

**EKONOMICKÁ UNIVERZITA V BRATISLAVE**  
**FAKULTA HOSPODÁRSKEJ INFORMATIKY**

Evidenčné číslo: 103003/I/2025/36124048430884612

**VYUŽITIE PROBLÉMU BATOHA PRI ALOKÁCII**  
**VEREJNÝCH ZDROJOV**

**Diplomová práca**

**Bratislava 2023**

**Bc. Stanislav Hanes**

**EKONOMICKÁ UNIVERZITA V BRATISLAVE**  
**FAKULTA HOSPODÁRSKEJ INFORMATIKY**

**VYUŽITIE PROBLÉMU BATOHA PRI ALOKÁCII  
VEREJNÝCH ZDROJOV**

**Diplomová práca**

**Študijný program:** Data Science v Ekonómii

**Študijný odbor:** Ekonómia a manažment

**Školiace pracovisko:** Katedra operačného výskumu a ekonometrie

**Vedúci záverečnej práce:** Ing. Pavel Gežík, PhD.

**Bratislava 2025**

**Bc. Stanislav Hanes**

## ABSTRAKT

HANES, Stanislav: *Využitie problému batoha pri alokácii verejných zdrojov* – Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra operačného výskumu a ekonometrie – Vedúci záverečnej práce: Ing. Pavel Gežík, PhD. – Bratislava: FHI EU, 2025, 64 s.

Záverečná práca je vypracovaná na tému využitie problému batoha pri alokácii verejných zdrojov.

Cieľom záverečnej práce je optimalizácia výberu projektov financovaných z verejných zdrojov v rámci Programu rozvoja vidieka SR 2014 – 2022, a to prostredníctvom aplikácie a porovnania viacerých prístupov založených na riešení viacnásobného 0 – 1 problému batoha. Porovnávané sú z hľadiska výpočtovej náročnosti, kvality riešenia a praktickej aplikácie. Údaje o žiadostiach o nenávratný finančný príspevok boli získali na základe oficiálnej žiadosti priamo od oddelenia monitoringu projektových podpôr Pôdohospodárskej platobnej agentúry. Výsledkom záverečnej práce je návrh vlastného modelu so spätným výberom projektov, ktorý bol porovnaný so súčasným mechanizmom alokácie zdrojov. Pridaná hodnota práce spočíva v aplikácii teoretického optimalizačného modelu na reálny proces výberu projektov verejnej správy a v možnosti jeho využitia pri nastavovaní alokačných mechanizmov v budúcich programových obdobiach. Práca je rozdelená do štyroch kapitol, ktoré sú doplnené dvanástimi obrázkami a desiatimi tabuľkami.

Prvá kapitola sa zaoberá teoretickými východiskami problému batoha, jeho variantmi a algoritmami riešenia. Zároveň je v tejto časti predstavená činnosť Pôdohospodárskej platobnej agentúry a mechanizmus výberu projektov v rámci Programu rozvoja vidieka SR 2014 – 2022. Druhá kapitola definuje hlavný cieľ záverečnej práce. Tretia kapitola obsahuje popis použitej metodiky a vysvetľuje, ako bol problém výberu projektov modelovaný ako viacnásobný 0 – 1 problém batoha. V štvrtej kapitole sú prezentované výsledky porovnania vybraných prístupov a rovnako tak navrhovaného modelu so spätným výberom projektov.

**Kľúčové slová:** problém batoha, pažravý algoritmus, algoritmus dynamického programovania, binárne celočíselne lineárne programovanie

## ABSTRACT

HANES, Stanislav: *Application of the Knapsack Problem in the allocation of public funds* – University of Economics in Bratislava, Faculty of Economic Informatics; Department of Operations Research and Econometrics – Thesis supervisor: Ing. Pavel Gežík, PhD. – Bratislava: FHI EU, 2025, 64 pages.

This thesis addresses how the knapsack problem can be used to optimize the allocation of public funds.

The aim of this thesis is to optimize the selection of projects funded by public sources under the Slovak Rural Development Programme 2014–2022, by applying and comparing several approaches based on solving the multiple 0–1 knapsack problem. The methods are compared in terms of computational complexity, solution quality, and practical applicability. Data on applications for non-repayable financial contributions were obtained directly from the Project Support Monitoring Department of the Agricultural Paying Agency, based on an official request. The result of the thesis is a proposed model with reverse project selection, which is compared to the current resource allocation mechanism. This thesis adds value by applying a theoretical optimization model in the context of public sector project selection, offering potential for use in future programming periods. The thesis is divided into four chapters, which are accompanied by twelve figures and ten tables.

The first chapter deals with the theoretical background of the knapsack problem, its variants, and solution methods. It also presents the role of the Agricultural Paying Agency and explains the project selection mechanism under the Slovak Rural Development Programme 2014–2022. The second chapter defines the main objective of the thesis. The third chapter describes the methodologies used and explains how the project selection problem was modelled as a multiple 0–1 knapsack problem. The fourth chapter presents the results of the comparison of the selected approaches, including the proposed reverse-selection model.

**Keywords:** knapsack problem, greedy algorithm, dynamic programming algorithm, binary integer linear programming

# OBSAH

|                                                                         |           |
|-------------------------------------------------------------------------|-----------|
| Úvod .....                                                              | 7         |
| <b>1 Súčasný stav riešenej problematiky doma a v zahraničí.....</b>     | <b>8</b>  |
| 1.1 Problém batoha .....                                                | 9         |
| 1.1.1 0 – 1 problém batoha .....                                        | 10        |
| 1.1.2 Problém batoha s čiastkovým súčtom.....                           | 10        |
| 1.1.3 Frakčný problém batoha .....                                      | 10        |
| 1.1.4 Ohraničený problém batoha.....                                    | 11        |
| 1.1.5 Neohraničený problém batoha .....                                 | 11        |
| 1.1.6 Viacnásobný problém batoha a jeho modifikácie .....               | 12        |
| 1.1.7 Nelineárne varianty problému batoha.....                          | 15        |
| 1.2 Výpočtová zložitosť problému batoha a algoritmy jeho riešenia ..... | 17        |
| 1.2.1 Exaktné metódy .....                                              | 17        |
| 1.2.2 Aproximačné metódy.....                                           | 19        |
| 1.2.3 Heuristické a metaheuristické metódy .....                        | 20        |
| 1.3 Pôdohospodárska platobná agentúra.....                              | 23        |
| 1.4 Program rozvoja vidieka Slovenskej republiky 2014 – 2022 .....      | 25        |
| 1.4.1 Proces výberu projektov na realizáciu .....                       | 25        |
| <b>2 Cieľ práce .....</b>                                               | <b>27</b> |
| <b>3 Metodika práce a metódy skúmania .....</b>                         | <b>28</b> |
| <b>4 Výsledky práce a diskusia .....</b>                                | <b>38</b> |
| 4.1 Dataset .....                                                       | 38        |
| 4.2 Porovnanie vybraných metód .....                                    | 41        |
| 4.2.1 Pažravý algoritmus .....                                          | 41        |
| 4.2.2 Dynamické programovanie .....                                     | 43        |
| 4.2.3 Binárne celočíselné lineárne programovanie .....                  | 46        |
| 4.2.4 Syntéza porovnania vybraných algoritmov .....                     | 48        |
| 4.3 Model so spätným využitím rozpočtu.....                             | 51        |
| 4.3.1 Komparácia modelu s výberom projektov PPA.....                    | 56        |
| <b>Záver .....</b>                                                      | <b>59</b> |
| <b>Zoznam použitej literatúry .....</b>                                 | <b>61</b> |
| <b>Prílohy.....</b>                                                     | <b>64</b> |

## Zoznam obrázkov a tabuliek

|                                                                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Obrázok č. 1:</b> Diagram modelu so spätným výberom projektov.....                                                                                          | 37 |
| <b>Obrázok č. 2:</b> Hlavná časť skriptu Pažravého algoritmu.....                                                                                              | 42 |
| <b>Obrázok č. 3:</b> Hlavná časť skriptu Dynamického programovania .....                                                                                       | 44 |
| <b>Obrázok č. 4:</b> Podmienka v spätnom výbere projektov dynamického programovania.....                                                                       | 45 |
| <b>Obrázok č. 5:</b> Hlavná časť skriptu binárneho celočíselného lineárneho programovania ...                                                                  | 47 |
| <b>Obrázok č. 6:</b> Porovnanie podielu získaných bodov v rámci vybraných výziev .....                                                                         | 48 |
| <b>Obrázok č. 7:</b> Porovnanie podielu alokovaného rozpočtu vybraných algoritmov .....                                                                        | 49 |
| <b>Obrázok č. 8:</b> 1. fáza modelu so spätným výberom projektov .....                                                                                         | 51 |
| <b>Obrázok č. 9:</b> 2. fáza modelu so spätným výberom projektov .....                                                                                         | 52 |
| <b>Obrázok č. 10:</b> Podiel alokovaného rozpočtu a vybraných projektov.....                                                                                   | 54 |
| <b>Obrázok č. 11:</b> Počet vybraných projektov v rámci 2. fázy algoritmu a počet nevybraných projektov, ktoré sa kumulujú .....                               | 55 |
| <b>Obrázok č. 12:</b> Porovnanie územného rozloženia dosiahnutých bodov za vybrané projekty a financovanie podľa krajov: BILP verzus reálna alokácia PPA ..... | 57 |
|                                                                                                                                                                |    |
| <b>Tabuľka č. 1:</b> Reprezentatívne projektové žiadosti o NFP v PRV 2014-2022 k 31. 12. 2024 .....                                                            | 40 |
| <b>Tabuľka č. 2:</b> Reprezentatívne výzvy v rámci PRV 2014 - 2022 .....                                                                                       | 40 |
| <b>Tabuľka č. 3:</b> Výsledky Pažravého algoritmu .....                                                                                                        | 42 |
| <b>Tabuľka č. 4:</b> Výsledky dynamického programovania .....                                                                                                  | 45 |
| <b>Tabuľka č. 5:</b> Výsledky dynamického programovania po pridaní dodatočnej podmienky                                                                        | 46 |
| <b>Tabuľka č. 6:</b> Výsledky binárneho celočíselného lineárneho programovania .....                                                                           | 47 |
| <b>Tabuľka č. 7:</b> Porovnanie výsledkov a výpočtovej náročnosti vybraných algoritmov .....                                                                   | 50 |
| <b>Tabuľka č. 8:</b> Výsledky porovnania riešiteľov pre model so spätným výberom projektov                                                                     | 52 |
| <b>Tabuľka č. 9:</b> Reprezentácia výsledkov na podklade prvých päť iterácii modelu .....                                                                      | 54 |
| <b>Tabuľka č. 10:</b> Porovnanie výberu modelu so spätným výberom projektov s výberom PPA.....                                                                 | 56 |
| <b>Tabuľka č. 11:</b> Porovnanie výberu modelu BILP a reálna alokácia PPA podľa kraja.....                                                                     | 58 |

## Úvod

Rozhodovanie o efektívnom rozdelení obmedzených zdrojov je mimoriadne dôležité, najmä v oblasti verejných financií. Toto rozhodovanie má priamy vplyv na rozvoj spoločnosti, stability a hospodársku výkonnosť krajiny. Jedným z prístupov, ktorý sa využíva pri modelovaní takýchto rozhodovacích problémov je problém batoha, ktorý patrí medzi klasické optimalizačné úlohy a umožňuje vybrať z množiny alternatív takú kombináciu, ktorá pri daných obmedzeniach maximalizuje celkový úžitok.

V práci uvidíme teoretické východiská problému batoha a jeho varianty, ktoré vychádzajú z rôznych praktických aplikácií a najčastejšie využívané algoritmy na jeho riešenie. Úvodná kapitola bude zároveň prezentovať pôsobnosť Pôdohospodárskej platobnej agentúry a opisovať mechanizmus výberu projektov v rámci Programu rozvoja vidieka Slovenskej republiky 2014 – 2022, ktorý tvorí aplikačný rámec tejto práce. Výber tohto procesu bol motivovaný snahou demonštrovať na reálnom príklade, ako môžu optimalizačné algoritmy prispieť k efektívnemu a správne rozdeleniu verejných zdrojov. V súčasnosti sa totiž výber projektov v rámci tohto procesu zameriava skôr na individuálny prínos jednotlivých žiadostí, bez zohľadnenia celkového prínosu v rámci dostupného rozpočtu.

V nasledujúcej časti práce bude definovaný cieľ práce a popis metodiky akým je výber projektov modelovaný ako viacnásobný 0–1 problém batoha. V tejto časti budú vysvetlené princípy použitých algoritmov a konštrukcia vlastného modelu so spätným výberom projektov.

V závere práce následne porovnáme výsledky dosiahnuté rôznymi optimalizačnými prístupmi, pričom dôraz bude kladený na kvalitu riešenia a možnosť aplikácie v podmienkach verejnej správy. Súčasťou bude aj predstavenie navrhovaného optimalizačného modelu založeného na najvhodnejšom algoritme so spätnou selekciou projektov, ktorý má potenciál priniesť významné zvýšenie efektivity a transparentnosti v procese výberu projektov. Výsledky tohto modelu budú následne komparované s aplikovaným postupom Pôdohospodárskej platobnej agentúry v súčasnosti. Predpokladaným prínosom práce je návrh aplikovateľného riešenia, ktoré môže zlepšiť rozhodovanie pri prerozdeľovaní verejných zdrojov a tak slúžiť ako metodologický základ pre pripravované programové obdobie.

# 1 Súčasný stav riešenej problematiky doma a v zahraničí

Rozhodovanie je neoddeliteľnou súčasťou všetkých oblastí ľudského života. V osobnej rovine býva často ovplyvnené emóciami, intuícou alebo individuálnymi preferenciami. V oblasti hospodárenia, plánovania alebo riadenia je však potrebné rozhodovať sa na základe racionálnych postupov, ktoré sú založené na jasne definovaných cieľoch a kvantifikovateľných ukazovateľoch. Takýto prístup zabezpečuje objektivitu, znižuje mieru neistoty a zvyšuje šancu na efektívne a opodstatnené rozhodnutia, najmä v situáciách, kde je potrebné efektívne narábať s obmedzenými zdrojmi. Práve na tomto princípe je postavený problém batoha, ktorý predstavuje najjednoduchší model v oblasti optimalizačných úloh. (Kellerer, Pferschy a Pisinger, 2004)

Už v roku 1957 ho George Dantzig ilustroval nasledovne: „*V tomto probléme si človek plánuje turistiku a rozhodol sa, že si nevezme viac ako 70 libier rôznych vecí, ako je napríklad spací vak, ..., konzervy s jedlom...*“<sup>1</sup> Môžeme si tak predstaviť, že kapacita „batoha“ je v tomto prípade 70 libier a chceme zvoliť takú kombináciu „vecí“, aby sme maximalizovali úžitok vybraných „vecí“.

Vďaka svojej prehľadnej štruktúre a schopnosti efektívne modelovať reálne rozhodovacie situácie sa problém batoha stal základom množstva praktických aplikácií. Tradične nachádza uplatnenie najmä v oblasti logistiky, kde slúži na optimalizáciu využitia kapacity pri preprave tovaru, ako aj v oblasti kapitálového rozpočtovania, kde sa rozhoduje o výbere investičných projektov v podmienkach obmedzeného rozpočtu. Rovnako sa využíva pri plánovaní výroby, riadení zásob a všeobecne pri efektívnom rozdeľovaní limitovaných zdrojov. Postupom času sa problém batoha začal uplatňovať aj v oblasti informačných technológií, najmä pri priradovaní výpočtových úloh, optimalizácii dátových prenosov a rozdelení výpočtového zaťaženia medzi dostupné uzly v sieťach. Uplatnenie nachádza aj pri návrhu šifrovacích mechanizmov, kde sa niektoré varianty problému batoha využívajú ako základ bezpečnostných algoritmov, známy je napríklad Merkle-Hellmanov kryptosystém. V súčasnosti nachádza problém batoha uplatnenie aj v špecializovaných oblastiach, ktoré s tradičnými optimalizačnými úlohami na prvý pohľad priamo nesúvisia.

---

<sup>1</sup> DANTZIG, George B. *Discrete-variable extremum problems* [online]. Santa Monica: The RAND Corporation, 1957, s. 14 [cit. 30. 03. 2025]. Dostupné na: [https://www.rand.org/content/dam/rand/pubs/research\\_memoranda/2008/RM1832.pdf](https://www.rand.org/content/dam/rand/pubs/research_memoranda/2008/RM1832.pdf)

Príkladom je riadenie spotreby energie v inteligentných domácnostiach, využitie v mediálnom prostredí pri výbere reklamných spotov pre obmedzený vysielací čas, ako aj aplikácie v odpadovom hospodárstve. (Biglar, 2018)

Keďže problém batoha patrí medzi klasické optimalizačné úlohy využívané pri rozhodovaní o alokácii zdrojov či rozpočtového plánovania, jeho využitie je opodstatnené aj pri rozdeľovaní verejných zdrojov, keďže tieto sú dostupné len v obmedzenom objeme. Efektívna alokácia preto predstavuje kľúčový proces, v rámci ktorého je potrebné určiť, ako čo najvýhodnejšie rozčleniť dostupné prostriedky s cieľom dosiahnuť čo najväčší spoločenský prínos. Ide o výber takej kombinácie výdavkov, ktorá najviac prispeje k napĺňaniu priorít verejnej politiky a potrieb obyvateľstva. Pri rozdeľovaní verejných prostriedkov je nevyhnutné zohľadniť nielen aktuálne rozpočtové obmedzenia, ale aj dlhodobé strategické zámery. Efektívna alokácia verejných zdrojov tak prispieva k hospodárskemu rastu, pomáha zmierňovať sociálne a regionálne nerovnosti a prispieva k stabilite verejných inštitúcií a dlhodobému zvyšovaniu kvality života. Transparentné a zodpovedné rozhodovanie pri využívaní verejných financií zároveň posilňuje dôveru občanov vo verejné inštitúcie, čo je zásadnou podmienkou pre fungovanie demokratického systému. (Cibík, 2023)

## **1.1 Problém batoha**

Základným predpokladom pre formuláciu problému batoha je schopnosť kvantifikovať parametre objektov, ktoré sú dôležité pre posúdenie jednotlivých alternatív. Tieto parametre môžu reflektovať rôzne charakteristiky, ako sú nároky na zdroje - váha, či očakávaný úžitok - hodnota. Iba v tomto prípade je tak možné implementovať optimalizačné metódy. V klasickej formulácii problému batoha následne uvažujeme množinu objektov, z ktorých každý je definovaný týmito parametrami. Jedným z nich je, ako bolo vyššie spomínané očakávaný úžitok a druhým nárok na zdroje. Cieľom je vybrať takú podmnožinu týchto objektov, ktorá maximalizuje celkový úžitok pri dodržaní kapacitného obmedzenia daného zdrojom. (Kellerer et al., 2004)

V reálnych aplikáciách často vstupujú do rozhodovania ďalšie faktory, ako sú rôzne typy zdrojov, závislosti medzi objektmi alebo špecifické obmedzenia. Tieto skutočnosti podnietili intenzívny výskum problému batoha a vznik mnohých jeho variantov. V nasledujúcej časti práce si preto stručne predstavíme najčastejšie varianty tohto problému.

### 1.1.1 0 – 1 problém batoha

Najbežnejším a zároveň najjednoduchším interpretovateľným variantom pre tento problém je 0 – 1 problém batoha. V tomto prípade uvažujeme, že máme  $n$  objektov a každý objekt  $j$  má úžitok  $v_j$  a váhu  $w_j$ . Rozhodnutie spočíva v tom, či daný objekt zaradíme do riešenia práve raz, alebo vôbec. V tomto variante je  $x_j$  počet kópii vybraného objektu  $j$ , pričom  $x_j$  predstavuje binárnu premennú, ktorá nadobúda hodnotu 1, ak je objekt zahrnutý do množiny riešenia, a hodnotu 0, ak nie je. Uvažujeme, že  $W$  je maximálna dostupná kapacita, pričom váhy a úžitok objektov sú nezáporné. Tento problém je daný nasledovne: (Assi a Haraty, 2018)

$$\max \sum_{j=1}^n v_j x_j$$

za podmienky:

$$\sum_{j=1}^n w_j x_j \leq W$$

kde:

$$x_{ij} \in \{0,1\}$$

Cieľom je maximalizovať súčet hodnôt vybraných objektov, pričom sa nepresiahne maximálna dostupná kapacita ( $W$ ).

### 1.1.2 Problém batoha s čiastkovým súčtom

Jedným z najbližších príbuzných vyššie spomínaného problému je problém batoha s čiastkovým súčtom. V tomto prípade sa zameriavame výlučne na váhu objektov  $w_j$ , pričom cieľom je nájsť takú podmnožinu objektov, ktorej celková váha nepresiahne alebo presne dosiahne danú kapacitu  $W$ . Úžitok objektov pritom nezohráva žiadnu úlohu. Úlohou je tak určiť, či existuje kombinácia objektov, ktorá rešpektuje kapacitné obmedzenia. Hľadáme teda rozhodovacie premenné, ktoré spĺňajú: (Gerasch a Wang, 1994)

$$\sum_{j=1}^n w_j x_j \leq W$$

### 1.1.3 Frakčný problém batoha

Za najstarší a najprirodzenejší spôsob modifikácie 0-1 problému batoha sa považuje jeho lineárna relaxácia, ktorá umožňuje brať z položiek aj zlomky. Tento prístup ako prvý

predstavil George B. Dantzig už v roku 1957. Pri tejto úprave sa odstraňuje podmienka binárnosti rozhodovacích premenných a teda nie je nutné, aby nadobúdali výlučne hodnoty 0 alebo 1. Premenné  $x_j$  môžu namiesto toho nadobúdať spojité hodnoty v intervale  $0 \leq x_j \leq 1$ . (Martello a Toth, 1990)

Frakčný problém batoha preto umožňuje, že jednotlivé objekty možno zaradiť do riešenia aj čiastočne, nielen úplne alebo vôbec. Tento prístup je vhodný najmä v prípadoch, kde sa pracuje s deliteľnými prvkami, ako napríklad preprava kvapalín, sypkých surovín alebo iného homogénneho materiálu.

#### 1.1.4 Ohraničený problém batoha

Častejšie ako pri delení jednotlivých objektov je možné sa v praxi stretnúť s prípadmi, kedy sa objekty vyskytujú vo viacerých identických kópiách. Typickým príkladom môže byť preprava tovaru, kde sú prepravované objekty s identickými parametrami – hmotnosťou  $w_j$  a hodnotou  $v_j$ .

V takomto prípade už nerozhodujeme iba o tom, či zaradiť alebo nezaradiť jeden konkrétny objekt, ale aj o počte identických kusov, ktoré sa zmestia do obmedzenej kapacity. Cieľom je, rovnako ako v predošlých prípadoch maximalizovať celkovú hodnotu vybraných objektov, v tomto prípade prevážaných, pričom rešpektujeme kapacitné obmedzenie. Takáto situácia vedie k formulácii ohraničeného problému batoha, v ktorom je počet zaradených kusov objektu vyjadrený celočíselnou premennou  $x_j$ , pričom celkový počet identických kópií daného objektu je označený ako  $b_j$ . Práve existencia hornej hranice  $b_j$ , ktorá obmedzuje maximálnu možnú hodnotu pre  $x_j$ , je dôvodom, prečo sa tento variant označuje ako ohraničený. Premenná  $x_j$  tak nadobúda hodnoty z intervalu  $0 \leq x_j \leq b_j$ . (Munapo, 2020)

#### 1.1.5 Neohraničený problém batoha

Oproti predošlej situácii, kde sme mali obmedzený počet dostupných kópií z každého typu objektu ohraničený konkrétnou hodnotou  $b_j$ , sa v tomto variante stretávame s opačným predpokladom. Každý objekt  $j$  je dostupný v neobmedzenom počte a je možné ho zaradiť do riešenia ľubovoľným počtom krát. Samozrejme za predpokladu, že je pri všetkých zaradených objektoch dodržaná kapacitná podmienka  $W$ . Rovnako ako v predchádzajúcich variantoch je cieľom maximalizovať celkovú hodnotu vybraných objektov. Rozdielom je, že premenná  $x_j$  v tomto variante môže nadobúdať ľubovoľnú hodnotu z množiny celých nezáporných čísel, teda: (Kellerer et al., 2004)

$$x_j \geq 0, \quad x_j \in \mathbb{Z}$$

Klasickým príkladom tohto variantu je problém výmeny mincí (*coin change problem*), ktorý zároveň predstavuje špeciálny typ problému batoha. Na rozdiel od klasickej formulácie problému batoha, kde je cieľom maximalizovať celkovú hodnotu vybraných objektov, sa v tomto prípade snažíme minimalizovať počet použitých objektov, teda mincí.

Ide teda o minimalizačnú úlohu, kde rozhodovacie premenné  $x_j$  vyjadrujú počet použití mincí  $j$ . Hodnoty  $w_j$ , ktoré v klasickom modeli predstavujú váhu obmedzujúcu kapacitu zdroja, tu vstupujú ako nominálne hodnoty mincí, ktorých súčet musí presne zodpovedať cieľovej sume  $S$ . Premenné  $x_j$  zároveň tvoria jadro cieľovej funkcie, ktorá vyjadruje celkový počet použitých mincí. Problém výmeny mincí teda spočíva v nájdení takej kombinácie dostupných nominálnych hodnôt, ktorá umožní presne vyjadriť cieľovú sumu  $S$ , pričom každú mincu možno použiť ľubovoľne veľakrát. Formálne, ak máme množinu mincí s hodnotami  $w_1, w_2, \dots, w_n$  a hľadáme celočíselné nezáporné premenné  $x_j$ , ktoré spĺňajú: (Kozen a Zaks, 1994)

$$\sum_{j=1}^n w_j x_j = S, \quad x_j \geq 0, \quad x_j \in \mathbb{Z}$$

$$\min \sum_{j=1}^n x_j$$

### 1.1.6 Viacnásobný problém batoha a jeho modifikácie

V prípadoch, kedy je potrebné rozšíriť klasický model batoha o viacero nezávislých kapacitných obmedzení, hovoríme o viacnásobnom probléme batoha. Tento variant vzniká v situáciách, kde je potrebné rozhodovať o alokácii objektov medzi viacero dostupných zdrojov. Typickými aplikáciami sú úlohy súvisiace s rozdeľovaním zásielok medzi viaceré dopravné prostriedky, alokáciou projektov medzi rôzne rozpočtové balíky alebo distribúciou výpočtových úloh v rámci IT infraštruktúry.

Na rozdiel od klasického problému batoha, kde optimalizujeme využitie jednej kapacity  $W$ , vo viacnásobnom probléme batoha uvažujeme  $m$  batohov, pričom každý z nich má vlastnú kapacitu  $W_i$ . Cieľom je tak rozhodnúť, ktoré objekty zaradiť a do ktorého batoha tak, aby sa maximalizoval celkový úžitok vybraných objektov a zároveň platilo, že každý objekt je možné zaradiť navyše jedenkrát. V dôsledku toho tak dochádza k modifikácii rozhodovacích premenných, ktoré sú reprezentované pomocou dvojice indexov  $x_{ij}$ . Index  $i$  označuje konkrétnu kapacitu batoha (zdroja), a index  $j$  konkrétny objekt (úlohu, položku

a podobne), pričom hodnota 1 reprezentuje zaradenie objektu  $j$  do batoha  $i$ , a hodnota 0 opak. Formálne môžeme viacnásobný problém batoha zapísať nasledovne: (Kellerer et al., 2004)

$$\max \sum_{i=1}^m \sum_{j=1}^n v_j x_{ij}$$

za podmienok:

$$\sum_{j=1}^n w_j x_{ij} \leq W_i \quad \text{pre všetky } i = 1, \dots, m$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad \text{pre všetky } j = 1, \dots, n$$

$$x_{ij} \in \{0,1\}$$

Tento model zároveň predstavuje východiskový rámec pre viaceré rozšírenia, ktoré umožňujú presnejšie zohľadniť praktickú aplikáciu. Medzi najčastejšie využívané patrí napríklad viacnásobný problém batoha s rovnakými kapacitami, pri ktorom sa predpokladá, že všetky dostupné zdroje majú rovnakú kapacitu  $W$ . V takom prípade je možné zjednodušiť kapacitné obmedzenia nasledovne: (Kellerer et al., 2004)

$$\sum_{j=1}^n w_j x_{ij} \leq W, \quad \text{pre všetky } i = 1, \dots, m.$$

Ďalším variantom je viacnásobný problém batoha s čiastkovým súčtom, kde úžitok každého objektu je rovný jej váhe a teda platí  $v_j = w_j$ . Cieľom v tomto prípade je maximalizovať celkové využitie dostupných kapacít: (Kellerer et al., 2004)

$$\max \sum_{i=1}^m \sum_{j=1}^n w_j x_{ij}$$

Oba prístupy možno kombinovať v modifikácii viacnásobného problému batoha s rovnakými kapacitami a čiastkovým súčtom, ktorý predpokladá identické kapacitné obmedzenia  $W$  a rovnosť medzi úžitkom a váhou jednotlivých objektov  $v_j = w_j$ .

Špecifickým rozšírením je takzvaná „bottleneck“ verzia viacnásobného problému batoha, pri ktorej je cieľom minimalizovať najviac zaťažený batoh, teda optimalizovať rovnomerné rozdelenie kapacity medzi jednotlivé zdroje. Takýto prístup vedie k nelineárnemu modelu, a preto účelová funkcia nadobúda tvar: (Caprara, Kellerer a Pferschy, 2000)

$$\min \left( \max_{i=1, \dots, m} \sum_{j=1}^n w_j x_{ij} \right)$$

Okrem uvedených variantov poznáme aj generalizovaný priradovací problém, ktorý v rámci každej kombinácie objektov a jednotlivých zdrojov definuje špecifický úžitok  $v_{ij}$  a zároveň individuálnu spotrebu kapacity  $w_j$ . Cieľom je maximalizovať celkový úžitok pri rešpektovaní individuálnych kapacít jednotlivých zdrojov pre jednotlivé kombinácie objektov a zdrojov: (Kellerer et al., 2004)

$$\max \sum_{i=1}^m \sum_{j=1}^n v_{ij} x_{ij}$$

za podmienok:

$$\sum_{j=1}^n w_{ij} x_{ij} \leq W_i, \quad \text{pre všetky } i = 1, \dots, m,$$

$$\sum_{i=1}^m x_{ij} \leq 1, \quad \text{pre všetky } j = 1, \dots, n,$$

$$x_{ij} \in \{0,1\}$$

Najkomplexnejší model, ktorý rozširuje viacnásobný problém batoha, je multidimenzionálny problém viacnásobného batoha. Tento variant kombinuje výber objektov z rôznych tried s viacerými kapacitnými obmedzeniami.

Predpokladáme, že existuje  $T$  tried, pričom každá trieda  $t$  obsahuje niekoľko objektov  $j$ . Každý objekt má priradený úžitok  $v_{tj}$  a v každej z  $L$  dimenzií má určitú spotrebu kapacity zdroja  $w_{tjk}$ . Celková dostupná kapacita v dimenzii  $k$  je označená ako  $W_k$ . Cieľom je z každej triedy vybrať najviac jeden objekt tak, aby celkový úžitok bol maximálny a zároveň neboli prekročené kapacity v žiadnej z dimenzií. Tento problém možno formulovať nasledovne: (Assi a Haraty, 2018)

$$\max \sum_{t=1}^T \sum_{j=1}^{n_t} v_{tj} x_{tj}$$

za podmienok:

$$\sum_{t=1}^T \sum_{j=1}^{n_t} w_{tjk} x_{tj} \leq W_k, \quad \text{pre všetky } k = 1, \dots, L$$

$$\sum_{j=1}^{n_t} x_{tj} = 1, \quad \text{pre všetky } t = 1, \dots, T, \quad x_{tj} \in \{0,1\}$$

### 1.1.7 Nelineárne varianty problému batoha

Nie všetky vzťahy, ktoré sa vyskytujú pri modelovaní reálnych problémov možno vyjadriť lineárne. Mnohé z nich sú nelineárne vzhľadom na správanie premenných, ich vzájomné závislosti alebo charakter vzťahov medzi veličinami. Práve preto vznikli modifikácie klasického problému batoha vo forme jeho nelineárnych variantov. Typickými príkladmi sú plánovanie výroby, kde prínos ďalších jednotiek produkcie prestáva rásť lineárne, napríklad z dôvodu presýtenia trhu, alebo alokácia marketingového rozpočtu, kde návratnosť investície rastie čoraz pomalšie kvôli klesajúcemu marginálnemu efektu jednotlivých cieľných skupín.

Nelineárny problém batoha predstavuje rozšírenie klasického problému batoha, v ktorom sa maximalizuje súčet individuálnych úžitkov prislúchajúcich jednotlivým objektom. V najvšeobecnejšej forme, ako ju uvádza Hochbaum (1995), ide o maximalizáciu súčtu funkcií  $v_j(x_j)$ , pričom každá z nich závisí od premennej  $x_j$ , za podmienky, že súčet príslušných funkcií  $w_j(x_j)$  neprekročí danú kapacitu  $W$ .

Funkcie  $v_j(x_j)$  sa predpokladajú ako konkávne a neklesajúce, zatiaľ čo funkcie  $w_j(x_j)$  sú konvexné a takisto neklesajúce. Takto formulovaný model umožňuje zachytiť situácie, v ktorých ďalšie jednotky vybraných objektov prinášajú čoraz menší prínos, zatiaľ čo ich spotreba zdrojov rastie čoraz rýchlejšie. Premenné  $x_j$  nadobúdajú celočíselné hodnoty v rozsahu od 0 po  $u_j$ , pričom kapacita  $W$  a horné ohraničenia  $u_j$  sa považujú za celé čísla. Matematicky potom možno tento problém formulovať: (Hochbaum, 1995)

$$\max \sum_{j=1}^n v_j(x_j)$$

za podmienok:

$$\sum_{j=1}^n w_j(x_j) \leq W$$

$$0 \leq x_j \leq u_j, \quad x_j \in \mathbb{Z}, \quad j = 1, 2, \dots, n$$

Medzi nelineárne rozšírenia problému batoha patrí takisto kvadratický problém batoha, v ktorom cieľová funkcia zohľadňuje nielen individuálne prínosy objektov, ale aj ich vzájomné interakcie. Cieľová funkcia bude potom mať tvar: (Kellerer et al., 2004)

$$\max \sum_{j=1}^n \sum_{k=1}^n v_{jk} x_j x_k$$

za podmienok:

$$\sum_{j=1}^n w_j x_j \leq W$$

$$x_j \in \{0,1\}, \quad j = 1, 2, \dots, n$$

kde:

- súčin  $x_j x_k$  nadobúda hodnotu 1 len vtedy, ak sú vybrané oba objekty.

Tento variant umožňuje realistickejšie modelovať situácie, v ktorých kombinácia vybraných objektov prináša väčší alebo menší efekt než jednoduchý súčet ich samostatných prínosov. Vďaka tejto vlastnosti sa kvadratický model využíva v prípadoch, kde je potrebné zohľadniť vzťahy medzi objektmi alebo dopady ich súčasného výskytu v riešení.

Výber uvedených variantov problému batoha bol zameraný na poskytnutie reprezentatívneho prierezu najčastejšie využívanými a v odbornej literatúre často diskutovanými prístupmi, ktoré nachádzajú uplatnenie v rôznych oblastiach optimalizačných úloh. Tieto základné formy tvoria fundamentálny rámec pre modelovanie situácií, v ktorých je potrebné efektívne rozhodovať pri obmedzených zdrojoch, a zároveň predstavujú vstupnú bránu k celému spektru pokročilých variantov, ktoré reagujú na špecifiká reálnych rozhodovacích situácií. Okrem klasických variantov existuje aj široká paleta rozšírených foriem, ako sú stochastické modely zohľadňujúce neistotu vstupov či dynamické modely pracujúce s rozhodovaním v čase. Táto rôznorodosť poukazuje na vysokú univerzálnosť problému batoha a jeho schopnosť prispôbiť sa rôznym kontextom, čím sa stal jedným z najskúmanejších a najaplikovanejších modelov v oblasti kombinatorickej optimalizácie.

## 1.2 Výpočtová zložitosť problému batoha a algoritmy jeho riešenia

Teória výpočtovej zložitosti sa zaoberá skúmaním náročnosti riešenia problémov z hľadiska objemu potrebných výpočtových prostriedkov. Neanalyzuje iba to, či je daný problém riešiteľný, ale predovšetkým to, ako efektívne sa dá vyriešiť vzhľadom na časovú a pamäťovú náročnosť pri zväčšujúcom sa vstupe. Za ústredný problém teórie výpočtovej zložitosti sa považuje známy problém P verzus NP, ktorý sa zaoberá vzťahom medzi dvomi triedami problémov. Trieda P zahŕňa úlohy, ktoré je možné vyriešiť pomocou deterministického algoritmu v polynomiálnom čase, zatiaľ čo trieda NP predstavuje množinu problémov, pri ktorých možno správnosť daného riešenia overiť v polynomiálnom čase. Základnou otázkou v rámci problému P verzus NP je, či existuje taký algoritmus, ktorý dokáže nájsť riešenie rovnako rýchlo, ako vieme overiť jeho správnosť. (Arora a Barak, 2009)

V oblasti kombinatorickej optimalizácie patrí problém batoha medzi najznámejšie a najčastejšie skúmané úlohy. Je klasickým príkladom NP-ťažkého problému, ktorý je zaradený aj medzi 21 NP-úplných problémov podľa Karpa, no zároveň je často považovaný za jeden z „ľahších“ problémov tejto triedy, pretože pre mnohé jeho varianty existuje pseudo-polynomiálne riešenie. To znamená, že časová zložitosť riešenia závisí od veľkosti vstupu, a to nielen z hľadiska počtu prvkov, ale aj od veľkosti číselných hodnôt, ktoré vstup obsahuje. (Pisinger, 2004)

### 1.2.1 Exaktné metódy

Exaktné algoritmy predstavujú triedu výpočtových metód, ktorých cieľom je nájsť optimálne riešenie. Tieto algoritmy prehľadávajú celý priestor možných riešení systematicky, čím zabezpečujú nájdenie optimálneho riešenia. Exaktné algoritmy okrem iného slúžia aj ako referenčný rámec pri hodnotení kvality aproximačných a heuristických metód a taktiež prispievajú k hlbšiemu teoretickému porozumeniu štruktúry a komplexity riešených problémov. Významným obmedzením ich praktického použitia však zostáva výpočtová náročnosť, ktorá pri NP-ťažkých úlohách často narastá exponenciálne so zväčšujúcim sa vstupom. (Fiveable) Napríklad pri riešení klasického 0 – 1 problému batoha pomocou metódy hrubej sily, teda prehľadávania všetkých možných kombinácií riešeného problému je časová zložitosť  $O(2^n)$ , keďže je potrebné preskúmať každú možnú podmnožinu objektov. (Assi a Haraty, 2018)

Ako už bolo spomenuté, klasický 0 – 1 problém batoha patrí medzi slabo NP-ťažké úlohy. To znamená, že neexistuje algoritmus s polynomiálnou časovou zložitou, pokiaľ

neplatí  $P = NP$ , avšak existuje pseudo-polynomiálne riešenie. Toto riešenie poskytuje algoritmus dynamického programovania, ktorého teoretický základ položil Richard Bellman už v roku 1957. Tento algoritmus má časovú zložitosť  $O(nW)$ , kde  $n$  reprezentuje počet objektov a  $W$  kapacitu zdroja. Keďže výpočtová zložitosť závisí od číselnej hodnoty vstupu, a nie výlučne od jeho veľkosti, klasifikujeme ho ako pseudo-polynomiálny algoritmus. Tento prístup umožňuje nájsť optimálne riešenie so stredne veľkými číselnými vstupmi. (Assi a Haraty, 2018)

„„Najjednoduchšie“ jednorozmerné varianty problému batoha (ako napríklad klasický 0–1 problém batoha, problém s čiastkovým súčtom, ohraničený, neohraničený či problém výmeny mincí - patria medzi slabo NP-ťažké úlohy, čo znamená, že ich možno riešiť v pseudo-polynomiálnom čase pomocou dynamického programovania. Väčšina rozšírení a všeobecnejších formulácií problému batoha sú však silno NP-ťažké, teda nie je možné ich riešiť pseudo-polynomiálnym algoritmom, pokiaľ by neplatila rovnosť  $P = NP$ .“<sup>2</sup>

Tým sa dostávame k ďalším exaktným metódam na riešenie problému batoha, ktoré sú aplikovateľné na viacero variantov. Často využívanými prístupmi sú algoritmy vetvenia a ohraničenia a vetvenia a rezania. Metóda vetvenia a ohraničenia (Branch & Bound) sa štandardne zakladá na relaxovaní celočíselného problému na lineárny model. Ak je optimálne riešenie tohto modelu celočíselné, je zároveň riešením pôvodného problému. Ak riešenie nie je celočíselné, algoritmus vyberie jednu z premenných, ktorá je neceločíselná a vytvorí z nej dva nové podproblémy (vetvenie) v jednom sa táto premenná nastaví tak, aby bola menšia alebo rovná najbližšiemu menšiemu celému číslu, v druhom naopak väčšia alebo rovná najbližšiemu väčšiemu celému číslu (ohraničenie). Týmto spôsobom sa zmenší priestor možných riešení, ale zároveň sa zachovávajú všetky potenciálne celočíselné riešenia. Tento postup sa opakuje, až kým algoritmus nenájde riešenie, ktoré spĺňa celočíselné podmienky. V štúdiu Eliasa Munapa z roku 2020 autor navrhuje reformuláciu tohto postupu, ktorá znižuje výpočtovú náročnosť pridaním vhodne definovaných obmedzení ešte pred samotným spustením algoritmu. (Munapo, 2020)

Pri riešení komplexnejších variantov problému batoha alebo iných kombinatorických optimalizačných úloh, kde je výpočtový priestor príliš rozsiahly, vznikla rozšírená metóda Vetvenie a rezanie (Branch & Cut), ktorá kombinuje klasické vetvenie s využitím rezných rovín (Cutting planes), ktoré reprezentujú postupné pridávanie lineárnych obmedzení do

---

<sup>2</sup> CACCHIANI, Calentina – IORI, Manuel – LOCATELLI, Alberto – MARTELLO, Silvano. *Knapsack problems — An overview of recent advances. Part I: Single knapsack problems*. In European Journal of Operational Research, 2022, roč. 302, č. 1, s. 2. DOI: 10.1016/j.ejor.2022.01.041

relaxovaného modelu. Týmto spôsobom sa efektívne zužuje priestor prehľadávania a výrazne sa znižuje počet generovaných podproblémov. (Mitchell, 2002)

Keďže je problém batoha formulovaný ako úloha celočíselného, respektíve binárneho lineárneho programovania, na jej riešenie je možné využiť aj moderné prístupy prostredníctvom výkonných optimalizačných riešiteľov, ako sú napríklad CPLEX, Gurobi, CBC, GLPK, SCIP, XPRESS, LINGO a ďalšie, ktoré sú určené na riešenie zmiešaných celočíselných úloh (MIP – *Mixed Integer Programming*). Tieto nástroje implementujú pokročilé algoritmy spomínané vyššie a často kombinujú exaktné a heuristické prístupy, vďaka čomu dokážu efektívne riešiť aj stredne veľké úlohy NP-ťažkých problémov. (Achterberg, 2007)

Exaktným riešením pre frakčný problém batoha, ktorý predstavuje lineárnu relaxáciu 0 – 1 verzie, je pažravý algoritmus (Greedy algorithm) vychádzajúci z povahy tohto uvoľneného problému. Umožňuje do riešenia zaradiť aj časti objektov, vďaka čomu možno dosiahnuť optimálne riešenie v krátkom čase. Algoritmus triedi položky podľa pomeru úžitku k hmotnosti a postupne ich vkladá do batoha až do naplnenia jeho kapacity pri časovej zložitosti  $O(n \log n)$ . (Cormen et al., 2009)

### 1.2.2 Aproximačné metódy

Napriek tomu, že exaktné algoritmy umožňujú nájsť optimálne riešenie, ich praktická použiteľnosť je výrazne limitovaná výpočtovou zložitou, najmä pri problémoch s rozsiahlym vstupom. V reálnych podmienkach však nie je vždy nevyhnutné získať optimálne riešenie a často postačuje riešenie, ktoré je dostatočne blízke optimálnemu, no výrazne rýchlejšie vypočítateľné. Ako uvádzajú Cormen a kol. (2009), jednou z možných stratégií, ako sa vyrovnat' s výpočtovou náročnosťou NP-úplných problémov, je práve hľadanie približných riešení v polynomiálnom čase. Tento prístup predstavuje základ aproximačných algoritmov, ktoré generujú prípustné riešenia s kvantifikovateľnou odchýlkou od optimálneho riešenia. V istom zmysle tak „platíme“ za kratší výpočtový čas znížením kvality riešenia, pričom výsledkom je riešenie, ktoré je suboptimálne, no stále dostatočne kvalitné. Aproximačné algoritmy sú preto mimoriadne dôležité pre prípady, keď exaktné metódy zlyhávajú alebo sú prakticky neaplikovateľné. (Kellerer et al., 2004)

„Aproximačná schéma pre optimalizačný problém je rodina aproximačných algoritmov, ktorý ako vstup prijíma nielen samotnú inštanciu problému, ale aj hodnotu  $\varepsilon > 0$ , pričom pre každé pevne zvolené  $\varepsilon$  ide o  $(1+\varepsilon)$  aproximačný algoritmus.“<sup>3</sup>

Táto skutočnosť znamená, že parameter  $\varepsilon$ , zvolený používateľom, reprezentuje akceptovateľnú mieru aproximácie, teda čím je jeho hodnota nižšia, tým bližšie je výstup k optimálnemu riešeniu, avšak za cenu vyššej výpočtovej náročnosti.

Historicky prvú aproximačnú schému pre problém batoha predstavili Ibarra a Kim, ktorí navrhli plne polynomiálnu aproximačnú schému (FPTAS). Ich práca sa stala základom celej línie následných výskumov, ktoré pretrvávajú dodnes. (Eberle et al., 2004) Základný princíp FPTAS je založený na kombinácii viacerých techník. Najprv sa využije pažravý algoritmus, ktorý poskytne odhad hodnoty optimálneho riešenia. Tento odhad sa následne použije na škálovanie úžitkov jednotlivých objektov, čím sa zníži rozsah hodnôt a tým aj výpočtová náročnosť. Na takto upravené dáta sa nakoniec aplikuje algoritmus dynamického programovania, ktoré umožní získať riešenie s garantovanou maximálnou odchýlkou  $\varepsilon$  od optimálneho riešenia. (Kellerer et al., 2004)

Kým klasický variant 0 - 1 problému batoha umožňuje FPTAS, s časovou zložitou  $O(n^2/\varepsilon)$ , čiže výpočtový čas rastie polynomiálne so vstupom  $n$  a zároveň s presnosťou  $1/\varepsilon$ , pre jeho komplexnejší variant, viacnásobný problém batoha, nie je známe riešenie pomocou FPTAS. Napriek tomu sú pre tento problém známe iné aproximačné schémy, ako napríklad polynomiálne aproximačné schémy (PTAS) s časovou zložitou  $n^{f(\varepsilon)}$ . Tu  $f(\varepsilon)$  reprezentuje ľubovoľnú rastúcu funkciu závislú od  $\varepsilon$ , čo pri snahe o vysokú presnosť vedie k výraznému nárastu výpočtového času. Vylepšením tohto prístupu sú efektívne polynomiálne aproximačné schémy (EPTAS), ktoré sa vyznačujú časovou zložitou  $f(\varepsilon) * n^{O(1)}$ . Znamená to, že náročnejšia časť výpočtu závisí len od požadovanej presnosti  $\varepsilon$ , zatiaľ čo výpočtová zložitost' vo vzťahu k veľkosti  $n$  ostáva polynomiálna. EPTAS tak predstavuje kompromis medzi presnosťou a výpočtovou náročnosťou, keďže umožňuje dosiahnuť vysokú presnosť riešenia bez dramatického nárastu výpočtového času. (Eberle et al., 2004)

### 1.2.3 Heuristické a metaheuristické metódy

Niektoré komplexnejšie varianty problému batoha, najmä pri veľkých vstupných dátach, nie je možné efektívne riešiť ani pomocou aproximačných schém. V takýchto prípadoch sa využívajú heuristické metódy, ktoré síce nezaručujú nájdenie optimálneho

---

<sup>3</sup> CORMEN, Thomas H. – LEISERSON, Charles E. – RIVEST, Ronald L. – STEIN, Clifford. *Introduction to Algorithms*. 3. vyd. Cambridge, Massachusetts : MIT Press, 2009. ISBN 978-0-262-03384-8., s. 1107

riešenia a ani neumožňujú presne kvantifikovať jeho odchýlku od optima, ich prednosť však spočíva v nízkej výpočtovej náročnosti a rýchlosti získania výsledku.

Najznámejším a najčastejšie využívaným heuristickým prístupom pre riešenie 0 – 1 problému batoha je pažravý algoritmus. „*Pažravý algoritmus možno označiť ako „slepý“. Vo svojej najjednoduchšej forme sa vždy pozerá iba dopredu a neberie do úvahy minulosť. Jeho jediným cieľom je získať čo najväčší aktuálny úžitok z dostupných riešení, bez ohľadu na to, či by iné riešenia mohli priniesť väčší prospech v inom čase v budúcnosti.*“<sup>4</sup>

Štandardný pažravý algoritmus prvýkrát predstavil George Dantzig už v roku 1957, pričom tento algoritmus bol pôvodne navrhnutý pre frakčný problém batoha, kde sa predmety môžu deliť na menšie časti. Jeho podstata spočíva v zoradení predmetov podľa pomeru hodnoty k hmotnosti a následnom postupnom pridávaní týchto predmetov do batoha, kým to kapacita umožňuje, s cieľom maximalizovať celkovú hodnotu.

V rámci štúdie Mishru a Perkinsa (2023) autori porovnali viaceré heuristické metódy pre riešenie 0 – 1 problému batoha, pričom najväčší dôraz kládli na rôzne varianty pažravých algoritmov. Jednotlivé varianty sa odlišujú najmä tým, ako hodnotia a triedia položky pred ich výberom, čo priamo ovplyvňuje výpočtový čas aj kvalitu výsledného riešenia. Okrem klasického štandardného pažravého algoritmu, založeného na pomere zisku k váhe, bol analyzovaný aj variant skórovací pažravý algoritmus (Scored Greedy), ktorý tento pomer nahrádza nelineárnou skórovacou funkciou, čím sa snaží lepšie vyvážiť úžitok a penalizáciu za váhu objektov. Ďalším typom je ťažký pažravý algoritmus (Heavy Greedy), ktorý triedi položky podľa absolútnej hodnoty úžitku, teda preferuje „drahé“ objekty bez ohľadu na ich váhu. Zaujímavou variáciou je obmedzený pažravý algoritmus (Limited Greedy), ktorý v počiatkových fázach uprednostňuje ťažšie položky a neskôr prechádza na ľahšie, zatiaľ čo defenzívny pažravý algoritmus (Defensive Greedy) prioritne vyberá čo najľahšie položky s cieľom naplniť batoh čo najväčším počtom objektov.

Okrem klasických pažravých algoritmov poznáme aj tzv. opačné pažravé algoritmy, ktoré predstavujú inverzný prístup ku konštrukcii riešenia. Tento typ algoritmu bol po prvýkrát definovaný Ilevom a Linkerom v roku 2006, kde bol označený ako „Greedy descent“, známy tiež pod názvom lakomý algoritmus (Stingy algorithm). Tieto algoritmy oproti pažravému algoritmu začínajú s batohom plným všetkých objektov a postupne z neho

---

<sup>4</sup> ETAWI, Namera Ali – ABUROMMAN, Fatima Thaher. 0/1 Knapsack Problem: Greedy vs. Dynamic-Programming. In: International Journal of Advanced Engineering and Management Research. 2020, roč. 5, č. 2, s. 2. [cit.09. 04. 2025]. Dostupné z: [https://www.ijaemr.com/uploads/pdf/archivepdf/2020/IJAEMR\\_393.pdf](https://www.ijaemr.com/uploads/pdf/archivepdf/2020/IJAEMR_393.pdf)

odstraňujú tie najmenej výhodné, kým riešenie nevyhovuje kapacitnému obmedzeniu. Tento prístup je tak možné využiť v prípadoch, že chceme eliminovať tie najmenej vhodné objekty. (Mishra a Perkins, 2023)

Kým heuristické metódy využívajú pevne stanovené pravidlá na rýchle generovanie riešení, metaheuristiky predstavujú ich rozšírenie v podobe stochastických, iteratívnych algoritmov, ktoré napodobňujú prírodné procesy a pomocou inteligentnej kombinácie stratégií a učenia efektívne prehľadávajú priestor riešení s cieľom nájsť suboptimálne riešenia. (Basheer a Algamal, 2019) V štúdiu „*Riešenie 0/1 problému batoha použitím metaheuristických techník*“ boli analyzované štyri prístupy: simulované žihanie, iterované lokálne prehľadávanie, genetický algoritmus a optimalizácia rojom častíc. Všetky testované algoritmy preukázali schopnosť generovať kvalitné riešenia, čo autorov viedlo k záveru, že „*získané výsledky ukazujú, že použitie metaheuristik je nepochybne vhodným prístupom pri riešení problémov s obmedzeniami, akým je aj problém batoha*“<sup>5</sup> Okrem týchto metaheuristických metód boli analyzované aj ďalšie algoritmy inšpirované prírodou v štúdiu s názvom „*Nature-inspired optimization algorithms in knapsack problem: A review*“ od autorov Ghalya Tawfeeq Basheer a Zakariya Yahya Algamal (2019).

---

<sup>5</sup> ADEYEMO, Hayatullahi – AHMED, Moataz. *Solving 0/1 Knapsack Problem Using Meta-heuristic Techniques*. In: 2017 9th IEEE-GCC Conference and Exhibition (GCCCE). IEEE, 2017, s. 6. ISBN 978-1-5386-2756-3. [cit. 09. 04. 2025]. Dostupné z: <https://ieeexplore.ieee.org/document/8448239>

### 1.3 Pôdohospodárska platobná agentúra

Pôdohospodárska platobná agentúra (ďalej ako PPA) bola zriadená zákonom č. 473/2003 Z. z. o Pôdohospodárskej platobnej agentúre a svoju činnosť začala oficiálne vykonávať 1. januára 2004. Slovenská republika ju zriadila ako centrálnu inštitúciu zodpovednú za administráciu finančných prostriedkov určených na podporu poľnohospodárstva a rozvoja vidieka.

PPA je rozpočtovou organizáciou napojenou na štátny rozpočet, pričom patrí do pôsobnosti Ministerstva pôdohospodárstva a rozvoja vidieka Slovenskej republiky. Jej vznik reflektoval potrebu efektívneho riadenia procesov súvisiacich s poskytovaním dotácii financovaných z národných aj európskych zdrojov, s cieľom zabezpečiť správne a účelné prerozdelenie verejných financií. Zohráva dôležitú úlohu v celom procese implementácie finančnej podpory. To zahŕňa nielen poskytovanie finančných prostriedkov, ale aj dohľad nad ich využitím. V rámci svojej kontrolnej a monitorovacej funkcie zabezpečuje transparentnosť financovania, sleduje dodržiavanie legislatívnych a administratívnych požiadaviek stanovených Európskou úniou a Slovenskou republikou a dohliada na správne a účelné čerpanie finančných dotácií. (Zákon č. 473/2003 Z. z.)

V súčasnosti PPA poskytuje finančnú podporu prostredníctvom viacerých mechanizmov, medzi ktoré patria priame podpory, organizácia trhu, štátna pomoc a projektové podpory.

Priame podpory predstavujú jeden z nástrojov Spoločnej poľnohospodárskej politiky Európskej únie, ktorého cieľom je zabezpečiť stabilitu príjmov poľnohospodárov, podporiť poľnohospodársku produkciu, udržateľné hospodárenie s pôdou, rozvoj vidieckych oblastí a generačnú obmenu v tomto sektore. Slovenskí poľnohospodári môžu o túto formu podpory žiadať každoročne, pričom podmienky jej poskytovania sú stanovené európskou a slovenskou legislatívou. Sú vyplácané pravidelne na základe stanovených podmienok, pričom prostriedky pochádzajú zo štátneho rozpočtu Slovenskej republiky, ale sú následne refundované z rozpočtu Európskej únie. To znamená, že štát najprv vyplatí finančné prostriedky žiadateľom a potom mu Európska únia preplatí časť alebo celú sumu podľa platných pravidiel. (PPA, Priame podpory)

Štátna pomoc v oblasti pôdohospodárstva je určená podnikateľským subjektom pôsobiacim v poľnohospodárskej prvovýrobe, lesnom a rybnom hospodárstve a potravinárstve. Poskytuje sa na investície do výroby, spracovania a marketingu poľnohospodárskych produktov, kompenzáciu škôd spôsobených nepriaznivými

klimatickými podmienkami, výskum a vývoj v tomto sektore či ochranu kultúrneho a prírodného dedičstva. Podľa § 6 zákona č. 358/2015 Z. z. o štátnej pomoci môže mať táto podpora priamu alebo nepriamu formu. Priama pomoc sa realizuje prostredníctvom dotácií, príspevkov a grantov, pričom môže zahŕňať aj úhradu časti úveru alebo úrokov z úveru. Nepriame formy pomoci zahŕňajú daňové a poplatkové úľavy, zvýhodnené úvery, alebo aj predaj majetku štátu za zvýhodnených podmienok. Okrem uvedených foriem môže byť podpora poskytovaná aj ako minimálna pomoc, ktorá sa vyznačuje nižšími objemami finančnej podpory a zjednodušenými pravidlami poskytovania. (PPA, Štátna pomoc)

Organizácia trhu zahŕňa rôzne programy a opatrenia, prostredníctvom ktorých PPA pomáha stabilizovať trh s poľnohospodárskymi produktami a podporovať výrobcov. Medzi najvýznamnejšie nástroje môžeme zahrnúť licencie, ktoré oprávňujú držiteľa na dovoz alebo vývoz určitej komodity, takisto školský program, napríklad Školské ovocie a zelenina, ktorá podporuje distribúciu týchto produktov do škôl. Súčasťou organizácie trhu je aj verejná intervencia, ktorá zahŕňa nákup, skladovanie a následný predaj intervenčných zásob na vnútornom trhu Európskej únie a takisto poskytovanie nenávratného finančného príspevku (ďalej ako NFP) na propagáciu poľnohospodárskych výrobkov, a to nielen na európskom trhu, ale aj na trhu tretích krajín. (PPA, Organizácia trhu)

Projektové podpory, ktoré zastrešuje Pôdohospodárska platobná agentúra sú súčasťou Programu rozvoja vidieka Slovenskej republiky. Tieto podpory majú formu NFP, ktoré sú pridelované na základe vyhlasovaných výziev v súlade s pravidlami tohto programu. V súčasnosti projektové podpory, ktoré PPA naplno implementuje sa nazýva Program rozvoja vidieka Slovenskej republiky 2014 – 2022.

## 1.4 Program rozvoja vidieka Slovenskej republiky 2014 – 2022

Program rozvoja vidieka Slovenskej republiky 2014 – 2022 (ďalej ako PRV SR 2014 – 2022) je strategický program financovaný z Európskeho poľnohospodárskeho fondu pre rozvoj vidieka a štátneho rozpočtu Slovenskej republiky. Zameriava sa na zvýšenie konkurencieschopnosti v oblasti poľnohospodárstva, potravinárstva a lesného hospodárstva, zlepšenie efektivity a udržateľnosti hospodárenia s prírodnými zdrojmi, prispôsobovanie sa klimatickým zmenám a podporu rovnomerného rozvoja vidieckych komunít a ekonomiky a taktiež k zachovaniu a tvorbe pracovných miest. Program nadväzuje na obdobie 2007 – 2013 a implementuje opatrenia v súlade so Spoločnou poľnohospodárskou politikou Európskej únie. (MPSR, 2023)

PPA v rámci tohto programu vyhlasuje výzvy na NFP na podporu rôznych opatrení zameraných na rozvoj vidieka, poľnohospodárstva a lesného hospodárstva. Konkrétne opatrenia zahŕňajú napríklad financovanie projektov zameraných na odborné vzdelávanie či poradenské služby, ako aj investície do hmotného majetku, infraštruktúry či modernizácie poľnohospodárskych subjektov, pričom dôraz sa kladie aj na zvyšovanie energetickej efektívnosti. Súčasťou opatrení je tiež prevencia a eliminácia škôd spôsobených prírodnými katastrofami a ich následnú obnovu. PPA taktiež alokuje finančné prostriedky na rozvoj podnikov v tomto sektore, ako aj na obnovu, zlepšovanie a rozširovanie všetkých druhov infraštruktúr. Opatrenia sú zamerané aj na zlepšovanie základných služieb pre vidiecke oblasti, vrátane voľnočasových aktivít, kultúrnych a rekreačných zariadení. Taktiež investície do lesného hospodárstva na zlepšenie jeho životaschopnosti, ako aj podpora miestnych rozvojových aktivít, ktoré prispievajú k trvalo udržateľnému rozvoju vidieckych oblastí. (MPSR, 2023)

PPA k 31. 12. 2024 v rámci týchto oblastí PRV SR 2014 – 2022 evidovala celkovo až 20 705 podaných žiadostí o NFP. Celková požadovaná suma týchto žiadostí dosiahla hodnotu 3 537 153 346 €. PPA tak ako implementačný orgán zabezpečuje komplexný proces poskytovania nenávratných finančných príspevkov v rámci Programu rozvoja vidieka SR 2014 – 2022. (PPA, Sumárne prehľady)

### 1.4.1 Proces výberu projektov na realizáciu

Vyššie spomínaný proces pozostáva z niekoľkých krokov. V prvom kroku PPA zverejňuje výzvy na podávanie žiadostí o projektovú podporu, pričom presne definuje na akú oblasť a podoblasť sa vzťahuje, podmienky oprávnenosti žiadateľov, hodnotiace

kritériá, objem plánovanej alokácie finančných prostriedkov a termíny na predkladanie žiadostí o NFP.

Po vyhlásení výzvy môžu oprávnení žiadatelia predkladať žiadosti o NFP. Po doručení žiadostí PPA vykonáva administratívnu kontrolu, pri ktorej preveruje formálnu správnosť a úplnosť podaných žiadostí, ich súlad s podmienkami oprávnenosti a hodnotí, či je projekt finančne a technicky realizovateľný.

Nasleduje odborné hodnotenie žiadostí, pričom žiadosti sú náhodne pridelované hodnotiteľom, pre zachovanie objektivity hodnotenia. Žiadosti sú posudzované na základe bodovacích kritérií, ktoré sú zverejnené pre každú projektovú výzvu individuálne priamo v podmienkach výzvy. Každé kritérium má stanovený maximálny počet bodov. Kritéria sú navrhované tak, aby čo najlepšie odrážali ciele podporovaných oblastí a podoblastí uvedených vyššie. Bodové hodnotenie sa zameriava na konkrétne ukazovatele, ktoré zabezpečujú, že projekty prispievajú k správne a efektívne využitiu v rámci príslušného podopatrenia. To znamená, že projekty s najvyšším bodovým hodnotením najlepšie reflektujú efektívne využitie finančných prostriedkov v rámci danej podoblasti a zároveň majú najvyššiu pravdepodobnosť na získanie finančnej podpory.

Na základe výsledkov bodového hodnotenia žiadostí sa zostavuje poradovník úspešných žiadateľov, pričom projekty sú zoradené zostupne podľa dosiahnutého bodového hodnotenia. PPA následne schvaľuje projekty v poradí od najvyššie hodnoteného, pričom výška žiadanej finančnej podpory neovplyvňuje poradie výberu. Tento proces pokračuje až do vyčerpania alokovaných finančných prostriedkov určených pre danú výzvu. Takýto spôsob výberu projektov zodpovedá možnému spôsobu aplikácie pažravého algoritmu.

Po schválení projektov PPA uzatvára s úspešnými žiadateľmi zmluvy o poskytnutí NFP. Následne príjemcovia realizujú projekty v súlade so schváleným harmonogramom a rozpočtom. PPA vykonáva priebežnú administratívnu aj fyzickú kontrolu oprávnenosti výdavkov, pričom financovanie sa uskutočňuje na základe preukázateľne vynaložených nákladov, ktoré sú v súlade s podmienkami výzvy a sú nevyhnutné na realizáciu projektu.

V zmluve je individuálne špecifikovaný počet žiadostí o platbu, ktoré môže žiadateľ predložiť. Po ukončení projektu, čím sa rozumie podanie poslednej žiadosti o platbu, PPA sleduje jeho prínos a efektivitu, ktorá nezahŕňa iba hospodárnosť a účelnosť využitia verejných zdrojov, ale aj širší spoločenský dopad. Najmä to, ako projekt prispel k rozvoju vidieckych oblastí, zvýšeniu zamestnanosti, podpore udržateľného rozvoja, zlepšenia infraštruktúry a kvality života na vidieku alebo prispel k ochrane správne využitiu prírodných zdrojov. Výsledky sú následne zahrnuté do hodnotiacich správ. (MPSR, 2023)

## 2 Cieľ práce

Cieľom záverečnej práce je predstaviť alternatívny spôsob alokácie finančných prostriedkov z verejných zdrojov založený na využití optimalizačných metód. Efektívne a správne využitie verejných zdrojov je dôležité najmä z dôvodu zabezpečenia ich maximálneho prínosu pre spoločnosť a ekonomiku. Keďže verejné zdroje sú disponibilné v obmedzenom množstve a ich alokácia zahŕňa výber z viacerých alternatív, bude tento problém modelovaný ako problém batoha.

Predmetom nášho skúmania budú žiadosti o nenávratné finančné príspevky v rámci projektovej podpory Programu rozvoja vidieka Slovenskej republiky 2014 – 2022, ktorú ako implementačný orgán spravuje Pôdohospodárska platobná agentúra. Práca sa zameriava na porovnanie troch prístupov k riešeniu viacnásobného 0–1 problému batoha, konkrétne heuristickej metódy reprezentovanej pažravým algoritmom, algoritmom dynamického programovania a modelu založeného na binárnom celočíselnom lineárnom programovaní. V tejto súvislosti budeme analyzovať optimálnosť ich riešenia, výpočtovú náročnosť a praktickú aplikovateľnosť.

Na základe výsledkov porovnania vyberieme najvhodnejší model, ktorý implementujeme na reálny súbor projektových žiadostí z vybraných výziev a navrhujeme vlastný model spätného výberu žiadostí. Ten následne porovnáme so súčasným spôsobom prerozdeľovania verejných zdrojov s cieľom preukázať prínos optimalizačného prístupu pri rozhodovaní o verejných zdrojoch. Práca tak môže poskytnúť nový koncept výberu projektov na realizáciu a zároveň môže slúžiť ako metodický základ pre nastavenie alokačných mechanizmov v pripravovanom programovom období.

Hlavným cieľom záverečnej práce je teda návrh riešenia, ktoré umožní maximalizovať celkový spoločenský a ekonomický prínos z rozpočtu Programu rozvoja vidieka Slovenskej republiky 2014 - 2022 tak, aby verejné zdroje boli využité efektívne a mali čo najväčší dopad na strategické ciele tohto programu.

### 3 Metodika práce a metody skúmania

Pri rozhodovaní o rozdelení verejných zdrojov v rámci projektových podpôr Programu rozvoja vidieka 2014-2022 čelíme kombinatorickému optimalizačnému problému, v ktorom je potrebné nájsť optimálnu kombináciu projektov tak, aby nebol prekročený dostupný rozpočet a zároveň bol maximalizovaný ich spoločenský a ekonomický prínos.

Každú žiadosť o nenávratný finančný príspevok tak možno prirovnať k predmetu s určitou „hmotnosťou“, ktorú predstavuje požadovaná výška finančných prostriedkov a „hodnotu“, ktorú vyjadruje jeho bodové hodnotenie. Obmedzenie rozpočtu v našom prípade reprezentujú projektové výzvy, ktoré môžeme prirovnať k maximálnej kapacite batoha, pričom cieľom je vybrať takú kombináciu projektov, aby bol celkový bodový zisk čo najvyšší bez prekročenia dostupných finančných zdrojov.

Pri výbere vhodných metód sme vychádzali z kľúčového predpokladu, že jednotlivé projekty nie je možné financovať čiastočne, čo znamená, že rozhodovací proces je striktné binárny. Projekt je buď schválený na financovanie v plnej výške, alebo zamietnutý. Tento faktor jednoznačne definuje problém ako 0 – 1 problém batoha. Pri aplikácii tohto prístupu na reálne podmienky sa dostávame k viacnásobnému problému batoha, keďže financovanie projektov prebieha v rámci viacerých samostatných rozpočtov, ktoré definujú vyhlasované výzvy. Každá vyhlásená výzva bude tak v našich modeloch predstavovať samostatný batoh, pričom jeho kapacita je daná maximálnou výškou alokovaných finančných prostriedkov pre danú výzvu. Táto situácia si vyžaduje efektívnu alokáciu zdrojov naprieč viacerými rozpočtovými obmedzeniami. Všeobecne môžeme tento problém zapísať nasledovne:

$$\max \sum_{i=1}^n \sum_{j=1}^m v_i x_{ij}$$

za podmienok:

$$\sum_{i=1}^n c_i x_{ij} \leq W_j, \quad \forall j \in \{1, \dots, m\}$$

$$\sum_{j=1}^m x_{ij} \leq 1, \quad \forall i \in \{1, \dots, n\}$$

$$x_{ij} \in \{0,1\}, \quad \forall i,j$$

kde:

- $x_{ij} \in \{0,1\}$  predstavuje binárnu premennú, ktorá nadobúda hodnotu 1, ak projekt  $i$  je priradený rozpočtu  $j$ , inak 0;
- $v_i$  predstavuje bodové hodnotenie projektu  $i$ ;
- $c_j$  je požadovaná výška financovania projektu  $i$ ;
- $W_j$  je maximálna kapacita rozpočtu  $j$ ;
- $n$  je počet projektov;
- $m$  je počet rozpočtov, respektíve výziev.

Na riešenie problému batoha sme využili tri odlišné prístupy. Každý z vybraných algoritmov má svoje výhody a obmedzenia, ktoré ovplyvňujú ich efektivitu pre riešenie tohto problému.

### **Pažravý algoritmus**

Za prvý vybraný algoritmus sme zvolili pažravý algoritmus, keďže predstavuje najjednoduchší a výpočtovo nenáročný prístup k riešeniu problému batoha. Vďaka svojej jednoduchšej štruktúre a rýchlej implementácii je vhodný pri spracovaní veľkého množstva žiadostí o projektovú podporu.

V rámci našej analýzy sme zvolili prístup založený na maximalizácii bodového hodnotenia na jednotku dostupného rozpočtu. Ak sú projekty vyberané výlučne na základe absolútneho bodového hodnotenia, čo možno interpretovať ako znak ich individuálneho vysokého prínosu, môže nastať situácia, v ktorej niekoľko veľkých projektov vyčerpá dostupný rozpočet. Preto k tomuto problému pristupujeme globálne, čo odzrkadľuje skutočnosť, že našim cieľom je maximalizovať celkový prínos rozdelených finančných prostriedkov. To znamená, že pri výbere uprednostňujeme projekty, ktoré poskytujú najlepší pomer získaných bodov k požadovanému financovaniu na projekt. Takto si teda vypočítame relatívnu efektivnosť  $e_i$  pre každý projekt:

$$e_i = \frac{v_i}{c_i}$$

kde:

- $v_i$  je bodové hodnotenie projektu  $i$ ;
- $c_i$  je požadovaná výška financovania projektu  $i$ .

Na základe vypočítanej relatívnej efektívnosti  $e_i$  zoradíme projekty v rámci každej výzvy  $j \in \{1, \dots, m\}$  s rozpočtom  $W_j$  zostupne. To môžeme matematicky vyjadriť ako:

$$e_{(1)} \geq e_{(2)} \geq \dots \geq e_{(n_j)}$$

Následne iteratívne vyberáme projekty, ktoré sa zmestia do disponibilného zostatku rozpočtu:

$$\sum_{p \in S_j} c_p + c_i \leq W_j$$

kde:

- $S_j$  predstavuje množinu projektov, ktoré boli do výzvy  $j$  vybrané;
- $p$  predstavuje id projektu.

Zaradenie projektu potom zapíšeme pomocou binárnej premennej:

$$x_{ij} = \begin{cases} 1; & \text{ak je projekt } i \text{ priradený výzve } j \\ 0; & \text{ak nie} \end{cases}$$

Cieľom tohto prístupu je maximalizovať počet bodov vybraných projektov v rámci každej výzvy.

$$\max \sum_{i=1}^n \sum_{j=1}^m v_i x_{ij}$$

Hoci tento algoritmus nezaručuje nájdenie globálne optimálneho riešenia, vďaka svojej výpočtovej nenáročnosti a rýchlosti predstavuje efektívnu heuristickú metódu vhodnú najmä pri veľkom počte žiadostí. Na aplikáciu chamtivého algoritmu sme využili programovací jazyk Python.

### **Dynamické programovanie**

Ako druhý prístup k alokácii verejných zdrojov sme vybrali dynamické programovanie, ktoré sa často využíva pri riešení problémov s postupnou rozhodovacou štruktúrou. Tento algoritmus je vhodný na riešenie sekvenčných optimalizačných úloh, kde výstup pre daný krok závisí od predchádzajúcich výpočtov. Túto skutočnosť by sme tak mohli v ďalšej časti aplikovať na spätný výber žiadostí v prípade nevyčerpania celého objemu rozpočtu výzvy. Vzhľadom na to, že algoritmus dynamického programovania pre 0–1 problém batoha bol podrobne popísaný v publikácii „*MIT 6.006 Introduction to Algorithms*“ (MIT, 2011), využili sme túto metodológiu ako východiskový rámec a následne ju prispôbobi na riešenie nášho problému.

Ako bolo uvedené vyššie, riešime viacnásobný problém batoha, kde má každá výzva  $j \in \{1, \dots, m\}$  s rozpočtom  $W_j$ , pridelenú samostatnú množinu projektov, ktorá môže byť financovaná. Na rozdiel od heuristického prístupu však dynamické programovanie skúma všetky možné kombinácie výberu projektov, čím zabezpečuje globálne optimálne riešenie.

Pre každú výzvu preto riešime samostatný 0 – 1 problém batoha, pričom pre každú výzvu vytvárame nezávislú tabuľku dynamického programovania, ktorá umožňuje nájsť optimálny výber projektov v rámci disponibilného rozpočtu.

Dynamické programovanie v klasickej forme vyplňa dvojrozmernú tabuľku, ktorej veľkosť determinuje počet riadkov  $(n+1) * (W+1)$ , kde:

- $n$  predstavuje počet žiadostí, respektíve projektov;
- $W$  je celkový dostupný rozpočet.

Pripočítanie jedného indexu v oboch smeroch slúži na zahrnutie takzvaného východiskového stavu, keď ešte nezvažujeme žiaden projekt (riadok s indexom 0) a zároveň nemáme k dispozícii žiadny rozpočet (stĺpec s indexom 0). Tieto počiatočné hodnoty tvoria základ pre budovanie riešenia – tabuľka sa následne vyplňa smerom od prípadu „žiadny projekt, žiadny rozpočet“ až po prípad, keď uvažujeme všetky projekty pri maximálnom rozpočte. Každý riadok tak reprezentuje rozhodovanie o tom, či projekt zaradiť alebo nie, a každý stĺpec zodpovedá určitej úrovni disponibilného rozpočtu  $w$ .

Každá bunka tejto  $dp[i][w]$  uchováva maximálny možný počet bodov prvých  $i$  projektov pri rozpočte  $w$ , teda pri medzistave rozpočtu konkrétnej výzvy  $W$ . Hodnoty v tabuľke sa vyplňajú iteratívne, pre každý projekt sa rozhodujeme, či ho zaradíme do výberu, alebo nie. Toto rozhodnutie závisí od toho, či máme dostatok rozpočtu na jeho zaradenie a ak áno, či jeho zahrnutie prinesie lepší výsledok než jeho vynechanie. Teda porovnávame dve možnosti:

1. projekt nezaradíme a zachováme rovnaký počet bodov ako v predchádzajúcom riadku pri rovnakej hodnote rozpočtu  $w$ ,
2. projekt zaradíme a pripočítame jeho body k riešeniu z predošlého riadka zníženého o požadovanú výšku finančných prostriedkov pri rovnakej hodnote rozpočtu  $w$ .

Toto rozhodovanie tak môžeme matematicky vyjadriť:

$$dp[i][w] \begin{cases} dp[i-1][w], & \text{ak } c_{i-1} > w, \\ \max(dp[i-1][w], dp[i-1][w - c_{i-1}] + v_{i-1}), & \text{ak } c_{i-1} \leq w. \end{cases}$$

Po naplnení tabuľky dynamického programovania sa optimálne riešenie nachádza v bunke  $dp[n][W]$ . Táto hodnota reprezentuje maximálny počet bodov, ktorý je možné získať výberom z  $n$  projektov pri celkovom rozpočte  $W_j$ . Na identifikáciu konkrétnych projektov, ktorých kombinácia tvorí optimálne riešenie je potrebné vykonať spätný prechod tabuľkou, pričom začíname v bunke  $dp[n][W]$  a postupne prechádzame až k bunke „žiadny projekt, žiadny rozpočet“, pričom kontrolujeme, ktoré projekty boli vybrané. Ak pre niektorý projekt  $i$  platí:

$$dp[i][w] \neq dp[i-1][w]$$

znamená to, že výber projektu  $i-1$  je súčasťou optimálneho riešenia. V takom prípade sa zníži medzirozpočet  $w$  o požadovanú výšku financovania tohto projektu:

$$w = w - c_{i-1}$$

Tento proces pokračuje až kým nedosiahneme začiatok tabuľky, čím získame finálny výber projektov.

Na praktickú implementáciu tohto algoritmu sme rovnako ako pri predošlom algoritme využili programovací jazyk Python.

### **Binárne celočíselné lineárne programovanie**

Tretím prístupom k alokácii verejných zdrojov je nami zvolené binárne celočíselné lineárne programovanie, keďže ako sme spomínali vyššie, riešime 0 – 1 problém batoha. Takto získané riešenie nám rovnako, ako pri dynamickom programovaní garantuje globálne optimálne riešenie. Výhodou tohto prístupu však je, že pri použití dostupných optimalizačných nástrojov je jeho implementácia často jednoduchšia a flexibilnejšia, najmä ak problém obsahuje viacero obmedzení alebo väčší počet vstupných premenných. V porovnaní s pažravým algoritmom, ktorý je schopný nájsť len suboptimálne riešenie, poskytuje tento prístup globálne optimálne riešenie.

Na riešenie tohto typu problému sme využili knižnicu PuLP v programovacom jazyku Python. Na rozdiel od knižnice `scipy.optimize`, ktorá sa rovnako využíva pri matematickej optimalizácii, knižnica PuLP umožňuje priame nástroje na prácu s binárnymi a celočíselnými premennými. Knižnica PuLP umožňuje taktiež priame definovanie maximalizačnej funkcie, zatiaľ čo `scipy.optimize` len minimalizačnej.

Knižnica PuLP podporuje široké spektrum riešiteľov (solverov), ktoré je možné využiť pri riešení úloh lineárneho a celočíselného programovania. V rámci našej práce sme okrem predvoleného voľne dostupného riešiteľa CBC využili aj dva výkonné komerčné

riešiteľe CPLEX a Gurobi, ktoré sa vyznačujú vysokou výpočtovou efektivitou a spoľahlivosťou.

### CBC (Coin-or Branch and Cut)

CBC je voľne dostupný riešiteľ zmiešaných celočíselných lineárnych úloh, ktorý bol vyvinutý v rámci projektu COIN-OR (Computational Infrastructure for Operations Research). Na jeho vývoji sa podieľali dvaja vývojári z výskumného oddelenia spoločnosti IBM.

CBC je dostupný vo forme samostatnej spustiteľnej aplikácie, ale prostredníctvom rozhraní je možné jeho využitie aj v iných programovacích jazykoch. Tento riešiteľ využíva branch-and-cut algoritmus, ktorý predstavuje kombináciu dvoch základných metód používaných pri riešení zmiešaných celočíselných úloh. Prvou metódou je metóda vetvenia a orezávania, ktorá najskôr rieši problém ako klasickú lineárnu úlohu, teda bez celočíselných obmedzení. Ak riešenie obsahuje premenné s neceločíselnými hodnotami, napríklad rozhodovacia premenná pre projekt má hodnotu 0,6, vyberie sa takáto premenná a na základe nej sa úloha rozvetví na dve nové s upravenými obmedzeniami. V jednej sa premenná zaokrúhli smerom nadol, v druhej smerom nahor. Týmto spôsobom sa vytvára rozhodovací strom, ktorý sa prehľadáva až kým sa nenájde riešenie, v ktorom všetky premenné nadobúdajú celočíselné hodnoty. Teda či projekt bude financovaný (1) alebo nie (0). Tento proces zároveň orezáva tie vety rozhodovacieho stromu, ktoré nemôžu viesť k lepšiemu riešeniu aké už bolo nájdené. Súčasne sa počas riešenia do modelu pridávajú dodatočné lineárne obmedzenia, takzvané rezné roviny, ktoré pomáhajú eliminovať neriešiteľné časti v riešenom priestore, čo predstavuje druhú metódu. (Forrest a Lougee-Heimer, 2005)

### CPLEX

CPLEX je výkonný riešiteľ, ktorého názov vznikol ako kombinácia jazyka C a simplexovej metódy (C+PLEX). V súčasnosti je súčasťou produktu od spoločnosti IBM ILOG CPLEX Optimization Studio. V rámci našej práce sme zaobstarali študentskú licenciu, keďže tento riešiteľ nie je voľne dostupný ako napríklad CBC. Za základ tohto riešiteľa by sme mohli pokojne označiť aj CBC, ktoré CPLEX rozširuje o ďalšie optimalizačné nástroje, čím zvyšuje výpočtovú efektivitu a spoľahlivosť pri riešení rozsiahlych úloh. Okrem vyššie spomínaných metód CPLEX disponuje metódami, ktoré slúžia na predspracovanie modelu, napríklad odstráni zjavne nerealizovateľné projekty,

ktoré by z dôvodu vysokej požadovanej sumy nemohli byť súčasťou optimálneho riešenia. Takisto využíva už v počiatočnej fáze heuristiky, ktoré síce nemusia nutne nájsť optimálne riešenie, ale môžu dopomôcť modelu k priblíženiu sa k nemu, napríklad rýchlejšou identifikáciou vetiev rozhodovacieho stromu. Taktiež využíva metódy ako duálny simplex či interiérový bodový algoritmus. (IBM)

Výhodou CPLEX-u je jeho adaptívnosť, teda schopnosť analyzovať správanie modelu a na základe jeho vlastností v priebehu výpočtov upravovať rôzne metódy. Riešiteľ teda vie rozhodnúť či bude v danom momente efektívnejšie pokračovať v rezaní vetiev alebo aplikovať heuristiku.

### Gurobi

Gurobi je spolu s CPLEX-om v súčasnosti jeden z najvýkonnejších riešiteľov a rovnako ako CPLEX ide o komerčný softvér. Na účely našej práce sme preto taktiež využili študentskú licenciu, ktorá nám umožnila ho aplikovať v programovacom jazyku Python. Gurobi vyvíja spoločnosť Gurobi Optimization, LLC. Tento riešiteľ podporuje rovnako ako CPLEX široké spektrum úloh.

Rozdiel medzi CPLEX a Gurobi nespočíva primárne v použitých metódach, keďže obidva využívajú rovnaké základné algoritmické prístupy, ale najmä v miere adaptivity a výkonnostnom doladení týchto techník na rôzne scenáre, ktoré sa počas výpočtov vyskytnú. Podľa výsledkov porovnávacích štúdií je tento riešiteľ označovaný ako jeden z najrýchlejších a najefektívnejších nástrojov na trhu a práve preto sme sa ho rozhodli zahrnúť do našej analýzy. (Gurobi, 2019)

### **Model so spätným výberom projektov**

Jedným z hlavných cieľov tejto práce je poskytnúť ucelený pohľad na to, ako by mohla vyzeráť systematická optimalizácia alokácie verejných zdrojov v rámci celého programového obdobia Programu rozvoja vidieka Slovenskej republiky. V reálnych podmienkach sa výzvy na nenávratný finančný príspevok vyhlasujú postupne v rôznom čase a disponujú samostatnými rozpočtami, ktoré sa nie vždy vyčerpajú v plnej miere.

Z tohto dôvodu sme sa rozhodli rozšíriť základný model o mechanizmus spätného výberu projektov, ktorý simuluje možnosť dodatočného využitia nealokovaných finančných prostriedkov z predchádzajúcich výziev, ktorý reflektuje reálne podmienky Pôdohospodárskej platobnej agentúry. V tomto prístupe sme aplikovali najefektívnejší z testovaných modelov, pričom sme do optimalizácie opätovne zaradili projekty, ktoré neboli financované v pôvodnej výzve a to len z dôvodu obmedzeného rozpočtu, nie však z dôvodu nedostatočného bodového hodnotenia jednotlivých projektov.

Tento rozšírený model tak môžeme nazvať dvojfázovým 0 – 1 viacnásobným problémom batoha. V prvej fáze sa v rámci každej výzvy vykoná výber optimálnej kombinácie projektov na základe dostupného rozpočtu pre danú výzvu:

$$\max \sum_{i=1}^{n_j} v_{ij} x_{ij}$$

za podmienok:

$$\sum_{i=1}^{n_j} c_i x_{ij} \leq W_j$$

$$\sum_{j=1}^m x_{ij} \leq 1, \quad \forall i \in \{1, \dots, n\}$$

$$x_{ij} \in \{0,1\}, \quad \forall i,j$$

kde:

- $x_{ij} \in \{0,1\}$  predstavuje binárnu premennú, ktorá nadobúda hodnotu 1, ak projekt  $i$  je priradený rozpočtu výzvy  $j$ , inak 0;
- $v_{ij}$  predstavuje bodové hodnotenie projektu  $i$  vo výzve  $j$ ;
- $c_i$  je požadovaná výška financovania projektu  $i$ ;
- $W_j$  je maximálny rozpočet výzvy  $j$ ;
- $n_j$  je počet projektov vo výzve  $j$ ;
- $m$  je počet rozpočtov, respektíve výziev.

Projekty, ktoré neboli zahrnuté do optimálneho riešenia pre jednotlivé výzvy  $j$  z dôvodu nedostatočného rozpočtu, čiže ich rozhodovacia premenná  $x_{ij}^* = 0$ , sa ukladajú do množiny nevybraných projektov:

$$N_j = N_{j-1} \cup \{(i, j) \mid x_{ij}^* = 0\}$$

Zároveň sa pre každú výzvu zaznamenáva nevyužitá časť rozpočtu po optimalizácii, ktorá sa kumulatívne ukladá do zostatku bude následne využitá v druhej fáze:

$$Z^{(j)} = Z^{(j-1)} + \left( W_j - \sum_{i=1}^{n_j} c_i x_{ij}^* \right)$$

kde:

- $x_{ij}^*$  je optimálne riešenie pre projekt  $i$  vo výzve  $j$  v 1. fáze;
- $N_j$  predstavuje množinu nevybraných projektov po  $j$ -tej výzve;
- $Z^{(j)}$  predstavuje nevyužitý kumulatívny rozpočet po  $j$ -tej výzve.

Po každej výzve sa následne spustí druhá fáza, v ktorej sa vykoná optimalizácia pre projekty z množiny  $N_j$  s využitím aktuálne dostupného nealokovaného rozpočtu  $Z^{(j)}$ :

$$\max \sum_{(i,j) \in N_j} v_{ij} y_{ij}$$

za podmienok:

$$\sum_{(i,j) \in N_j} c_i y_{ij} \leq Z^{(j)}$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in N_j$$

kde:

- $y_{ij}$  predstavuje binárnu premennú, ktorá nadobúda hodnotu 1, ak projekt  $i$  je dodatočne financovaný v rámci nealokovaného rozpočtu, inak 0.

Dôležité je pritom poznamenať, že po každej iterácii, teda po druhej fáze každej výzvy tohto modelu, sa identifikuje množina dodatočne financovaných projektov  $A_j$ :

$$A_j = \{(i, j) \in N_j \mid y_{ij}^* = 1\}$$

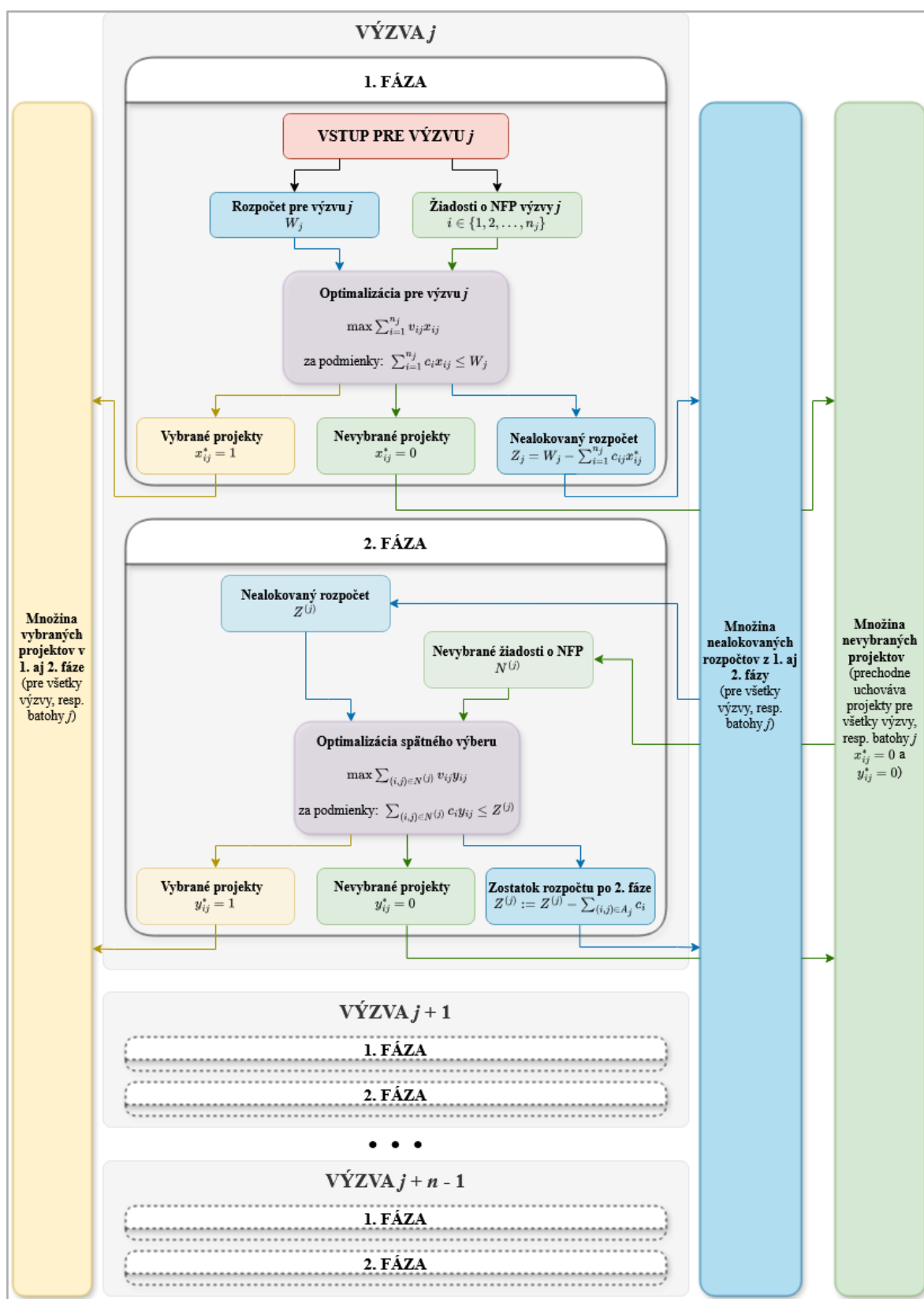
Na jej základe sa následne aktualizuje množina nevybraných projektov:

$$N_j = N_j \setminus A_j$$

a rozpočet:

$$Z^{(j)} = Z^{(j)} - \sum_{(i) \in A_j} c_i$$

Obrázok č. 1: Diagram modelu so spätným výberom projektov



Zdroj: vlastné spracovanie v softvéri draw.io

## 4 Výsledky práce a diskusia

### 4.1 Dataset

Dáta, ktoré budeme využívať sme získali priamo od PPA, konkrétne od Oddelenia monitoringu projektových podpôr na základe oficiálnej žiadosti a komunikácie s vedúcou oddelenia. V rámci tejto spolupráce nám boli poskytnuté nielen samotné dáta, ale aj doplňujúce vysvetlenia, ktoré nám umožnili hlbšie pochopiť vzájomné súvislosti medzi jednotlivými premennými a pozorovaniami. Dataset obsahuje komplexné informácie o všetkých projektových žiadostiach v rámci PRV 2014 – 2022 k 31. 12. 2024.

Hlavným účelom využitia tohto datasetu je aplikácia problému batoha na optimalizáciu výberu projektov, resp. žiadostí s cieľom efektívnej alokácie finančných prostriedkov. Tento dataset obsahuje nasledujúce kľúčové premenné:

- id projektu – ktoré reprezentuje unikátne identifikačné číslo projektu, ktoré bolo z dôvodu ochrany osobných údajov anonymizované interným šifrovacím kľúčom PPA;
- výzva – označenie výzvy, v rámci ktorej bola konkrétna žiadosť registrovaná;
- kód opatrenia a podopatrenia (KP) – špecifikácia účelu, pre ktorú je výzva určená;
- región financovania – kategorizácia alokovaných finančných prostriedkov podľa územnej príslušnosti:
  - *MMR (Menej rozvinuté regióny)* – reprezentujú všetky regióny Slovenska okrem Bratislavského kraja;
  - *OR (Ostatné regióny)* – pre Bratislavský kraj;
- požadovaná výška finančných prostriedkov – suma, ktorú žiadateľ uviedol v žiadosti;
- schválená výška finančných prostriedkov – suma, ktorú PPA po posúdení žiadosti skutočne schválila;
- status žiadosti – informácia o výsledku posúdenia žiadosti, pričom môže nadobúdať nasledujúce obmeny:
  - *schválená* – žiadosť bola vybraná na financovanie;
  - *neštandardne ukončená* – napríklad z dôvodu odstúpenia žiadateľa alebo nepredloženia poslednej žiadosti o platbu v stanovenom termíne, čo vedie k povinnosti vrátiť už poskytnuté finančné prostriedky;
  - *vyradená* – z dôvodov, akými sú napríklad nedostatok finančných prostriedkov, nekompletnosť žiadosti o NFP, nesplnenie podmienok výzvy, alebo späťvzatie žiadosti zo strany žiadateľa;

- počet získaných bodov – bodové hodnotenie pridelené na základe odborného posúdenia, čo bolo opísané vyššie.

Okrem uvedených premenných dataset pôvodne obsahoval aj informácie o počte platieb a ich dátumoch, ako aj o vyplatených sumách pri neštandardne ukončených projektoch, oprávnených výdavkoch a výške financovania z vlastných zdrojov. Vzhľadom na to, že tieto premenné nie sú relevantné pre optimalizáciu v rámci našej práce a zároveň predpokladáme, že všetky žiadosti prešli administratívnym a odborným hodnotením zo strany PPA, čím bola zabezpečená ich správnosť a súlad s podmienkami výzvy, nebudeme sa ďalej zaoberať ich validitou, ale budeme vychádzať najmä zo statusu výzvy, ktorá nám tieto informácie poskytne už sumárne. Rovnako pri optimalizácii vychádzame z požadovanej výšky finančných prostriedkov, keďže v prípade vyradených žiadostí nie je známe, akú sumu by PPA mohla konkrétnej žiadosti potenciálne prideliť. Z tohto dôvodu boli vyššie uvedené premenné z ďalšieho spracovania vylúčené.

Taktiež sme z datasetu odstránili pozorovania týkajúce sa Bratislavského regiónu (OR), keďže ich zahrnutie by viedlo k zdvojnásobeniu výsledkov optimalizácie. Cieľom tejto práce totižto nie je vytvoriť komplexné riešenie pre celé programové obdobie, ale predovšetkým predstaviť alternatívny prístup k už existujúcemu spôsobu výberu projektov, ktorý umožní efektívnu alokáciu dostupných finančných prostriedkov. Rovnako sme tak odstránili pozorovania týkajúce sa žiadostí, ktoré boli vyradené z iných dôvodov ako je nedostatok finančných prostriedkov a to z toho hľadiska, že pokiaľ bol iný dôvod vyradenia žiadosti, tá následne nie je relevantná pre našu optimalizáciu. Z analýzy boli vylúčené aj výzvy a im prislúchajúce žiadosti, ktoré k 31. 12. 2024 neboli uzavreté, pretože žiadostiam chýbali údaje o bodovom hodnotení alebo ich konečnom statuse. Bez týchto informácií nie je možné overiť splnenie podmienok výzvy, a ich zahrnutie by mohlo spôsobiť skreslenie výsledkov. Rovnako niektoré výzvy určovali okrem oblastí a podoblastí aj konkrétny účel využitia, avšak tieto údaje neboli súčasťou poskytnutého datasetu a preto sme sa rozhodli riešiť problém sumárne pre všetky účely v rámci výzvy.

Touto úpravou sme získali celkovo 9 834 žiadostí o NFP v celkovej výške 2 039 869 739,62 €, pričom tieto žiadosti spadali pod 44 rôznych výziev a 23 podoblastí. Okrem žiadostí, ktoré boli určené pre vyhlásené výzvy v rámci tohto programového obdobia sme evidovali aj 228 žiadostí, ktoré pochádzajú z predchádzajúceho programového obdobia, avšak boli financované z prostriedkov PRV 2014 – 2022.

**Tabuľka č. 1: Reprezentatívne projektové žiadosti o NFP v PRV 2014-2022 k 31. 12. 2024**

| ID Projektu | Dátum prijatia | Výzva       | KP   | MRR | Požadovaná výška finančného príspevku z verejných zdrojov | Status žiadosti | Schválený príspevok | Ak vyradený - dôvod          | Dosiahnutý počet bodov |
|-------------|----------------|-------------|------|-----|-----------------------------------------------------------|-----------------|---------------------|------------------------------|------------------------|
| PROJ-13134  | 29.7.2015      | 4/PRV/2015  | 19.1 | A   | 8 899,99 €                                                | s               | 8 899,99 €          |                              | 100,00                 |
| PROJ-07473  | 21.9.2015      | 7/PRV/2015  | 6.4  | A   | 456 482,40 €                                              | v               |                     | nedostatok fin. prostriedkov | 84,00                  |
| PROJ-00033  | 14.9.2015      | 6/PRV/2015  | 4.1  | A   | 59 880,00 €                                               | v               |                     | nedostatok fin. prostriedkov | 94,00                  |
| PROJ-14892  | 21.12.2021     | 43/PRV/2019 | 1.1  | A   | 50 560,00 €                                               | s               | 48 096,45 €         |                              | 73,00                  |
| PROJ-16211  | 14.9.2015      | 6/PRV/2015  | 4.1  | A   | 71 769,20 €                                               | s               | 71 769,20 €         |                              | 98,00                  |
| PROJ-13559  | 13.9.2019      | 40/PRV/2019 | 8.6  | A   | 200 000,00 €                                              | n               | 200 000,00 €        |                              | 81,00                  |

**Zdroj:** PPA, Oddelenie monitoringu projektových podpôr

Aby sme zabezpečili prehľadnosť a čitateľnosť reprezentácie tohto datasetu a priblížili jeho štruktúru, vybrali sme kľúčové ukazovatele, ktoré najlepšie vystihujú jeho podstatu. „Status žiadosti“ nám hovorí o schválených (s), vyradených (v) a neštandardne ukončených žiadostiach (n). V stĺpci „Schválený príspevok“ je hodnota uvedená iba pri žiadostiach, ktoré boli schválené a neštandardne ukončené, čo znamená, že projekt bol vybraný na financovanie. Tieto informácie neskôr využijeme pri porovnaní výsledkov.

Okrem datasetu projektových žiadostí sme zostavili aj doplnkový dataset, ktorý obsahuje informácie o jednotlivých projektových výzvach, a to najmä o výške alokovaných finančných prostriedkov na NFP, ktoré sú zverejňované na webovej stránke PPA.

**Tabuľka č. 2: Reprezentatívne výzvy v rámci PRV 2014 - 2022**

| Výzva      | Dátum vyhlásenia výzvy | Dátum uzavretia výzvy | Opatrenie | Výška finančných prostriedkov MRR |
|------------|------------------------|-----------------------|-----------|-----------------------------------|
| 1/PRV/2015 | 18.05.2015             | 23.10.2015            | 4.3       | 29 500 000,00 €                   |
| 2/PRV/2015 | 18.05.2015             | 16.10.2015            | 8.3       | 51 000 000,00 €                   |
| 3/PRV/2015 | 22.05.2015             | 18.09.2015            | 8.4       | 19 300 000,00 €                   |
| 4/PRV/2015 | 22.05.2015             | 28.07.2015            | 19.1      | 1 455 000,00 €                    |
| 5/PRV/2015 | 28.05.2015             | 15.10.2015            | 4.3       | 40 000 000,00 €                   |
| 6/PRV/2015 | 12.06.2015             | 30.10.2015            | 4.1       | 144 380 000,00 €                  |

**Zdroj:** PPA, <https://www.apa.sk/indikativny-harmonogram-vyziev>

Niektoré výzvy boli v priebehu programového obdobia aktualizované, pričom na zostavenie tohto datasetu sme vychádzali z ich najnovších aktualizácií. Zmeny sa týkali predovšetkým predĺžením lehoty na podávanie žiadostí a úpravy výšky alokovaných finančných prostriedkov. K týmto zmenám došlo v dôsledku nižšieho záujmu zo strany žiadateľov a pomalšej implementácie niektorých podopatrení a práve z tohto dôvodu boli uvoľnené finančné prostriedky presmerované do iných podopatrení. (PPA, Indikatívny harmonogram výziev)

## 4.2 Porovnanie vybraných metód

Porovnanie vybraných algoritmov sme realizovali na podklade údajov prvých šiestich výziev PRV 2014 – 2022. Výber práve prvých šiestich výziev bol motivovaný snahou o zachovanie časovej postupnosti a zároveň skutočnosťou, že výzva 6/PRV/2015 zahŕňala až 1 156 žiadostí o NFP, čo poskytuje dostatočne veľký rozsah pozorovaní pre objektívne porovnanie metód. Tak vysoký počet pozorovaní zároveň umožňuje posúdiť správanie algoritmov pri rozsiahlejších datasetoch a overiť ich presnosť a robustnosť v rôznych podmienkach.

Pre každú z vybraných metód boli sledované kľúčové kritéria ako celkový bodový zisk vybraných projektov v rámci jednotlivých výziev, počet vybraných projektov z každej výzvy, podiel alokovaného rozpočtu na celkovom rozpočte a takisto čas potrebný na vyriešenie optimalizačnej úlohy pre všetky sledované výzvy. Na účely porovnania výpočtovej efektivity sme využili knižnicu *time*, ktorá umožňuje zaznamenávať trvanie výpočtového procesu.

V rámci tejto podkapitoly budú predstavené len kľúčové časti implementovaných algoritmov, ktoré vystihujú ich logiku a princíp fungovania. Kompletne zdrojové kódy sú z dôvodu prehľadnosti a rozsahu práce uvedené v prílohách.

### 4.2.1 Pažravý algoritmus

Hlavná časť kódu predstavuje cyklus, ktorý postupne iteruje cez všetky výzvy. V prvom kroku vyfiltruje iba žiadosti pre aktuálnu výzvu, teda pre jeden batoh. Týmto krokom zabezpečíme, že algoritmus nebude vyberať žiadosti určené pre inú projektovú výzvu. Následne sa z tabuľky výziev načíta pridelený rozpočet pre aktuálnu výzvu, ktorý slúži ako kapacita batoha.

V ďalšom kroku kvantifikuje relatívnu efektivitu jednotlivých žiadostí. Projekty sa následne zoradia zostupne podľa vypočítanej relatívnej efektivity, aby sa zabezpečilo, že priorita výberu projektov bude podmienená vyšším prínosom na jednotku alokovaných finančných prostriedkov. Následne algoritmus iteruje cez zoradený zoznam od najefektívnejších projektov a postupne ich pridáva do zoznamu financovaných projektov, pričom sleduje aby celková suma už vybraných žiadostí neprekročila disponibilný rozpočet pre danú výzvu. Pri každom schválenom projekte sa alokované finančné prostriedky priebežne aktualizujú, čím sa zabezpečí, že financovanie neprekročí stanovený rozpočet.

**Obrázok č. 2:** Hlavná časť skriptu Pažravého algoritmu

```
summary_data = []
vyzva_results = {}
start_time = time.time()

for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    rozpocet = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]

    projekty["Pomer bodov k €"] = projekty["Dosiachnutý počet bodov"] / projekty["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]

    projekty = projekty.sort_values(by="Pomer bodov k €", ascending=False)

    vybrane = []
    suma = 0

    for _, p in projekty.iterrows():
        if suma + p["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"] <= rozpocet:
            vybrane.append(p["ID Projektu"])
            suma += p["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]
```

**Zdroj:** vlastné spracovanie v programovacom jazyku Python

V tabuľke nižšie môžeme vidieť výsledky optimalizácie aplikáciou Pažravého algoritmu. Ako je z výstupov zrejmé, v žiadnom z prípadov nedošlo k prekročeniu stanoveného rozpočtu, čo potvrdzuje správnu funkčnosť algoritmu. Zároveň možno pozorovať, že dostupné rozpočty boli vo väčšine prípadov takmer úplne využité, pričom objem nealokovaných prostriedkov zostáva minimálny.

**Tabuľka č. 3:** Výsledky Pažravého algoritmu

| Výzva      | Maximálny rozpočet (EUR) | Použitý rozpočet (EUR) | Nealokované prostriedky (EUR) | Získané body |
|------------|--------------------------|------------------------|-------------------------------|--------------|
| 1/PRV/2015 | 29 500 000,00            | 29 476 100,98          | 23 899,02                     | 6 948,00     |
| 2/PRV/2015 | 51 000 000,00            | 40 949 621,59          | 10 050 378,41                 | 5 747,28     |
| 3/PRV/2015 | 19 300 000,00            | 19 212 370,66          | 87 629,34                     | 7 328,19     |
| 4/PRV/2015 | 1 455 000,00             | 1 153 751,10           | 301 248,90                    | 8 640,00     |
| 5/PRV/2015 | 40 000 000,00            | 39 580 773,07          | 419 226,93                    | 5 705,00     |
| 6/PRV/2015 | 144 380 000,00           | 144 370 558,86         | 9 441,14                      | 92 892,09    |

**Zdroj:** vlastné spracovanie

Najväčšiu hodnotu nealokovaných zdrojov môžeme pozorovať pri výzve 2/PRV/2015, kde táto hodnota predstavuje až 10 050 378,41 €. Po výpise výsledkov a analýze vybraných a nevybraných žiadostí sme zistili, že boli vybraté všetky projekty až na projekt s id projektu PROJ-05282, ktorý síce získal 93 bodov, no požadovaná výška finančných prostriedkov predstavovala až 16 015 140,00€. Napriek vysokému bodovému hodnoteniu tento projekt nebol financovaný, pretože algoritmus uprednostnil efektivitu v zmysle pomeru požadovaných prostriedkov k získaným bodom. Dôsledkom tohto prístupu je, že boli vybraté projekty s bodovým hodnotením už od 53 bodov, avšak tieto projekty mali nižšie požiadavky na financovanie.

Pre ilustráciu si tak môžeme uviesť žiadosť s id PROJ-07973. Táto žiadosť dosiahla 53,56 bodov a v rámci nej bolo žiadaných 931 001,51€. Relatívna efektivita v tomto prípade predstavuje 0,00005754 bodov/€, zatiaľ čo v prípade nevybraného projektu je tento pomer 0,00000581 bodov/€. Práve skutočnosť, že projekt s nižším bodovým hodnotením dosiahol takmer desaťnásobne vyššiu relatívnu efektivitu, viedla k jeho uprednostneniu pred projektom s vyšším bodovým ziskom, no výrazne vyššou požiadavkou na financovanie.

#### 4.2.2 *Dynamické programovanie*

Dynamické programovanie v klasickej forme vyžaduje dvojrozmernú tabuľku, ktorej veľkosť je závislá od počtu projektov a dostupného rozpočtu. Pri práci s reálnymi finančnými údajmi je však nutné zabezpečiť, aby algoritmus operoval výlučne s celočíselnými hodnotami. Dôvodom je, že dynamické programovanie v programovacom jazyku Python nedokáže indexovať desatinné čísla a keďže finančné údaje v datasete obsahujú aj tie, je potrebné upraviť dáta tak, aby boli kompatibilné s týmto algoritmom.

Prvotná úprava predstavovala pre násobenie všetkých ukazovateľov hodnotou 100, čím sa zabezpečilo, že všetky číselné hodnoty predstavovali celé čísla. Tento krok však viedol k zväčšeniu tabuľky dynamického programovania (ďalej ako DP tabuľky), čo spôsobilo prekročenie kapacity operačnej pamäte a znemožnilo dokončenie výpočtu. Na základe postupného ladenia tejto metódy na naše podmienky sme tak dospeli k finálnej verzii, pri ktorej sme hodnoty škálovali faktorom 1000. Tento krok zmenšil počet stĺpcov v DP tabuľke, čím sa výrazne znížila pamäťová náročnosť algoritmu a tento prístup sa tak stal výpočtovo realizovateľný. Nevýhodou tohto riešenia je avšak mierna strata presnosti.

Podobne ako pri predchádzajúcom algoritme, aj pri dynamickom programovaní sme pre každú výzvu najskôr filtrovali relevantné projekty, ktoré vstupovali do výpočtu pre danú výzvu. Taktiež sme konvertovali dôležité premenné na zoznamy na zabezpečenie menšej výpočtovej náročnosti.

Po úprave všetkých vstupných premenných algoritmus vytvorí 2D tabuľku, ktorá uchováva maximálne dosiahnuteľné bodové hodnotenie pre rôzne kombinácie projektov pri rôznych rozpočtových obmedzeniach. Tieto obmedzenia reprezentujú kapacitu batohov, pričom každý rozpočet funguje ako samostatný batoh s určitou kapacitou.

**Obrázok č. 3:** Hlavná časť skriptu *Dynamického programovania*

```
for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    rozpocet = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]

    # škálovanie
    scale = 1000
    rozpocet = int(rozpocet / scale)

    ids = projekty["ID Projektu"].tolist()
    costs = (projekty["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"] /
             scale).astype(int).tolist()
    scores = projekty["Dosiahnutý počet bodov"].tolist()
    n = len(ids)

    # DP
    dp = [[0] * (rozpocet + 1) for _ in range(n + 1)]

    # DP tabuľka
    for i in range(1, n + 1):
        for w in range(rozpocet + 1):
            if costs[i - 1] <= w:
                dp[i][w] = max(dp[i - 1][w], dp[i - 1][w - costs[i - 1]] + scores[i - 1])
            else:
                dp[i][w] = dp[i - 1][w]

    # Spätne získanie vybraných projektov
    w = rozpocet
    vybrane = []
    for i in range(n, 0, -1):
        if w <= 0:
            break
        if dp[i][w] != dp[i - 1][w]:
            vybrane.append(ids[i - 1])
            w -= costs[i - 1]
```

**Zdroj:** *vlastné spracovanie v programovacom jazyku Python*

Ako môžeme vidieť táto tabuľka má rozmery  $(n+1) * (budget+1)$ , kde  $n$  predstavuje počet projektov v danej výzve a *budget* je maximálny dostupný rozpočet (po škálovaní). Napríklad, ak uvažujeme výzvu 1/PRV/2015, ktorá obsahuje 107 relevantných žiadostí o NFP a jej rozpočet je 29 500 000,00 €, tabuľka dynamického programovania tak bude mať 108 riadkov a 29 500 stĺpcov. Algoritmus postupne iteruje cez všetky projekty v rámci aktuálnej výzvy a vyhodnocuje, či ich možno zaradiť do financovania v rámci dostupného rozpočtu. Ak sa projekt zmestí do rozpočtu, môžeme si vybrať medzi dvomi alternatívami – buď ho zaradíme, alebo nie, pričom vždy volíme variant, ktorý vedie k vyššiemu bodovému ohodnoteniu. Ak sa projekt nezmesť do rozpočtu, pokračujeme bez jeho zahrnutia.

Po naplnení tabuľky sa vykoná spätná rekonštrukcia optimálneho riešenia, ktorá identifikuje konkrétne projekty vybrané na financovanie. Je avšak dôležité poznamenať, že projekty sa na konci vyberajú podľa ich pôvodných *id projektu* pred škálovaním. Týmto spôsobom zabránime tomu, aby boli výsledky skreslené zaokrúhľovaním a stratou presnosti, ku ktorej mohlo dôjsť práve pri úprave týchto finančných údajov.

**Tabuľka č. 4: Výsledky dynamického programovania**

| Výzva      | Maximálny rozpočet (EUR) | Použitý rozpočet (EUR) | Nealokované prostriedky (EUR) | Získané body |
|------------|--------------------------|------------------------|-------------------------------|--------------|
| 1/PRV/2015 | 29 500 000,00            | 29 511 719,98          | -11 719,98                    | 6 950,00     |
| 2/PRV/2015 | 51 000 000,00            | 40 949 621,59          | 10 050 378,41                 | 5 747,28     |
| 3/PRV/2015 | 19 300 000,00            | 19 337 621,14          | -37 621,14                    | 7 349,21     |
| 4/PRV/2015 | 1 455 000,00             | 1 153 751,10           | 301 248,90                    | 8 640,00     |
| 5/PRV/2015 | 40 000 000,00            | 39 970 461,34          | 29 538,66                     | 5 719,00     |
| 6/PRV/2015 | 144 380 000,00           | 144 938 311,57         | -558 311,57                   | 92 935,99    |

**Zdroj:** vlastné spracovanie

Z tabuľky výsledkov môžeme pozorovať, že väčšina výziev bola spracovaná s vysokou mierou efektivity, pričom využitie dostupných rozpočtov bolo takmer úplné. Výnimku však predstavujú tri výzvy: 1/PRV/2015, 3/PRV/2015 a 6/PRV/2015, pri ktorých došlo k miernemu prekročeniu rozpočtu. Najväčšia odchýlka sa prejavila pri výzve 6/PRV/2015 vo výške približne 558 000 €, pričom v rámci tejto výzvy (s rozpočtom 144 380 000 €) bolo vybraných až 1 025 projektov na financovanie. Táto odchýlka je dôsledkom zaokrúhľovania hodnôt vznikajúceho pri škálovaní a následnej spätnej rekonštrukcii reálnych hodnôt. V globále ide o zanedbateľné prekročenie (0,39%), ktoré by sa pri aplikácii na celé programové obdobie pravdepodobne vykompenzovalo. Avšak pre zabezpečenie presnosti výstupov sme algoritmus upravili o podmienku v spätnom výbere projektov, ktorá zaručí, že v prípade prekročenia dostupného rozpočtu pre danú výzvu dôjde k postupnému odstraňovaniu projektov v opačnom poradí ich výberu až dovtedy, kým nie je splnená podmienka obmedzenia rozpočtu. Tento dodatočný krok v algoritme zabezpečuje dodržanie stanovených rozpočtových limitov aj po spätnej rekonštrukcii skutočných hodnôt.

**Obrázok č. 4: Podmienka v spätnom výbere projektov dynamického programovania**

```
while nealokovany_rozpocet < 0 and len(vybrane) > 0:
    posledny_id = vybrane.pop()
    vybrane_df = vybrane_df[vybrane_df["ID Projektu"] != posledny_id]

    posledny_proj = projekty[projekty["ID Projektu"] == posledny_id][["ID Projektu", "VUC",
        "Výzva", "Dosiadnutý počet bodov", "Požadovaná výška finančného príspevku z
        verejných zdrojov (v EUR)"]]

    # prepočet po odstranení projektu
    nevybrane_df = pd.concat([nevybrane_df, posledny_proj], ignore_index=True)
    pouzity_rozpocet = vybrane_df["Požadovaná výška finančného príspevku z verejných zdrojov (v
        EUR)"].sum()
    nealokovany_rozpocet = rozpozet * scale - pouzity_rozpocet
```

**Zdroj:** vlastné spracovanie v programovacom jazyku Python

**Tabuľka č. 5:** *Výsledky dynamického programovania po pridaní dodatočnej podmienky*

| Výzva      | Maximálny rozpočet (EUR) | Použitý rozpočet (EUR) | Nealokované prostriedky (EUR) | Získané body |
|------------|--------------------------|------------------------|-------------------------------|--------------|
| 1/PRV/2015 | 29 500 000,00            | 29 225 884,81          | 274 115,19                    | 6 861,00     |
| 2/PRV/2015 | 51 000 000,00            | 40 949 621,59          | 10 050 378,41                 | 5 747,28     |
| 3/PRV/2015 | 19 300 000,00            | 19 041 566,42          | 258 433,58                    | 7 263,21     |
| 4/PRV/2015 | 1 455 000,00             | 1 153 751,10           | 301 248,90                    | 8 640,00     |
| 5/PRV/2015 | 40 000 000,00            | 39 970 461,34          | 29 538,66                     | 5 719,00     |
| 6/PRV/2015 | 144 380 000,00           | 144 373 766,02         | 6 233,98                      | 92 378,99    |

**Zdroj:** *vlastné spracovanie*

Po opätovnom spustení optimalizácie s implementovanou dodatočnou podmienkou spätného výberu projektov možno konštatovať, že algoritmus už plne rešpektuje rozpočtové limity definované pre jednotlivé výzvy. Takto upravený model eliminuje riziko prekročenia rozpočtových limitov a stáva sa vhodným nástrojom pre optimalizáciu aj tam, kde nie je prípustné prekračovať alebo dodatočne upravovať rozpočty jednotlivých výziev, čo je pri verejných zdrojoch dôležité pre zachovanie transparentnosti

#### *4.2.3 Binárne celočíselné lineárne programovanie*

Rovnako ako pri predošlých dvoch algoritmoch, najskôr pre každú výzvu iteratívne vyfiltrujeme príslušné projekty a načítame jej rozpočet. Tieto údaje slúžia ako vstupy do optimalizačného modelu, ktorý je formulovaný pomocou knižnice PuLP v programovacom jazyku Python. V ďalšom kroku vytvárame binárne rozhodovacie premenné, kde každá premenná zodpovedá jednému projektu podľa *id projektu* a nadobúda hodnotu 1, ak je projekt zaradený do optimálneho riešenia a 0, ak zaradený nie je. Cieľom optimalizácie je maximalizovať celkový počet bodov získaných z vybraných projektov, pričom jediným obmedzením je dostupný rozpočet výzvy, teda súčet požadovaných príspevkov z verejných zdrojov nesmie prekročiť stanovený rozpočet.

**Obrázok č. 5:** Hlavná časť skriptu binárneho celočíselného lineárneho programovania

```
for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    rozpocet = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]

    # inicializacia problemu
    problem = LpProblem(f"Knapsack_{vyzva}", LpMaximize)

    # definovanie binarnych premennych
    decision_vars = {p: LpVariable(f"proj_{p}", cat="Binary") for p in projekty["ID Projektu"]}

    # cielova funkcia
    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Dosiadnutý počet bodov"].values[0] for p in decision_vars)

    # obmedzenie rozpoctu
    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"].values[0] for p in decision_vars) <= rozpocet

    problem.solve()
```

**Zdroj:** vlastné spracovanie v programovacom jazyku Python

Po formulácii cieľovej funkcie a definovaní rozpočtového obmedzenia sa pomocou príkazu *problem.solve()* spustí optimalizačný proces, ktorý vyhladá takú kombináciu projektov, ktorá maximalizuje bodový zisk v rámci dostupného rozpočtu.

V rámci nášho porovnania sme sa najskôr rozhodli aplikovať iba predvolený riešiteľ CBC na prvých šesť výziev. Následne si v ďalšej časti bližšie prezentujeme výsledky pri všetkých 42 dostupných výzvach aj so spätným výberom projektov a alokácie nevyčerpaného rozpočtu. Tento krok sme vykonali z toho dôvodu, že pri takom malom objeme výziev sme nedospeli k markantnému rozdielu a nevedeli tak porovnať výpočtovú náročnosť medzi jednotlivými riešiteľmi.

**Tabuľka č. 6:** Výsledky binárneho celočíselného lineárneho programovania

| Výzva      | Maximálny rozpočet (EUR) | Použitý rozpočet (EUR) | Nealokované prostriedky (EUR) | Získané body |
|------------|--------------------------|------------------------|-------------------------------|--------------|
| 1/PRV/2015 | 29 500 000               | 29 476 100,98          | 23 899,02                     | 6 948,00     |
| 2/PRV/2015 | 51 000 000               | 40 949 621,59          | 10 050 378,41                 | 5 747,28     |
| 3/PRV/2015 | 19 300 000               | 19 194 618,53          | 105 381,47                    | 7 346,19     |
| 4/PRV/2015 | 1 455 000                | 1 153 751,10           | 301 248,90                    | 8 640,00     |
| 5/PRV/2015 | 40 000 000               | 39 936 744,36          | 63 255,64                     | 5 719,00     |
| 6/PRV/2015 | 144 380 000              | 144 365 294,02         | 14 705,98                     | 92 901,65    |

**Zdroj:** vlastné spracovanie

Na základe výsledkov môžeme konštatovať, že optimalizačný model pracoval spoľahlivo a v súlade s rozpočtovými obmedzeniami. Ani v jednom prípade nedošlo k prekročeniu stanoveného rozpočtu, čo potvrdzuje správnu formuláciu a aplikáciu rozpočtového obmedzenia v rámci modelu.

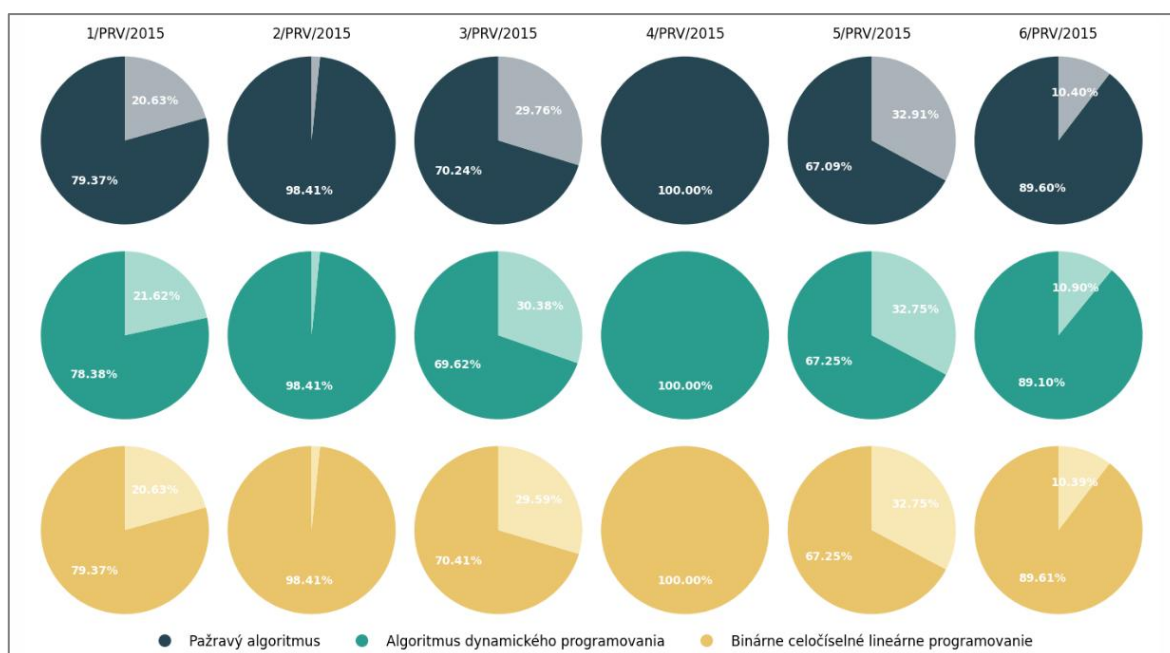
Zároveň možno konštatovať, že väčšina výziev bola spracovaná s vysokou mierou rozpočtovej efektivity, objem nealokovaných prostriedkov je minimálny a pohybuje sa len v desiatkach tisíc eur, čo je vzhľadom na výšku rozpočtov zanedbateľné. Výnimkou je výzva 2/PRV/2015, kde zostalo nevyužitých viac ako 10 miliónov eur. Tento jav však už bol vysvetlený pri predchádzajúcej analýze.

Celkovo môžeme model vyhodnotiť ako stabilný a schopný generovať optimálne riešenie, keďže dokázal vybrať také kombinácie projektov, ktoré priniesli vysoký bodový zisk a zároveň rešpektovali rozpočtové obmedzenia jednotlivých výziev.

#### 4.2.4 Syntéza porovnania vybraných algoritmov

V tejto podkapitole si zhrnieme výsledky vybraných algoritmov, pričom sa zameriame na optimálnosť riešenia a výpočtovú náročnosť.

**Obrázok č. 6:** Porovnanie podielu získaných bodov v rámci vybraných výziev



**Zdroj:** vlastné spracovanie

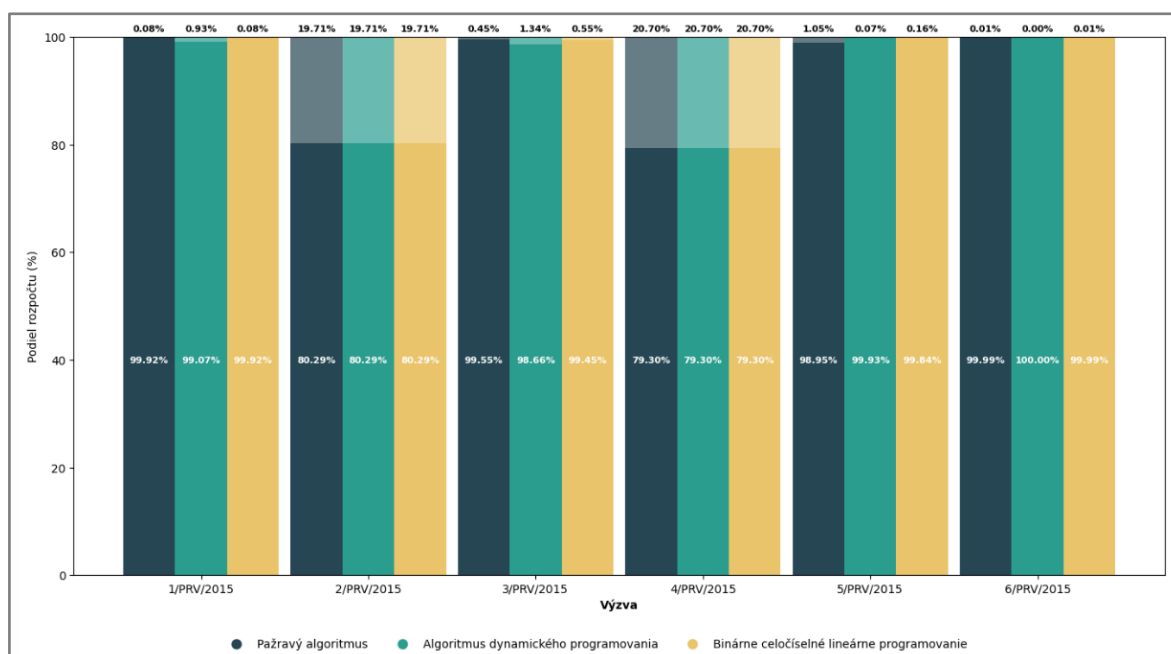
Na základe výsledkov optimalizácie možno konštatovať, že pri výzve 2/PRV/2015 dosiahli všetky použité algoritmy rovnaké riešenie. Ani jeden z nich nezaradil do optimálneho riešenia konkrétnu žiadosť o NFP vo výške 16 015 140,00 € pričom namiesto toho uprednostnili väčší počet projektov, so síce nižším bodovým hodnotením, ale za to s nižšou výškou žiadaného NFP. Podobná zhoda nastala aj pri výzve 4/PRV/2015, kde boli podporené všetky žiadosti. Dôvodom bola skutočnosť, že objem dostupných prostriedkov v tejto výzve preyšoval celkovú sumu požadovaných finančných prostriedkov.

Zhodné výsledky sme zaznamenali pri využití pažravého algoritmu a binárneho celočíselného lineárneho programovania (BILP) s využitím predvoleného riešiteľa CBC pri výzve 1/PRV/2015.

Pažravý algoritmus dosiahol pri výzve 3/PRV/2015 lepšie výsledky v porovnaní s algoritmom dynamického programovania, ktorý však vo väčšine prípadov vykazoval takmer identickú úspešnosť. Výnimkou v tomto prípade je výzva 5/PRV/2015, kde dosiahol lepšie výsledky. Rovnako tak tento algoritmus dosiahol najlepšie výsledky s pomedzi všetkých algoritmov práve pri výzve 1/PRV/2015.

Algoritmus BILP dosiahol najvyšší bodový zisk v dvoch výzvach, čo predstavuje najlepší celkový výkon spomedzi všetkých porovnávaných prístupov.

**Obrázok č. 7:** Porovnanie podielu alokovaného rozpočtu vybraných algoritmov



**Zdroj:** vlastné spracovanie

Pri analýze využitia dostupných rozpočtov v jednotlivých výzvach, teda pri „kapacitách jednotlivých batohov“, ktoré predstavujú jediné obmedzenie našich modelov identifikujeme, že najnižší podiel nealokovaných prostriedkov bol zaznamenaný pri výzvach 1/PRV/2015 a 6/PRV/2015. Vo všetkých aplikovaných algoritmoch bol v týchto prípadoch rozpočet využitý takmer na 100 %. Vzhľadom na túto skutočnosť nie je možné len na základe grafického porovnania jednoznačne konštatovať, že model BILP disponuje viditeľne lepšou optimalizačnou schopnosťou než heuristická metóda pažravého algoritmu. Ako už bolo uvedené pri interpretácii bodových výsledkov, pri výzve 5/PRV/2015 dosiahol algoritmus dynamického programovania najvyšší podiel získaných bodov, pričom zároveň najlepšie

využil dostupný rozpočet. Ďalej budeme prezentovať konkrétne číselné výstupy, ktoré nám umožnia jednoznačne posúdiť, ktorý z vybraných prístupov bude najvhodnejší pre ďalšie modelovanie.

**Tabuľka č. 7:** Porovnanie výsledkov a výpočtovej náročnosti vybraných algoritmov

| Výzva                          | Získané body |           |             | Nealokované prostriedky (EUR) |               |               |
|--------------------------------|--------------|-----------|-------------|-------------------------------|---------------|---------------|
|                                | PA           | DP        | BILP        | PA                            | DP            | BILP          |
| 1/PRV/2015                     | 6 948,00     | 6 861,00  | 6 948,00    | 23 899,02                     | 274 115,19    | 23 899,02     |
| 2/PRV/2015                     | 5 747,28     | 5 747,28  | 5 747,28    | 10 050 378,41                 | 10 050 378,41 | 10 050 378,41 |
| 3/PRV/2015                     | 7 328,19     | 7 263,21  | 7 346,19    | 87 629,34                     | 258 433,58    | 105 381,47    |
| 4/PRV/2015                     | 8 640,00     | 8 640,00  | 8 640,00    | 301 248,90                    | 301 248,90    | 301 248,90    |
| 5/PRV/2015                     | 5 705,00     | 5 719,00  | 5 719,00    | 419 226,93                    | 29 538,66     | 63 255,64     |
| 6/PRV/2015                     | 92 892,09    | 92 378,99 | 92 901,65   | 9 441,14                      | 6 233,98      | 14 705,98     |
| <b>Celkový čas spracovania</b> | <b>PA</b>    |           | <b>DP</b>   | <b>BILP</b>                   |               |               |
|                                | 0,2413 s.    |           | 125,2942 s. | 1,6320 s.                     |               |               |

**Zdroj:** vlastné spracovanie

Ako môžeme vidieť, BILP dosiahol najväčší počet bodov a to pri všetkých vybraných výzvach PRV 2014-2022 (zelenou). Z tohto hľadiska možno konštatovať, že algoritmus BILP dosahuje najvyšší bodový zisk, a teda prináša najlepší výsledok z pohľadu maximalizácie hodnotenia projektov, čo je dôležité pre alokáciu verejných zdrojov a ich celkový prínos pre strategické ciele tohto programu.

Naopak, najnižšie bodové hodnotenie dosiahol Pažravý algoritmus (PA) a algoritmus dynamického programovania (DP) (červenou). Z hľadiska objemu nealokovaných prostriedkov si však najlepšie počínal algoritmus dynamického programovania, ktorý pri výzve 1/PRV/2015 nealokoval až 274 115,19 €. V porovnaní s tým boli zostatky pri zvyšných dvoch algoritmoch výrazne nižšie, konkrétne 23 899,02 €. Tento prípad tak poukazuje na to, že vyšší počet bodov nemusí korešpondovať s efektívnym využitím rozpočtu.

Z hľadiska časovej, respektíve výpočtovej náročnosti dosiahol najkratší výpočtový čas PA, zatiaľ čo najdlhší čas spracovania zaznamenalo DP, čo bolo vzhľadom na jeho charakter a komplexitu aj očakávané. Na základe týchto výsledkov možno konštatovať, že BILP predstavuje z hľadiska výpočtovej náročnosti kompromisné riešenie.

V ďalšej časti záverečnej práce preto budeme pracovať práve s týmto modelom a rozšírime ho o mechanizmus spätného výberu projektov, ktorý umožní efektívne dočerpanie disponibilných finančných prostriedkov v prípade, že rozpočet výzvy nebol vyčerpaný úplne.

### 4.3 Model so spätným využitím rozpočtu

PPA v rámci implementácie PRV 2014 – 2022 uplatňuje flexibilný prístup k alokácii verejných prostriedkov. V prípade, že sa implementácia konkrétneho podopatrenia, viazaného na určitú výzvu, realizuje pomaly, respektíve nedôjde k celkovému vyčerpaniu plánovaných alokovaných finančných prostriedkov, PPA presúva tieto prostriedky do iného podopatrenia. Tento presun je zvyčajne smerovaný do výzvy, v ktorej je zvýšený dopyt po financovaní projektov.

V súlade s logikou tohto systému sme navrhli vlastný optimalizačný prístup, ktorého cieľom je zabezpečiť čo najefektívnejšie využitie verejných zdrojov prostredníctvom systematického prehodnocovania zvyšného rozpočtu a nevybraných projektov. Navrhnutý prístup je formalizovaný prostredníctvom dvojfázového viacnásobného 0 – 1 problému batoha. V prvej fáze dochádza k výberu projektov na financovanie v rámci jednotlivých výziev podľa vopred vyhlásených objemov alokovaných prostriedkov.

**Obrázok č. 8:** 1. fáza modelu so spätným výberom projektov

```
for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    rozpocet = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]

    # inicializacia problemu
    problem = LpProblem(f"Knapsack_{vyzva}", LpMaximize)

    # definovanie binarnych premennych
    decision_vars = {p: LpVariable(f"proj_{p}", cat="Binary") for p in projekty["ID Projektu"]}

    # cielova funkcia
    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Dosiachnutý počet bodov"].values[0] for p in decision_vars)

    # obmedzenie rozpoctu
    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"].values[0] for p in decision_vars) <= rozpocet

    problem.solve()
```

**Zdroj:** Vlastné spracovanie v programovacom jazyku Python

Projekty, ktoré neboli vybrané z dôvodu obmedzeného rozpočtu sú následné ukladané do takzvanej čakajúcej množiny. V druhej fáze tento model vyhodnocuje túto množinu projektov a v kombinácii so zostatkovými prostriedkami opätovne spúšťa optimalizáciu. Dôležité je pritom poznamenať, že po každej výzve sa množina projektov, ktoré neboli financované kumuluje, takisto ako aj objem nečerpaných rozpočtov.

**Obrázok č. 9:** 2. fáza modelu so spätným výberom projektov

```

if not nevybrane_nevybrane.empty and zostatok_zostatok > 0:
    problem = LpProblem(f"Knapsack_{vyzva}_faza2", LpMaximize)
    decision_vars = {p: LpVariable(f"proj_{p}", cat="Binary") for p in
        nevybrane_nevybrane["ID Projektu"]}
    problem += lpSum(decision_vars[p] * nevybrane_nevybrane.loc[nevybrane_nevybrane["ID
        Projektu"] == p, "Dosiahnutý počet bodov"].values[0] for p in decision_vars)
    problem += lpSum(decision_vars[p] * nevybrane_nevybrane.loc[nevybrane_nevybrane["ID
        Projektu"] == p, "Požadovaná výška finančného príspevku z verejných zdrojov (v
        EUR)"].values[0] for p in decision_vars) <= zostatok_zostatok
    problem.solve()

# Výber projektov po druhej fáze
faza2_vybrate = [p for p in decision_vars if decision_vars[p].value() is not None and
    decision_vars[p].value() == 1]
faza2_vybrate_df = nevybrane_nevybrane[nevybrane_nevybrane["ID
    Projektu"].isin(faza2_vybrate)]
# Aktualizácia zvyšných nevybraných projektov
nevybrane_nevybrane = nevybrane_nevybrane[~nevybrane_nevybrane["ID
    Projektu"].isin(faza2_vybrate)]

# Korekcia nealokovaných prostriedkov
pouzity_faza2_rozpocet = faza2_vybrate_df['Požadovaná výška finančného príspevku z
    verejných zdrojov (v EUR)'].sum()
nealokovany_faza2_rozpocet = max(0, zostatok_zostatok - pouzity_faza2_rozpocet)

# Aktualizácia tabuľky s 2 fazou
summary_data.append([f"{vyzva} - faza2", zostatok_zostatok, pouzity_faza2_rozpocet,
    nealokovany_faza2_rozpocet, faza2_vybrate_df['Dosiahnutý počet bodov'].sum(),
    len(faza2_vybrate_df), len(nevybrane_nevybrane), LpStatus[problem.status]])

# Aktualizácia zostatkového rozpočtu po druhej optimalizácii
zostatok_zostatok = nealokovany_faza2_rozpocet

```

**Zdroj:** vlastné spracovanie v programovacom jazyku Python

V rámci tejto implementácie sme využili tri rôzne riešiteľe - CBC, CPLEX a Gurobi a porovnali si výsledky z hľadiska výpočtovej náročnosti a spoľahlivosti riešenia. Vďaka modularite knižnice PuLP, ktorá poskytuje jednotné rozhranie pre viacero riešiteľov, bolo možné realizovať toto porovnanie veľmi jednoducho. V príkaze *problem.solve()* tak jedinou úpravou bola zmena z prednastaveného riešiteľa CBC na zvyšné spomínané dva, a tak spustiť model bez potreby zásahu do samotnej štruktúry modelu:

- `problem.solve();`
- `problem.solve(CPLEX_CMD(msg=True));`
- `problem.solve(pulp.GUROBI_CMD(msg=True)).`

**Tabuľka č. 8:** Výsledky porovnania riešiteľov pre model so spätným výberom projektov

| Riešiteľ | Požadovaná výška finančných príspevkov v € | Body       | Počet vybraných projektov | Čas v sekundách |
|----------|--------------------------------------------|------------|---------------------------|-----------------|
| CBC      | 1 376 194 725,6463685                      | 759 202.95 | 9203                      | 5 062,2329      |
| CPLEX    | 1 376 275 314,4363685                      | 759 428.31 | 9207                      | 104,8908        |
| Gurobi   | 1 376 310 496,9463687                      | 759 200.37 | 9203                      | 122,4753        |

**Zdroj:** vlastné spracovanie

Ako môžeme vidieť, rôzne optimalizačné nástroje deklarovali rozdielne hodnoty účelovej funkcie (dosiahnuté body) ako optimálne riešenie. Na základe analýzy spôsobu

vyhodnocovania týchto riešiteľov sme identifikovali, že pri riešení úloh zmiešaného celočíselného programovania používajú takzvaný MIP GAP. Je to parameter, ktorý definuje prípustnú toleranciu medzi hodnotou najlepšie nájdeného celočíselného riešenia a najlepším známym teoretickým ohraničením cieľovej funkcie. V predvolenom nastavení tak ide o malé nenulové číslo určujúce hranicu, pri ktorej je možné riešenie považovať za optimálne. (Petersen, 2013). Ak vyjadríme rozdiel medzi dvomi riešeniami:

$$\text{CPLEX} : 759\,428,31 - \text{Gurobi} : 759\,200,37 = 227,94;$$

$$\text{relatívna odchýlka predstavuje} = \frac{227,94}{759\,428,31} \approx 0,0003 = 0,03\%.$$

To znamená, že tieto riešitele ukončili výpočet už pri tejto hodnote, pretože sa nachádza v tolerancii parametra  $\varepsilon$ . Ako uvádza oficiálna dokumentácia Gurobi (2023): „V každom prípade, pre danú toleranciu MIP Gap  $\varepsilon$ , po dokončení výpočtu máte zaručené, že optimálne riešenie sa nachádza v  $\varepsilon$ -okolí riešenia nájdeného riešiteľom.“<sup>6</sup>

Z uvedeného dôvodu sa týmto rozdielom nebudeme detailnejšie zaoberať. Ak budeme analyzovať využité kapacitné obmedzenie v našom modeli, ktoré reprezentuje objem alokovaných finančných prostriedkov, dosiahol najlepšie výsledky práve CPLEX. Do optimálneho riešenia zaradil až 9 207 projektov, čo je zároveň najviac zo všetkých testovaných. Aj keď je Gurobi, ako sme uviedli v metodologickej časti, často prezentovaný ako jeden z najefektívnejších riešiteľov, v našom prípade sa jeho výpočtová efektivita nepotvrdila. Dokonca aj prednastavený riešiteľ CBC dosiahol v porovnaní s Gurobi lepšie výsledky z pohľadu kvality získaného riešenia.

Z hľadiska výpočtového času dosiahli komerčné riešitele CPLEX a Gurobi výrazne lepšie výsledky než prednastavený riešiteľ CBC. Kým CBC potreboval na spracovanie približne 4 996 sekúnd (83 minút), CPLEX a GUROBI vyriešili úlohu medzi 104 a 123 sekúnd, teda viac než 45-krát rýchlejšie.

Na základe uvedených výsledkov možno konštatovať, že najvhodnejším riešiteľom pre náš model je CPLEX. V ďalšej časti preto budeme analyzovať výsledky z tohto riešiteľa a v závere porovnáme výsledky získané optimalizačným modelom s reálnymi výsledkami v alokácii verejných zdrojov, ktoré boli dosiahnuté po výbere projektov v rámci PRV 2014 – 2022.

---

<sup>6</sup> GUROBI HELP CENTER. *What is the MIPGap?* [online]. 2023. [cit. 26. 03. 2025]. Dostupné z: <https://support.gurobi.com/hc/en-us/articles/8265539575953-What-is-the-MIPGap>

S cieľom ilustrovať správanie modelu pri aplikácii na reálne dáta uvádzame výstup pre prvých päť iterácií, ktoré zahŕňajú prvých päť výziev a zodpovedajúce druhé fázy. Úplný výpis všetkých výsledkov je dostupný v prílohe č. 5.

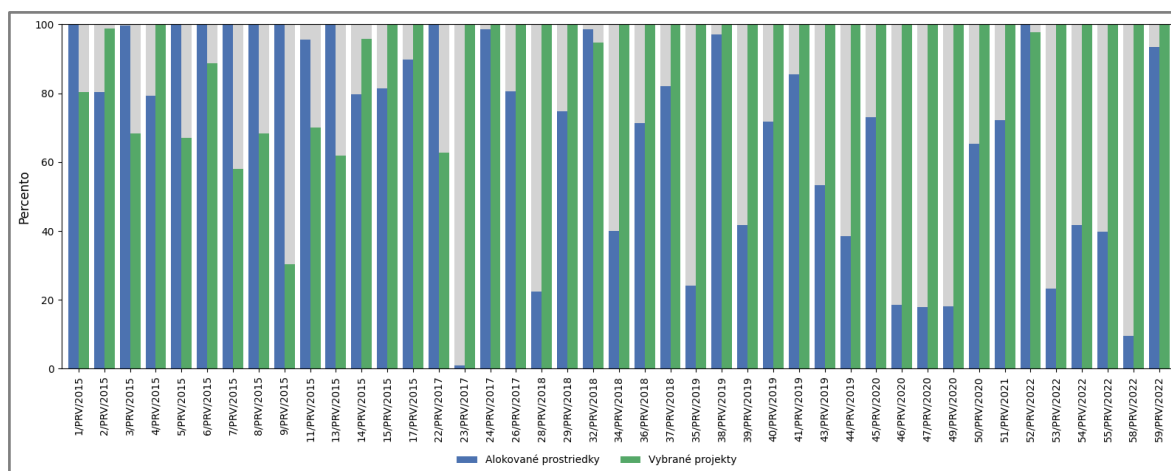
**Tabuľka č. 9:** Reprezentácia výsledkov na podklade prvých päť iterácii modelu

| Výzva      | Maximálny rozpočet (EUR) | Použitý rozpočet (EUR) | Nealokované prostriedky (EUR) | Získané body | Počet vybraných projektov | Celkový počet projektov | Stav optim. |
|------------|--------------------------|------------------------|-------------------------------|--------------|---------------------------|-------------------------|-------------|
| 1/PRV/2015 | 29 500 000,00            | 29 476 100,98          | 23 899,02                     | 6 948,00     | 86                        | 107                     | Optimal     |
| 1 - faza2  | 23 899,02                | 22 092,05              | 1 806,97                      | 300,00       | 3                         | 246                     | Optimal     |
| 2/PRV/2015 | 51 000 000,00            | 40 949 621,59          | 10 050 378,41                 | 5 747,28     | 77                        | 78                      | Optimal     |
| 2 - faza2  | 10 052 185,38            | 9 938 423,21           | 113 762,16                    | 14 100,00    | 141                       | 106                     | Optimal     |
| 3/PRV/2015 | 19 300 000,00            | 19 194 618,53          | 105 381,47                    | 7 346,19     | 93                        | 136                     | Optimal     |
| 3 - faza2  | 219 143,63               | 190 072,39             | 29 071,24                     | 100,00       | 1                         | 148                     | Optimal     |
| 4/PRV/2015 | 1 455 000,00             | 1 153 751,10           | 301 248,90                    | 8 640,00     | 112                       | 112                     | Optimal     |
| 4 - faza2  | 330 320,14               | 235 281,13             | 95 039,01                     | 100,00       | 1                         | 147                     | Optimal     |
| 5/PRV/2015 | 40 000 000,00            | 39 936 744,36          | 63 255,64                     | 5 719,00     | 63                        | 94                      | Optimal     |
| 5 - faza2  | 158 294,65               | 0,00                   | 158 294,65                    | 0,00         | 0                         | 178                     | Optimal     |

**Zdroj:** vlastné spracovanie

Zároveň je vhodné pripomenúť, že ako sme uviedli už pri analýze datasetu žiadostí o NFP, PPA v rámci PRV 2014 – 2022 financovala aj žiadosti pochádzajúce z predchádzajúceho programového obdobia. Táto skutočnosť sa prejavuje v druhom riadku, keď sa po prvej výzve v druhej fáze objavuje 228 „nových“ žiadostí, ktoré nadväzujú na predchádzajúce obdobie. Rovnako môžeme v riadkoch označených ako „*i*-faza 2“ (kde  $i \in \{1, \dots, 59\}$ ), v stĺpci „Počet vybraných projektov“ pozorovať počet projektov, ktoré boli dodatočne vybrané na základe kumulovaného nealokovaného rozpočtu z predchádzajúcich výziev. (Príloha č. 5)

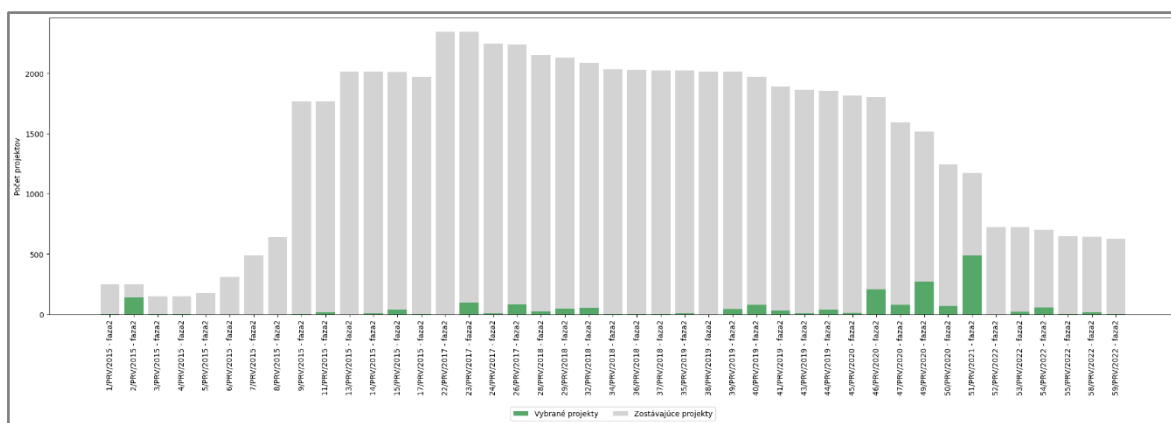
**Obrázok č. 10:** Podiel alokovaného rozpočtu a vybraných projektov



**Zdroj:** vlastné spracovanie

Na grafe sú zobrazené percentuálne podiely alokovaných finančných prostriedkov (modrou farbou) a podiely vybraných žiadostí o NFP, teda projektov (zelenou farbou) v rámci jednotlivých výziev. Tento grafický prehľad umožňuje lepšie porozumieť efektívnosti využitia dostupných rozpočtov. Z grafu je zároveň možné pozorovať, že pri viacerých výzvach v konečnej fáze programového obdobia nedochádza k úplnému čerpaniu disponibilných finančných prostriedkov, pričom podiel podporených projektov dosahuje 100%. Táto skutočnosť poukazuje na existenciu nevyužitého rozpočtu, ktorý predstavuje priestor na uplatnenie navrhovaného optimalizačného modelu. Práve z týchto výziev boli vo viacerých prípadoch spätne financované projekty z predchádzajúcich výziev, ktoré pôvodne vybrané neboli. Túto skutočnosť následne môžeme vidieť na obrázku č. 11:

**Obrázok č. 11:** Počet vybraných projektov v rámci 2. fázy algoritmu a počet nevybraných projektov, ktoré sa kumulujú



Zdroj: vlastné spracovanie

Na uvedenom grafe možno pozorovať, koľko pôvodne neuspokojených projektov bolo následne dodatočne financovaných. Kde zelenou sú dodatočne financované projekty (teda projekty financované v 2. fáze) a šedou „čakajúce projekty“. Najvýraznejší nárast počtu neuspokojených žiadostí možno identifikovať vo výzve 9/PRV/2015, v rámci ktorej ostalo bez podpory až 1 614 projektov. Podporených bolo len 490 projektov, pričom celková požadovaná výška finančných prostriedkov dosiahla 24 500 000,00 €. Od výzvy 23/PRV/2017 je možné pozorovať postupné znižovanie počtu nepodporených projektov, čo súvisí s ich následným financovaním zo zostatkových rozpočtových prostriedkov v rámci neskorších výziev. Tento vývoj zároveň naznačuje praktické uplatňovanie mechanizmu spätného výberu projektov, ktorého cieľom je efektívne využitie nevyčerpaných verejných zdrojov.

#### 4.3.1 Komparácia modelu s výberom projektov PPA

Na účely vyhodnotenia efektivity navrhovaného optimalizačného modelu založeného na binárnom celočíselnom lineárnom programovaní (BILP) sme porovnali jeho výstupy s reálnymi výsledkami alokácie prostriedkov realizovanými PPA v rámci PRV 2014 – 2022.

**Tabuľka č. 10:** Porovnanie výberu modelu so spätným výberom projektov s výberom PPA

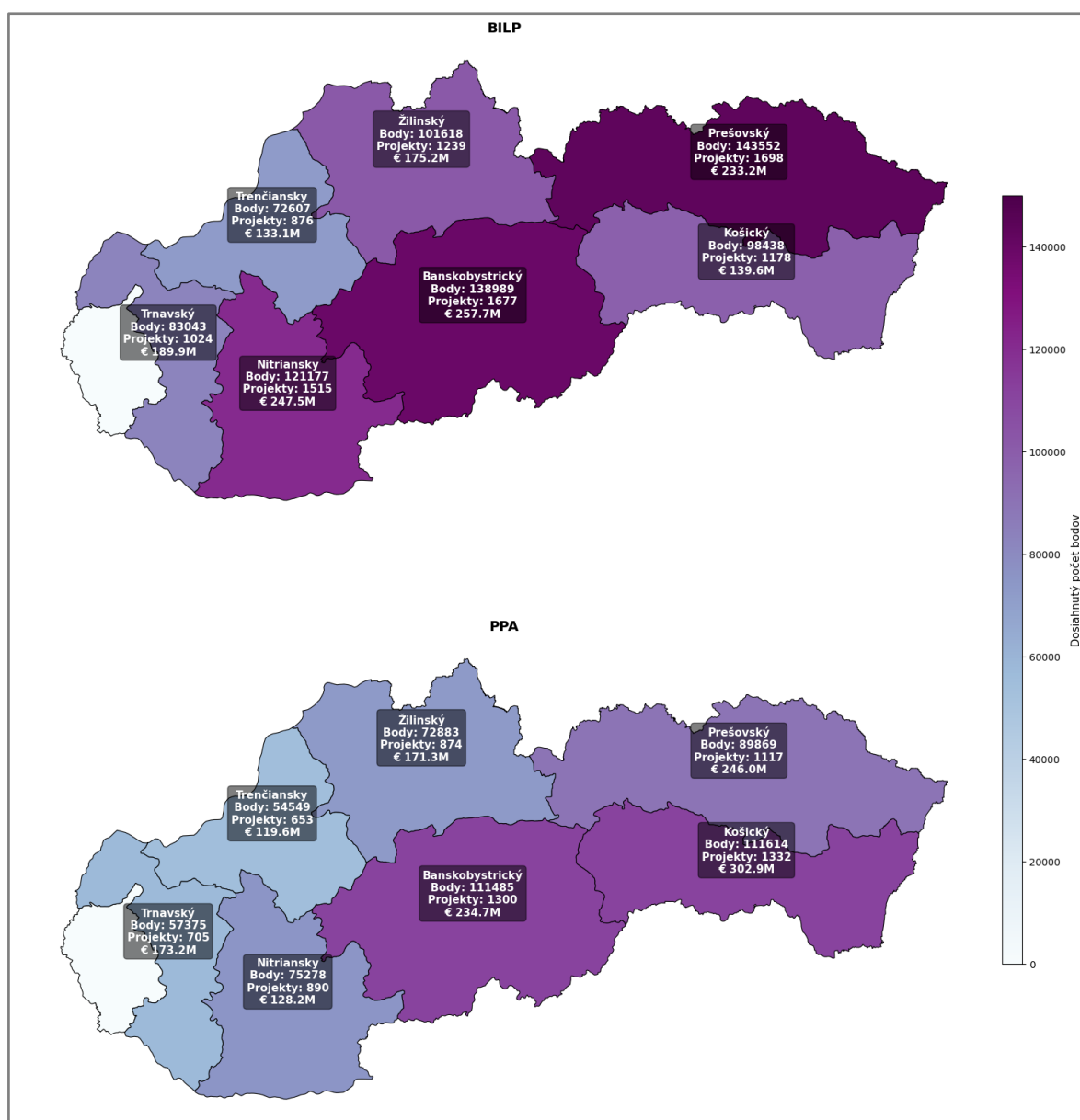
| Ukazovateľ                                | BILP model       | PPA              | Rozdiel            |
|-------------------------------------------|------------------|------------------|--------------------|
| Požadovaná výška finančných príspevkov(€) | 1 376 275 314,44 | 1 375 804 560,76 | <b>+470 753,68</b> |
| Celkový rozpočet (€)                      | 1 376 344 683,03 |                  | -                  |
| Získané body                              | 759 428,31       | 573 056,70       | <b>+186 113,61</b> |
| Celkové dostupné body                     | 808 000,49       |                  | -                  |
| Podiel získaných bodov (%)                | 93,99            | 70,95            | <b>+23,04</b>      |
| Počet vybraných projektov                 | 9 207            | 6 871            | <b>+2 333</b>      |
| Celkový počet žiadostí                    | 9 834            |                  | -                  |

**Zdroj:** vlastné spracovanie

Výsledky jednoznačne poukazujú na vyššiu účinnosť nášho modelu v niekoľkých kľúčových ukazovateľoch. Z hľadiska výšky požadovaných finančných príspevkov nami navrhovaný model alokoval o 470 753,68 € viac prostriedkov ako bolo prerozdelené v rámci výziev PPA, čo znamená vyššie využitie dostupného rozpočtu a lepšie zhodnotenie disponibilných verejných zdrojov. Výrazný rozdiel však nastal pri sume bodového hodnotenia projektov. Optimalizačný model BILP dosiahol celkový súčet bodov 93,99 % z maximálne možného počtu bodov. Naproti tomu, výber realizovaný PPA dosiahol len 70,95 % z celkového možného. Rozdiel 186 371,61 bodov poukazuje na schopnosť modelu efektívne maximalizovať celkový bodový zisk v rámci rovnakých rozpočtových obmedzení a teda model alokoval dostupné prostriedky spôsobom, ktorý viedol k vyššiemu súhrnnému hodnoteniu podporených projektov. Ak zoberieme do úvahy, že jedna žiadosť o NFP v rámci PRV 2014 – 2022 mohla získať hodnotenie v rozsahu od 1 do 100 bodov, potom si tento rozdiel môžeme ilustrovať ako ekvivalent približne 1862 projektov, ktoré by v hypotetickom prípade dosiahli maximálnych 100 bodov. Na tento fakt poukazuje aj rozdiel v počte vybraných projektov. Kým nami navrhovaný model do celkového optimálneho riešenia zahrnul 9 207 projektov, v prípade PPA ich bolo podporených len 6 871. To predstavuje rozdiel 2 336 projektov, ktoré mohli byť financované navyše pri zachovaní všetkých rozpočtových obmedzení a zároveň s vyšším celkovým bodovým ziskom.

Na záver, s cieľom detailnejšie ilustrovať spôsob alokácie verejných zdrojov sme výsledky rozčlenili podľa jednotlivých krajov, čím získavame presnejší obraz o ich regionálnom rozložení a počte dosiahnutých bodov v rámci každého kraja Slovenskej republiky. Táto vizualizácia umožňuje porovnať nielen počet podporených projektov, ale aj celkový bodový zisk a výšku prerozdelených finančných prostriedkov v jednotlivých regiónoch.

**Obrázok č. 12:** Porovnanie územného rozloženia dosiahnutých bodov za vybrané projekty a financovanie podľa krajov: BILP verzus reálna alokácia PPA



**Zdroj:** vlastné spracovanie

Z porovnania je zrejmé, že model BILP v každom kraji okrem Košického dosiahol vyšší bodový zisk ako reálne výsledky PPA, pričom najvýraznejší rozdiel možno pozorovať v Prešovskom, Banskobystrickom a Nitrianskom kraji. V Košickom kraji sa pritom ukázalo,

že aj napriek výrazne nižšej alokácii finančných prostriedkov dokázal model podporiť porovnateľný počet projektov. To naznačuje vyššiu efektivitu pri výbere projektov, ktoré prinášajú vyšší úžitok pri nižších nákladoch. Z tohto vyplýva, že navrhnutý optimalizačný model preukázal lepšiu schopnosť efektívne využívať dostupné finančné zdroje, maximalizovať kvalitu výberu projektov v rámci dostupných rozpočtov a zároveň zvýšiť mieru podpory bez prekročenia dostupných rozpočtov. Takýto prístup má potenciál významne zefektívniť rozhodovanie o verejných výdavkoch v rámci dotačných mechanizmov.

**Tabuľka č. 11:** Porovnanie výberu modelu BILP a reálna alokácia PPA podľa kraja

| Kraj            | Model | Získané body | Počet vybraných projektov | Požadovaná výška finančných prostriedkov ( € ) |
|-----------------|-------|--------------|---------------------------|------------------------------------------------|
| Bratislavský    | BILP  | 0,00         | 0                         | 0,00                                           |
|                 | PPA   | 0,00         | 0                         | 0,00                                           |
| Banskobystrický | BILP  | 138 989,00   | 1677                      | 257 737 201,53                                 |
|                 | PPA   | 111 485,72   | 1300                      | 234 650 228,25                                 |
| Košícký         | BILP  | 98 438,00    | 1178                      | 139 567 599,92                                 |
|                 | PPA   | 111 614,52   | 1332                      | 302 948 585,32                                 |
| Nitriansky      | BILP  | 121 177,00   | 1515                      | 247 523 617,95                                 |
|                 | PPA   | 75 278,11    | 890                       | 128 173 127,92                                 |
| Prešovský       | BILP  | 143 552,00   | 1698                      | 233 204 345,82                                 |
|                 | PPA   | 89 869,33    | 1117                      | 245 993 278,16                                 |
| Trenčiansky     | BILP  | 72 607,00    | 876                       | 133 134 813,76                                 |
|                 | PPA   | 54 549,98    | 653                       | 119 551 897,02                                 |
| Trnavský        | BILP  | 83 043,00    | 1024                      | 189 861 117,41                                 |
|                 | PPA   | 57 375,46    | 705                       | 173 201 834,34                                 |
| Žilinský        | BILP  | 101 618,00   | 1239                      | 175 246 618,04                                 |
|                 | PPA   | 72 883,58    | 874                       | 171 285 609,74                                 |

**Zdroj:** vlastné spracovanie

Výsledky porovnania zdôrazňujú, že nami navrhovaný model so spätným výberom projektov môže výrazne ovplyvniť alokáciu verejných zdrojov, a to spôsobom, ktorý vedie k vyššiemu celkovému bodovému zisku, väčšiemu počtu podporených projektov a zároveň prispieva k strategickejšej distribúcii verejných zdrojov z programu PRV 2014 - 2022.

## Záver

Problém batoha nachádza široké uplatnenie v rôznych oblastiach a odvetviach, pričom jeho varianty umožňujú flexibilné využitie podľa konkrétnych potrieb. V práci boli predstavené najdôležitejšie varianty tohto problému, spolu s vhodnými metódami ich riešenia, ktoré sa líšia mierou optimality dosiahnutého výsledku, ako aj výpočtovou náročnosťou. Pozornosť sme zamerali na problematiku alokácie verejných zdrojov, pričom sme na modelovanie výberu optimálnej kombinácie žiadostí o nenávratný finančný príspevok v rámci Programu rozvoja vidieka Slovenskej republiky 2014 – 2022 využili viacnásobný 0 – 1 problém batoha. Tento prístup reflektuje reálne podmienky rozhodovania, keďže umožňuje zohľadniť rôzne kapacitné obmedzenia súvisiace s dostupným rozpočtom jednotlivých výziev počas celého programového obdobia.

V práci boli porovnané tri algoritmicke prístupy, a to konkrétne pažravý algoritmus, algoritmus dynamického programovania a binárne celočíselné lineárne programovanie. Porovnanie bolo vykonané na podklade reálnych údajov žiadostí o nenávratný finančný príspevok, ktoré nám na účely práce poskytla Pôdohospodárska platobná agentúra ako orgán zodpovedný za kompletnú implementáciu programu.

Z výsledkov porovnania vybraných algoritmov sme dospeli k záveru, že výber konkrétneho optimalizačného prístupu má zásadný vplyv na efektivitu alokácie verejných zdrojov, ktoré sú v rámci jednotlivých výziev pridelené na podporu projektov.

V prípade vyhodnocovania projektov Pôdohospodárskou platobnou agentúrou, ktorý je založený na jednoduchej prioritizácii projektov podľa počtu získaných bodov a iteratívnym prístupom k ich výberu, sa dostávame iba k individuálnemu prínosu konkrétneho projektu. V kontexte verejných zdrojov je avšak žiaduce, aby sa na alokáciu finančných prostriedkov pozeralo globálne a teda aby bola zabezpečená celková spoločenská návratnosť v rámci daných rozpočtových možností.

Ako prvý bol v rámci našej práce analyzovaný pažravý algoritmus, ktorý triedi žiadosti na základe pomeru bodového hodnotenia k požadovanej výške finančného príspevku. Tento prístup dosiahol dostačujúce výsledky, najmä vďaka svojej jednoduchej implementácii. Vzhľadom na uvedené, je možno tento algoritmus považovať za vhodný nástroj pre výber projektov v prostredí s vysokým počtom projektových žiadostí, napriek tomu, že ide o heuristickú metódu.

Druhým aplikovaným algoritmom bolo dynamické programovanie, ktoré je schopné nájsť globálne optimum. Pri jeho implementácii na reálne dáta sa však prejavili viaceré

obmedzenia súvisiace s prostredím, v ktorom sa tento algoritmus vykonával a následne k nedodržaniu rozpočtových obmedzení. Tento nedostatok bol následne eliminovaný dodatočným mechanizmom, ktorý iteratívne odstraňoval projekty prekračujúce rozpočet z výslednej množiny financovaných žiadostí. Napriek tomu však vysoká výpočtová náročnosť a zložitosť jeho implementácie znižuje jeho praktickú využiteľnosť pri riešení rozsiahlych optimalizačných úloh s finančnými dátami.

Ako najvhodnejší prístup v rámci riešeného problému sa ukázal model založený na binárnom celočíselnom lineárnom programovaní. Na základe analýzy rôznych riešiteľov sme dospeli k záveru, že najvhodnejším riešiteľom pre náš model je CPLEX, a preto sme ho v ďalšej časti našej práci rozšírili o mechanizmus spätného výberu projektov, ktoré neboli v prvej fáze financované. Týmto spôsobom vznikol dvojfázový model 0 – 1 viacnásobného problému batoha, ktorý dokáže efektívne kumulovať zvyšné rozpočtové prostriedky a takisto neuspokojené projekty a následne ich dodatočne financovať. Hlavný prínos práce spočíva predovšetkým v jej praktickej využiteľnosti, pretože navrhnutý optimalizačný model predstavuje optimálny spôsob výberu projektov počas celého programového obdobia, pričom reflektuje reálne podmienky rozhodovacieho procesu.

Porovnaním tohto prístupu s reálnymi výsledkami Pôdohospodárskej platobnej agentúry v rámci Programu rozvoja vidieka Slovenskej republiky 2014 – 2022 sme preukázali, že navrhovaný model dokáže výrazne zvýšiť celkové bodové hodnotenie, počet financovaných projektov a takisto aj využiť rozpočet v čo najväčšej miere. Jeho aplikácia by tak mohla významne prispieť k transparentnejšiemu a strategickjšiemu alokovaniu verejných financií. Navrhovaný prístup zároveň predstavuje relevantný podklad pre metodologické zmeny v rámci pripravovaného programového obdobia. V prípade, že by sa štátne inštitúcie a samotná Pôdohospodárska agentúra, ako zodpovedný orgán za implementáciu Programu rozvoja vidieka Slovenskej republiky 2014-2022 rozhodli implementovať podobný optimalizačný prístup, mohlo by to viesť k zvýšeniu efektivity verejných výdavkov, zvýšeniu spoločenského prínosu a celkovému zlepšeniu kvality života obyvateľstva v dotknutých regiónoch.

## Zoznam použitej literatúry

### Knihy a monografie:

ARORA, Sanjeev – BARAK, Boaz. *Computational Complexity: A Modern Approach*. Cambridge : Cambridge University Press, 2009. 489 s. ISBN 978-0-521-42426-4.

CIBÍK, Jozef. *Financovanie verejnej správy*. Trnava : Univerzita sv. Cyrila a Metoda v Trnave, 2023. 142 s. ISBN 978-80-572-0361-6.

CORMEN, Thomas H. – LEISERSON, Charles E. – RIVEST, Ronald L. – STEIN, Clifford. *Introduction to Algorithms*. 3. vyd. Cambridge, Massachusetts : MIT Press, 2009. 1292 s. ISBN 978-0-262-03384-8.

KELLERER, Hans – PFERSCHY, Ulrich – PISINGER, David. *Knapsack Problems*. Berlin Heidelberg : Springer, 2004. 546 s. ISBN 978-3-540-24777-7.

MARTELLO, Silvano – TOTH, Paolo. *Knapsack problems: algorithms and computer implementations*. Chichester : Wiley, 1990. 296 s. ISBN 0-471-92420-2.

MITCHELL, John E. *Branch-and-Cut Algorithms for Combinatorial Optimization Problems*. In: PARDALOS, P. M. – RESENDE, M. G. C., eds. *Handbook of Applied Optimization*. Oxford : Oxford University Press, 2002, s. 65–77. ISBN 978-0-19-512594-8.

### Články:

ADEYEMO, Hayatullahi – AHMED, Moataz. *Solving 0/1 Knapsack Problem Using Meta-heuristic Techniques*. In: 2017 9th IEEE-GCC Conference and Exhibition. IEEE, 2017, 6 s. ISSN 2473-9391. [cit. 09.03.2025]. Dostupné na: <https://ieeexplore.ieee.org/document/8448239>

ACHTERBERG, Tobias. *Constraint Integer Programming*. Berlín, 2007. Dizertačná práca (PhD). Technische Universität Berlin. 412 s. Dostupné z: [https://opus4.kobv.de/opus4-zib/files/1112/Achterberg\\_Constraint\\_Integer\\_Programming.pdf](https://opus4.kobv.de/opus4-zib/files/1112/Achterberg_Constraint_Integer_Programming.pdf)

ASSI, Maram – HARATY, Ramzi A. *A Survey of the Knapsack Problem*. Beirut : Lebanese American University, 2018. 6 s. [cit. 09.03.2025]. Dostupné na: <https://cafetarjome.com/wp-content/uploads/1004/translation/05dde51ad89d21dd.pdf>

BASHEER, Ghalya Tawfeeq – ALGAMAL, Zakariya Yahya. *Nature-inspired optimization algorithms in knapsack problem: A review*. In: *Iraqi Journal of Statistical Science*, 2019, roč. 30, s. 55–72. ISSN 2071-0311. [cit. 04. 04. 2025]. Dostupné na: <https://scispace.com/pdf/nature-inspired-optimization-algorithms-in-knapsack-problem-51uth9b50r.pdf>

CAPRARA, Alberto – KELLERER, Hans – PFERSCHY, Ulrich. *The Multiple Subset Sum Problem*. In: *SIAM Journal on Optimization*, 2000, roč. 11, č. 2, s. 308–319. ISSN 1052-6234. [cit. 09.03.2025]. Dostupné na: <https://doi.org/10.1137/S1052623498348481>

DANTZIG, George B. *Discrete-variable extremum problems* [online]. Santa Monica: The RAND Corporation, 1957, s. 14 [cit. 30. 03. 2025]. Dostupné na: [https://www.rand.org/content/dam/rand/pubs/research\\_memoranda/2008/RM1832.pdf](https://www.rand.org/content/dam/rand/pubs/research_memoranda/2008/RM1832.pdf)

EBERLE, Franziska – MEGOW, Nicole – NÖLKE, Lukas – SIMON, Bertrand – WIESE, Andreas. *Fully Dynamic Algorithms for Knapsack Problems with Polylogarithmic Update Time*. 2021. 52 s. [cit. 02. 04. 2025]. Dostupné z: <https://arxiv.org/abs/2007.08415>

ETAWI, Namera Ali – ABUROMMAN, Fatima Thaher. 0/1 Knapsack Problem: Greedy vs. Dynamic-Programming. In *International Journal of Advanced Engineering and Management Research*. 2020, roč. 5, č. 2, s. 2. ISSN 2456-3676. [cit. 04. 04. 2025]. Dostupné na: [https://www.ijaemr.com/uploads/pdf/archivepdf/2020/IJAEMR\\_393.pdf](https://www.ijaemr.com/uploads/pdf/archivepdf/2020/IJAEMR_393.pdf)

GERASCH, Thomas E. – WANG, Pearl Y. *A survey of parallel algorithms for one-dimensional integer knapsack problems*. In *INFOR*, August 1994, vol. 32, no. 3, s. 163–176. ISSN 0315-5986. [cit. 09. 03. 2025]. Dostupné na: <https://doi.org/10.1080/03155986.1994.11732247>

HOCHBAUM, Dorit S. A nonlinear knapsack problem. In *Operations Research Letters*, 1995, roč. 17, č. 3, s. 103–110. ISSN 0167-6377. [cit. 30. 03. 2025]. Dostupné na: [https://doi.org/10.1016/0167-6377\(95\)00009-7](https://doi.org/10.1016/0167-6377(95)00009-7)

KOZEN, Dexter – ZAKS, Shmuel. Optimal bounds for the change-making problem. In *Theoretical Computer Science*, 1994, roč. 123, č. 2, s. 377–388. ISSN 0304-3975. [cit. 09.03.2025]. Dostupné na: <https://www.cs.cornell.edu/~kozen/Papers/change.pdf>

MISHRA, Sanatan – PERKINS, David. Using Heuristic Algorithms to Solve the 0-1 Knapsack Problem. In *Journal of Student Research*. 2023, roč. 12, č. 4. 14 s. ISSN 2167-1907. [cit. 04. 04. 2025]. Dostupné na: <https://doi.org/10.47611/jsrhs.v12i4.5660>

MUNAPO, Elias. Improvement of the Branch and Bound Algorithm for Solving the Knapsack Linear Integer Problem. In: *Eastern-European Journal of Enterprise Technologies*. 2020, roč. 3, č. 4, [cit. 30. 03. 2025]. 11 s. ISSN 1729-3774. Dostupné na: <https://doi.org/10.15587/1729-4061.2020.198849>

PISINGER, David. Where are the hard knapsack problems? In *Computers & Operations Research*. Copenhagen : University of Copenhagen, 2004. 14 s. [cit. 01. 04. 2025]. Dostupné na: <https://www.dcs.gla.ac.uk/~pat/cpM/jchoco/knapsack/papers/hardInstances.pdf>

### Internetové zdroje:

BIGLAR, Abbas. *Some applications of Knapsack problem* [online]. Tehran: Islamic Azad University, 2018 [cit. 12. 3. 2025]. Dostupné na: <https://www.researchgate.net/publication/322951612>

FIVEABLE. *Exact Algorithms*. [online]. Fiveable. [cit. 02. 04. 2025]. Dostupné na: <https://library.fiveable.me/combinatorial-optimization/unit-1/exact-algorithms/study-guide/ODz4iuGe8nVSpZV2>

FORREST, John – LOUGEE-HEIMER, Robin. *CBC User Guide* [online]. IBM Corporation, 2005. [cit. 22. 03. 2025]. Dostupné na: <https://www.coin-or.org/Cbc/cbcuserguide.html>

GUROBI HELP CENTER. *What is the MIPGap?* [online]. 2023. [cit. 26. 03. 2025]. Dostupné na: <https://support.gurobi.com/hc/en-us/articles/8265539575953-What-is-the-MIPGap>

GUROBI OPTIMIZATION, LLC. *Benchmarks* [online]. Houston: Gurobi Optimization, 2019. 23 s. [cit. 22. 03. 2025]. Dostupné na: <https://assets.gurobi.com/pdfs/benchmarks.pdf>

IBM. *IBM ILOG CPLEX Optimization Studio* [online]. [cit. 22. 03. 2025]. Dostupné na: <https://www.ibm.com/products/ilog-cplex-optimization-studio>

MPSR. *Program rozvoja vidieka SR 2014 – 2022*. Verzia 13.1, prijaté Európskou komisiou, 18. 12. 2023. [online]. 769 s. [cit. 10. 02. 2025]. Dostupné na: <https://www.apa.sk/prv-2014-2020-prv-vlastny-dokument>

MIT Computer Science and Artificial Intelligence Laboratory. *Recitation 21: Dynamic Programming – Knapsack Problem*. 6.006 Introduction to Algorithms, Fall 2011. Massachusetts Institute of Technology. Dostupné z: [https://courses.csail.mit.edu/6.006/fall11/rec/rec21\\_knapsack.pdf](https://courses.csail.mit.edu/6.006/fall11/rec/rec21_knapsack.pdf)

PETERSEN, Paul Rubin. *Different solvers, different solutions?* [online]. 2013. [cit. 26. 03. 2025]. Dostupné na: <https://orinanobworld.blogspot.com/2013/08/different-solvers-different-solutions.html>

PÔDOHOSPODÁRSKA PLATOBNÁ AGENTÚRA. *Indikatívny harmonogram výziev* [online]. [cit. 12. 02. 2025]. Dostupné na: <https://www.apa.sk/indikativny-harmonogram-vyziev>

PÔDOHOSPODÁRSKA PLATOBNÁ AGENTÚRA. *Organizácia trhu* [online]. [cit. 09. 02. 2025]. Dostupné na: <https://www.apa.sk/organizacia-trhu>

PÔDOHOSPODÁRSKA PLATOBNÁ AGENTÚRA. *Priame podpory* [online]. [cit. 09. 02. 2025]. Dostupné na: <https://www.apa.sk/priame-a-agro-environmentalne-podpory>

PÔDOHOSPODÁRSKA PLATOBNÁ AGENTÚRA. *Sumárne prehľady* [online]. [cit. 06. 02. 2025]. Dostupné na: <https://www.apa.sk/sumarne-prehlady-1>

PÔDOHOSPODÁRSKA PLATOBNÁ AGENTÚRA. *Štátna pomoc* [online]. [cit. 09. 02. 2025]. Dostupné na: <https://www.apa.sk/statna-pomoc>

### **Právne normy:**

Zákon č. 473/2003 Z. z. o Pôdohospodárskej platobnej agentúre a o podpore podnikania v pôdohospodárstve. [cit. 09. 02. 2025]. Dostupné na: <https://www.epi.sk/zz/2003-473>

Zákon č. 358/2015 Z. z. o úprave niektorých vzťahov v oblasti štátnej pomoci a minimálnej pomoci a o zmene a doplnení niektorých zákonov (zákon o štátnej pomoci). [cit. 12. 02. 2025]. Dostupné na: <https://www.slov-lex.sk/pravne-predpisy/SK/ZZ/2015/358/>

# Prílohy

## Príloha č. 1: Pažravý algoritmus

```
summary_data = []
vyzva_results = {}

start_time = time.time()

for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    rozpocet = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]

    projekty["Pomer bodov k €"] = projekty["Dosiahnutý počet bodov"] / projekty[
        "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]

    projekty = projekty.sort_values(by="Pomer bodov k €", ascending=False)

    vybrane = []
    suma = 0

    for _, p in projekty.iterrows():
        if suma + p["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"] <= rozpocet:
            vybrane.append(p["ID Projektu"])
            suma += p["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]

    vybrane_df = projekty[projekty["ID Projektu"].isin(vybrane)][["ID Projektu", "VUC",
        "Výzva", "Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)",
        "Ak vyradený - dôvod vyradenia"]]
    nevybrane_df = projekty[~projekty["ID Projektu"].isin(vybrane)][["ID Projektu", "VUC", "Výzva",
        "Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)", "Ak
        vyradený - dôvod vyradenia"]]

    pouzity_rozpocet = suma
    nealokovany_rozpocet = rozpocet - pouzity_rozpocet
    celkove_skore = vybrane_df["Dosiahnutý počet bodov"].sum()
    max_skore = projekty["Dosiahnutý počet bodov"].sum()

    pocet_vybranych_projektov = len(vybrane_df)
    celkovy_pocet_projektov = len(projekty)

    summary_data.append([
        vyzva, rozpocet, pouzity_rozpocet, nealokovany_rozpocet,
        celkove_skore, max_skore,
        pocet_vybranych_projektov, celkovy_pocet_projektov])

    vybrane_df["Stav"] = "Vybrané"
    nevybrane_df["Stav"] = "Nevybrané"
    vyzva_results[vyzva] = pd.concat([vybrane_df, nevybrane_df], ignore_index=True)

summary_greedy = pd.DataFrame(summary_data, columns=[
    "Výzva", "Maximálny rozpočet (EUR)", "Použitý rozpočet (EUR)",
    "Nealokované prostriedky (EUR)", "Získané body", "Maximálne možné body",
    "Počet vybraných projektov", "Celkový počet projektov"])

final_results_greedy = pd.concat(vyzva_results.values(),
    keys=vyzva_results.keys()).reset_index(level=0).rename(columns={"level_0": "Výzva"})

end_time = time.time()
elapsed_time = end_time - start_time

print(f"Celkový čas spracovania algoritmu: {elapsed_time:.4f} sekúnd.")

final_results_greedy
summary_greedy
```

**Zdroj:** Vlastné spracovanie v programovacom jazyku Python

## Príloha č. 2: Algoritmus dynamického programovania

```
summary_data = []
vyzva_results = {}

start_time = time.time()

for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    rozpocet = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]

    scale = 1000
    rozpocet = int(rozpocet / scale)

    ids = projekty["ID Projektu"].tolist()
    costs = (projekty["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"] /
scale).astype(int).tolist()
    scores = projekty["Dosiahnutý počet bodov"].tolist()
    n = len(ids)

    dp = [[0] * (rozpocet + 1) for _ in range(n + 1)]

    for i in range(1, n + 1):
        for w in range(rozpocet + 1):
            if costs[i - 1] <= w:
                dp[i][w] = max(dp[i - 1][w], dp[i - 1][w - costs[i - 1]] + scores[i - 1])
            else:
                dp[i][w] = dp[i - 1][w]

    w = rozpocet
    vybrane = []
    for i in range(n, 0, -1):
        if w <= 0:
            break
        if dp[i][w] != dp[i - 1][w]:
            vybrane.append(ids[i - 1])
            w -= costs[i - 1]

    vybrane_df = projekty[projekty["ID Projektu"].isin(vybrane)][["ID Projektu", "VUC", "Výzva",
"Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]]
    nevybrane_df = projekty[~projekty["ID Projektu"].isin(vybrane)][["ID Projektu", "VUC", "Výzva",
"Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]]

    pouzity_rozpocet = vybrane_df["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"].sum()
    nealokovany_rozpocet = rozpocet * scale - pouzity_rozpocet

    while nealokovany_rozpocet < 0 and len(vybrane) > 0:
        posledny_id = vybrane.pop()
        vybrane_df = vybrane_df[vybrane_df["ID Projektu"] != posledny_id]

        posledny_proj = projekty[projekty["ID Projektu"] == posledny_id][["ID Projektu", "VUC", "Výzva",
"Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]]
        nevybrane_df = pd.concat([nevybrane_df, posledny_proj], ignore_index=True)

        pouzity_rozpocet = vybrane_df["Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"].sum()
        nealokovany_rozpocet = rozpocet * scale - pouzity_rozpocet

    celkove_skore = vybrane_df["Dosiahnutý počet bodov"].sum()
    max_skore = projekty["Dosiahnutý počet bodov"].sum()

    pocet_vybranych_projektov = len(vybrane_df)
    celkovy_pocet_projektov = len(projekty)

    summary_data.append([
        vyzva, rozpocet * scale, pouzity_rozpocet, nealokovany_rozpocet,
        celkove_skore, max_skore,
        pocet_vybranych_projektov, celkovy_pocet_projektov])

    vybrane_df["Stav"] = "Vybrané"
    nevybrane_df["Stav"] = "Nevybrané"
    vyzva_results[vyzva] = pd.concat([vybrane_df, nevybrane_df], ignore_index=True)

summary_dp = pd.DataFrame(summary_data, columns=[
    "Výzva", "Maximálny rozpočet (EUR)", "Použitý rozpočet (EUR)",
    "Nealokované prostriedky (EUR)", "Získané body", "Maximálne možné body",
    "Počet vybraných projektov", "Celkový počet projektov"])

final_results_dp = pd.concat(vyzva_results.values(),
    keys=vyzva_results.keys()).reset_index(level=0).rename(columns={"level_0": "Výzva"})

end_time = time.time()
elapsed_time = end_time - start_time

print(f"Celkový čas spracovania algoritmu: {elapsed_time:.4f} sekúnd.")

final_results_dp
summary_dp
```

**Zdroj:** vlastné spracovanie v programovacom jazyku Python

### Príloha č. 3: Binárne celočíselné lineárne programovanie – predvolený riešiteľ CBC

```
summary_data = []
vyzva_results = {}
start_time = time.time()

for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    rozpocet = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]

    problem = LpProblem(f"Knapsack_{vyzva}", LpMaximize)

    decision_vars = {p: LpVariable(f"proj_{p}", cat="Binary") for p in projekty["ID Projektu"]}

    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Dosiahnutý počet bodov"].values[0] for p in decision_vars)

    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"].values[0] for p in decision_vars) <= rozpocet

    problem.solve()

    vybrane = [p for p in decision_vars if decision_vars[p].value() == 1]

    vybrane_df = projekty[projekty["ID Projektu"].isin(vybrane)][["ID Projektu", "VUC", "Výzva", "Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]]

    nevybrane_df = projekty[~projekty["ID Projektu"].isin(vybrane)][["ID Projektu", "VUC", "Výzva", "Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]]

    pouzity_rozpocet = vybrane_df['Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)'].sum()
    nealokovany_rozpocet = rozpocet - pouzity_rozpocet
    celkove_skore = vybrane_df['Dosiahnutý počet bodov'].sum()

    pocet_vybranych_projektov = len(vybrane_df)
    celkovy_pocet_projektov = len(projekty)
    max_skore = projekty['Dosiahnutý počet bodov'].sum()

    summary_data.append([vyzva, rozpocet, pouzity_rozpocet, nealokovany_rozpocet, celkove_skore, max_skore, pocet_vybranych_projektov, celkovy_pocet_projektov, LpStatus[problem.status]])

    vybrane_df["Stav"] = "Vybrané"
    nevybrane_df["Stav"] = "Nevybrané"
    vyzva_results[vyzva] = pd.concat([vybrane_df, nevybrane_df], ignore_index=True)

summary_ilp = pd.DataFrame(summary_data, columns=[
    "Výzva", "Maximálny rozpočet (EUR)", "Použitý rozpočet (EUR)",
    "Nealokované prostriedky (EUR)", "Získané body", "Maximálne možné body",
    "Počet vybraných projektov", "Celkový počet projektov", "Stav optimalizácie"])

final_results_ilp = pd.concat(vyzva_results.values(),
    keys=vyzva_results.keys()).reset_index(level=0).rename(columns={"level_0": "Výzva"})

end_time = time.time()
elapsed_time = end_time - start_time

print(f"Celkový čas spracovania algoritmu: {elapsed_time:.4f} sekúnd.")

final_results_ilp
summary_ilp
```

**Zdroj:** vlastné spracovanie v programovacom jazyku Python

#### Príloha č. 4: Model so spätným výberom projektov – BILP; riešiteľ CPLEX

```
summary_data = []
vyzva_results = {}
zostatok_zostatok = 0
nevybrane_nevybrane = pd.DataFrame(columns=["ID Projektu", "VUC", "Výzva",
      "Dosiahnutý počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"])
nevybrane_nevybrane = pd.concat([nevybrane_nevybrane, ziadosti[ziadosti["Výzva"] == "Pokračujúce projekty PRV
7-13 financ. z PRV 14-20"]], ignore_index=True)

start_time = time.time()

for vyzva in vyzvy_list:
    projekty = ziadosti[ziadosti["Výzva"] == vyzva].copy()
    budget = vyzvy.loc[vyzvy["Výzva"] == vyzva, "výška finančných prostriedkov MRR"].values[0]
    print(vyzva)
    problem = LpProblem(f"Knapsack_{vyzva}", LpMaximize)

    decision_vars = {p: LpVariable(f"proj_{p}", cat="Binary") for p in projekty["ID Projektu"]}

    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Dosiahnutý počet
bodov"].values[0] for p in decision_vars)

    problem += lpSum(decision_vars[p] * projekty.loc[projekty["ID Projektu"] == p, "Požadovaná výška finančného
príspevku z verejných zdrojov (v EUR)"].values[0] for p in decision_vars) <= budget

    problem.solve(CPLEX_CMD(msg=True))

    vybrata = [p for p in decision_vars if decision_vars[p].value() is not None and decision_vars[p].value()
==1]

    vybrata_df = projekty[projekty["ID Projektu"].isin(vybrata)][["ID Projektu", "VUC", "Výzva", "Dosiahnutý
počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]]
    nevybrata_df = projekty[~projekty["ID Projektu"].isin(vybrata)][["ID Projektu", "VUC", "Výzva", "Dosiahnutý
počet bodov", "Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"]]

    pouzity_budget = vybrata_df['Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)'].sum()
    nealokovany_budget = max(0, budget - pouzity_budget)
    celkove_skore = vybrata_df['Dosiahnutý počet bodov'].sum()

    summary_data.append([vyzva, budget, pouzity_budget, nealokovany_budget, celkove_skore, len(vybrata_df),
len(projekty), LpStatus[problem.status]])

    nevybrane_nevybrane = pd.concat([nevybrane_nevybrane, nevybrata_df], ignore_index=True)
    zostatok_zostatok += nealokovany_budget

# 2. faza
if not nevybrane_nevybrane.empty and zostatok_zostatok > 0:
    problem = LpProblem(f"Knapsack_{vyzva}_faza2", LpMaximize)
    decision_vars = {p: LpVariable(f"proj_{p}", cat="Binary") for p in nevybrane_nevybrane["ID Projektu"]}
    problem += lpSum(decision_vars[p] * nevybrane_nevybrane.loc[nevybrane_nevybrane["ID Projektu"] == p,
"Dosiahnutý počet bodov"].values[0] for p in decision_vars)
    problem += lpSum(decision_vars[p] * nevybrane_nevybrane.loc[nevybrane_nevybrane["ID Projektu"] == p,
"Požadovaná výška finančného príspevku z verejných zdrojov (v EUR)"].values[0] for p in decision_vars) <=
zostatok_zostatok
    problem.solve(CPLEX_CMD(msg=True))

    faza2_vybrata = [p for p in decision_vars if decision_vars[p].value() is not None and
decision_vars[p].value() == 1]
    faza2_vybrata_df = nevybrane_nevybrane[nevybrane_nevybrane["ID Projektu"].isin(faza2_vybrata)]

    nevybrane_nevybrane = nevybrane_nevybrane[~nevybrane_nevybrane["ID Projektu"].isin(faza2_vybrata)]

    pouzity_faza2_budget = faza2_vybrata_df['Požadovaná výška finančného príspevku z verejných zdrojov (v
EUR)'].sum()
    nealokovany_faza2_budget = max(0, zostatok_zostatok - pouzity_faza2_budget)

    summary_data.append([f"{vyzva} - faza2", zostatok_zostatok, pouzity_faza2_budget,
nealokovany_faza2_budget, faza2_vybrata_df['Dosiahnutý počet bodov'].sum(), len(faza2_vybrata_df),
len(nevybrane_nevybrane), LpStatus[problem.status]])

    zostatok_zostatok = nealokovany_faza2_budget

summary_ilp = pd.DataFrame(summary_data, columns=["Výzva", "Maximálny rozpočet (EUR)", "Použitý rozpočet
(EUR)", "Nealokované prostriedky (EUR)", "Získané body", "Počet vybraných projektov", "Celkový počet
projektov", "Stav optimalizácie"])

end_time = time.time()

summary_ilp
```

Zdroj: vlastné spracovanie v programovacom jazyku Python

**Príloha č. 5: Výstup z optimalizácie – model so spätným výberom projektov; BILP - CPLEX**

| Výzva       | Maximálny rozpočet (EUR) | Použitý rozpočet (EUR) | Nealokované prostriedky (EUR) | Získané body | Počet vybraných projektov | Celkový počet projektov | Stav optim. |
|-------------|--------------------------|------------------------|-------------------------------|--------------|---------------------------|-------------------------|-------------|
| 1/PRV/2015  | 29 500 000,00            | 29 476 100,98          | 23 899,02                     | 6 948,00     | 86                        | 107                     | Optimal     |
| 1 - faza2   | 23 899,02                | 22 092,05              | 1 806,97                      | 300,00       | 3                         | 246                     | Optimal     |
| 2/PRV/2015  | 51 000 000,00            | 40 949 621,59          | 10 050 378,41                 | 5 747,28     | 77                        | 78                      | Optimal     |
| 2 - faza2   | 10 052 185,38            | 9 938 423,21           | 113 762,16                    | 14 100,00    | 141                       | 106                     | Optimal     |
| 3/PRV/2015  | 19 300 000,00            | 19 194 618,53          | 105 381,47                    | 7 346,19     | 93                        | 136                     | Optimal     |
| 3 - faza2   | 219 143,63               | 190 072,39             | 29 071,24                     | 100,00       | 1                         | 148                     | Optimal     |
| 4/PRV/2015  | 1 455 000,00             | 1 153 751,10           | 301 248,90                    | 8 640,00     | 112                       | 112                     | Optimal     |
| 4 - faza2   | 330 320,14               | 235 281,13             | 95 039,01                     | 100,00       | 1                         | 147                     | Optimal     |
| 5/PRV/2015  | 40 000 000,00            | 39 936 744,36          | 63 255,64                     | 5 719,00     | 63                        | 94                      | Optimal     |
| 5 - faza2   | 158 294,65               | 0,00                   | 158 294,65                    | 0,00         | 0                         | 178                     | Optimal     |
| 6/PRV/2015  | 144 380 000,00           | 144 369 209,93         | 10 790,07                     | 92 899,01    | 1025                      | 1156                    | Optimal     |
| 6 - faza2   | 169 084,72               | 0,00                   | 169 084,72                    | 0,00         | 0                         | 309                     | Optimal     |
| 7/PRV/2015  | 72 000 000,00            | 71 998 794,77          | 1 205,23                      | 20 930,00    | 247                       | 426                     | Optimal     |
| 7 - faza2   | 170 289,95               | 0,00                   | 170 289,95                    | 0,00         | 0                         | 488                     | Optimal     |
| 8/PRV/2015  | 85 600 000,00            | 85 590 213,87          | 9 786,13                      | 26 901,00    | 327                       | 478                     | Optimal     |
| 8 - faza2   | 180 076,08               | 0,00                   | 180 076,08                    | 0,00         | 0                         | 639                     | Optimal     |
| 9/PRV/2015  | 24 500 000,00            | 24 500 000,00          | 0,00                          | 43 251,60    | 490                       | 1614                    | Optimal     |
| 9 - faza2   | 180 076,08               | 150 000,00             | 30 076,08                     | 252,00       | 3                         | 1760                    | Optimal     |
| 11/PRV/2015 | 20 000 000,00            | 19 098 757,76          | 901 242,24                    | 647,00       | 7                         | 10                      | Optimal     |
| 11 - faza2  | 931 318,32               | 900 000,00             | 31 318,32                     | 1 512,00     | 18                        | 1745                    | Optimal     |
| 13/PRV/2015 | 30 540 000,00            | 30 539 782,48          | 217,52                        | 38 737,00    | 437                       | 707                     | Optimal     |
| 13 - faza2  | 31 535,84                | 0,00                   | 31 535,84                     | 0,00         | 0                         | 2015                    | Optimal     |
| 14/PRV/2015 | 1 900 000,00             | 1 581 521,87           | 318 478,13                    | 3 722,00     | 47                        | 47                      | Optimal     |
| 14 - faza2  | 350 013,97               | 350 000,00             | 13,97                         | 588,00       | 7                         | 2008                    | Optimal     |
| 15/PRV/2015 | 11 000 000,00            | 8 955 915,69           | 2 044 084,31                  | 2,00         | 2                         | 2                       | Optimal     |
| 15 - faza2  | 2 044 098,28             | 2 044 022,83           | 75,45                         | 3 362,00     | 40                        | 1968                    | Optimal     |
| 17/PRV/2015 | 2 000 000,00             | 1 793 905,20           | 206 094,80                    | 7,00         | 7                         | 7                       | Optimal     |
| 17 - faza2  | 206 170,25               | 200 000,00             | 6 170,25                      | 336,00       | 4                         | 1964                    | Optimal     |
| 22/PRV/2017 | 58 150 000,00            | 58 148 967,48          | 1 032,52                      | 56 637,00    | 636                       | 1014                    | Optimal     |
| 22 - faza2  | 7 202,77                 | 0,00                   | 7 202,77                      | 0,00         | 0                         | 2342                    | Optimal     |
| 23/PRV/2017 | 4 830 000,00             | 45 000,00              | 4 785 000,00                  | 207,00       | 3                         | 3                       | Optimal     |
| 23 - faza2  | 4 792 202,77             | 4 791 233,23           | 969,54                        | 7 915,00     | 95                        | 2247                    | Optimal     |
| 24/PRV/2017 | 34 610 630,00            | 34 112 160,00          | 498 470,00                    | 130,00       | 130                       | 130                     | Optimal     |
| 24 - faza2  | 499 439,54               | 498 394,87             | 1 044,67                      | 754,00       | 9                         | 2238                    | Optimal     |
| 26/PRV/2017 | 21 631 000,00            | 17 411 825,66          | 4 219 174,34                  | 10 521,21    | 129                       | 129                     | Optimal     |
| 26 - faza2  | 4 220 219,01             | 4 200 000,00           | 20 219,01                     | 6 894,95     | 84                        | 2154                    | Optimal     |
| 28/PRV/2018 | 1 490 000,00             | 333 687,01             | 1 156 312,99                  | 7,00         | 7                         | 7                       | Optimal     |
| 28 - faza2  | 1 176 532,00             | 1 150 000,00           | 26 532,00                     | 1 886,00     | 23                        | 2131                    | Optimal     |
| 29/PRV/2018 | 9 538 000,00             | 7 122 101,15           | 2 415 898,85                  | 747,00       | 8                         | 8                       | Optimal     |
| 29 - faza2  | 2 442 430,85             | 2 442 388,59           | 42,26                         | 3 895,03     | 48                        | 2083                    | Optimal     |
| 32/PRV/2018 | 66 133 700,00            | 63 426 451,14          | 2 707 248,86                  | 3 095,00     | 35                        | 38                      | Optimal     |
| 32 - faza2  | 2 707 291,12             | 2 700 000,00           | 7 291,12                      | 4 374,00     | 54                        | 2032                    | Optimal     |
| 34/PRV/2018 | 488 540,00               | 195 871,43             | 292 668,57                    | 8,00         | 8                         | 8                       | Optimal     |

|             |                |                |               |            |      |      |         |
|-------------|----------------|----------------|---------------|------------|------|------|---------|
| 34 - faza2  | 299 959,69     | 299 523,81     | 435,88        | 414,00     | 5    | 2027 | Optimal |
| 36/PRV/2018 | 607 000,00     | 432 362,15     | 174 637,85    | 10,00      | 10   | 10   | Optimal |
| 36 - faza2  | 175 073,73     | 150 000,00     | 25 073,73     | 243,00     | 3    | 2024 | Optimal |
| 37/PRV/2018 | 178 000,00     | 146 143,94     | 31 856,06     | 5,00       | 5    | 5    | Optimal |
| 37 - faza2  | 56 929,79      | 50 000,00      | 6 929,79      | 81,00      | 1    | 2023 | Optimal |
| 35/PRV/2019 | 614 475,12     | 147 825,73     | 466 649,39    | 2,00       | 2    | 2    | Optimal |
| 35 - faza2  | 473 579,18     | 450 000,00     | 23 579,18     | 729,00     | 9    | 2014 | Optimal |
| 38/PRV/2019 | 195 700,00     | 189 837,77     | 5 862,23      | 2,00       | 2    | 2    | Optimal |
| 38 - faza2  | 29 441,41      | 0,00           | 29 441,41     | 0,00       | 0    | 2014 | Optimal |
| 39/PRV/2019 | 3 718 300,00   | 1 549 500,00   | 2 168 800,00  | 48,00      | 48   | 48   | Optimal |
| 39 - faza2  | 2 198 241,41   | 2 198 049,24   | 192,17        | 3 491,00   | 43   | 1971 | Optimal |
| 40/PRV/2019 | 14 331 000,00  | 10 298 851,35  | 4 032 148,65  | 6 336,04   | 77   | 77   | Optimal |
| 40 - faza2  | 4 032 340,82   | 4 000 000,00   | 32 340,82     | 6 428,31   | 80   | 1891 | Optimal |
| 41/PRV/2019 | 10 000 000,00  | 8 557 882,19   | 1 442 117,81  | 1 186,50   | 13   | 13   | Optimal |
| 41 - faza2  | 1 474 458,63   | 1 450 000,00   | 24 458,63     | 2 291,00   | 29   | 1862 | Optimal |
| 43/PRV/2019 | 1 003 559,05   | 534 193,67     | 469 365,38    | 621,00     | 8    | 8    | Optimal |
| 43 - faza2  | 493 824,01     | 492 678,53     | 1 145,48      | 717,00     | 9    | 1853 | Optimal |
| 44/PRV/2019 | 3 250 000,00   | 1 248 592,59   | 2 001 407,41  | 805,00     | 10   | 10   | Optimal |
| 44 - faza2  | 2 002 552,89   | 2 000 000,00   | 2 552,89      | 3 160,00   | 40   | 1813 | Optimal |
| 45/PRV/2020 | 2 698 279,80   | 1 968 337,10   | 729 942,70    | 2 579,46   | 31   | 31   | Optimal |
| 45 - faza2  | 732 495,59     | 700 000,00     | 32 495,59     | 1 106,00   | 14   | 1799 | Optimal |
| 46/PRV/2020 | 12 650 000,00  | 2 350 000,00   | 10 300 000,00 | 4 036,00   | 47   | 47   | Optimal |
| 46 - faza2  | 10 332 495,59  | 10 300 000,00  | 32 495,59     | 15 913,17  | 206  | 1593 | Optimal |
| 47/PRV/2020 | 4 800 000,00   | 855 000,00     | 3 945 000,00  | 4 734,00   | 57   | 57   | Optimal |
| 47 - faza2  | 3 977 495,59   | 3 950 000,00   | 27 495,59     | 5 938,94   | 79   | 1514 | Optimal |
| 49/PRV/2020 | 19 000 000,00  | 3 430 000,00   | 15 570 000,00 | 140,00     | 140  | 140  | Optimal |
| 49 - faza2  | 15 597 495,59  | 15 597 096,12  | 399,47        | 19 962,00  | 270  | 1244 | Optimal |
| 50/PRV/2020 | 20 000 000,00  | 13 060 201,69  | 6 939 798,31  | 32 360,00  | 401  | 401  | Optimal |
| 50 - faza2  | 6 940 197,78   | 6 939 769,42   | 428,36        | 6 111,00   | 71   | 1173 | Optimal |
| 51/PRV/2021 | 234 093 081,00 | 169 076 609,09 | 65 016 471,91 | 38 097,00  | 460  | 460  | Optimal |
| 51 - faza2  | 65 016 900,27  | 65 012 932,12  | 3 968,15      | 42 191,00  | 486  | 687  | Optimal |
| 52/PRV/2022 | 268 729 543,06 | 268 715 264,20 | 14 278,86     | 139 127,00 | 1583 | 1621 | Optimal |
| 52 - faza2  | 18 247,01      | 0,00           | 18 247,01     | 0,00       | 0    | 725  | Optimal |
| 53/PRV/2022 | 4 900 000,00   | 1 140 000,00   | 3 760 000,00  | 7 034,00   | 76   | 76   | Optimal |
| 53 - faza2  | 3 778 247,01   | 3 774 349,00   | 3 898,01      | 1 859,00   | 22   | 703  | Optimal |
| 54/PRV/2022 | 27 648 400,00  | 11 550 000,00  | 16 098 400,00 | 20 534,00  | 231  | 231  | Optimal |
| 54 - faza2  | 16 102 298,01  | 16 066 710,79  | 35 587,22     | 5 360,00   | 56   | 647  | Optimal |
| 55/PRV/2022 | 1 500 000,00   | 597 031,60     | 902 968,40    | 350,62     | 4    | 4    | Optimal |
| 55 - faza2  | 938 555,62     | 926 413,32     | 12 142,30     | 236,00     | 3    | 644  | Optimal |
| 58/PRV/2022 | 8 500 000,00   | 814 222,32     | 7 685 777,68  | 1 077,00   | 13   | 13   | Optimal |
| 58 - faza2  | 7 697 919,98   | 7 553 581,27   | 144 338,71    | 1 562,00   | 16   | 628  | Optimal |
| 59/PRV/2022 | 7 880 475,00   | 7 355 468,55   | 525 006,45    | 3 232,00   | 39   | 39   | Optimal |
| 59 - faza2  | 669 345,16     | 599 976,57     | 69 368,59     | 100,00     | 1    | 627  | Optimal |

**Zdroj:** *vlastné spracovanie*