Solving of linear programming problems with the use of software tools.* – University of Economics in Bratislava. Faculty of Business Informatics; Department of Operations Research and Econometrics. – The supervisor: doc. Ing. Michaela Chocholatá, PhD. Bratislava: FHI, 2023, 52 pages.

The aim of this bachelor thesis is to zoom in on area of linear programming, introduce chosen types of tasks from this area and focus on possibilities of their solving with use of chosen software tools. Bachelor thesis is divided into five chapters. It includes 7 tables and 12 pictures. The first chapter is dedicated towards operational research, mathematical programming, linear programming and methods of solving tasks of linear programming. The subject of the second chapter is formulating the aim of the thesis, third chapter has overview of methodology of the thesis and used methods of investigation - introduced is also the formation of tasks of mathematical programming with special emphasis on tasks of linear programming, attention is also given to chosen types of tasks of linear programming including chosen options of software solving. In the fourth chapter we present the results of solving chosen tasks of linear programming including description of their software result. The fifth chapter is about discussion. The bachelor thesis is ended with a part called Conclusion.

Key words: operations research, mathematical programming, linear programming

Obsah

Úvod

V dnešnej dobe všetko niečo stojí či už čas, alebo peniaze. Firmy sa snažia nájsť správny pomer medzi minimalizáciou nákladov a maximalizáciou tržieb, tak aby dosahovali maximálne zisky. Spotrebitelia sa zase snažia nájsť správny pomer medzi hodnotou voľného času a času určeného na prácu, tak aby mali dostatok peňažných prostriedkov na nákup čo najväčšieho množstva tovarov a služieb za čo najmenšiu cenu. Vo všeobecnosti sa všetky subjekty národného hospodárstva snažia rozhodnúť medzi viacerými možnými riešeniami v rozmedzí určitých kritérií a nájsť optimálne riešenie. Otázkami optimalizácie sa zaoberá napríklad lineárne programovanie.

Lineárne programovanie je špeciálny typ matematického programovania, t. j. hľadanie optimálnych riešení rozhodovacích problémov za predpokladu dodržania obmedzujúcich podmienok. Teda jednou z metód operačného výskumu.

V tejto bakalárskej práci si špecifikujeme každú zo spomínaných oblastí: lineárne programovanie, matematické programovanie aj operačný výskum. Na začiatku spomenieme podstatu a históriou operačného výskumu. Potom charakterizujeme matematické programovanie a s ním spojené pojmy: matematické modely a fázy rozhodovacieho procesu.

Následne definujeme lineárne programovanie a metódy jeho riešenia. Pozrieme sa na tvorbu formulácií úloh matematického programovania a úloh lineárneho programovania všeobecne, ale aj pre konkrétne typy úloh lineárneho programovania, ktoré neskôr použijeme v praktickej časti. Tiež si bližšie popíšeme vybraté spôsoby riešenia týchto úloh, a to doplnok Riešiteľ v MS Exceli a funkciu linprog z knižnice SciPy jazyka Python.

V praktickej časti budeme riešiť konkrétne typy úloh lineárneho programovania, pre ktoré sme zadefinovali formulácie. Riešiť ich budeme vybranými softvérovými nástrojmi a následne porovnáme výsledky získané pomocou nich.

1 Súčasný stav riešenej problematiky doma a v zahraničí

V prvej kapitole sa budeme venovať definovaniu a charakteristikám jednotlivých pojmov. Predstavíme si pojem operačný výskum, jeho vlastnosti a históriu. Ďalej si charakterizujeme matematické programovanie a jeho špeciálny typ, lineárne programovanie. Nakoniec si zadefinujeme metódy riešenia úloh lineárneho programovania, konkrétne simplexovú metódu.

1.1 Operačný výskum

Podľa Jablonského (2007) operačný výskum nie je len samostatná vedná disciplína, ale skôr pozostáva zo súboru viacerých samostatných disciplín zameraných na analýzu rôznych rozhodovacích problémov. Fábry (2011) dodáva, že tieto rozhodovacie problémy sa riešia väčšinou v zmysle nájdenia najlepšieho, tzv. o*ptimálneho riešenia*.

Singla (2016) v medzinárodnom internetovom časopise uvádza, že operačný výskum je analytická, logická a systematická metóda riešenia problémov a rozhodovania, ktorá je nápomocná pri riadení organizácií. Rozdeľuje termín operačný výskum na pojmy operačný a výskum. Pojem operačný opisuje ako fungovanie v určitom slede alebo pripravenosť na použitie. Slovo výskum, podľa neho znamená systematické skúmanie a štúdium materiálov a zdrojov, s cieľom zistiť fakty a dospieť k novým záverom. Čiže operačný výskum je aplikácia vedeckých a matematických techník na štúdium a analýzu problémov, ktorá nám poskytuje údaje na prijatie rozhodnutia (Singla, 2016).

Podstatu operačného výskumu je možné lepšie pochopiť prostredníctvom termínu: „výskum operácií“. Z tohto názvu je zrejmé, že operačný výskum sa aplikuje všade tam, kde ide o problémy týkajúce sa spôsobu vedenia a koordináciu operácii (t. j. činností) v rámci organizácie. Povaha organizácie je pri tom nepodstatná, operačný výskum sa môže uplatniť kdekoľvek, napríklad vo výrobe, doprave, stavebníctve, telekomunikáciách, finančnom plánovaní, zdravotnej starostlivosti, armáde či verejných službách (Hillier, Lieberman, 2010).

Vidíme, že operačný výskum je veľmi rozsiahly, keďže sa dá aplikovať na širokú škálu oblastí a každá z nich má svoje špecifické problémy. Preto neexistuje jednoznačná definícia, ktorá by úplne vystihovala túto „vednú disciplínu“. V zahraničnej literatúre sa používajú

alternatívne názvy: Operational Research (v Anglicku), Operations Research (v Amerike), Management Science, Operations Analysis, Quantitative Analysis. V Českej republike sa ujal alternatívny názov operačná analýza, na Slovensku sa používajú názvy operačný výskum i operačná analýza (Fábry, 2011).

Viacerí autori sa zhodujú na týchto základných vlastnostiach operačného výskumu (Agarwal, 2019b; Khan, 2019):

- **Systémová orientácia**. Operačný výskum študuje situáciu alebo problém ako celok. To znamená, že akákoľvek akcia alebo aktivita, má určitý účinok na zvyšok organizácie. Optimálny výsledok jednej časti systému nemusí byť optimálny pre inú časť. Preto na vyhodnotenie akéhokoľvek rozhodnutia je potrebné identifikovať všetky možné interakcie a určiť ich vplyv na organizáciu ako celok.
- **Interdisciplinárny tímový prístup**. Operačný výskum má interdisciplinárny charakter a vyžaduje tímový prístup k riešeniu problému. Manažérske problémy obsahujú ekonomické, fyzické, psychologické, biologické, sociologické a inžinierske aspekty. Žiadny jednotlivec nemôže mať dôkladné znalosti o všetkých z nich. Vyžaduje si to množstvo ľudí, ktorí majú odborné znalosti v oblasti matematiky, štatistiky, inžinierstva, ekonómie, manažmentu, informatiky atď.
- **Vedecký prístup**. Na riešenie problémov operačný výskum využíva vedecké metódy. Aplikuje ich za účelom analýzy a riešenia zložitých problémov. Je to formalizovaný proces uvažovania, kde treba jasne definovať problém. Teda v tomto prístupe nie je miesto pre dohady a osobnú zaujatosť osoby s rozhodovacou právomocou.
- **Rozhodovanie**. Pomocou operačného výskumu sa zvyšuje efektivita manažérskych rozhodnutí. Operačný výskum môžeme teda považovať za rozhodovaciu vedu, ktorá pomáha manažmentu robiť lepšie rozhodnutia. Hlavným predpokladom operačného výskumu je rozhodovanie bez ohľadu na to, aká je daná situácia.

  Rozhodovanie je systematický proces a pozostáva z nasledujúcich krokov:

  a) Diagnóza problému a stanovenie kritéria. Kritériom môže byť maximalizácia ziskov, užitočnosť a minimalizácia nákladov atď.
  b) Výber alternatívneho postupu na posúdenie.
  c) Určenie modelu, ktorý sa má použiť.

d) Hodnotenie rôznych alternatív.
e) Výber najlepšej, t. j. optimálnej alternatívy.

- **Používanie počítača**. Na vyriešenie zložitého matematického modelu alebo na vykonanie veľkého počtu výpočtov je nápomocné použitie počítača. Takže počítač je neoddeliteľnou súčasťou prístupu operačného výskumu.
- **Ciele**. Operačný výskum sa vždy snaží nájsť optimálne riešenie problému. Na tento účel sú definované a analyzované ciele organizácie. Tieto ciele sa potom použijú ako základ na porovnanie alternatívnych postupov.
- **Kvantitatívne riešenie**. Poskytnutím kvantitatívneho základu pre rozhodovanie operačný výskum podporuje manažment. Cieľom je poskytovať systematický a vedecko-racionálny prístup ku kvantitatívnym riešeniam rôznych manažérskych problémov.
- **Ľudský úsudok**. Ľudské faktory zohrávajú dôležitú úlohu pri určovaní kvantitatívnych riešení. V niektorých problémoch sú prehliadané, čo je chyba, štúdie operačného výskumu sa bez ľudských faktorov nezaobídu.

1.1.1 História operačného výskumu

Prvé náznaky operačného výskumu sa objavili počas druhej svetovej vojny. Mnohé strategické a taktické problémy spojené s vojenským úsilím boli príliš komplikované na to, aby sa dalo očakávať adekvátne riešenie od jednotlivca alebo dokonca od jednej disciplíny. V reakcii na tieto zložité problémy boli vedci s rôznym vzdelaním zoskupení do špeciálnych jednotiek v rámci ozbrojených síl. Tieto tímy vedcov začali spolupracovať, aplikovali svoje interdisciplinárne znalosti na riešenie takých problémov, ako je rozmiestnenie radarov, riadenie protilietadlovej paľby, rozmiestnenie lodí na minimalizáciu strát z nepriateľských ponoriek a stratégie protivzdušnej obrany. Keďže tieto tímy boli vo všeobecnosti pridelené veliteľom zodpovedným za vojenské operácie, boli nazývané „tímy operačného výskumu“. Prirodzene disciplína dostala názov operačný výskum (Ravindran, 2008).

Po vojne nastal priemyselný boom, do popredia sa dostávali problémy spôsobené narastajúcou zložitosťou a špecializáciou organizácii. Mnohí vedci, ktorí pracovali v jednotkách vojenského operačného výskumu, začali aplikovať metódy operačného výskumu na riešenie problémov rôznych odvetví v týchto organizáciách. Vďaka tomu nasledoval podstatný pokrok pri zdokonaľovaní techník operačného výskumu. Skvelým

príkladom je *simplexová metóda* na riešenie problémov lineárneho programovania, ktorú vyvinul George Dantzig v roku 1947. Mnohé ďalšie štandardné nástroje operačného výskumu boli relatívne dobre vyvinuté pred koncom 50. rokov 20. storočia (Hillier, Lieberman, 2010).

Ďalší faktor, ktorý dal impulz rastu tohto odboru, bol nápor počítačovej revolúcie. Na riešenie zložitých problémov je zvyčajne potrebné enormné množstvo výpočtov. Robiť ich ručne by trvalo veľmi dlho a to nepripadalo do úvahy. Preto obrovským prínosom bol vývoj elektronických digitálnych počítačov so schopnosťou vykonávať aritmetické výpočty miliónkrát rýchlejšie, než to dokáže ľudská bytosť (Hillier, Lieberman, 2010).

Pokiaľ sa na rozvoj operačného výskumu pozrieme cez jednotlivé štáty, hlavným strediskom sa stali Spojené štáty americké a nie Veľká Británia, v ktorej boli založené tímy operačného výskumu. Keďže priemysel ešte nebral operačný výskum vážne, rozhodujúci bol rozdielny prístup oboch krajín ku výskumu obrany. Anglicko znižovalo výdavky na výskum obrany, čo viedlo k prepúšťaniu mnohých pracovníkov operačného výskumu z armády. Naopak USA po vojne zvýšilo svoj záujem o výskum obrany. Potvrdilo sa to vznikom novej spoločnosti MORS (Military Operations Research Society), ktorá neskôr úzko spolupracovala s tiež novozaloženou spoločnosťou ORSA (The Operations Research Society of America). Začalo narastať povedomie o operačnom výskume a jeho aplikáciou do priemyslu vznikol nový pojem *management science*, na základe ktorého sa vytvorila národná spoločnosť TIMS (The Institute of Management Sciences) na podporu vedeckých poznatkov v chápaní a praxi manažmentu. Neskôr sa ORSA a TIMS zlúčili do INFORMS (Institute of Operations Research and Management Sciences), ktorý funguje dodnes (Ravindran, 2008; Theintactone, 2019).

Postupom času začali vznikať národné spoločnosti operačného výskumu takmer v každej krajine. Európske krajiny vytvorili asociáciu spoločností operačného výskumu EURO, ktorej členom je aj Slovenská spoločnosť pre operačný výskum SSOR. Všetci členovia EURA a ostatné národné spoločnosti operačného výskumu nepatriace sem, zhrňuje organizácia IFORS (The International Federation of Operational Research Societies). Založená bola v roku 1959 na konferencii, ktorej účastníkmi boli už spomínaná spoločnosť ORSA, spoločnosť Veľkej Británie ORS a Francúzska spoločnosť SOFRO. Hlavným účelom

IFORS je rozvoj operačného výskumu ako jednotnej vedy a jeho šírenie vo všetkých národoch sveta (IFORS, 2023; EURO, 2023).

1.2 Matematické programovanie

Hneď na začiatok treba zdôrazniť, že matematické programovanie nie je programovanie v zmysle kódovanie. Nástup matematického programovania nastal v počiatkoch rozvoja výpočtových zariadení a slovo „program“ sa nevzťahoval na počítačové programy, ako sú známe dnes, ale skôr na navrhovaný plán logistických operácií. Preto vznikol trochu mätúci názov matematické programovanie. Čiže v termíne matematické programovanie je „programovanie“ brané ako „plánovanie“ (Williams, 2013; Singh, Eisner, 2023).

Matematické programovanie je jednou zo základných metód operačného výskumu. Ide o transformovanie reálnych ekonomických operácii a procesov do matematických modelov. Model vo všeobecnosti je zjednodušený obraz reálneho systému (Brezina, Ivaničová, Pekár, 2007).

Matematické modely majú tieto výhody (Jablonský, 2007):

- ich použitie umožní vytvoriť štruktúru systému a špecifikuje všetky možné varianty stavu systému, ktorých môže byť často neobmedzené množstvo,
- modely poskytujú možnosť analyzovať správania systému v skrátenom čase – procesy trvajúce v realite dni, mesiace či roky, môžu byť simulované na počítačoch v zlomkoch sekúnd,
- s modelmi ide ľahko manipulovať a vykonávať veľký počet experimentov pomocou zmien ich parametrov,
- s konštrukciou a využitím modelu sú síce tiež spojené náklady, ale sú o mnoho menšie, skoro až zanedbateľné oproti nákladom vznikajúcich pri experimentovaní s reálnym systémom.

Williams (2013) uvádza, že motívov na tvorbu matematických modelov je viacero:

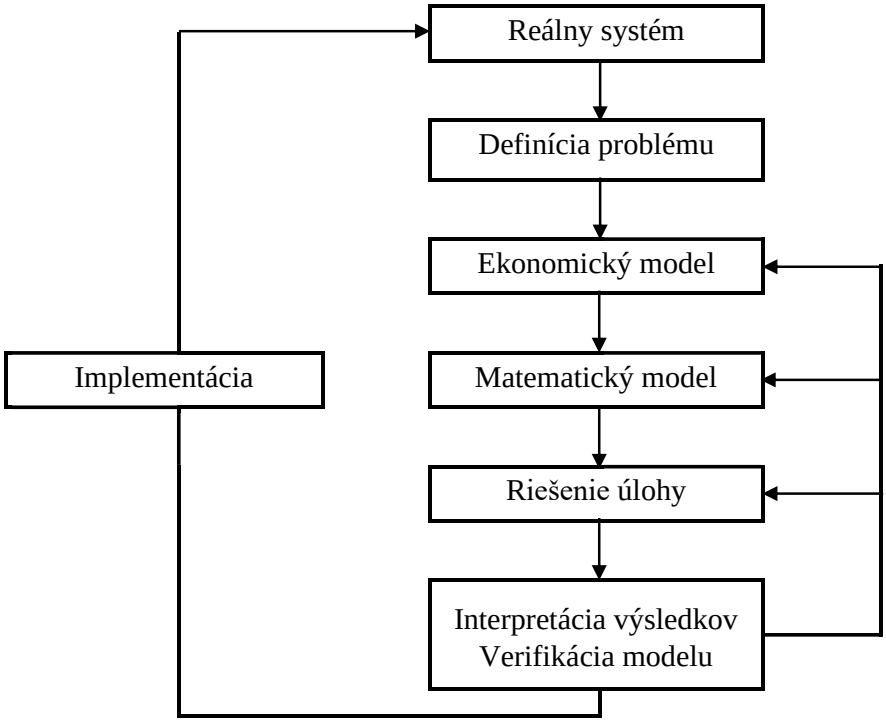
1. Skutočné budovanie modelu často odhalí vzťahy, ktoré mnohým ľuďom neboli zrejmé. Výsledkom je, že sa dosiahne lepšie pochopenie modelovaného objektu.

2. Po zostavení modelu je zvyčajne možné ho matematicky analyzovať, aby sme pomohli navrhnúť postupy, ktoré by inak nemuseli byť zrejmé.
3. S modelom je možné experimentovať, pričom väčšinou nie je možné, priam až nežiaduce experimentovať s objektom, pre ktorý tvoríme model. Bolo by zjavne politicky ťažké a tiež nevhodné experimentovať s nekonvenčnými ekonomickými opatreniami v krajine, ak by existovala vysoká pravdepodobnosť katastrofálneho zlyhania. Snaha o takéto odvážne experimenty by bola prijateľnejšia na matematickom modeli.

V rámci operačného výskumu, resp. matematického programovania sa možno stretnúť s nasledujúcim rozdelením modelov (Agarwal, 2019a):

- podľa účelu /prospešnosti:
    - deskriptívny/opisný – opisuje nejaký aspekt situácie na základe pozorovaní, prieskumov, výsledkov dotazníkov alebo iných dostupných údajov. Napr. výsledok prieskumu verejnej mienky,
    - prediktívny/predpovedajúci – odpovedá na otázku typu „čo ak?“, t. j. môže predpovedať určité udalosti,
    - normatívny/nariaďujúci – ak bol prediktívny model opakovane úspešný, možno ho určiť ako „overený zdroj“. Na základe neho sa môžu vytvárať odporúčania až nariadenia, čo by sa malo urobiť, aby sa dosiahol určitý cieľ.
- podľa povahy prostredia:
    - deterministický – predpokladá podmienky úplnej istoty a dokonalého poznania,
    - stochastický/pravdepodobnostný – týka sa situácií, v ktorých nie je možné s istotou predvídať dôsledky alebo návratnosť určitých krokov. Je však možné predpovedať rámec udalostí, ktoré môžu pri daných krokoch nastať.
- podľa správania:
    - statický – nezohľadňuje vplyv zmien, ku ktorým dochádza v horizonte plánovania, t. j. je nezávislý od času. Tiež v takomto modeli je potrebné iba jedno rozhodnutie počas trvania daného časového obdobia,
    - dynamický - považuje čas za jednu z dôležitých premenných a pripúšťa vplyv zmien generovaných časom. V dynamických modeloch sa v horizonte plánovania vyžaduje nielen jedno, ale séria vzájomne závislých rozhodnutí.

Lepšie porozumieť pojem *matematické programovanie* možno pomocou popisu jednotlivých fáz rozhodovacieho procesu (Obr. 1.1). V prvom rade si zadefinujeme dve základné role: *rozhodovací subjekt a analytik*. Rozhodovací subjekt, resp. rozhodovateľ je ten, ktorý má rozhodovaciu právomoc a zadáva problém potrebný vyriešiť. Analytik je riešiteľom daného problému. Ak analytik nájde riešenie predstaví ho rozhodovaciemu subjektu. Ten sa ho rozhodne prijať alebo ho odmietne a vráti analytikovi spolu s poopraveným zadaním problému na prepracovanie. Je jasné, že komunikácia medzi rozhodovacím subjektom a analytikom hrá významnú úlohu (Fábry, 2011).



Obr. 1.1 – Priebeh rozhodovacieho procesu, Zdroj: (Fábry, 2011)

Rozhodovací proces môže mať tieto fázy (Fábry, 2011; McCombes, George, 2022; Plevný, Žižka, 2010):

1. **Definovanie problému**. Najprv treba rozpoznať existenciu problému v reálnom systéme. Jeho včasné zistenie môže rozhodovaciemu subjektu pomôcť zabrániť už nenávratným dôsledkom alebo len ušetriť finančné prostriedky. Samozrejme, že nestačí na problém len upozorniť, ale je potrebné dokázať problém jasne a presne definovať.

   Vo svete podnikania tento krok zodpovedá napísaniu správy o probléme, v ktorej daný problém dávame do určitého kontextu. Pri praktickom výskume sa zodpovedajú otázky

typu: Kde a kedy problém vzniká? Koho sa problém týka? Aké pokusy boli podniknuté na vyriešenie problému? Z teoretického hľadiska sa pozerá na to, čo už je o probléme známe a ako je definovaný v odborných literatúrach.

2. **Formulovanie ekonomického modelu**. Ako bolo vyššie spomínané, model je zjednodušený obraz reality, takže popis daného problému nesmie byť príliš zložitý a mal by vystihovať hlavne jeho podstatné rysy.

   Ekonomický model je verbálny popis problému v reálnom systéme. Najprv potrebujeme vyjadriť cieľ, ktorý chceme dosiahnuť. Potom procesy, ktorými možno ovplyvniť výsledný efekt. Nakoniec činitele, ktoré akýmkoľvek spôsobom obmedzujú realizáciu týchto procesov a my ich v rámci riešenia daného problému nevieme ovplyvniť.
3. **Formulovanie matematického modelu**. Matematický model predstavuje zjednodušené formalizované zobrazenie ekonomických zákonitosti pomocou matematických vzťahov. Čiže, ak ekonomický model zapíšeme pomocou matematických vzťahov, dostávame matematický model. Jednotlivé časti ekonomického modelu sa stávajú parametrami, premennými, funkciami, rovnicami, nerovnicami a pod.
4. **Riešenie úlohy**. Akonáhle je problém vyjadrený pomocou matematických vzťahov, nie je komplikované získať jeho riešenie, ktoré je už len technickou záležitosťou. Vyberie sa jeden zo štandardných algoritmov operačného výskumu a ten sa aplikuje pomocou niektorého z množstva ľahko dostupných softvérových balíkov.
5. **Interpretácia výsledkov a verifikácia modelu**. Ak sa získané riešenie úlohy z predchádzajúceho kroku vyjadrí slovne, alebo sa vysvetlia číselné výsledky, hovoríme o *interpretácií výsledkov*. Výstup je zväčša v podobe počítačovej zostavy a je na analytikovi nájsť v nej odpovede na to, ako by sa mal riešiť daný problém. Následne analytik tieto výsledky odprezentuje termínmi použitými v ekonomickom modeli. Nepoužíva iné termíny či prvky, ktoré zaviedol počas formulácie matematického modelu. Robí to z dôvodu, aby im rozhodovací subjekt rozumel.

   Pred odprezentovaním výsledkov nasleduje *verifikácia modelu.* Pod verifikáciou sa rozumie overenie správnosti a reálnosti získaných výsledkov zostaveného modelu. Veľa analytikov v prípade získania takých výsledkov, ktoré u nich vyvolávajú uspokojenie so svojou prácou, tento krok vynechávajú alebo podcenia, čo môže viesť k nežiadúcim následkom. Aj u jednoduchších modelov môžu byť výsledky ovplyvnené chybnou formuláciou modelu. Takže skúsený analytik venuje pozornosť všetkým výsledkom,

a ak nájde nezrovnalosti, vráti sa späť ku kroku, kde zadával matematický model do softvéru, prípadne až k samotnej formulácii matematického modelu.

6. **Implementácia**. Zavŕšením celého rozhodovacieho procesu je zavedenie výsledkov do reálneho systému. Ak teda rozhodovací subjekt získal od analytika výsledky, je len na ňom uviesť ich do praxe. Cieľom implementácie je zlepšiť fungovanie systému. Teda nasleduje ešte jeden dôležitý krok a tým je kontrola. Kontroluje sa či urobená zmena naozaj priniesla očakávaný prínos.

Matematické programovanie zahŕňa teóriu, použitie a výpočtové riešenie matematických modelov, ktoré pomáhajú pri rozhodovaní. Podľa týchto modelov vieme bližšie špecifikovať, o ktorý druh matematického programovania presne ide. Ak tieto modely zahŕňajú iba lineárne funkcie, t. j. lineárne programovanie; modely, v ktorých všetky premenné musia mať celočíselné hodnoty, t. j. celočíselné programovanie; modely zahŕňajúce všeobecnejšie funkcie, t. j. nelineárne programovanie; modely so spojitými aj diskrétnymi hodnotami, t. j. zmiešané celočíselné programovanie; a modely zahŕňajúce náhodné premenné, t. j. stochastické programovanie (Singh, Eisner, 2023).

Vyššie bolo uvedené, že lineárne programovanie je súčasťou matematického programovania. Potvrdzujú to aj Brezina s Pekárom (2018, s. 17) svojím tvrdením: „Matematické programovanie je riešenie optimalizačných úloh s cieľom nájsť extrém účelovej funkcie na základe stanoveného cieľa (kritéria) pri určitých ohraničujúcich podmienkach. Ak sú všetky vzťahy v úlohe matematického programovania lineárne (účelová funkcia aj ohraničujúce podmienky), hovoríme o lineárnom programovaní“.

1.3 Lineárne programovanie

Lineárne programovanie je jednoduchým prístupom, ktorý sa používa na zobrazenie komplikovaných vzťahov v reálnom svete pomocou lineárnej funkcie. Vykonáva sa lineárna optimalizácia tak, aby sa dosiahol najlepší výsledok minimalizáciou, alebo maximalizáciou tejto funkcie. Všetky prvky v matematickom modeli majú navzájom lineárny vzťah. Daná lineárna funkcia, alebo inak povedané účelová funkcia je lineárna v neznámych a obmedzenia pozostávajú z lineárnych rovníc a lineárnych nerovníc, resp. ich sústav (Luenberger, Ye, 2016).

Znamená to, že sa v modeli nesmie vyskytovať žiadna nelineárna závislosť použitých premenných. Inými slovami (Plevný, Žižka, 2010):

- všetky premenné sa vyskytujú len v prvej mocnine,
- nie sú argumentom žiadnej funkcie (logaritmické, mocninné, exponenciálne atď.),
- nesmú sa násobiť medzi sebou.

V tabuľke 1.1 môžeme vidieť niektoré výhody a nevýhody lineárneho programovania.

Výhody
- umožňuje čo najlepšie využiť aktíva podniku. To znamená, že poskytuje informáciu, ako optimálne využiť výrobné faktory (práca, pôda a kapitál) a ako a kam ich rozdeliť. - zlepšuje kvalitu rozhodovania. Rozhodovací prístup používateľa sa v dôsledku použitia tejto techniky stáva objektívnejším a menej subjektívnym. - odráža bariéry vo výrobnom procese. Napríklad určitý stroj nedokáže uspokojiť dopyt, zatiaľ čo iný áno a vie aj ušetriť čas.
Nevýhody
- účelová funkcia a ohraničenia musia byť lineárne. - neexistuje záruka celočíselného riešenia. - neberie ohľad na vplyvy neistoty a času.

Tab. 1.1 – výhody a nevýhody lineárneho programovania, Zdroj: Agarwal, 2022

1.4 Metódy riešenia úloh lineárneho programovania

Rozhodujúcim medzníkom v histórii riešenia úloh lineárneho programovania sa stal rok 1947, kedy George Dantzig vyvinul simplexovú metódu. Počas druhej svetovej vojny jeho modely lineárneho programovania pomohli vojenským silám s problémami v doprave a plánovaní. Od objavenia simplexovej metódy sa viacerí autori pokúšali prispieť do tejto oblasti. Napríklad v roku 1979 sovietsky vedec Leonid Khachian vyvinul elipsoidovú metódu, ktorá mala byť revolučná. Časom sa však ukázalo, že táto metóda, nebola o nič lepšia ako simplexová metóda. V roku 1984 vedecký pracovník Narendra Karmarkar vyvinul Karmarkarov algoritmus, alebo inak nazývaný algoritmus vnútorného bodu, ktorý sa ukázal byť štyrikrát rýchlejší ako simplexová metóda pre určité problémy. Ale simplexová metóda stále funguje najlepšie pre väčšinu problémov a používa sa dodnes (Sekhon, Bloom, 2022).

Simplexová metóda sa používa na hľadanie optimálneho riešenia úloh lineárneho programovania. Predstavuje iteračný postup, ktorý skúma len bázické a nie všetky prípustné riešenia. Po určitom počte krokov (iterácií) určí optimálne riešenie alebo poskytne informáciu, že optimálne riešenie neexistuje. V každej iterácii sa testuje optimálnosť získaného prípustného bázického riešenia. Ak nie je optimálne, algoritmus pristúpi ku skúmaniu ďalšieho prípustného bázického riešenia, ktoré nie je horšie ako predchádzajúce skúmané riešenie. Opakovanie postupu sa ukončí, ak prípustné bázické riešenie sa určí ako optimálne alebo sa zistí, že optimálne riešenie neexistuje (Brezina, Pekár, 2018).

Simplexová metóda obsahuje dva základné algoritmy - primárny (priamy) algoritmus a duálny algoritmus. Existuje pár modifikácií ako primárno-duálny algoritmus, primárny revidovaný algoritmus a pod. Ďalej si bližšie zadefinujeme základné typy (Brezina, Pekár, 2018).

Primárny algoritmus funguje na už spomenutom princípe - po konečnom počte krokov sa určí optimálne riešenie alebo sa stanoví, že optimálne riešenie úlohy neexistuje. Duálny algoritmus je postavený na tom, že ku každému matematickému modelu úlohy lineárneho programovania možno určitým spôsobom priradiť iný matematický model, ktorý s ním úzko súvisí. Toto priradenie sa nazýva dualita (podvojnosť, dvojitosť). Pôvodný model, resp. pôvodná úloha lineárneho programovania sa nazýva primárny matematický model, resp. *primárna úloha*. Priradenú úlohu ku pôvodnej úlohe budeme označovať *duálna úloha*. V duálnej úlohe sa rozhodovacie premenné nazývajú *duálne premenné* a označujú sa u_i pre $i = 1, 2, ..., m$ (Chovanová, 2016).

Ku každej primárnej úlohe možno sformulovať duálnu úlohu, alebo naopak (ku duálnej primárnu). Na ich formuláciu vieme použiť pravidlá zobrazené v tabuľkách 1.2 a 1.3, ak ohraničenia matematického modelu primárnej úlohy sú buď rovnice, alebo nerovnice rovnakého typu, poprípade ich upravíme do tohto tvaru (Chovanová, 2016).

Primárna úloha	**Duálna úloha**
maximalizovať	minimalizovať
koeficienty účelovej funkcie	prvky pravej strany
prvky pravej strany	koeficienty účelovej funkcie
i-tý riadok matice technologických koeficientov	*i*-tý stĺpec matice technologických koeficientov

j-ty stĺpec matice technologických koeficientov	j-ty riadok matice technologických koeficientov
i-té ohraničenie typu „≤“	i-tá premenná $u_i \geq 0$
i-té ohraničenie typu „≥“	i-tá premenná $u_i \leq 0$
i-té ohraničenie typu „=“	i-tá premenná voľná
j-ta premenná $x_i \leq 0$	j-te ohraničenie typu „≤“
j-ta premenná $x_i \geq 0$	j-te ohraničenie typu „≥“
j-ta premenná voľná	j-te ohraničenie typu „=“

Tab. 1.2 - pravidlá pre primárnu maximalizačnú úlohu a duálnu minimalizačnú úlohu,

Zdroj: Brezina, Pekár, 2018

Primárna úloha	**Duálna úloha**
minimalizovať	maximalizovať
koeficienty účelovej funkcie	prvky pravej strany
prvky pravej strany	koeficienty účelovej funkcie
i-tý riadok matice technologických koeficientov	i-tý stĺpec matice technologických koeficientov
j-tý stĺpec matice technologických koeficientov	j-tý riadok matice technologických koeficientov
i-té ohraničenie typu „≤“	i-tá premenná $u_i \leq 0$
i-té ohraničenie typu „≥“	i-tá premenná $u_i \geq 0$
i-té ohraničenie typu „=“	i-tá premenná voľná
j-tá premenná $x_i \leq 0$	j-té ohraničenie typu „≥“
j-tá premenná $x_i \geq 0$	j-té ohraničenie typu „≤“
j-tá premenná voľná	j-té ohraničenie typu „=“

Tab. 1.3 - pravidlá pre primárnu minimalizačnú úlohu a duálnu maximalizačnú úlohu,

Zdroj: Brezina, Pekár, 2018

2 Cieľ práce

Cieľom práce je priblížiť oblasť lineárneho programovania. Predovšetkým sa zamerať na riešenia konkrétnych úloh s využitím softvérových nástrojoch. My sme si vybrali MS Excel a jeho doplnok Riešiteľ a jazyk Python a jeho funkciu linprog z knižnice SciPy.

Cieľom teoretickej časti práce je opísať podstatu a históriu operačného výskumu. Následne charakterizovať matematické programovanie a matematické modely. Postupne sa prepracovať až ku lineárnemu programovaniu a metódam riešenia úloh lineárneho programovania. Potom zadefinovať formulácie úloh matematického programovania a úloh lineárneho programovania všeobecne, ale aj pre tri konkrétne typy úloh lineárneho programovania (úloha výrobného plánovania, úloha o výžive a rezný problém) a popísať vybraté možnosti softvérových riešení týchto úloh.

V praktickej časti ilustrovať riešenia už predtým spomenutých konkrétnych typov úloh lineárneho programovania pomocou doplnku Riešiteľ v MS Exceli a funkcie linprog v jazyku Python. Na záver analyzovať a porovnať riešenia úloh prostredníctvom vybratých softvérových nástrojov.

3 Metodika práce a metódy skúmania

Ďalej sa pozrieme na formulácie úloh matematického a potom lineárneho programovania. Tiež si rozoberieme typy úloh lineárneho programovania, konkrétne úlohu výrobného plánovania, úlohu o výžive a rezný problém. Nakoniec si predstavíme formulácie konkrétnych úloh lineárneho programovania a ukážeme postupy softvérových riešení týchto úloh.

3.1 Formulácia úlohy matematického programovania

Formulácia matematického modelu vychádza z presného verbálneho popisu daného ekonomického problému v podobe ekonomického modelu, jeho hlavné prvky sú uvedené v tabuľke 3.1 (Fábry, 2011).

	Ekonomický model	Matematický model	Príklady
Čo chceme dosiahnuť?	cieľ	účelová funkcia	maximalizácia množstva produkcie; maximalizácia zisku; minimalizácia nákladov; minimalizácia rizika
Čo môžeme ovplyvniť?	procesy	premenné	množstvo produkcie; veľkosť zisku; veľkosť nákladov; využitie strojov; množstvo odpadu
Aké sú prekážky?	činitele	ohraničujúce podmienky	obmedzené materiálové zdroje; finančné zdroje; kapacita výrobných technológií; využitie výrobných zdrojov

Tab. 3.1 – Prvky ekonomického a matematického modelu, Zdroj: Fábry, 2011

Cieľ (napr. zvýšenie zisku podniku) súvisí s hlavným impulzom k vykonaniu analýzy systému a väčšinou pramení z nespokojnosti rozhodovacieho subjektu so súčasným stavom. V určitom hodnotovom ukazovateli sa nájde extrém a ten sa vyjadrí prostredníctvom tzv. *účelovej funkcie (kritéria).* Nástrojmi na dosiahnutie cieľa sú *procesy*, ktoré sú realizované v danom systéme (napr. vyrábané výrobky), a v modeli ich reprezentujú *premenné*. Presnejšie by bolo používať názov *rozhodovacie premenné (decision variables)*, aby sme zdôraznili fakt, že prostredníctvom týchto premenných rozhodovací subjekt

realizuje svoje rozhodnutia doporučené analytikom. Naviac tento prívlastok pomáha odlíšiť ich od ostatných premenných ako sú doplnkové, umelé, fiktívne a pod. *Činitele* (ohraničenia) pôsobia na prebiehajúce procesy a určitým spôsobom ich obmedzujú (napr. množstvo vyprodukovaných výrobkov závisí od množstva materiálu a tiež musí rešpektovať odbytové požiadavky). Vyjadrujú sa pomocou ohraničujúcich podmienok alebo iným názvom štruktúrne ohraničenia. Výsledkom sú výstupy, ktoré predstavujú hodnoty rozhodovacích premenných (Brezina, Pekár, 2018; Fábry, 2011).

Na zapísanie všeobecného modelu matematického programovania, zavedieme tieto označenia (Brezina, Ivaničová, Pekár, 2007):

n - počet rozhodovacích premenných,

m - počet ohraničujúcich podmienok (štruktúrnych ohraničení),

k - počet účelových funkcií,

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

f_s - reálne funkcie rozhodovacích premenných $x_1, x_2, \dots, x_n, s = 1, 2, \dots, k$, ktoré opisujú ciele optimalizácie,

g_i - reálne funkcie rozhodovacích premenných $x_1, x_2, \dots, x_n, i = 1, 2, \dots, m$, ktoré opisujú ohraničujúce podmienky,

D_j - množiny tých hodnôt reálnych čísiel, ktoré môžu nadobúdať rozhodovacie premenné $x_j,\ j = 1, 2, \dots, n$.

Pri zápise úloh matematického programovania postupujeme na základe schémy v tabuľke 3.2.

Sledované ciele	$f_1(x_1, x_2, \dots, x_n)$ $f_2(x_1, x_2, \dots, x_n)$ ... $f_k(x_1, x_2, \dots, x_n)$	Účelové funkcie
Podmienky	$g_1(x_1, x_2, \dots, x_n) \geq 0$ $g_2(x_1, x_2, \dots, x_n) \geq 0$ ... $g_m(x_1, x_2, \dots, x_n) \geq 0$	Štruktúrne podmienky rozhodnutia (štruktúrne ohraničenia)
Špeciálne podmienky pre premenné	$x_j \geq 0$	Podmienky nezápornosti
	$x_j \in D_j$	Dodatočné podmienky

Tab. 3.2 – Schéma modelu matematického programovania, Zdroj: Brezina, Pekár, 2018

Špeciálne podmienky pre premenné vyjadrujú dodatočné predpoklady, ktoré treba pri konštrukcii úlohy matematického programovania dodržať. Podmienky nezápornosti nám hovoria o tom, že hľadané hodnoty rozhodovacích premenných nemôžu nadobúdať záporné hodnoty (nemožno napr. vyrobiť záporné množstvo nejakých výrobkov). Dodatočné podmienky reprezentujú niektoré špeciálne požiadavky, ktoré je nutné zahrnúť do konštrukcie úlohy matematického programovania (Brezina, Ivaničová, Pekár, 2007).

3.2 Formulácia úlohy lineárneho programovania

Lineárne programovanie je optimalizačný problém, v ktorom je účelová funkcia lineárna v neznámych a ohraničenia pozostávajú z lineárnych rovníc a nerovníc. Presná forma týchto ohraničení sa môže líšiť od jedného problému k druhému, ale každú úlohu lineárneho programovania možno transformovať do základného tvaru modelu lineárneho programovania (Luenberger, Ye, 2016).

Základný tvar modelu lineárneho programovania (Brezina, Pekár, 2018):

maximalizovať, resp. minimalizovať

$$f(x) = z = \sum_{j=1}^{n} c_j \, x_j, \tag{3.1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \begin{Bmatrix} \leq \\ = \\ \geq \end{Bmatrix} b_i, \qquad i = 1, 2, \dots m, \tag{3.2}$$

$$x_j \geq 0, \qquad j = 1, 2, \dots n. \tag{3.3}$$

kde

c_j - koeficienty účelovej funkcie, $j = 1, 2, \dots n$,

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

b_i - koeficienty pravej strany, $i = 1, 2, \dots m$,

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

Pri riešení zodpovedajúcich úloh lineárneho programovania hľadáme takú hodnotu rozhodovacích premenných, aby hodnota účelovej funkcie (3.1) bola maximálna (resp. minimálna) pri splnení podmienok (3.2) a (3.3). Podmienky (3.2) vyjadrujú činitele definované v ekonomickom modeli úlohy. Každému činiteľovi zodpovedá jedna rovnica alebo nerovnica. Tiež ich označujeme ako vlastné obmedzenia. Okrem vlastných obmedzení

poznáme ďalšie podmienky (3.3), ktoré zabezpečujú nezápornosť všetkých premenných. Premenné v matematickom modeli majú takú ekonomickú interpretáciu, že by záporné hodnoty neboli vhodné (napr. objem produkcie). Tieto podmienky sa nazývajú podmienky nezápornosti (Jablonský, 2007; Brezina, Pekár, 2018).

Maticový zápis modelu lineárneho programovania (Jablonský, 2007):

maximalizovať, resp. minimalizovať

$$z = \mathbf{c}^{\mathrm{T}}\mathbf{x},$$

$$\mathbf{A}\mathbf{x}\begin{Bmatrix}\leq\\=\\\geq\end{Bmatrix}\mathbf{b},$$

$$\mathbf{x} \geq \mathbf{0}.$$

kde

$\mathbf{c}^{\mathrm{T}} = (c_1, c_2, \dots, c_n)$ je n – rozmerný riadkový vektor koeficientov účelovej funkcie,

$\mathbf{x} = (x_1, x_2, \dots, x_n)^{\mathrm{T}}$ je n – rozmerný stĺpcový vektor rozhodovacích premenných,

$\mathbf{b} = (b_1, b_2, \dots, b_m)^{\mathrm{T}}$ je m – rozmerný stĺpcový vektor koeficientov pravej strany,

$\mathbf{0} = (0, 0, \dots, 0)^{\mathrm{T}}$ je n – rozmerný stĺpcový nulový vektor,

A je matica technologických koeficientov sústavy ohraničení rozmerov *m*n*.

3.3 Typy úloh lineárneho programovania

V praxi sa stretávame s rôznymi problémami, ktoré sa definujú ako úlohy lineárneho programovania. Fábry (2011) zaraďuje medzi základné typy úloh lineárneho programovania nasledujúce:

- Úloha výrobného plánovania
- Úloha o výžive
- Rezný problém
- Optimalizácia portfólia
- Dopravná úloha
- Priraďovací problém

Ďalej sa pozrieme špeciálne na prvé tri typy úloh a ich formulácie.

3.3.1 Úloha výrobného plánovania (The Production Planning)

Stredobodom tejto úlohy je riadiaci alebo výrobný systém, povedzme napr. v továrni alebo rafinérii. Komodity ako suroviny, materiál, kapitál a práca sú vstupy, ktoré pôsobením rôznych výrobných procesov sa premenia na výstup, zväčša výrobok. Základným problémom je tento systém prevádzkovať tak, aby sa maximalizoval zisk s použitím obmedzených zdrojov alebo minimalizovali náklady pri splnení špecifikovaných výrobných požiadaviek, alebo určitá kombinácia týchto cieľov (Thie, Keough, 2008).

Špeciálnym prípadom tejto úlohy je *úloha výrobného plánovania s polotovarmi*. Plánuje sa v nich nielen produkcia výrobkov, ale aj polotovarov, ktoré buď slúžia k ďalšej výrobe, alebo k priamemu predaju (Fábry, 2011).

Všeobecná formulácia úlohy výrobného plánovania ma takýto tvar (Brezina, Ivaničová, Pekár, 2007):

$$\max\ z(x) = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad i = 1, 2, \dots, m$$

$$x_j \geq 0 \qquad j = 1, 2, \dots, n$$

kde

n - počet druhov výrobkov,

m - počet výrobných faktorov (surovín, kapacít strojov, pracovných hodín a pod.), ktoré sa pri výrobe používajú,

x_j - množstvo produkcie j-tého výrobku, $j = 1, 2, \dots, n$,

c_j - ocenenie j-tého výrobku (jednotkový zisk, cena výrobku a pod.), $j = 1, 2, \dots, n$,

a_{ij} - počet jednotiek i-tého výrobného faktora spotrebovaného na výrobu jednotky j-tého výrobku, $i = 1, 2, \dots, m,\ j = 1, 2, \dots, n$,

b_i - disponibilné množstvo i-tého výrobného faktora, $i = 1, 2, \dots, m$.

3.3.2 Úloha o výžive (The Diet Problem)

Úloha o výžive je jednou z klasických ilustrácií problému, ktorý vedie k modelu lineárneho programovania. Problém sa týka poskytovania stravy s minimálnymi nákladmi,

ktorá je dostatočná na to, aby sa človek uživil. Jednoducho povedané, aký je najlacnejší spôsob kombinovania rôznych množstiev dostupných potravín v strave, ktorá spĺňa nutričné požiadavky človeka?

V reálnom svete sa na vytvorenie matematického modelu tohto problému najprv zvažujú rôzne aspekty problému. Napríklad: Ako určiť základné výživové požiadavky? Treba zvážiť vek, pohlavie, veľkosť a aktivitu nášho subjektu? Mení sa cena potravín v dôsledku sezónnych a geografických rozdielov? Ktoré potraviny sú dostupné? Ako určiť nutričné hodnoty týchto potravín (Thie, Keough, 2008)?

Pozrieme sa teraz na jednoduchšiu formuláciu problému výživy. Predpokladajme, že na trhu je dostupných n rôznych potravín a že j-tá potravina sa predáva za cenu c_j za jednotku. Okrem toho existuje m základných nutričných zložiek a na dosiahnutie vyváženej stravy musí každý jedinec prijať aspoň b_i jednotky i-tej živiny denne. Každá jednotka potraviny j obsahuje a_{ij} jednotiek i-tej živiny. Počet jednotiek potravy j v strave označíme x_j, ktoré vyberáme tak, aby sa minimalizovali celkové náklady (Luenberger, Ye, 2016).

Formulácia takéhoto zadania úlohy vyzerá nasledovne (Brezina, Ivaničová, Pekár, 2007):

$$\min\ z(x) = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \qquad i = 1, 2, \dots, m$$

$$x_j \geq 0 \qquad j = 1, 2, \dots, n$$

kde

n - počet druhov potravín,

m - počet nutričných zložiek dôležitých na výživu,

x_j - množstvo potrebnej j-tej potraviny, $j = 1, 2, \dots, n$,

c_j - cena j-tej potraviny, $j = 1, 2, \dots, n$,

a_{ij} - množstvo jednotiek i-tej nutričnej zložky dôležitej na výživu, ktoré obsahuje j-tá potravina, $i = 1, 2, \dots, m,\ j = 1, 2, \dots, n$,

b_i - minimálne požadované množstvo i-tej nutričnej zložky dôležitej na výživu, $i = 1, 2, \dots, m$.

3.3.3 Rezný problém (The Cutting Stock Problem)

Rezný problém, alebo úloha o optimálnom delení zásob, sa vyznačuje tým, že existuje obmedzené množstvo zásob originálnych dielov (dosiek, trubiek, tyčí, kotúčov papiera, oceľových plechov a pod.), ktoré je potrebné rozdeliť na menšie časti (Fábry, 2011).

Existuje niekoľko variácií rezného problému. Ak každý ďalší kus, ktorý chceme získať, vyžaduje jeden rez, nazýva sa to 1D alebo jednorozmerný rezný problém (One Dimensional Cutting Stock Problem) Príklady sú napr. rezanie papierových roliek, látkových roliek a kovových tyčí. Ak rezanie zahŕňa obdĺžnikový útvar narezaný na malé obdĺžniky požadovaných veľkostí, nazýva sa to 2D alebo dvojrozmerný rezný problém (Two Dimensional Cutting Stock Problem). Ukážkovým príkladom je rezanie sklenených tabúľ alebo kovových plechov (Ehsan, 2020).

V bakalárskej práci je ďalej rozoberaný jednorozmerný typ. Všeobecná formulácia tohto typu úlohy je (Brezina, Ivaničová, Pekár, 2007):

$$\min\ z(x) = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad i = 1, 2, \dots, m$$

$$x_j \geq 0 \qquad j = 1, 2, \dots, n$$

kde

n - počet spôsobov rezania,

m - počet druhov menších častí, ktoré treba narezať z originálnych dielov,

x_j - množstvo originálnych dielov, ktoré treba rezať j-tým spôsobom, $j = 1, 2, \dots, n$,

c_j - odpad vznikajúci j-tým spôsobom rezania originálnych dielov, $j = 1, 2, \dots, n$,

a_{ij} - počet menších častí i-tého druhu, ktoré sa narežú z originálnych dielov j-tým spôsobom rezania, $i = 1, 2, \dots, m,\ j = 1, 2, \dots, n$,

b_i - požadované množstvo menších častí i-tého druhu, $i = 1, 2, \dots, m$.

3.4 Softvérové riešenia úloh lineárneho programovania

Na riešenie úloh lineárneho programovania existuje mnoho softvérových nástrojov. V tejto bakalárskej práci sme sa sústredili na dva z nich. Konkrétne doplnok Riešiteľ v MS Exceli a funkciu linprog z knižnice SciPy v jazyku Python.

3.4.1 Doplnok Riešiteľ v MS Exceli

Excel je tabuľkový procesor od spoločnosti Microsoft. Používateľom umožňuje formátovať, organizovať a počítať údaje v tabuľkách. Organizovaním údajov pomocou softvéru, ako je Excel, môžu dátoví analytici a iní používatelia zjednodušiť zobrazovanie informácií pri pridávaní alebo zmene údajov. Microsoft Excel obsahuje veľké množstvo polí nazývaných bunky, ktoré sú usporiadané v riadkoch a stĺpcoch. Údaje sú umiestnené v týchto bunkách. Excel je súčasťou balíkov Microsoft Office a Office 365 a je kompatibilný s ostatnými aplikáciami balíka Office. Je k dispozícii pre platformy Windows, MacOS, Android a iOS (Gillis, 2021).

Doplnok Riešiteľ patrí do špeciálnej sady príkazov, ktoré sa často označujú ako nástroje analýzy What-if. Primárne je určený na simuláciu a optimalizáciu rôznych obchodných a inžinierskych modelov. Aj keď Riešiteľ nedokáže vyriešiť každý možný problém, je skutočne užitočný pri riešení všetkých druhov optimalizačných problémov, kde potrebujeme urobiť to najlepšie rozhodnutie (Cheusheva, 2023).

Spúšťa sa na karte Údaje kliknutím na tlačidlo Riešiteľ. Ukáže sa nové okno, kde sa zadávajú parametre Riešiteľa. Nastavujú sa 3 základné parametre: účelová funkcia, premenné modelu a ohraničujúce podmienky. Riešiteľ nájde optimálnu hodnotu (maximálnu, minimálnu alebo špecifikovanú) pre vzorec v bunke *Účelová funkcia* zmenou hodnôt v bunkách *Premenné modelu* a podlieha obmedzeniam v bunkách *Ohraničujúce podmienky*. Ohraničujúce podmienky umožňujú nasledujúce vzťahy medzi odkazovanou bunkou a obmedzením: menší alebo rovný ≤, väčší alebo rovný ≥ alebo len rovný =. Tiež ide nastaviť pre premenné podmienky celočíselnosti, binárnosť a rozdielne hodnoty daných premenných. Špeciálne sa zaznačujú podmienky nezápornosti a vyberá sa metóda, ktorou má Riešiteľ zadaný problém vyriešiť. Riešiteľ ponúka tieto metódu riešenia: simplexovú metódu pre optimalizačné úlohy lineárneho programovania, gradientnú metódu pre hladké nelineárne problémy a evoločný algoritmus pre nehladké nelineárne problémy (Cheusheva, 2023).

3.4.2 Python a funkcia linprog

Python je počítačový programovací jazyk používaný pri vývoji webových stránok, vývoji softvérov, automatizácii rôznych úloh, analýze údajov a vizualizácii údajov. Python je univerzálny jazyk, čo znamená, že môže byť použitý na vytváranie rôznych programov a nie je špecializovaný na žiadne špecifické problémy. Táto všestrannosť, spolu s jeho prívetivosťou pre začiatočníkov, z neho urobili jeden z najpoužívanejších programovacích jazykov súčasnosti (Coursera, 2023).

Python je populárny z viacerých dôvodov (Coursera, 2023):

- má jednoduchú syntax, ktorá napodobňuje prirodzený jazyk, takže je ľahšie čitateľný a zrozumiteľný,
- je vhodný pre začiatočníkov, vďaka čomu je obľúbený u programátorov na základnej úrovni,
- je open source, čo znamená, že ho možno bezplatne používať a distribuovať aj na komerčné účely,
- obsahuje archív modulov a knižníc – zväzky kódov, ktoré používatelia tretích strán vytvorili na rozšírenie možností Pythonu – sú rozsiahle a neustále sa rozrastajú.

V tejto bakalárskej práci sa zaoberáme riešením úloh lineárneho programovania, jazyk Python ponúka viaceré knižnice na riešenie takýchto úloh, napr. SciPy, PuLP alebo Pyomo. My špecifikujeme knižnicu SciPy (Scientific Python), ktorá predstavuje univerzálny balík pre vedecké výpočty. Obsahuje viaceré submodely, pre nás je zaujímavý submodel scipy.optimize, ktorý sa používa na lineárnu a nelineárnu optimalizáciu a obsahuje aj spomínanú funkciu linprog.

Funkcia linprog rieši úlohy lineárneho programovania v tejto forme (The SciPy community, 2023):

$$\min_x c^T x$$
$$A_{ub} x \leq b_{ub}$$
$$A_{eq} x = b_{eq}$$
$$l \leq x \leq u$$

kde x je vektor rozhodovacích premenných; c, b_{ub}, b_{eq}, l, u sú vektory; a A_{ub} a A_{eq}sú matice.

Funkcia liprog má takúto syntax (The SciPy community, 2023):

scipy.optimize.linprog (c, A_ub=None, b_ub=None, A_eq=None, b_eq=None, bounds=None, method='highs', integrality=None)

kde

c = koeficienty účelovej funkcie

A_ub = koeficienty nerovníc ľavej strany ohraničení

b_ub = koeficienty nerovníc pravej strany ohraničení

A_eq = koeficienty rovníc ľavej strany ohraničení

b_eq = koeficienty rovníc pravej strany ohraničení

bounds = určuje interval pre rozhodovacie premenné. Hovorí nám o tom, aká môže byť minimálna a maximálna hodnota rozhodovacej premennej. Preddefinovaný príkaz *None* sa používa, ak nie sú žiadne obmedzenia. Najčastejšie sa tento parameter využíva na zabezpečenie nezápornosti premenných príkazom (0,None).

method = určenie algoritmu použitého na vyriešenie problému. Podporované sú metódy: 'highs' (preddefinovaná), 'highs-ds', 'highs-ipm', 'interior-point' (staršia), 'revised simplex' (staršia) a 'simplex' (staršia). Staršie metódy sa prestali používať od verzie 1.11.0. Napr. namiesto metódy 'simplex' sa používa metóda 'highs', ktorá je rýchlejšia a spoľahlivejšia. Metóda 'highs' si automaticky vyberá medzi 'highs-ds' a 'highs-ipm'. Metóda 'highs-ds' rieši úlohu pomocou duálneho simplexového algritmu a 'highs-ipm' pomocou metódy vnútorného bodu.

integrality = číselná hodnota označujúca typ rozhodovacej premennej:

0: spojitá premenná; bez intervalu

1: celočíselná premenná; rozhodovacia premenná musí byť celé číslo obmedzená parametrom ***bounds***,

2: polo-spojitá premenná; rozhodovacia premenná musí byť v medziach ***bounds*** alebo nadobudnúť hodnotu 0,

3: polo-celočíselná premenná; rozhodovacia premenná musí byť celé číslo v medziach ***bounds*** alebo nadobudnúť hodnotu 0.

Štandardne sú všetky premenné spojité.

Zavolať si možno tieto hodnoty (The SciPy community, 2023):

x = optimálne hodnoty rozhodovacích premenných,

fun = optimálna hodnota účelovej funkcie,

slack = rozdiel pravej a ľavej strany ohraničení v tvare nerovníc,

con = rozdiel pravej a ľavej strany ohraničení v tvare rovníc,

success = *True* ak algoritmus prebehol úspešne,

status = číselná hodnota predstavujúca stav ukončenia algoritmu:

0: Optimalizácia bola úspešne ukončená.

1: Bol dosiahnutý stanovený počet iterácií.

2: Úloha nemá prístupné riešenie.

3: Úloha má neohraničené riešenie.

4: Vyskytli sa numerické ťažkosti.

message = slovný popis ***statusu***,

nit = celkový počet iterácií vykonaných vo všetkých fázach algoritmu.

4 Výsledky práce

Na základe predchádzajúcich informácií a zistení sa ďalej v práci realizujú výpočty pre už vyššie špecifikované 3 typy úloh lineárneho programovania v prostredí MS Excel pomocou nástroja Riešiteľ a v jazyku Python. Údaje a hodnoty v príkladoch sú vymyslené a slúžia len na ukázanie riešenia týchto úloh a porovnanie riešení v MS Exceli a jazyka Python. V úlohe o výžive sme čerpali údaje zo stránky Lenntech (2023).

4.1 Úloha výrobného plánovania

Úlohy výrobného plánovania sú zamerané na navrhovanie takého výrobného plánu, aby sa dosahoval maximálny zisk, minimálne náklady, maximálne tržby a pod. My sa pozrieme na príklad s cieľom maximalizovať tržby, ktorý označíme ako príklad 1. V ďalšom texte tejto podkapitoly sa budeme venovať riešeniu tohto príkladu – po uvedení jeho slovnej formulácie, budeme prezentovať jeho matematickú formuláciu a softvérové riešenie (Fábry, 2011).

Príklad 1: Firma vyrába dva druhy drevených hračiek: skladací vláčik a stavebné kocky. Vláčik predáva za 10 € a sadu kociek za 20 €. Na výrobu 1 vláčika potrebuje 2 kg dreva, 2 plechovky rôznych farieb a 2 hodiny rezbárskej práce. Výroba 1 sady kociek vyžaduje 3 kg dreva, 5 plechoviek rôznych farieb a 1 hodinu rezbárskej práce. Každý mesiac je k dispozícii 900 kg dreva, 1300 plechoviek rôznych farieb a 500 hodín rezbárskej práce. Firma chce zistiť, koľko má vyrobiť jednotlivých hračiek, aby maximalizovala svoje tržby.

Riešenie: Ak chceme z *ekonomického modelu*, čiže len slovného popisu problému, vytvoriť *matematický model*, musíme problém vyjadriť matematickými prostriedkami. Najprv si zadefinujeme premenné (Šmerek, Moučka, 2008).

Premenné: Vystupujú tu dve rozhodovacie premenné:

$x_1 =$ počet vláčikov vyrobených za mesiac,

$x_2 =$ počet sád kociek vyrobených za mesiac.

Presná definícia všetkých premenných je nevyhnutná pre záverečnú interpretáciu výsledkov, preto je vhodné stanoviť i časové obdobie, ku ktorému sa vzťahujú (Fábry, 2011).

Z obsahu premenných je zrejmé, že musia platiť podmienky nezápornosti (Šmerek, Moučka, 2008):

$$x_1, x_2 \geq 0$$

Pripadali by do úvahy aj podmienky celočíselnosti, ale určitým spôsobom komplikujú riešiteľnosť modelu. Úlohy s takýmito podmienkami je možné riešiť pomocou špeciálnych metód *celočíselného lineárneho programovania*. Ďalej sa budeme zaoberať modelom bez tejto podmienky (Fábry, 2011).

Účelová funkcia: Akonáhle sú známe všetky premenné, je možné pristúpiť k formulácii účelovej funkcie. V našom prípade ide o hodnotu tržieb v závislosti od výšky produkcie. Tržby sa vo všeobecnosti vypočítajú ako počet predaných výrobkov vynásobených ich predajnou cenou. Ak predpokladáme, že firma predá všetky vyrobené hračky, tak možno funkciu celkových mesačných tržieb vyjadriť nasledovne:

$$\text{tržby} = 10x_1 + 20x_2$$

Ohraničujúce podmienky: Najprv zaručíme, aby sa neprekročilo množstvo dreva, ktoré je k dispozícii:

$$2x_1 + 3x_2 \leq 900$$

Pravá strana nerovnosti predstavuje možnú spotrebu dreva za mesiac. Na ľavej strane máme súčin dreva potrebného na jednotlivé hračky a množstva danej hračky.

Rovnako zostavíme podmienky pre farby a rezbársku prácu:

$$2x_1 + 5x_2 \leq 1300$$
$$2x_1 + x_2 \leq 500$$

Úplná formulácia matematického modelu je nasledovná:

$$\begin{aligned} \max\ z(x) = {} & 10x_1 + 20x_2 \\ 2x_1 + {} & 3x_2 \leq 900 \\ 2x_1 + {} & 5x_2 \leq 1300 \\ 2x_1 + {} & x_2 \leq 500 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Pripomeňme si všeobecnú formuláciu úlohy výrobného plánovania:

$$\max\ z(x) = \sum_{j=1}^{n} c_j x_j$$

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad j = 1, 2, \ldots, n$$

Všimnime si, že všetky ohraničenia formulácie z príkladu 1 obsahujú nerovnosť ≤ tak ako je to zadefinované vo všeobecnej formulácii úlohy výrobného plánovania, čím sa tento príklad stáva typickým príkladom úlohy výrobného plánovania.

Našim cieľom je nájsť optimálny výrobný plán, pri ktorom firma bude maximalizovať svoje tržby. Poďme sa pozrieť na riešenie v prostredí MS Excel.

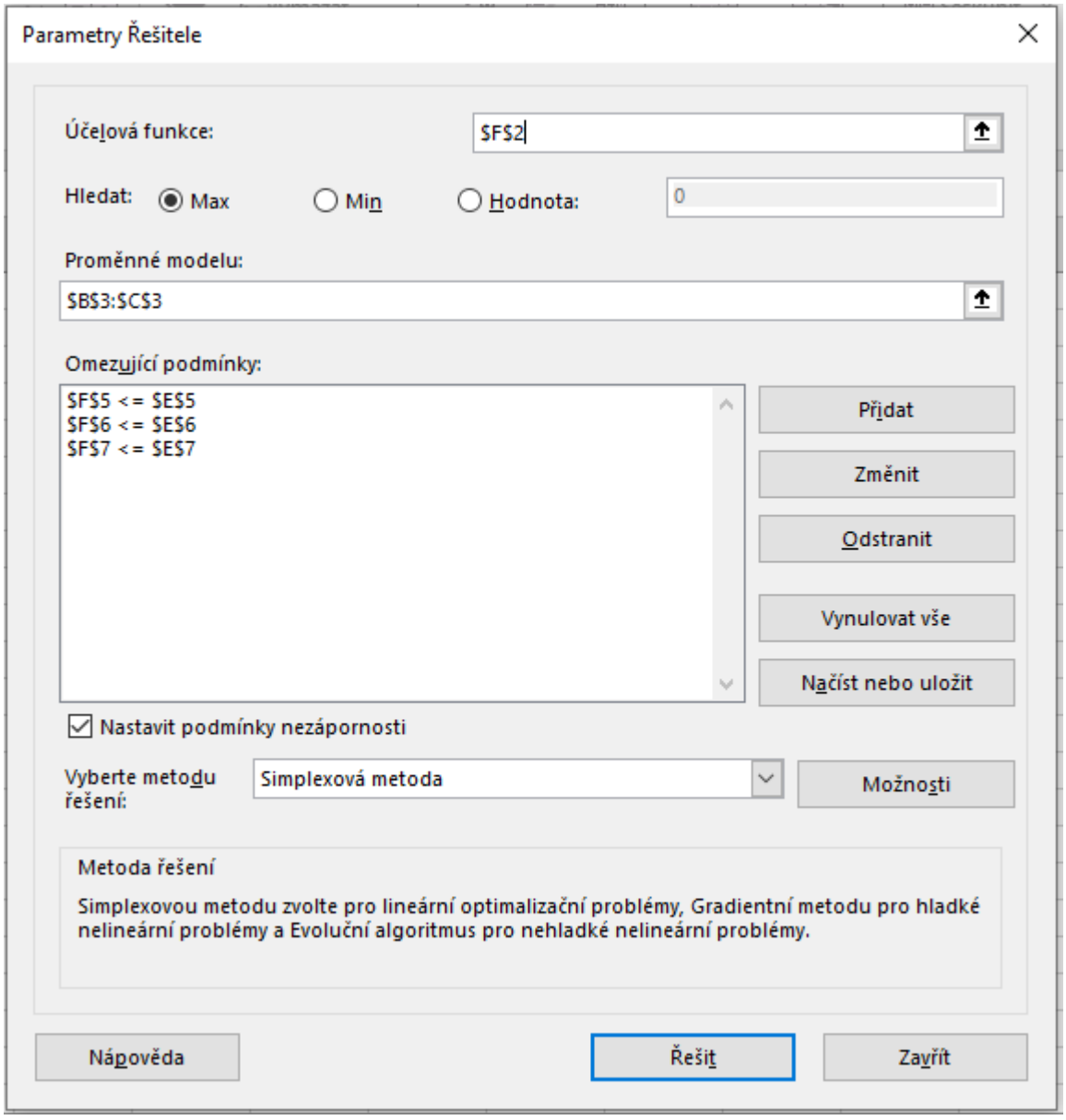
	A	B	C	D	E	F	G	H
1	max	vláčik	kocky					
2	cena	10	20		tržby	0		
3	výroba	0	0			spotrebované	nespotrebované	
4						zdroje	zdroje	
5	drevo	2	3	<=	900	0	900	
6	farby	2	5	<=	1300	0	1300	
7	práca	2	1	<=	500	0	500	
8								

Obr. 4.1 - Zápis úlohy výrobného plánovania do MS Excelu, Zdroj: vlastné spracovanie

Najprv na základe formulácie matematického modelu sme vytvorili tabuľku (Obr. 4.1). Druhý riadok v tejto tabuľke zodpovedá koeficientom účelovej funkcie, čiže cenám jednotlivých výrobkov. Ďalší riadok zase počtu vyrobených, resp. predaných výrobkov. Keďže sa doteraz nič nevyrobilo, tak sa sem zatiaľ zapísali nuly. Nižšie sme zaznačili ohraničujúce podmienky aj so znamienkami nerovností. Pridajú sa dva stĺpce spotrebované a nespotrebované zdroje. Do stĺpca spotrebované zdroje sme vypočítali koľko daného zdroja sme použili na výrobu oboch výrobkov. Keďže v bunkách B3 a C3 sú nuly, aj tu nám musia zatiaľ vyjsť nuly. Použili sme funkciu SUMPRODUCT, vynásobili sme množstvo výrobných zdrojov s počtom vyrobených výrobkov, napr. v bunke F5 sa použil takýto vzorec: =SUMPRODUCT(B5:C5,B3:C3). Tržby v bunke F2 sme vypočítali rovnakým spôsobom len namiesto ohraničenia výrobného zdroja sme použili predajnú cenu. Vzorec bude takýto: =SUMPRODUCT(B2:C2,B3:C3). V stĺpci nespotrebovaných zdrojov porovnávame ľavú a pravú stranu jednotlivých ohraničujúcich podmienok. Ukazuje, či sme využili všetky zdroje, alebo nám ešte nejaké ostali. Zistili sme to jednoduchým spôsobom - odpočítali sme

použité množstvo zdrojov na výrobu od disponibilného množstva jednotlivých výrobných zdrojov. Napr. v bunke G5 bude vzorec takýto: =E5-F5.

Na nájdenie riešenia sme použili doplnok Riešiteľ, ktorý možno vidieť na obrázku 4.2. Do účelovej funkcie sme zadali bunku F2 zodpovedajúcu tržbám. Zaznačili sme, že ich chceme maximalizovať. Premenné sú počet vyrobených výrobkov (bunky B3:C3). Ohraničenie je, aby spotrebované zdroje (bunky F5:F7) nepresiahli disponibilné množstvo jednotlivých výrobných zdrojov (bunky E5:E7). Tiež sme zaznačili podmienky nezápornosti a vybrali metódu riešenia úlohy. V našom prípade sme použili simplexovú metódu.



Obr. 4.2 - Zápis ohraničujúcich podmienok v nástroji Riešiteľ, Zdroj: vlastné spracovanie

	A	B	C	D	E	F	G	H
1	max	vláčik	kocky					
2	cena	10	20		tržby	5500		
3	výroba	150	200			spotrebované	nespotrebované	
4						zdroje	zdroje	
5	drevo	2	3	<=	900	900	0	
6	farby	2	5	<=	1300	1300	0	
7	práca	2	1	<=	500	500	0	
8								

Obr. 4.3 - Výsledok úlohy výrobného plánovania, Zdroj: vlastné spracovanie

Na obrázku 4.3 sa nachádza optimálne riešenie, ktoré vypočítal nástroj Riešiteľ. Možno z neho vyčítať spotrebu dreva, farieb aj rezbárskej práce (bunky F5:F7), množstvo vyrobených výrobkov (bunky B3:C3) a maximálne tržby (bunka F2).

Navrhovaný plán výroby: Optimálnym výrobným plánom je vyrábať 150 kusov vláčikov a 200 sád kociek za mesiac. Na ich výrobu sa spotrebuje 900 kg dreva, 1 300 plechoviek rôznych farieb a 500 hodín rezbárskej práce. Spoločnosť využíva naplno svoje zdroje, keďže jej žiadne nezvýšia, a dosahuje maximálne tržby v hodnote 5 500 €.

Pozrime sa teraz na riešenie tejto úlohy pomocou jazyka Python. Využijeme funkciu linprog z knižnice SciPy už spomínanú v kapitole 3.4.2. V tejto kapitole bolo spomínané, že táto funkcia podporuje len minimalizačné úlohy, tak sme prerobili našu formuláciu takto:

$$\begin{aligned} \min \; -\mathrm{z}(x) = -10x_1 - 20x_2 & \\ 2x_1 + 3x_2 &\leq 900 \\ 2x_1 + 5x_2 &\leq 1300 \\ 2x_1 + x_2 &\leq 500 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Teraz sa môže pristúpiť k tvorbe zdrojového kódu v jazyku Python (Obr. 4.4). Koeficienty účelovej funkcie sme vložili do premennej *cena*. Premenná *lava_strana* zodpovedá ľavej strane našich nerovníc a premenná *prava_strana* zodpovedá pravej strane nerovníc. Parametre A_eq a b_eq sme nepoužili, keďže naša formulácia neobsahuje rovnice. Ponechali sme preddefinovanú metódu *highs*, ktorá vyriešila úlohu pomocou duálneho simplexového algoritmu. Nechali sme si vypísať všetky rozhodovacie premenné a rozdiely

pravých a ľavých strán ohraničení. Upravili sme účelovú funkciu na kladnú hodnotu, tak aby interpretácia dávala zmysel.

```
# vlozenie funkcie z kniznice
from scipy.optimize import linprog

# koeficienty ucelovej funkcie
cena = [-10,-20]

# lava strana nerovnic
lava_strana = [[2,3],
               [2,5],
               [2,1]]

# prava strana nerovnic
prava_strana = [900,
                1300,
                500]

# riesenie ulohy
opt = linprog(c=cena, A_ub=lava_strana, b_ub=prava_strana)
print("Optimalny vyrobny plan:")
print("Vlacikov sa vyrobi: ", opt.x[0])
print("Sad kociek sa vyrobi: ", opt.x[1])
print("Nespotrebuje sa ", opt.slack[0], " kilogramov dreva")
print("Nespotrebuje sa ", opt.slack[1], " plechoviek farieb")
print("Nespotrebuje sa ", opt.slack[2], " hodin rezbarskej prace")
print("Pri danom vyrobnom plane firma dosiahne trzby ",-opt.fun, " €")
```

Obr. 4.4 - Zdrojový kód úlohy výrobného plánovania v jazyku Python, Zdroj: vlastné spracovanie

```
Optimalny vyrobny plan:
Vlacikov sa vyrobi:  150.0
Sad kociek sa vyrobi:  200.0
Nespotrebuje sa  0.0  kilogramov dreva
Nespotrebuje sa  0.0  plechoviek farieb
Nespotrebuje sa  0.0  hodin rezbarskej prace
Pri danom vyrobnom plane firma dosiahne trzby  5500.0  €
```

Obr. 4.5 – Riešenie v prostredí IDLE

Na obrázku 4.5 je riešenie úlohy pomocou jazyka Python. Optimálne by sa za mesiac malo vyrobiť 150 kusov vláčikov a 200 sád kociek. Na výrobu sa spotrebuje všetko drevo, všetky plechovky farieb aj všetky možné hodiny rezbárskej práce. Ak firma bude postupovať podľa tohto výrobného plánu, dosiahne maximálnu hodnotu tržieb 5 500 €.

4.2 Úloha o výžive

Úloha o výžive súvisí s nutričnými požiadavkami jednotlivca. Sleduje sa počet kalórií, množstvo vitamínov, minerálov, tukov a pod. My sa zameriame na úlohu s množstvom vitamínov v ovocí, ktorý označíme ako príklad 2 (Jablonský, 2007).

Príklad 2: V ekonomike existujú len 3 druhy ovocia (jablká, broskyne, hrušky), ktorými sa dopĺňajú potrebné vitamíny. Tabuľka 4.1 obsahuje údaje o množstve vitamínov v miligramoch v jednom kuse ovocia a mesačný limit vitamínov, ktoré potrebuje človek na normálne fungovanie. Cena jablka je 0,50 €/ks, broskyne 0,40 €/ks a hrušky 0,45 €/ks. Za predpokladu, že vitamíny sa nedajú prijať z iných potravín, je potrebné zistiť koľko a aký druh ovocia potrebuje človek mesačne zjesť, tak aby minimalizoval svoje náklady?

Vitamín	Jablko	Broskyňa	Hruška	Mesačný limit
A	0,005	0,016	0,002	0,6
B1	0,02	0,01	0,01	1,4
B2	0,01	0,02	0,01	1,6
B6	0,05	0,02	0,02	2

Tab. 4.1 – Vitamíny v miligramoch na jeden kus jednotlivého ovocia

Ďalej sa pozrieme na riešenie príkladu 2, ktorého slovná formulácia je rozpísaná vyššie. Nasledovať bude matematická formulácia, softvérové riešenie a interpretácia výsledkov.

Riešenie: Premenné:

$x_1 =$ počet kusov jabĺk za mesiac,

$x_2 =$ počet kusov broskýň za mesiac,

$x_3 =$ počet kusov hrušiek za mesiac.

Tieto premenné musia spĺňať podmienky nezápornosti:

$$x_1, x_2, x_3 \geq 0$$

Účelová funkcia: Minimalizujeme náklady na nákup jednotlivých druhov ovocia:

$$\min \ \mathrm{z}(x) = \ 0{,}5x_1 + \ 0{,}4x_2 + 0{,}45x_3$$

Ohraničujúce podmienky: Mesačný limit nám predstavuje najmenšie množstvá jednotlivých vitamínov, ktoré by mal človek z uvedených troch druhov ovocia v strave prijať, aby mohol normálne fungovať, čiže napr. podmienka pre vitamín A bude vyzerať takto:

$$0{,}005x_1 + 0{,}016x_2 + 0{,}002x_3 \geq 0{,}6$$

Celková matematická formulácia tejto úlohy po zohľadnení ohraničujúcich podmienok pre ďalšiu trojicu vitamínov i podmienok nezápornosti má potom nasledujúci tvar:

$$\begin{aligned} \min\ z(x) = \quad & 0{,}5x_1 + \quad 0{,}4x_2 + \quad 0{,}45x_3 \\ & 0{,}005x_1 + 0{,}016x_2 + 0{,}002x_3 \geq 0{,}6 \\ & 0{,}02x_1 + \quad 0{,}01x_2 + \quad 0{,}01x_3 \geq 1{,}4 \\ & 0{,}01x_1 + \quad 0{,}02x_2 + \quad 0{,}01x_3 \geq 1{,}6 \\ & 0{,}05x_1 + \quad 0{,}02x_2 + \quad 0{,}02x_3 \geq 2 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Pozrime sa teraz na riešenie v MS Exceli. Zhotovili sme tabuľku podľa tabuľky 4.1 a pridali sme riadky *cena* jednotlivých ovocí v €/ks a *množstvo* potrebné mesačne zjesť. Stĺpce sme pridali tiež dva. Prvý udáva, aké je skutočné množstvo prijatých vitamínov z danej kombinácie ovocí. Prenásobuje množstvá skonzumovaných druhov ovocia s obsahom daného vitamínu (v mg) v 1 ks príslušného druhu ovocia. Druhý porovnáva mesačný limit so skutočným množstvom prijatých vitamínov a ukazuje množstvo vitamínov, ktoré by sme pri danej kombinácii prijali navyše. Hlavnou úlohou je minimalizovať náklady na kúpu ovocia, ktoré sú vyjadrené súčinom množstva ovocia a cien jednotlivých ovocí, za predpokladu, že vitamíny sa nedajú prijať z iných potravín. Použité vzorce môžeme vidieť na obrázku 4.6.

	A	B	C	D	E	F	G
1	vitamín	jablko	broskyňa	hruška	mesačný limit	skutočné množstvo vitamínov	vitamíny navyše
2	A	0.005	0.016	0.002	0.6	=SOUČIN.SKALÁRNÍ(B2:D2,$B$7:$D$7)	=F2-E2
3	B1	0.02	0.01	0.01	1.4	=SOUČIN.SKALÁRNÍ(B3:D3,$B$7:$D$7)	=F3-E3
4	B2	0.01	0.02	0.01	1.6	=SOUČIN.SKALÁRNÍ(B4:D4,$B$7:$D$7)	=F4-E4
5	B6	0.05	0.02	0.02	2	=SOUČIN.SKALÁRNÍ(B5:D5,$B$7:$D$7)	=F5-E5
6	cena v €/ks	0.5	0.4	0.45			
7	množstvo	40	60	0		náklady	=SOUČIN.SKALÁRNÍ(B6:D6,$B$7:$D$7)
8							

Obr. 4.6 - Úloha o výžive v MS Exceli, Zdroj: vlastné spracovanie

Poznámka: SOUČIN.SKALÁRNÍ je SUMPRODUCT len v českom preklade.

Riešenie sme získali pomocou nástroja Riešiteľ. Do účelovej funkcie sme zadali náklady (bunka G7), ktoré chceme minimalizovať. Premenné zodpovedajú množstvu ovocia (bunky B7:D7) a ohraničenia: mesačný limit (bunky E2:E5) má byť menší ako skutočné

množstvo prijatých vitamínov (bunky F2:F5). Tiež sme zaznačili podmienky nezápornosti a vybrali simplexovú metódu.

	A	B	C	D	E	F	G
1	vitamín	jablko	broskyňa	hruška	mesačný limit	skutočné množstvo vitamínov	vitamíny navyše
2	A	0.005	0.016	0.002	0.6	1.16	0.56
3	B1	0.02	0.01	0.01	1.4	1.4	0
4	B2	0.01	0.02	0.01	1.6	1.6	0
5	B6	0.05	0.02	0.02	2	3.2	1.2
6	cena v €/ks	0.5	0.4	0.45			
7	množstvo	40	60	0		náklady	44
8							

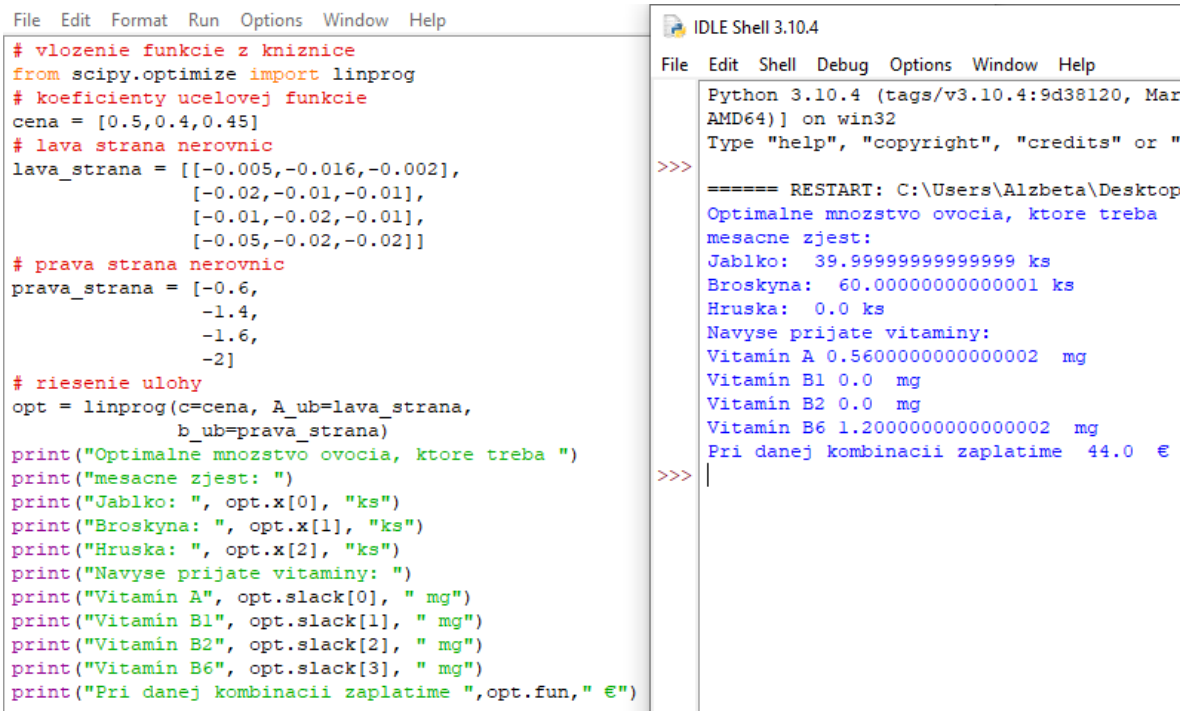
Obr. 4.7 – Výsledok riešenia úlohy o výžive, Zdroj: vlastné spracovanie

Z obrázku 4.7 vieme vyčítať riešenie úlohy. Mesačne by sme mali zjesť 40 jabĺk, 60 broskýň a žiadnu hrušku (bunky B7:D7), aby sme získali všetky potrebné vitamíny. Vitamínov B1 a B2 (bunky G3, G4) získame presné množstvo, vitamín A (bunka G2) o 0,56 mg viac a vitamínu B6 (bunka G5) až o 1,2 mg viac. Bude nás to stáť 44 € (bunka G7).

Riešenie v jazyku Python bude obdobné ako v predchádzajúcej úlohe. Zase sme upravili formuláciu, keďže sme narazili na ďalší nedostatok funkcie linprog. Prerobili sme znamienka nerovnosti $\geq$ na $\leq$, keďže funkcia linprog podporuje len tento typ nerovnosti. Formulácia bude vyzerať nasledovne:

$$\begin{aligned} \min\ z(x) = \quad & 0{,}5x_1 + \quad 0{,}4x_2 + \quad 0{,}45x_3 \\ & -0{,}005x_1 - 0{,}016x_2 - 0{,}002x_3 \leq -0{,}6 \\ & -0{,}02x_1 - \quad 0{,}01x_2 - \quad 0{,}01x_3 \leq -1{,}4 \\ & -0{,}01x_1 - \quad 0{,}02x_2 - \quad 0{,}01x_3 \leq -1{,}6 \\ & -0{,}05x_1 - \quad 0{,}02x_2 - \quad 0{,}02x_3 \leq -2 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

V zdrojovom kóde jazyka Python sme vložili do premennej *cena* koeficienty účelovej funkcie, do premennej *lava_strana* ľavú stranu nerovníc a do premennej *prava_strana* pravú stranu nerovníc. Úlohu sme vyriešili duálnym simplexovým algoritmom, ktorý vybrala preddefinovaná metóda *highs*. Nechali sme si vrátiť hodnoty účelovej funkcie, rozhodovacích premenných a rozdiely pravých a ľavých strán nerovníc.

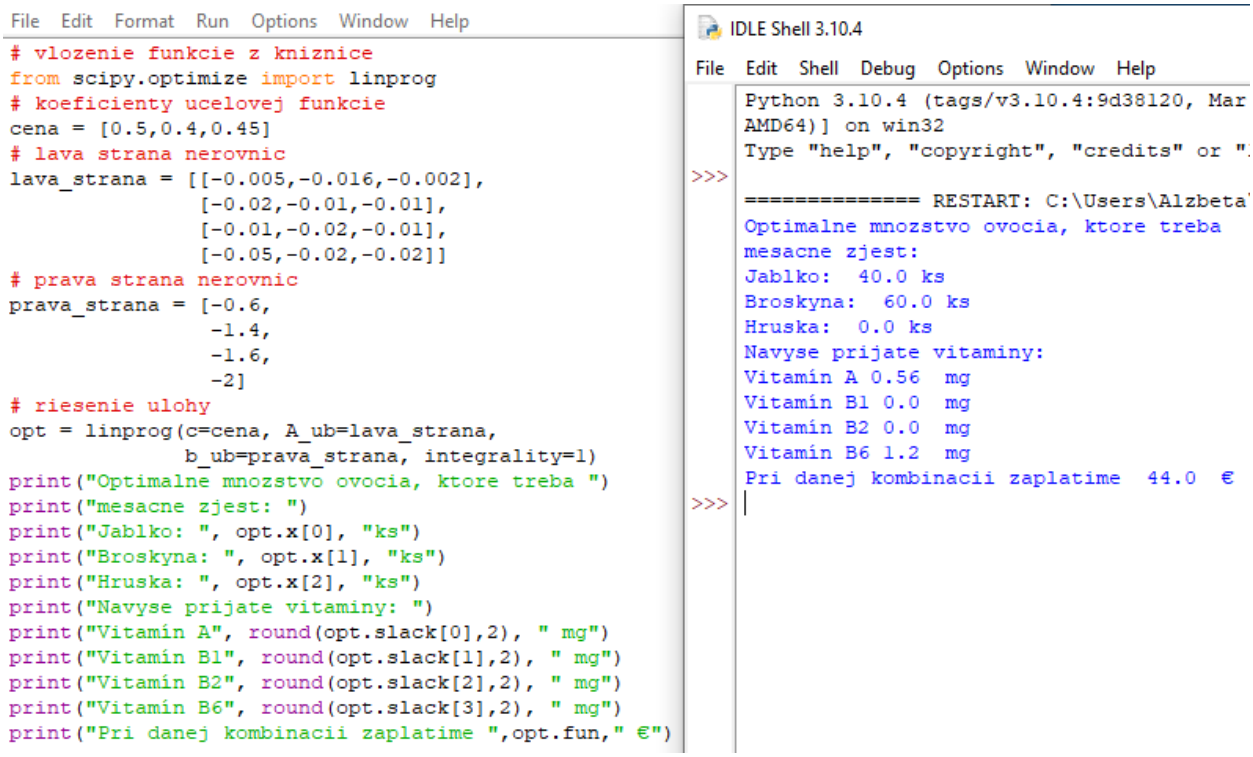


Obr. 4.8 - Zdrojový kód a riešenie úlohy o výžive, Zdroj: vlastné spracovanie

Zdrojový kód a optimálne riešenie úlohy je na obrázku 4.8. Môžeme vidieť, že množstvá jednotlivých ovocí sú zapísané v desatinných číslach, čo z hľadiska interpretácie nie je moc vyhovujúce.

Upravili sme zdrojový kód a pridali sme do funkcie linprog parameter *integrality* na povolenie iba celočíselných hodnôt pre rozhodovacie premenné, ktorý bol bližšie popísaný v podkapitole 3.4.2. Pre lepšiu prehľadnosť sme pridali aj zaokrúhlenie na 2 desatinné čísla vo rozdieloch ľavých a pravých strán jednotlivých ohraničení.

Na obrázku 4.9 možno vidieť upravený zdrojový kód a nové riešenie danej úlohy. Optimálne množstvo ovocia za mesiac je 40 jabĺk a 60 broskýň. Pri danej kombinácií ovocia sa príjme presne minimálne potrebné množstvo vitamínov B1 a B2. Vitamínu A získame o 0,56 mg viac a vitamínu B6 až o 1,2 mg viac. Zaplatíme minimálne náklady 44 €.



Obr. 4.9 – Upravený zdrojový kód a riešenie úlohy o výžive, Zdroj: vlastné spracovanie

4.3 Rezný problém

Rezným problémom sa prvýkrát začal zaoberať priemysel s papierom. Kotúče papiera sa vyrábali s pevne stanovenou šírkou, ale odberatelia požadovali aj menšie veľkosti. Hľadalo sa riešenie ako rezať kotúče na menšiu šírku, aby sa splnili požiadavky a minimalizovalo sa množstvo odpadu (Neos Guide, 2023).

Ďalej si ukážeme ako sa rieši takýto typ úlohy, ktorý označíme ako príklad 3. Uvedieme jeho slovnú formuláciu, vysvetlíme tvorbu matematického modelu zodpovedajúcej úlohy lineárneho programovania a jej softvérové riešenie.

Príklad 3: Papiernická firma má dvojmetrovú štandardnú formu na výrobu kotúčov papiera. Potom vyrobené kotúče orezáva podľa objednávky. Posledná objednávka bola 500 ks kotúčov šírky 1 m, 1 200 ks šírky 80 cm a 700 ks šírky 40 cm. Akým spôsobom má firma orezať štandardnú veľkosť kotúča, tak aby minimalizovala odpad? Maximálna povolená šírka odpadu je 20 cm.

Predtým ako pristúpime ku samotnej formulácii danej úlohy, potrebujeme zistiť, akým spôsobom možno profily s dĺžkou 200 cm rezať tak, aby boli splnené podmienky zo zadania.

Vytvorili sme tabuľku 4.2, ktorá bude slúžiť ako podklad. Nazýva sa plán rezania resp. rezný plán (Brezina, Pekár, 2018).

Šírka kotúča	Spôsoby rezania						Požadovaný počet kusov
	S1	S2	S3	S4	S5	S6	
100 cm	2	1	1	0	0	0	500
80 cm	0	1	0	2	1	0	1200
40 cm	0	0	2	1	3	5	700
Odpad	0	20	20	0	0	0	

Tab. 4.2 – Rezný plán

Spôsob rezania S1 znamená, že z dvojmetrovej štandardnej šírky kotúča narežeme 2 kusy 100 cm širokých kotúčov a nezostane nám žiaden odpad. Spôsobom S2 zase narežeme 1 kus 100 cm širokého kotúča a 1 kus 80 cm širokého kotúča a ostane nám odpad široký 20 cm. Význam ostatných spôsobov môžeme interpretovať analogicky (Šmerek, Moučka, 2008).

Rozhodovacie premenné budú ukazovať počty rezaní podľa spôsobu S1, S2, S3, S4, S5 a S6 (iné kombinácie rezaní neexistujú, tak aby sme nepresiahli maximálne povolenú šírku odpadu). Účelová funkcia bude označovať množstvo odpadu v cm vznikajúceho z daného spôsobu rezania. Ohraničenia zase množstvo kotúčov určitej šírky. Potom možno na základe zadaných podmienok vytvoriť takúto formuláciu úlohy:

$$\begin{aligned} \min\ z(x) = \quad & 20x_2 + 20x_3 \\ 2x_1 + \ x_2 + \ x_3 \quad & = 500 \\ x_2 + \ 2x_4 + \ x_5 \quad & = 1200 \\ 2x_3 + \ x_4 + 3x_5 + 5x_6 \quad & = 700 \\ x_1, x_2, x_3, x_4, x_5, x_6 \quad & \geq 0 \end{aligned}$$

Po matematickej formulácii úlohy sa môže pristúpiť k hľadaniu riešenia danej úlohy. Do Excelu sme zadali tabuľku 4.2. Pridali sme navyše nulový riadok: *počet rezaní* a navyše stĺpec: *narezaný počet kusov*, ktorý obsahuje funkciu SUMPRODUCT. Tento stĺpec násobí počet spôsobov rezaní s počtom kusov vznikajúcich daným rezaním a povie nám koľko narežeme kotúčov určitej šírky. Zväčšili sme tabuľku ešte o jeden stĺpec: *prebytok kotúčov*. Porovnáva požadovaný počet kusov v objednávke a skutočný počet narezaných kotúčov. V Riešiteli sme do účelovej funkcie vložili minimálny odpad (bunka J7). Premenné sú počet

rezaní (bunky B7:G7) a ohraničujúce podmienky sú rovnosti medzi požadovaným (bunky H3:H5) a narezaným počtom kusov kotúčov (bunky J3:J5). Tiež sme označili podmienky nezápornosti a zvolili simplexovú metódu.

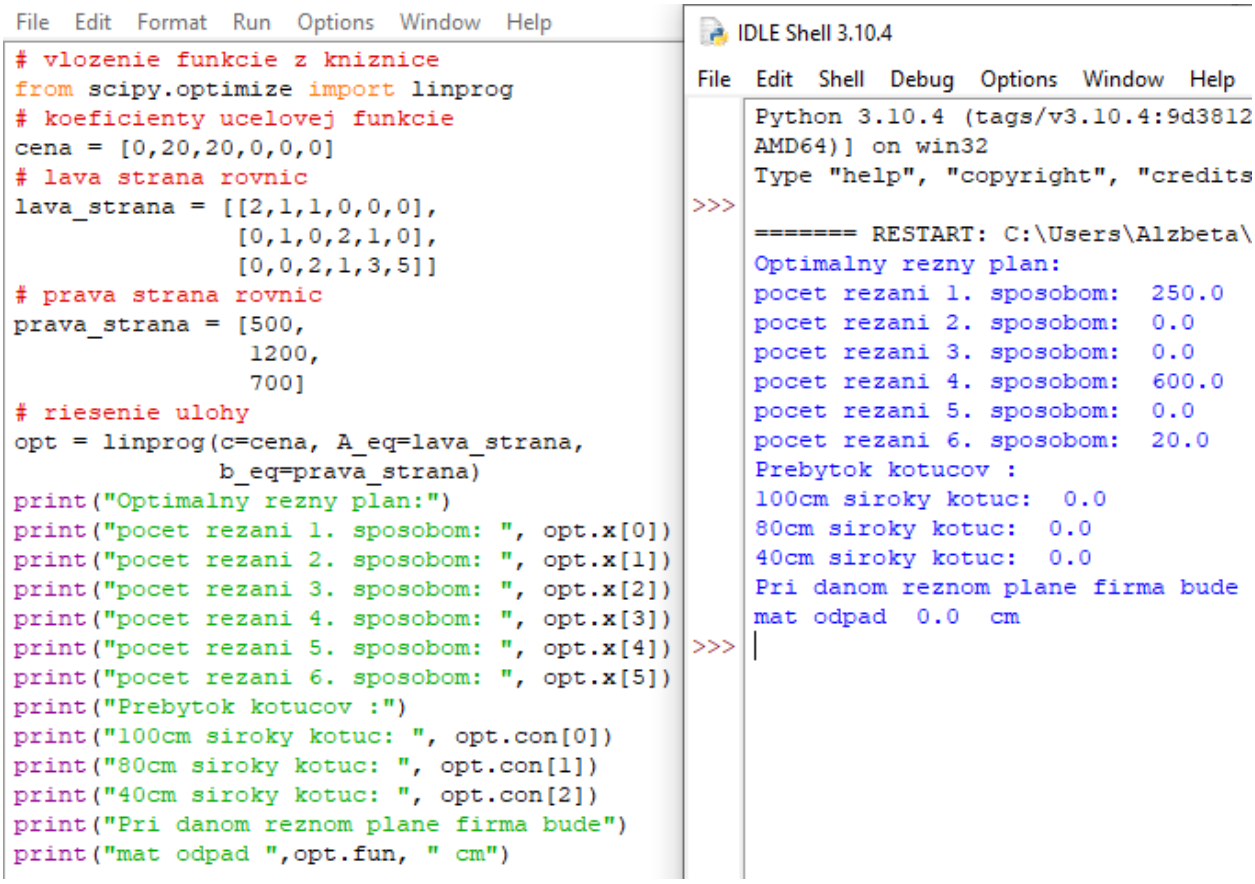
	A	B	C	D	E	F	G	H	I	J	K
1	Šírka kotúča	Spôsoby rezania						Požadovaný		Narezaný	Prebytok
2		S1	S2	S3	S4	S5	S6	počet kusov		počet kusov	kotúčov
3	100 cm	2	1	1	0	0	0	500	"=	500	0
4	80 cm	0	1	0	2	1	0	1200	"=	1200	0
5	40 cm	0	0	2	1	3	5	700	"=	700	0
6	Odpad	0	20	20	0	0	0				
7	počet rezaní	250	0	0	600	0	20	min .odpad		0	

Obr. 4.10 - Výsledok riešenia rezného problému, Zdroj: vlastné spracovanie

Výsledok získaný pomocou nástroja Riešiteľ je zobrazený na obrázku 4.10. Firma oreže 250 kotúčov prvým spôsobom rezania, 600 štvrtým a 20 šiestym. Takýmito rezaniami nezostane žiaden odpad a získa sa presný počet objednaných kotúčov každej šírky.

Teraz si ukážeme riešenie cez jazyk Python. Tak ako v predošlých dvoch úlohách aj v tejto úlohe sme využili funkciu linprog. Do premennej *cena* sme vložili koeficienty účelovej funkcie. Premenné *lava_strana* a *prava_strana* zodpovedajú v tomto prípade ľavej a pravej strane rovníc, nie nerovníc. Tak sme tieto premenné pridali do parametrov A_eq a b_eq. Použili sme preddefinovanú metódu *highs*, ktorá vyriešila úlohu pomocou duálneho simplexového algritmu. Nechali sme vypísať hodnoty účelovej funkcie, premenných a rozdielov pravých a ľavých strán rovníc. Takisto aj tu sme použili iný príkaz na vypísanie hodnôt. Namiesto *slack* sme použili *con*, ktorý zodpovedá vráteniu rozdielu v rovniciach.

Obrázok 4.11 zobrazuje zdrojový kód a výsledok riešenia rezného problému. Optimálny rezný plán je 250 rezaní prvým spôsobom, 600 rezaní druhým a 20 šiestym. Nebudeme mať prebytok kotúčov a ani nevznikne žiaden odpad z rezaní.



Obr. 4.11 - Zdrojový kód a riešenie rezného problému, Zdroj: vlastné spracovanie

5 Diskusia

Keď si porovnáme výsledky získané doplnkom Riešiteľ v MS Exceli a funkciou linprog v jazyku Python zistíme, že výsledky sú identické. Čo nám aj malo vyjsť, keďže sme v oboch prípadoch použili simplexovú metódu. V MS Exceli sme aplikovali primárny simplexový algoritmus a v jazyku Python duálny simplexový algoritmus. Ide o rovnakú metódu, ale iný algoritmus.

Hlavný rozdiel medzi dvoma použitými postupmi výpočtov predstavuje nie tak časová náročnosť výpočtu, ale technické zručnosti pri práci, hlavne v jazyku Python. Pomocou jazyka Python získame rovnaké hodnoty ako v MS Exceli, avšak Python vyžaduje, aby mal používateľ určité skúsenosti s programovaním. Tiež je potrebné poznať syntax funkcie linprog, jej jednotlivé parametre a ako zadať príkazy, ktoré nám vrátia požadované hodnoty.

V MS Exceli je to jednoduchšie. V doplnku Riešiteľ sa nastavujú základné parametre, ktoré sa nazývajú rovnako ako pri formulácii matematického modelu úlohy. Čiže, ak používateľ vie vytvoriť formuláciu, riešenie úlohy v tomto programe by malo byť bezproblémové.

Jediný problém môže nastať pri prezentácii výsledkov. V MS Exceli sa zmenia bunky obsahujúce rozhodovacie premenné a účelovú funkciu, inak výsledok je rovnaký ako zapísanie zadania úlohy do tabuľky. Čiže už pri tvorbe tabuľky musí používateľ myslieť na to, aby aj nezainteresovaná osoba sa vedela vyznať v danej tabuľke. V jazyku Python sa používateľ, po zadaní správnych parametrov funkcie, môže vyhrať s vzhľadom výstupu a dať vypísať hodnoty, ktoré sú naozaj potrebné pre ďalšie používanie.

Existuje mnoho ďalších softvérových nástrojov alebo knižníc jazyka Python, ktoré dokážu nájsť riešenia úloh lineárneho programovania. V tejto bakalárskej práci sme prezentovali iba niektoré. Je len na riešiteľovi problému vybrať si tú možnosť, ktorá mu najviac vyhovuje.

Záver

Cieľom bakalárskej práce bolo priblížiť oblasť lineárneho programovania, predstaviť vybrané typy úloh z tejto oblasti a poukázať na možnosti ich riešenia s využitím vybraných softvérových nástrojov. Stanovený cieľ bol splnený, nakoľko sme vyriešili 3 konkrétne úlohy pomocou doplnku Riešiteľ v MS Exceli a pomocou funkcie linprog jazyka Python.

V teoretickej časti sme opísali podstatu a históriu operačného výskumu. Ďalej sme sa venovali matematickému programovaniu a matematickým modelom. Následne sme charakterizovali lineárne programovanie a metódy riešenia úloh lineárneho programovania. Potom sme zadefinovali formulácie pre vybraté typy úloh lineárneho programovania a popísali vybraté možnosti riešenia úloh softvérovými nástrojmi.

Praktická časť bakalárskej práce je venovaná ukážke využitia lineárneho programovania, ktorá pozostávala zo slovnej i matematickej formulácie rôznych rozhodovacích problémov a ich softvérového riešenia. V tejto časti sme riešili tri rôzne typy úloh, konkrétne úlohu výrobného plánovania, úlohu o výžive a rezný problém, v MS Exceli pomocou doplnku Riešiteľ a v jazyku Python pomocou funkcie linprog z knižnice SciPy. Pri oboch nástrojoch sme použili simplexovú metódu, v MS Exceli sa aplikoval skôr primárny simplexový algoritmus a v jazyku Python zase duálny simplexový algoritmus.

Porovnaním výsledkov oboch spôsobov sme zistili, že výsledky daných úloh sú totožné. Rozdiel spočíva v nároku na technické zručnosti pri práci s jazykom Python. Výpočet pomocou funkcie linprog predpokladá určité znalosti tejto funkcie ako jej syntax a príkazy na vrátenie požadovaných hodnôt.

Použitá literatúra

1. AGARWAL, Lata. 2022. *Advantages and Disadvantages of Linear Programming Problem* [online]. Aktualizované 2022-02-03 [cit. 2023-04-07]. Dostupné na: <https://prinsli.com/advantages-and-disadvantages-of-linear-programming-problem-lpp/>.
2. AGARWAL, Lata. 2019a. *Classification of modelling in Operations Research.* [online]. Aktualizované 2019-01-17 [cit. 2023-04-06]. Dostupné na: <https://prinsli.com/classification-of-modelling-in-operations-research/>.
3. AGARWAL, Lata. 2019b. *Significant features / Nature / Characteristics of Operation Research.* [online]. Aktualizované 2019-01-15 [cit. 2023-04-11]. Dostupné na: <https://prinsli.com/characteristics-significant-features-of-operation-research/>.
4. BREZINA, Ivan - IVANIČOVÁ, Zlatica – PEKÁR, Juraj. 2007. *Operačná analýza*. Bratislava : Iura Edition, 2007. 243 s. ISBN 978-80-8078-176-7.
5. BREZINA, Ivan - PEKÁR, Juraj. 2018. *Úvod do Operačného výskumu I*. Bratislava : Letra Edu, 2018. 170 s. ISBN 978-80-89962-18-1.
6. CHEUSHEVA, Svetlana. 2023. *How to use Solver in Excel with examples* [online]. Aktualizované 2023-03-21 [cit. 2023-04-26]. Dostupné na: < https://www.ablebits.com/office-addins-blog/excel-solver-examples/>.
7. COURSERA. 2023. *What Is Python Used For? A Beginner's Guide* [online]. Aktualizované 2023-04-14 [cit. 2023-04-24]. Dostupné na: <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>.
8. EHSAN, Emad. 2020. *Cutting Stock Problem - 1D* [online]. Aktualizované 2020-07-18 [cit. 2023-04-27]. Dostupné na: <https://medium.com/analytics-vidhya/cutting-stock-problem-1d-df976f7263cd>.
9. EURO. 2023. *EURO* [online]. 2023 [cit. 2023-04-19]. Dostupné na: <https://www.euro-online.org/web/pages/95/euro>.
10. FÁBRY, Jan. 2011. *Matematické modelování*. Praha : PROFESSIONAL PUBLISHING, 2011. 180 s. ISBN 978-80-7431-066-9.

11. GILLIS, Alexander S. 2021. *DEFINITION Excel* [online]. Aktualizované november 2021 [cit. 2023-04-24]. Dostupné na: <https://www.techtarget.com/searchenterprisedesktop/definition/Excel>.
12. HILLIER, Frederick S. - LIEBERMAN, Gerald J. 2010. *Introduction to Operations Research*. 9th ed. New York : The McGraw-Hill Companies, 2010. 1047p. ISBN 978-0-07-337629-5.
13. CHOVANOVÁ, Henrieta Hrablik. 2016. *OPERAČNÁ ANALÝZA. NÁVODY NA CVIČENIA I*. Trnava : AlumniPress - MTF STU, 2016. 164s. ISBN 978-80-8096-240-1
14. IFORS. 2023. *IFORS History* [online]. 2023 [cit. 2023-04-19]. Dostupné na: <https://www.ifors.org/history/>.
15. JABLONSKÝ, Josef. 2007. *Operační výzkum* : Kvantitativní modely pro ekonomické rozhodování. 3. vyd. Praha : PROFESSIONAL PUBLISHING, 2007. 323 s. ISBN 978-80-86946-44-3.
16. KHAN, M. Shadab. 2019. *Chapter -01 Operations Research* [online]. Jún 2019 [cit. 2023-04-11]. Dostupné na: <https://www.researchgate.net/publication/333748649_Chapter_-01_Operations_Research>.
17. LENNTECH, 2023. *Vitamin content of fruit and vegetables* [online]. 2023 [cit. 2023-03-30]. Dostupné na: <https://www.lenntech.com/fruit-vegetable-vitamin-content.htm>.
18. LUENBERGER, David G. - YE, Yinyu. 2016. *Linear and Nonlinear Programming*. 4th ed. Cham (Switzerland) : Springer, 2016. 546 p. ISBN 978-3-319-18842-3.
19. MCCOMBES, Shona - GEORGE, Tegan. 2022. *How to Write a Problem Statement | Guide & Examples* [online]. Publikované 2022-11-06, Aktualizované 2022-11-28 [cit. 2023-04-04]. Dostupné na: <https://www.scribbr.com/research-process/problem-statement/>.
20. NEOS GUIDE. 2023. *The Cutting Stock Problem* [online]. 2023 [cit. 2023-03-27]. Dostupné na: <https://neos-guide.org/case-studies/sc/mfg/the-cutting-stock-problem/>.
21. PLEVNÝ, Miroslav - ŽIŽKA, Miroslav. 2010. *Modelování a optimalizace v manažerském rozhodování*. Plzeň : Vydavatelství Západočeské univerzity, 2010. 298 s. ISBN 978-80-7043-933-3.

22. RAVINDRAN, A. Ravi. 2008. History of Operations Research and Management Science. In *Operations Research and Management Science Handbook*. Boca Raton (Florida) : CRC Press, 2008. ISBN 978-0-8493-9721-9. p. XXI-XXII.
23. SEKHON, Rupinder - BLOOM, Roberta. 2022. *Maximization by The Simplex Method* [online]. Aktualizované 2022-07-18 [cit 2023-03-31]. Dostupné na: <https://math.libretexts.org/Bookshelves/Applied_Mathematics/Applied_Finite_Mathematics_(Sekhon_and_Bloom)/04%3A_Linear_Programming_The_Simplex_Method/4.02%3A_Maximization_By_The_Simplex_Method>.
24. SINGH, Bismark - EISNER, Mark. 2023. *A Brief History of Optimization and Mathematical Programming* [online]. 2023 [cit. 2023-03-18]. Dostupné na: <https://www.informs.org/Explore/History-of-O.R.-Excellence/O.R.-Methodologies/Optimization-Mathematical-Programming>.
25. SINGLA, Saurabh. 2016. Operational Research: A Study of Decision Making Process. In *Journal of Multidisciplinary Engineering Science and Technology* [online]. May 2016, vol. 3, no. 5 [cit. 2023-03-17], pp. 5336-5338. Dostupné na: <https://www.jmest.org/wp-content/uploads/JMESTN42351698.pdf>. ISSN 2458-9403.
26. ŠMEREK, Michal - MOUČKA Jiří. 2008. *Ekonomicko-matematické metody: učební text pro distanční studium*. Brno: Univerzita obrany, 2008. ISBN 978-80-7231-526-0.
27. THEINTACTONE. 2019. *Operations Research: Introduction, Historical Background* [online]. 2019-03-03 [cit. 2023-03-17]. Dostupné na: <https://theintactone.com/2019/03/03/qtm-u1-topic-1-operations-research-introduction-historical-background/>.
28. THE SCIPY COMMUNITY. 2023. *Scipy.optimize.linprog* [online]. Aktualizované 2023-02-19 [cit. 2023-04-28]. dostupné na: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html#scipy.optimize.linprog>.
29. THIE, Paul R. - KEOUGH, Gerard E. 2008. *An Introduction to Linear Programming and Game Theory*. 3rd ed. Hoboken (New Jersey) : John Wiley & Sons, 2008. 460 p. ISBN 978-0-470-23286-6.
30. WILLIAMS, H. Paul. 2013. *Model Building in Mathematical Programming*. 5th ed. Chichester (UK) : John Wiley & Sons, 2013. 411 p. ISBN 978-1-118-44333-0.