

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**

**FAKULTA EKONOMICKÁ**

**Bakalářská práce**

**Aplikace matematického programování pro řešení  
problému podnikové praxe**

**The use of mathematical programming for solving  
problems in business practice**

**Jan Bílek**

**Plzeň 2026**



## **Čestné prohlášení**

Prohlašuji, že jsem diplomovou práci na téma

*„Aplikace matematického programování pro řešení problému podnikové praxe“*

vypracoval/a samostatně pod odborným dohledem vedoucí/vedoucího diplomové práce za použití pramenů uvedených v příložené bibliografii.

Plzeň dne 20. 4. 2026

v. r. *Jan Bílek*

## **Zásady pro vypracování práce**

Zásada

1. Proved'te teoretický úvod do oblasti matematického programování.
2. Představte vybranou společnost, zejména problematiku výroby a dopravy.
3. Řešte vybraný podnikový problém s využitím vhodně zkonstruovaného matematického modelu.
4. Navrhněte pro podnik možná doporučení a zformulujte závěr.

## **Studijní program**

Projektové řízení

## **Poděkování**

Rád bych poděkoval panu Ing. Zdeňku Kresovi za odborné vedení, cenné připomínky a za vstřícný přístup a podporu při zpracování této bakalářské práce.

# Obsah

Úvod .....	7
<b>1 Rozhodovací proces .....</b>	<b>8</b>
1.1 Kvalitativní analýza .....	9
1.2 Kvantitativní analýza .....	9
<b>2 Operační výzkum .....</b>	<b>11</b>
2.1 Historie operačního výzkumu .....	11
2.2 Matematické programování .....	12
2.2.1 Postup řešení úloh v matematickém programování .....	14
<b>3 Lineární programování .....</b>	<b>17</b>
3.1 Obecný tvar lineárního modelu .....	17
3.2 Grafické řešení .....	19
3.3 Simplexová metoda .....	21
3.4 Celočíselné programování .....	23
3.5 Přiřazovací úloha .....	24
3.6 Dopravní úloha .....	25
3.7 Možnosti softwarového řešení optimalizačních úloh .....	27
<b>4 Podnik Prefab Factory a.s. ....</b>	<b>34</b>
<b>5 Model optimalizace výběru zakázek při omezené výrobní kapacitě .....</b>	<b>39</b>
5.1.1 Formulace zadání problému .....	42
5.1.2 Výsledný matematický model .....	49
5.1.3 Řešení matematického modelu pomocí jazyka Python .....	50
5.1.4 Intepretace výsledků .....	55
5.2 Analýza výsledků a možné doporučení pro podnik .....	59
<b>Závěr .....</b>	<b>65</b>

<b>Seznam obrázků.....</b>	<b>68</b>
<b>Seznam tabulek .....</b>	<b>69</b>
<b>Zdroje a literatura .....</b>	<b>70</b>
<b>Přílohy</b>	
<b>Abstrakt</b>	
<b>Abstract</b>	

# Úvod

V podnikovém prostředí se firmy stále setkávají se situacemi, kdy je nutné rozhodovat o omezených zdrojích v časovém tlaku. Ve výrobních podnicích se tyto problémy projevují zejména při plánování kapacit, organizaci navazujících procesů a při rozhodování o tom, které zakázky je podnik schopen a současně vhodné přijmout k realizaci. V takových případech nemusí být rozhodování založené pouze na zkušenosti nebo odhadu dostatečné a je vhodné jej podpořit systematickým analytickým přístupem. Jedním z možných přístupů je využití metod operačního výzkumu, zejména matematického programování. Tyto metody umožňují převést reálný rozhodovací problém do formální podoby a následně hledat řešení, které respektuje zadaná omezení a současně vede k co nejlepšímu výsledku podle zvoleného kritéria. V podnikovém prostředí tak mohou sloužit jako užitečný nástroj pro podporu manažerského rozhodování.

Cílem této práce je řešit vybraný podnikový problém společnosti Prefab Factory prostřednictvím vhodně sestaveného matematického modelu. Pozornost je zaměřena na rozhodování o výběru zakázek při omezené výrobní kapacitě podniku. Smyslem je navrhnout takový model, který umožní posoudit jednotlivé zakázky nejen z hlediska jejich ekonomického přínosu, ale také s ohledem na jejich kapacitní náročnost, strukturu modulů a míru standardizace výroby, a na tomto základě určit pro podnik nejvhodnější řešení. Práce je rozdělena na teoretickou a praktickou část. Teoretická část se věnuje základním východiskům rozhodovacího procesu, operačního výzkumu, matematického modelování a vybraným optimalizačním metodám, včetně stručného přehledu možností jejich softwarového řešení. Praktická část představuje podnik Prefab Factory a.s., popisuje souvislosti jeho výrobního procesu a navazuje formulací optimalizačního modelu zaměřeného na výběr zakázek při omezené výrobní kapacitě. Model je následně implementován v programovém prostředí Python, vyřešen a interpretován z hlediska jeho přínosu pro rozhodování podniku. Na základě dosažených výsledků a jejich zhodnocení jsou poté pro podnik formulována doporučení.

# 1 Rozhodovací proces

Každý podnik se při své činnosti dostává do situací, kdy je nutné volit mezi několika možnými variantami. Taková rozhodnutí se mohou týkat využití zdrojů, organizace činností, nebo volby nejvhodnějšího řešení. Z tohoto důvodu je vhodné nejprve vymezit samotný proces volby, jeho podstatu a základní fáze.

Rozhodovací proces je systematický postup, jehož předmětem je manažerský problém a jehož cílem je zvolit nejvhodnější variantu řešení určitého problému z více dostupných možností. V kontextu manažerského a ekonomického rozhodování představuje klíčový nástroj, který propojuje analýzu informací, hodnocení alternativ a volbu optimálního řešení s ohledem na stanovené cíle a omezení.

Podle Jablonského (2007) je rozhodovací proces základem manažerské činnosti, protože většina rozhodnutí v organizaci má kvantitativní charakter, vyžaduje tedy analýzu dat, tvorbu modelů a výběr optimální varianty pomocí metod operačního výzkumu. Rozhodování bývá často složité, neboť se odehrává v podmínkách nejistoty, rizika a omezených zdrojů.

Plevný a Žižka (2010) zdůrazňují, že efektivní rozhodování vyžaduje kombinaci kvantitativních analytických postupů s kvalitativním posouzením reálného kontextu. Manažer či analytik v tomto procesu stanovuje cíle rozhodování, vymezuje rozhodovací proměnné a omezující podmínky a následně využívá ekonomicko-matematické modely jako nástroj pro systematickou analýzu variant řešení, jejichž interpretace však musí být zasazena do širších souvislostí rozhodovaného problému.

## **Fáze rozhodovacího procesu**

Autoři jako Gros (2003) či Jablonský (2007) se shodují, že rozhodovací proces probíhá v několika logických krocích, a to:

- Identifikace problému – přesné určení, co je třeba rozhodnout a jaké jsou cíle.
- Analýza situace – shromáždění relevantních dat a informací o podmínkách rozhodování.
- Formulace modelu – vytvoření zjednodušeného popisu reality, který umožňuje kvantitativní analýzu.
- Stanovení kritérií a variant – definování hodnotících ukazatelů a možných řešení.

- Výběr optimální varianty – využití matematických a statistických metod (např. optimalizačních modelů, vícekriteriální analýzy).
- Implementace a kontrola výsledků – uvedení rozhodnutí do praxe a sledování jeho efektů.

## 1.1 Kvalitativní analýza

Kvalitativní analýza se zaměřuje na nečíselné, popisné aspekty zkoumaného problému. V manažerském rozhodování představuje první fázi poznávání situace, kdy je nutné pochopit souvislosti, vztahy a faktory, které mohou ovlivnit výsledek rozhodnutí.

Podle Grose (2003) kvalitativní analýza umožňuje identifikovat podstatné prvky systému – například cíle, možné alternativy, omezení, rizika či zájmy zainteresovaných stran. Nepracuje primárně s číselnými daty, ale spíše s informacemi, zkušenostmi, úsudkem odborníků nebo pozorováním reality.

Plevný a Žižka (2010) zdůrazňují, že kvalitativní analýza má průzkumný charakter, tedy slouží k pochopení problému a k formulaci jeho struktury předtím, než je možné přistoupit k matematickému modelování. Pomáhá stanovit rozhodovací proměnné, cíle a omezující podmínky, čímž vytváří základ pro pozdější kvantitativní zpracování.

Mezi běžné nástroje kvalitativní analýzy patří např. SWOT analýza, analýza procesů, brainstorming nebo rozhovory s odborníky.

## 1.2 Kvantitativní analýza

Kvantitativní analýza představuje přesnější, matematicky formulovaný přístup k rozhodování. Podle Jablonského (2007) spočívá řešení problému v jeho převodu do číselné podoby prostřednictvím matematického modelu, v němž jsou definovány rozhodovací proměnné a omezující podmínky, přičemž následná analýza je prováděna s využitím metod operačního výzkumu, statistiky a optimalizace. Plevný a Žižka (2010) vysvětlují, že kvantitativní analýza umožňuje měřit, porovnávat a vyhodnocovat alternativy podle stanovených kritérií (např. zisk, náklady, čas, riziko). Výsledkem je objektivní základ pro rozhodnutí, které lze obhájit racionálními argumenty a výpočty. V souvislosti s optimalizačními metodami je kvantitativní přístup klíčový tam, kde je možné jasně vyjádřit vztahy mezi proměnnými a cílem – například ve financích, logistice, výrobním plánování nebo řízení portfolia. Využívají se zde modely lineární, nelineární či

dynamické optimalizace (Cornuéjols & Tütüncü 2006). Gros (2003) doplňuje, že kvantitativní analýza poskytuje měřitelný a opakovatelný základ pro rozhodování. Pomocí matematických modelů lze testovat různé scénáře, analyzovat rizika a vybrat variantu s nejlepšími parametry podle zvoleného kritéria.

Postup kvantitativní analýzy lze podle Plevného a Žižky (2010) stručně popsat takto:

- Pozorování systému – seznámení se s reálným systémem, jeho strukturou, procesy a vztahy mezi prvky.
- Definice problému – přesné vymezení cíle analýzy a stanovení klíčových proměnných, které problém ovlivňují.
- Konstrukce modelu – vytvoření matematického nebo simulačního modelu, který zjednodušeně popisuje chování systému.
- Příprava dat – shromáždění, úprava a ověření dat potřebných pro práci s modelem.
- Řešení úlohy – aplikace vhodných optimalizačních metod k nalezení řešení.
- Interpretace výsledků – vyhodnocení výsledků ve vztahu k reálnému systému, ověření jejich platnosti a formulace doporučení pro rozhodování.

### **Porovnání a propojení přístupů**

Kvalitativní a kvantitativní analýza nejsou konkurenční, ale komplementární přístupy. Kvalitativní analýza pomáhá pochopit a formulovat problém, zatímco kvantitativní analýza umožňuje tento problém měřitelně vyhodnotit. Efektivní rozhodovací proces proto obvykle kombinuje oba přístupy:

- Kvalitativní část definuje, co se má zkoumat a jaké faktory jsou důležité.
- Kvantitativní část následně zjišťuje, jak moc tyto faktory ovlivňují výsledky a jaký je nejlepší způsob řešení.

Rozhodování v reálných podmínkách se opírá jak o kvalitativní úsudek, tak o kvantitativní analytické přístupy. Pro řešení složitějších rozhodovacích situací se proto uplatňují metody, které umožňují převést tyto úvahy do formální podoby a systematicky s nimi pracovat. Další část se zaměřuje na vývoj těchto metod, jejich hlavní oblasti využití a roli, kterou hrají při podpoře rozhodování.

## 2 Operační výzkum

S rostoucí složitostí v rozhodování již často nestačí vycházet pouze ze zkušenosti nebo intuitivního úsudku. V praxi je proto vhodné opřít rozhodování o přístupy, které umožňují problém systematicky analyzovat a porovnávat jednotlivé varianty řešení. Jedním z takových přístupů je právě operační výzkum.

Operační výzkum lze charakterizovat jako množinu kvantitativních postupů, které nám umožňují chápat složité systémy a lze s jejich pomocí nalézt nejlepší možné varianty řešení (Jablonský 2007).

Podle Grose (2003) je operační výzkum zaměřen na efektivní využívání omezených zdrojů prostřednictvím přesně definovaných modelů a analytických metod.

Podstatou operačního výzkumu je převod reálných problémů do matematické podoby, jelikož v takové podobě lze lépe porovnávat alternativy a vybrat nejvíce uspokojivé řešení (Plevný & Žižka 2010).

### 2.1 Historie operačního výzkumu

Operační výzkum vznikl ve 20. století jako reakce na potřebu racionálně řídit složité procesy, především během druhé světové války. Jeho základy však sahají až do 19. století, kdy se objevily první snahy o systematickou analýzu výroby, dopravy a řízení procesů (Britannica, n.d.; Church & Ackoff, 1957). Ve 30. letech 20. století začali britští vědci vedení A. P. Rowem a Robertem Watsonem-Wattem využívat vědecké metody při organizaci radarové obrany. Právě tehdy došlo ke spojení mezi vědou a optimalizací reálných vojenských operací, což položilo základ moderního operačního výzkumu (INFORMS, n.d.).

Během druhé světové války se operační výzkum rozvinul jako samostatná disciplína. Vědecké týmy analyzovaly efektivní nasazení radarů, ochranu konvojů, protiletadlovou obranu i logistiku vojenských sil (Shrader, C. R., 2006). Po válce se díky úspěchům ve vojenské oblasti tyto metody rychle rozšířily do civilního sektoru, zejména do průmyslu, dopravy a vládní správy (Öztürk, 2013). V roce 1948 byl na Massachusettském technologickém institutu (MIT) otevřen první kurz operačního výzkumu mimo armádu a v 50. letech se z operačního výzkumu stal samostatný akademický obor (Britannica, n.d.).

V tomto období se rozvíjely klíčové metody, jako lineární programování, teorie front, simulační modelování či rozhodovací analýza. Současně vznikly odborné společnosti, například britská Operational Research Society (1948) a americká Operations Research Society of America (1952), které významně přispěly k rozvoji oboru (The OR Society, n.d.; INFORMS, n.d.).

Dnes je operační výzkum chápán jako interdisciplinární oblast propojující matematiku, statistiku, informatiku a ekonomii. Jeho metody nacházejí uplatnění v logistice, zdravotnictví, energetice, dopravě i financích (Öztürk, 2013). V podnikovém prostředí pomáhá řídit procesy, plánovat výrobu, optimalizovat dopravní trasy a efektivně alokovat zdroje. Jeho hlavní přínos spočívá v podpoře rozhodování založeného na datech a matematických modelech namísto intuice (Kislingerová, 2007). Jak uvádí Kubišová (2015), rozvoj informačních technologií umožňuje řešit stále složitější modely s vysokou přesností, což zvyšuje využitelnost operačního výzkumu v praxi. Ten se tak stal významným nástrojem moderního řízení, který propojuje teoretické poznatky s praktickými aplikacemi a pomáhá manažerům převádět složité problémy do kvantitativní formy pro efektivní rozhodování (Jablonský, 2007; Kislingerová, 2007; Kubišová, 2015).

## **2.2 Matematické programování**

Matematické programování je klíčovou oblastí operačního výzkumu, zabývající se optimalizací řešení u rozhodovacích úloh, které jsou vyjádřeny pomocí matematického modelu. Cílem je nalézt takové hodnoty rozhodovacích proměnných, které minimalizují nebo maximalizují modelem stanovený záměr reprezentovaný účelovou funkcí. Zároveň je nutné, aby byla splněna veškerá omezení, které popisují reálná omezení systému (Jablonský 2007).

### **Ekonomický model**

Ekonomický model představuje zjednodušený popis ekonomické situace, který je vyjádřen slovně i numericky prostřednictvím proměnných a podmínek. Slouží jako výchozí předloha pro tvorbu matematického modelu, a proto musí obsahovat všechny informace potřebné k jeho pozdějšímu formálnímu sestavení. Model tak zahrnuje vymezení požadovaného výsledného stavu, popis hlavních prvků systému, jejich

vzájemných vztahů a omezení, která určují, co je v systému možné. Součástí ekonomického modelu je také charakteristika procesů, které v systému probíhají, jelikož tyto údaje vytvářejí základ pro přesnou formulaci matematického modelu. (Gros, 2003; Plevný, Žižka 2010)

### **Model v matematickém programování**

Model je zjednodušený popis reálného systému, který popisuje jeho vlastnosti a vztahy mezi prvky, aby bylo možné systému porozumět a předvídat jeho chování.

Podle Jablonského (2007) je model nástrojem, který umožňuje nahradit zkoumání složité reality její jednodušší reprezentací, přičemž zachovává rozhodující prvky důležité pro řešení daného problému.

Model je vždy určité zjednodušení skutečnosti, které musí být dostatečně přesné pro dosažení cíle analýzy, ale zároveň dostatečně jednoduché, aby s ním bylo možné pracovat a vystihovalo pouze ty vlastnosti, které jsou důležité z hlediska řešení problému (Plevný a Žižka 2010).

### **Rozhodovací proměnné**

Podle Plevného a Žižky (2010) vyjadřují rozhodovací proměnné říditelné vstupy reálného problému, tedy prvky, které lze cíleně ovlivnit (např. objem výroby, velikost investice). Pomocí rozhodovacích proměnných se v modelu promítá volba, což určuje strukturu matematického popisu modelu. Strukturou je přitom míněno uspořádání a vzájemné vazby rozhodovacích proměnných v účelové funkci a omezujících podmínkách, které vymezují tvar modelu a charakter řešeného rozhodovacího problému. Proměnné modelu musí být jasně definované, měřitelné a interpretovatelné. Každá proměnná má svoji množinu hodnot, kterých může nabývat neboli definiční obor. Definiční obor je často stanoven na ekonomických a technických omezení reálného systému (Jablonský 2007; Gros 2003). Vhodná identifikace a volba rozhodovacích proměnných je stěžejní krok pro tvorbu modelu, jelikož proměnné určují výslednou hodnotu účelové funkce. Nesprávně zvolená nebo neúplná sada proměnných nebude schopna odhalit reálnou podstatu problému. (Nayak, et al. 2023)

## **Účelová funkce**

Základním prvkem matematického modelu je účelová funkce, jenž vyjadřuje cíl optimalizace a formuluje, jakého extrému kritérium má daný model sledovat. Nejčastěji se jedná o úlohy maximalizační (kritéria jako zisk, efektivita, produkt) nebo minimalizační (náklady, čas). Účelová funkce převádí reálný problém do kvantitativní podoby, ve které lze porovnat a analyzovat různé varianty řešení dle jednoho kritéria (Hillier & Lieberman, 2021). Williams (2013) píše, že je správná formulace účelové funkce rozhodující fází, jelikož představuje, jakým způsobem model odráží realitu.

## **Omezující podmínky modelu**

Omezující podmínky matematického modelu vymezují reálné limity, pro rozhodovací proměnné, tedy přípustný obor řešení, v jehož rámci mohou proměnné nabývat svých hodnot. Nutnou vlastností podmínek je co nejpřesněji reflektovat reálná omezení systému, tak aby model co nejvíce odpovídal realitě a bylo možné nalézt řešení. Důležité tedy je, že omezující podmínky nejen vymezují přípustnou oblast řešení, ale zároveň určují, jakým způsobem jsou rozhodovací proměnné v modelu vzájemně provázány, neboť právě prostřednictvím těchto podmínek jsou stanoveny hranice, v jejichž rámci je hledáno optimální řešení. (Nayak, et al. 2023).

### **2.2.1 Postup řešení úloh v matematickém programování**

Řešení rozhodovacích úloh v rámci operačního výzkumu lze podle Grose (2003), Jablonského (2007), Navaka, Varshneva a Shekhara (2023) a Plevného a Žižky (2010) systematicky členit do několika na sebe navazujících fází:

#### **Vymezení problému**

Řešení úloh v operačním výzkumu začíná jednoznačným určením a pochopením problému, kdy se objasňují hlavní cíle, podmínky rozhodování (např. dostupné informace, omezení reálného prostředí, vnější vlivy) a důvody, proč je potřeba danou situaci analyzovat, což vytváří základní orientaci v celém rozhodovacím procesu.

## **Analýza a pochopení systému**

V další fázi se podrobně zkoumá prostředí, shromažďují relevantní informace, popisují se klíčové vazby mezi prvky systému a identifikují faktory, které mohou ovlivnit budoucí rozhodnutí, čímž vzniká kvalitativní podklad pro následné matematické zpracování.

## **Formulace matematického modelu**

Následuje tvorba modelu, při níž se reálný problém převádí do matematické struktury tvořené rozhodovacími proměnnými, cílovou funkcí a omezujícími podmínkami, což umožňuje pracovat s problémem pomocí kvantitativních metod a zachytit jeho podstatu ve zjednodušené, ale prakticky využitelné podobě.

## **Výběr metody řešení**

Po sestavení matematického modelu je nezbytné zvolit vhodnou analytickou či výpočetní metodu jeho řešení, která odpovídá typu již definovaného modelu a charakteru úlohy. Volba metody se přitom týká konkrétních postupů, například simplexové metody, gradientních metod nebo dalších algoritmů určených pro danou třídu optimalizačních problémů.

## **Výpočtové řešení modelu**

V této fázi se zvolený model řeší pomocí analytických technik nebo softwarových nástrojů, čímž se získá optimální nebo přijatelně efektivní varianta řešení, která vychází z přesně definovaných vstupů modelu.

## **Interpretace a ověření výsledků**

Jakmile je řešení nalezeno, je nezbytné posoudit jeho význam, zhodnotit jeho realizovatelnost v praxi a ověřit citlivost výsledků na změny vstupních parametrů, protože teprve tak lze vyhodnotit kvalitu a spolehlivost navrženého rozhodnutí.

## **Implementace a doporučení pro praxi**

Závěrečnou část tvoří návrh konkrétních kroků, které je třeba podniknout při zavedení doporučeného řešení do reálného prostředí, přičemž může dojít k úpravám modelu nebo doplnění dalších informací, pokud se ukáže, že skutečné podmínky neodpovídají původním předpokladům.

Přestože jednotliví autoři mohou klást důraz na odlišné aspekty procesu nebo používat mírně rozdílné členění, shodují se na základní posloupnosti kroků vedoucích od analýzy reálného problému přes formulaci matematického modelu a volbu vhodné metody řešení až po interpretaci výsledků a jejich praktickou implementaci.

Matematické programování představuje obecný rámec pro formální popis rozhodovacích situací a vztahů mezi jejich jednotlivými prvky. V praxi je však často žádoucí pracovat s takovou podobou modelu, která umožňuje efektivní analytické zpracování a interpretaci výsledků. Z tohoto důvodu se pozornost dále soustředí na modely, u nichž jsou vztahy mezi proměnnými vyjádřeny lineárně, což umožňuje využití specifických optimalizačních postupů.

### 3 Lineární programování

Lineární programování (LP) je jednou z nejvyužívanějších a nejpropracovanějších metod operačního výzkumu, neboť pracuje s lineární účelovou funkcí a lineárními omezujícími podmínkami, což umožňuje najít řešení rychle a poměrně jednoduše. Pokud lze problém formulovat do tvaru lineárního modelu, existují k získání výsledku velmi dobré řešící prostředky. Modely LP operují pod přímou úměrou parametrů (tj. konstantní výnosy z rozsahu) a neobsahují náhodné jevy, tedy jsou deterministické.

Modely LP zobrazují reálný systém s určitou nepřesností, jelikož vztahy mezi parametry nejsou v realitě čistě lineární. I tak dokážou modely LP poskytnout dobré výsledky pro podporu rozhodování v různých variantách manažerských problémů, přičemž nelineární závislosti lze v praxi často s dostatečnou přesností aproximovat lineárními vztahy, zejména v omezeném rozsahu hodnot rozhodovacích proměnných. (Bertsimas & Tsitsiklis, 1997; Gros, 2007; Plevný & Žižka, 2010; Šubrt et al., 2011)

#### 3.1 Obecný tvar lineárního modelu

Sestavení matematického modelu úlohy lineárního programování vychází z formálního popisu reálného rozhodovacího problému. Základními kroky tohoto procesu jsou definice rozhodovacích proměnných, konstrukce účelové funkce a formulace omezujících podmínek.

##### Definice proměnných

Rozhodovací proměnné reprezentují jednotlivé procesy nebo aktivity zahrnuté do modelu, které lze ovlivnit (řiditelné vstupy). Tyto proměnné se obvykle označují symbolem  $x_i$ , kde  $i = 1, 2, \dots, n$ , přičemž je nutné stanovit také jejich jednotky vyjádření (Šubrt et al., 2015).

##### Konstrukce účelové funkce

Účelová funkce vyjadřuje cíl optimalizace, například maximalizaci zisku nebo minimalizaci nákladů. V lineárním programování má účelová funkce lineární tvar:

$$Z_{max} = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (1)$$

Koeficienty  $c_i$  vyjadřují příspěvek jednotlivých proměnných k hodnotě účelové funkce (Lunáček & Heralecký, 2009).

### Formulace omezujících podmínek

Gros (2007) říká, že omezující podmínky vymezují přípustné hodnoty rozhodovacích proměnných a zachycují reálná omezení modelovaného systému, zejména kapacitní a požadavkové limity. Obecně je lze zapsat ve tvaru:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \quad (2)$$

kde koeficienty  $a_{ij}$  vyjadřují vztah mezi proměnnými a jednotlivými omezeními a  $b_i$  představují hodnoty pravých stran podmínek (Bertsimas & Tsitsiklis, 1997). Součástí modelu jsou také tzv. obligátní podmínky, typicky podmínka nezápornosti proměnných:

$$x_i \geq 0, \quad (3)$$

$$i = 1, 2, \dots, n$$

Na základě výše uvedených kroků lze úlohu lineárního programování shrnout do obecného modelu, který má tvar optimalizační úlohy s lineární účelovou funkcí a soustavou lineárních omezujících podmínek:

maximalizuj

$$Z_{max} = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

za podmínek:

$$\sum_{j=1}^n a_{ij} x_j \leq b_j \quad i = 1, 2, \dots, m \quad (4)$$

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

kde:

$x_j$  = j. rozhodovací proměnná,

$n$  = počet rozhodovacích proměnných,

$c_j$  = koeficient účelové funkce,

$a_{ij}$  = koeficient intenzity vazby mezi  $j$ . rozhodovací proměnnou a  $i$ . omezujícím faktorem,

$b_i$  = kapacita dostupného zdroje (určuje přípustnost  $i$ . omezení),

$m$  = počet omezení

(Bertsimas & Tsitsiklis, 1997; Plevný & Žižka, 2010).

Na základě vymezení obecného matematického modelu lze dále věnovat pozornost způsobům jeho řešení. Následující kapitoly se zabývají vybranými postupy uplatňovanými při řešení optimalizačních úloh, a to jak u problémů menšího rozsahu, tak u rozsáhlejších a specificky strukturovaných případů. Tyto přístupy ilustrují, jak lze obecně formulovaný model analyzovat a řešit v praxi.

## 3.2 Grafické řešení

Grafické řešení lineárního modelu představuje základní metodu řešení úloh lineárního programování se dvěma rozhodovacími proměnnými. Princip spočívá v geometrickém znázornění omezujících podmínek a účelové funkce v kartézském souřadnicovém systému, kde osy systému znázorňují dané proměnné modelu (Bertsimas & Tsitsiklis, 1997; Jablonský, 2007).

Každá omezující podmínka je zobrazena jako přímka, která rozděluje rovinu na dvě poloroviny (pokud se jedná o omezující podmínku nerovnosti). Průnik všech polorovin (a případně přímek, pokud existuje podmínka ve tvaru rovnosti) určených omezujícími podmínkami tvoří množinu přípustných řešení (MPŘ), která je v případě lineárního programování konvexním polyedrem (Plevný & Žižka, 2010). Pouze body ležící v této oblasti splňují všechna omezení modelu.

Vlastnost konvexnosti znamená, že pro libovolné dva body ležící v dané množině patří do množiny i úsečka, která tyto body spojuje (Plevný & Žižka, 2010). MPŘ má geometrickou strukturu, jejíž hranice jsou tvořeny hranicemi polorovin (omezující přímky), které odpovídají jednotlivým omezujícím podmínkám modelu. Zásadní vlastností konvexního polyedru v lineárním programování je skutečnost, že se optimální řešení lineární optimalizační úlohy nachází v některém z jeho extrémních bodů (vrcholů). (Bertsimas & Tsitsiklis, 1997; Jablonský, 2007).

Účelová funkce je v grafickém řešení znázorněna pomocí množiny rovnoběžných přímek (hladin), přičemž každá z těchto přímek odpovídá určité hodnotě účelové funkce (Gros, 2003).

Při změně pravé strany (parametr  $b$ ) z rovnice účelové funkce (1)

$$\mathbf{a}_1\mathbf{x}_1 + \mathbf{a}_2\mathbf{x}_2 = \mathbf{b} \quad (1)$$

dochází k rovnoběžnému posunu příslušné přímky, aniž by se změnil její sklon, který je dán koeficienty  $a_1$  a  $a_2$ . Změna hodnoty parametru  $b$  tedy neovlivňuje orientaci přímky v rovině, ale pouze její polohu. Tímto způsobem vzniká množina rovnoběžných přímek (hladina) účelové funkce (Gros, 2003).

V grafickém řešení maximalizační úlohy se tyto přímky posouvají ve směru rostoucích hodnot účelové funkce. Optimum úlohy je poté poslední přímka z množiny určené účelovou funkcí, která stále tvoří průnik s MPŘ, což znamená, že se optimum nachází na hranici MPŘ a je dosaženo v krajním bodě, přičemž v případě více optimálních řešení může optimum tvořit celá hrana (Plevný & Žižka, 2010; Jablonský, 2007).

Při řešení úlohy lineárního programování mohou nastat následující základní situace, které vyplývají z geometrických vlastností MPŘ a tvaru účelové funkce (Jablonský, 2007; Plevný & Žižka, 2010):

- Úloha má jednoznačné právě jedno optimální řešení, pokud účelová funkce dosahuje své maximální nebo minimální hodnoty v jediném extrémním bodě MPŘ. Tato situace je v praxi nejčastější a odpovídá typickému průběhu grafického i algoritmického řešení (Bertsimas & Tsitsiklis, 1997).
- Existence více řešení (nekonečna řešení, pokud je MPŘ spojitá) pokud je hladina účelové funkce v optimální poloze rovnoběžná s hranou MPŘ. Všechny body ležící na této hraně poskytují stejnou optimální hodnotu účelové funkce (Gros, 2003; Jablonský, 2007).
- Úloha je nepřipustná, pokud neexistuje žádný bod, který by splňoval všechny omezující podmínky současně. Geometricky to znamená, že průnik polorovin určených omezeními je prázdný (Plevný & Žižka, 2010).

- Úloha je neomezená, pokud hodnota účelové funkce může růst (nebo klesat) neomezeně, aniž by došlo k porušení omezujících podmínek. V takovém případě neexistuje konečné optimální řešení (Bertsimas & Tsitsiklis, 1997).

### 3.3 Simplexová metoda

#### Podstata a princip metody

Simplexová metoda je algoritmický postup pro řešení úloh lineárního programování, jehož cílem je nalézt optimální hodnoty rozhodovacích proměnných při splnění všech omezujících podmínek. Metoda je založena na skutečnosti, že optimální řešení lineárního programovacího problému leží v některém z krajních bodů (vrcholů) přípustné oblasti, která je tvořena konvexním polyedrem (Bertsimas & Tsitsiklis, 1997; Jablonský, 2007).

Základní princip simplexové metody spočívá v systematickém přechodu mezi sousedními krajními body přípustné oblasti, přičemž v každém kroku dochází ke zlepšení hodnoty cílové funkce, dokud není dosaženo optima (Gros, 2003; Šubrt et al., 2011).

Výchozím bodem aplikace simplexové metody je převedení úlohy do kanonického tvaru. Podle Plevného a Žižky (2010) se kanonický tvar vyznačuje tím, že všechny omezující podmínky jsou vyjádřeny rovnicemi, pravé strany těchto rovnic jsou nezáporné a matice koeficientů obsahuje jednotkovou submatici odpovídající bázi. Při tomto převodu se obvykle nejprve transformuje minimalizační úloha na maximalizační (5) změnou znamének koeficientů cílové funkce, dále se zajistí nezápornost pravých stran (6) omezení a nerovnosti se převádějí na rovnosti zavedením přídavných proměnných (7). Tyto proměnné reprezentují buď nevyužitou kapacitu zdrojů, nebo naopak překročení stanoveného limitu. V závěrečné fázi se případně doplňují pomocné (umělé) proměnné (8) tak, aby v matici koeficientů vznikla jednotková báze.

$$Z_{min} = c_1x_1 + c_2x_2 \rightarrow Z_{max} = -(c_1x_1 + c_2x_2) \quad (5)$$

$$a_{11}x_1 + a_{12}x_2 \leq b, kde b > 0 \rightarrow a_{11}x_1 + a_{12}x_2 \leq b \quad (6)$$

$$a_{11}x_1 + a_{12}x_2 \leq b, kde b < 0 \rightarrow -a_{11}x_1 - a_{12}x_2 \geq -b$$

$$-a_{11}x_1 - a_{12}x_2 \geq -b \rightarrow -a_{11}x_1 - a_{12}x_2 - y_1 = -b \quad (7)$$

$$-a_{11}x_1 - a_{12}x_2 - y_1 = -b \rightarrow -a_{11}x_1 - a_{12}x_2 - y_1 + u_1 = -b \quad (8)$$

Bazické přípustné řešení (11) úlohy lineárního programování s  $n$  proměnnými a  $m$  omezujícími podmínkami je takové řešení, v němž nejvýše  $m$  proměnných nabývá

kladných hodnot, zatímco ostatní proměnné jsou rovny nule. Těmto bazickým proměnným odpovídají jednotkové vektory v matici soustavy a jejich hodnoty jsou dány pravými stranami omezení (Šubrt et al., 2011).

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 &\leq b_2 \\ x_1, x_2 &\geq 0 \end{aligned} \quad (9)$$

$$\begin{aligned} \mathbf{a}_{11}\mathbf{x}_1 + \mathbf{a}_{12}\mathbf{x}_2 + y_1 &= \mathbf{b}_1 \\ \mathbf{a}_{21}\mathbf{x}_1 + \mathbf{a}_{22}\mathbf{x}_2 + y_2 &= \mathbf{b}_2 \end{aligned} \quad (10)$$

$$\vec{x} = (\mathbf{x}_1, \mathbf{x}_2, y_1, y_2) = (\mathbf{0}, \mathbf{0}, \mathbf{b}_1, \mathbf{b}_2) \quad (11)$$

- proměnné  $x_1, x_2$  nejsou v bázi a mají nulovou hodnotu,
- proměnné  $y_1, y_2$  jsou bazické a jejich hodnoty jsou rovny pravým stranám omezení.

Z geometrického hlediska platí, že pokud má úloha LP optimální řešení, nachází se (některé z nich) v jednom z krajních bodů MPŘ. Právě těmto krajním bodům odpovídají přípustná bazická řešení. Tato skutečnost je formalizována základní větou lineárního programování, podle níž má každá úloha LP s optimálním řešením alespoň jedno optimální řešení, které je zároveň bazickým přípustným řešením (Plevný & Žižka, 2010).

Princip simplexové metody tedy spočívá v systematickém prohledávání bazických přípustných řešení. Algoritmus začíná nalezením jednoho výchozího bazického řešení a následně přechází k dalším bazickým řešením tak, aby v každém kroku docházelo ke zlepšení hodnoty cílové funkce. Tyto přechody jsou realizovány prostřednictvím simplexové tabulky a odpovídajících algebraických úprav. Proces je ukončen ve chvíli, kdy již není možné nalézt řešení s lepší hodnotou cílové funkce, přičemž dosažené řešení je považováno za optimální (Plevný & Žižka, 2010).

### Možné výsledky řešení

Použitím simplexového algoritmu se lze dostat k následujícím situacím:

- Existence jednoznačného optimálního řešení, kdy cílová funkce dosahuje maxima v jediném krajním bodě.
- Existence alternativních optimálních řešení, kdy cílová funkce nabývá stejné optimální hodnoty ve více krajních bodech.
- Neomezenost úlohy, kdy lze hodnotu cílové funkce zvyšovat bez omezení.

- Neexistence přípustného řešení, které nastává, pokud soustavě omezení nevyhovuje žádné řešení.

Tyto možnosti jsou přirozeným důsledkem geometrické struktury přípustné oblasti (Jablonský, 2007; Šubrt et al., 2011).

### **Výhody a omezení simplexové metody**

Mezi hlavní výhody simplexové metody patří její praktická efektivita, interpretovatelnost výsledků a možnost citlivostní analýzy, díky čemuž nachází široké uplatnění v manažerském rozhodování (Gros, 2003; Plevný & Žižka, 2010). Omezením metody je její použitelnost výhradně pro lineární modely, možnost výskytu degenerace a skutečnost že při velmi velkém počtu proměnných a omezujících podmínek se zvyšují nároky na výpočetní prostředky a paměť, což může omezovat její praktickou aplikovatelnost pro velmi rozsáhlé úlohy (Bertsimas & Tsitsiklis, 1997; Cornuéjols & Tütüncü, 2006).

### **Degenerace řešení**

Specifickým jevem při použití simplexové metody je degenerace řešení, při níž některé základní proměnné nabývají nulových hodnot. V důsledku toho může docházet k situacím, kdy algoritmus sice mění bázi, avšak hodnota řešení se nemění.

Degenerace může vést ke zpomalení algoritmu, k opakovanému přecházení mezi stejnými bázemi (cyklování) a v krajním případě k nutnosti použití speciálních pravidel pro výběr pivota. Tyto situace nepředstavují chybu modelu, ale vyplývají z geometrických vlastností přípustné oblasti a z algebraické reprezentace řešeného problému. (Bertsimas & Tsitsiklis, 1997; Jablonský, 2007)

## **3.4 Celočíselné programování**

Celočíselné programování představuje rozšíření lineárního programování, v němž je u části nebo u všech rozhodovacích proměnných požadováno, aby nabývaly celočíselných hodnot. Tento typ modelů je využíván zejména v případech, kdy proměnné vyjadřují diskrétní rozhodnutí, jako je počet výrobků nebo volbu ano/ne (Jablonský, 2007; Plevný & Žižka, 2010).

Zvláštním případem celočíselného programování jsou modely s binárními proměnnými (12), které mohou nabývat pouze hodnot 0 nebo 1. Tyto proměnné se používají k popisu

rozhodnutí typu ano/ne. Díky tomu nachází celočíselné programování široké uplatnění v manažerském rozhodování, logistice, plánování výroby či rozvrhování. (Gros, 2003; Šubrt et al., 2011)

$$x \in \{0; 1\}, kde \begin{cases} 1, & \text{pokud jev nastane} \\ 0, & \text{pokud jev nenastane} \end{cases} \quad (12)$$

Na rozdíl od lineárního programování je řešení celočíselných modelů výpočetně náročnější, protože množina přípustných řešení již netvoří spojitou oblast, ale diskrétní množinu bodů. Optimální řešení proto nelze obecně nalézt pomocí jednoduchých analytických postupů a je nutné využívat speciální algoritmy, jako je například metoda větví a hranic. (Bertsimas & Tsitsiklis, 1997; Cornuéjols & Tütüncü, 2006)

Navzdory vyšší výpočetní náročnosti má celočíselné programování významnou praktickou hodnotu, neboť umožňuje modelovat reálné rozhodovací situace s vyšší mírou věrnosti než čistě lineární modely. Výsledná řešení jsou zpravidla snadno interpretovatelná a přímo použitelná v praxi, což činí celočíselné programování důležitým nástrojem moderní optimalizace. (Plevný & Žižka, 2010)

### 3.5 Přiřazovací úloha

Přiřazovací úloha představuje specifický typ optimalizačního problému, který je řešen v rámci celočíselného lineárního programování s využitím binárních rozhodovacích proměnných. Podstatou úlohy je nalezení takového přiřazení mezi dvěma množinami prvků, aby každý prvek z jedné množiny byl přiřazen právě jednomu prvku z množiny druhé, přičemž je optimalizováno zvolené kritérium, nejčastěji celkové náklady nebo čas. (Jablonský, 2007; Plevný & Žižka, 2010)

#### Základní předpoklady přiřazovací úlohy

Aby bylo možné úlohu formulovat jako přiřazovací, musí být splněny následující předpoklady:

- rozhodování je popsáno pomocí binárních proměnných,
- každý prvek z první množiny je přiřazen právě jednomu prvku z druhé množiny,
- počet zdrojů a požadavků je shodný, případně je úloha doplněna o fiktivní prvky pro dosažení rovnováhy.

(Jablonský, 2007)

## Obecná matematická formulace

- $c_{ij}$  označuje například náklady spojené s přiřazením prvku  $i$  k prvku  $j$
- $x_{ij}$  je binární proměnná, která nabývá hodnoty 1, pokud je přiřazení prvku  $i$  k prvku  $j$  realizováno, a 0 v opačném případě.

$$\min z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (13)$$

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= 1, i = 1, 2, \dots, n \\ \sum_{i=1}^n x_{ij} &= 1, j = 1, 2, \dots, n \\ x_{ij} &\in \{0, 1\} \end{aligned} \quad (14)$$

Účelová funkce (13) minimalizuje celkovou hodnotu optimalizovaného kritéria, zatímco omezující podmínky (14) zajišťují, že každý prvek je přiřazen právě jednou a nedochází k duplicitám přiřazení. (Plevný & Žižka, 2010; Bertsimas & Tsitsiklis, 1997)

### 3.6 Dopravní úloha

Dopravní úloha patří mezi základní speciální úlohy lineárního programování a zaměřuje se na optimalizaci přepravy homogenní komodity mezi soustavou zdrojů a cílových míst. Jejím cílem je stanovit takový plán přepravy, který dodrží kapacitní omezení zdrojů i požadavky odběratelů a zároveň minimalizuje celkové přepravní náklady nebo jinou zvolenou optimalizační veličinu. (Jablonský, 2007; Plevný & Žižka, 2010)

Z ekonomického hlediska dopravní úloha modeluje rozhodovací situace, v nichž je nutné rozdělit dostupné zdroje mezi více odběratelů při omezených kapacitách a rozdílných přepravních nákladech. Tyto náklady mohou reprezentovat finanční výdaje, časové nároky nebo vzdálenost mezi zdrojem a cílovým místem. Model tak umožňuje systematicky posoudit alternativní varianty distribuce a vybrat řešení s nejnižšími celkovými náklady. (Gros, 2003; Šubrt et al., 2011)

Rozhodovací proměnné  $x_{ij}$  vyjadřují objem přepravy mezi  $i$ . zdrojem a  $j$ . cílovým místem a tvoří základ pro formulaci omezení a účelové funkce dopravní úlohy. (Mallia et al., 2021)

### Vyrovnaný a nevyrovnaný dopravní problém

Z hlediska vztahu mezi celkovou nabídkou a poptávkou jsou rozlišovány dva základní typy dopravní úlohy. Vyrovnaný (či též vybilancovaný) dopravní problém nastává tehdy, když platí rovnost mezi celkovými kapacitami zdrojů a celkovými požadavky cílových míst (15).

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j. \quad (15)$$

V tomto případě jsou všechny požadavky přesně uspokojeny a veškeré dostupné kapacity plně využity (Jablonský, 2007).

Pokud tato rovnost neplatí, jedná se o nevyrovnaný (nevybilancovaný) dopravní problém. Při převaze nabídky zůstává část kapacit nevyužita, zatímco při převaze poptávky nelze uspokojit všechny požadavky. Takový problém lze převést na vyrovnaný doplněním fiktivního zdroje nebo fiktivního odběratele, přičemž přepravní náklady ve vztahu k fiktivním prvkům jsou obvykle nulové, jelikož nepředstavují skutečnou dopravu. V některých případech však mohou být těmto fiktivním vazbám přiřazeny sankční náklady (pokuty), zejména tehdy, když je cílem modelu zohlednit negativní důsledky neuspokojení poptávky a zajistit, aby byl odběratel uspokojen v co nejvyšší možné míře (Jablonský, 2007; Šubrt et al., 2011).

### Obecná matematická formulace

$m$  = počet zdrojů

$j$  = index cílového místa

$n$  = počet cílových míst

$a_i$  = kapacita zdroje  $i$

$i$  = index zdroje

$b_j$  = požadavek cílového místa  $j$

$x_{ij}$  = rozhodovací proměnná vyjadřující objem přepravy ze zdroje  $i$  do cílového místa  $j$ .

$c_{ij}$  = jednotkové přepravní náklady spojené s přepravou

$$\min z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (16)$$

$$\sum_{j=1}^n x_{ij} = a_i, i = 1, 2, \dots, m \quad (17)$$

$$\sum_{i=1}^m x_{ij} = b_j, j = 1, 2, \dots, n \quad (18)$$

$$x_{ij} \geq 0. \quad (19)$$

Uvedený matematický zápis představuje klasickou formulaci vyrovnané dopravní úlohy v LP. Účelová funkce (16) minimalizuje celkové náklady spojené s přepravou mezi jednotlivými zdroji a cílovými místy. První skupina omezujících podmínek (17) zajišťuje, že celkový objem přepravy z každého zdroje odpovídá jeho kapacitě, zatímco druhá skupina podmínek (18) vyjadřuje splnění požadavků jednotlivých cílových míst. Nezápornost rozhodovacích proměnných (19) odpovídá reálné interpretaci. (Bertsimas & Tsitsiklis, 1997; Mallia et al., 2021; Plevný & Žižka, 2010)

### 3.7 Možnosti softwarového řešení optimalizačních úloh

Optimalizační úlohy lze v praxi řešit pomocí různých softwarových nástrojů, které se liší svými možnostmi, způsobem práce i vhodností pro konkrétní typ problému. V této části jsou proto stručně představeny vybrané přístupy k softwarovému řešení optimalizačních úloh, zejména v prostředí MS Excel a v programovacím jazyce Python

#### Řešení optimalizačních úloh v prostředí MS Excel

Pro řešení lineárních optimalizačních úloh lze v prostředí MS Excel využít doplněk řešitel (*Solver*). Podle Microsoftu (n.d.) tento nástroj slouží k nalezení maximální nebo minimální hodnoty cílové buňky při současném splnění omezení. Využití řešitele proto předpokládá, že je model nejprve vhodně sestaven přímo v pracovním listu a teprve poté je zadán do dialogového okna řešitele.

Základní logika sestavení modelu je zachycena v tabulce 1, která zobrazuje obecnou strukturu optimalizační úlohy. V horní části tabulky jsou uvedeny názvy rozhodovacích proměnných  $x_1, x_2, \dots, x_n$ , pod nimi koeficienty účelové funkce  $c_1, c_2, \dots, c_n$  a dále oblast buněk určená pro samotné rozhodovací proměnné. Hodnota účelové funkce  $Z$  je pak vypočtena jako lineární kombinace koeficientů a proměnných, tedy ve tvaru  $Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$ . Podle Microsoftu (n.d.) musí být cílová buňka definována vzorcem, který je závislý na měněných buňkách modelu.

Tabulka 1: Struktura matematického modelu v MS excel

<b>Účelová funkce</b>	$\max/\min Z = c_1 \cdot x_1 + c_2 \cdot x_2 + \dots + c_n \cdot x_n$			
Označení proměnných	$x_1$	$x_2$	...	$x_n$
Koeficienty proměnných	$c_1$	$c_2$	...	$c_n$
Výsledné hodnoty $x_i$				
Hodnota účelové funkce $Z$	=skalární.součin( $c_1:c_n$ ; $x_1:x_n$ )			
Omezení 1	$a_{11}$	$a_{12}$	...	$a_{1n}$
Omezení 2	$a_{21}$	$a_{22}$	...	$a_{2n}$
...	...	...	...	...
Omezení $m$	$a_{m1}$	$a_{m2}$	...	$a_{mn}$
<b>Levá strana</b>	<b>Znaménko</b>		<b>Pravá strana</b>	
$a_{11} \cdot x_1 + a_{12} \cdot x_2 + \dots + a_{1n} \cdot x_n$	<= / >=		$b_1$	
$a_{21} \cdot x_1 + a_{22} \cdot x_2 + \dots + a_{2n} \cdot x_n$	<= / >=		$b_2$	
...	...		...	
$a_{m1} \cdot x_1 + a_{m2} \cdot x_2 + \dots + a_{mn} \cdot x_n$	<= / >=		$b_m$	

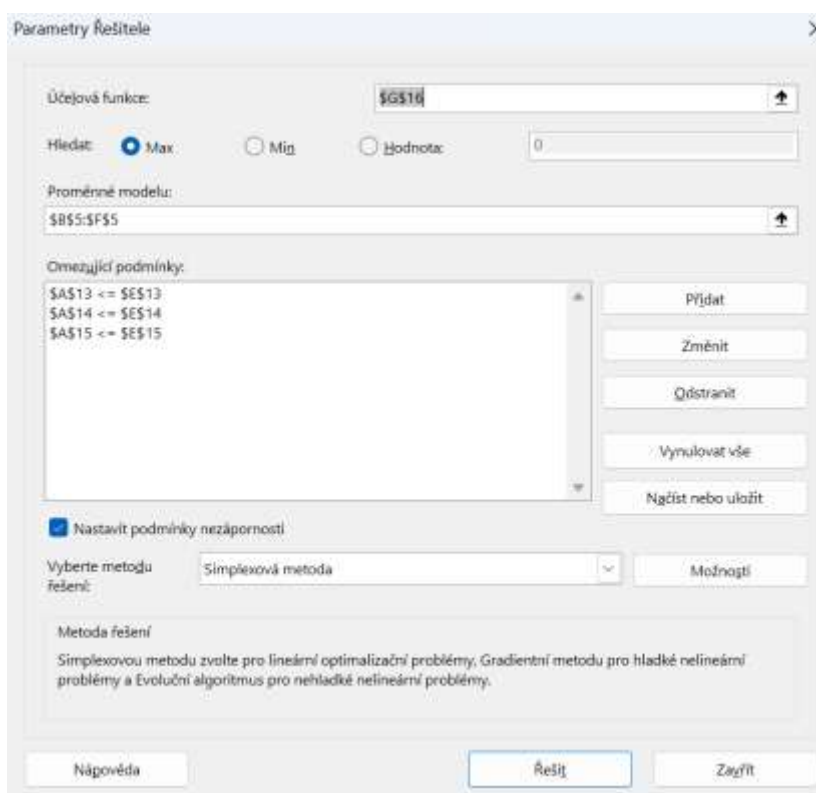
Zdroj: vlastní zpracování 2026

Spodní část tabulky zachycuje obecnou podobu omezení modelu. Každé omezení je zde reprezentováno vlastním řádkem koeficientů, například  $a_{11}, a_{12}, \dots, a_{1n}$  pro první omezení nebo  $a_{m1}, a_{m2}, \dots, a_{mn}$  pro omezení poslední. Z těchto koeficientů a z rozhodovacích proměnných  $x_1, x_2, \dots, x_n$  se skládá levá strana omezení. Ta je následně porovnána s pravou stranou omezení, označenou obecně jako  $b_i$ , pomocí relačního znaménka ( $\leq, \geq, =$ ). Frontline Systems (n.d.) uvádí, že omezení se v řešiteli zadávají právě jako vztah mezi vypočtenou levou stranou a příslušnou mezí na straně pravé.

Nejprve je tedy nutné připravit v listu všechny výpočty, které bude řešitel využívat. Do tabulky se nezadáva jen matematický zápis v symbolické podobě, ale především konkrétní struktura buněk. Tabulka 1 právě tuto logiku zachycuje v obecné podobě, tedy bez vazby na konkrétní numerický příklad. Takto připravený list představuje základ pro následné nastavení řešitele. (Microsoft, n.d.)

Po sestavení tabulky následuje zadání modelu do okna Parametry řešitele, které je zachyceno na obrázku 2. Do pole účelové funkce se vloží buňka obsahující výpočet hodnoty  $Z$ , do pole proměnných modelu oblast buněk odpovídající rozhodovacím proměnným a do části omezujících podmínek se zapisují vztahy mezi levými a pravými stranami omezení. Frontline Systems (n.d.) popisuje tento krok tak, že se nejprve vybere oblast buněk představující levou stranu omezení, následně se určí relační znaménko a poté oblast nebo hodnota pravé strany.

Obrázek 1: Parametry řešitele



Zdroj: vlastní zpracování 2026

Důležitou součástí nastavení je také volba metody řešení (pro lineární metody tedy Simplexová metoda). Postup tedy spočívá v tom, že se nejprve v pracovním listu připraví struktura odpovídající tabulce 1 a poté se její klíčové části přenesou do parametrů řešitele podle obrázku 2. Výsledkem úspěšného řešení jsou hodnoty rozhodovacích proměnných, hodnota účelové funkce a současně i možnost ověřit, zda jsou všechna omezení splněna. (Microsoft, n.d.)

Z hlediska využití je řešitel vhodný zejména pro menší a středně rozsáhlé modely, u nichž je důležitá názornost a přímá vazba mezi matematickým zápisem a tabulkovým

uspořádáním (Frontline Systems, n.d.). Pro rozsáhlejší úlohy je možné do MS excel stáhnout doplněk OpenSolver, který disponuje vyšším limitem počtu možných proměnných a relací napříč modelem.

### **Řešení v Pythonu s využitím PuLP**

Pro složitější optimalizační úlohy je vhodné použít programové řešení v jazyce Python. Výhodou tohoto přístupu je zejména to, že model lze rozdělit na vstupní data, samotnou formulaci úlohy, spuštění řešitele a následné zpracování výsledků. Oproti tabulkovému řešení tak vzniká přehlednější struktura, která je lépe použitelná i pro rozsáhlejší modely s větším počtem proměnných a omezení.

Samotná formulace modelu probíhá pomocí knihovny PuLP, která slouží k modelování úloh lineárního a smíšeného celočíselného programování v jazyce Python. Po sestavení je tento model dále předán řešiteli, který provede vlastní výpočet optimálního řešení. Využít lze řešitel CBC, tj. COIN-OR Branch and Cut, což je open-source řešič pro úlohy smíšeného celočíselného programování, jehož princip spočívá v postupném větvení úlohy na menší podproblémy a ve zpřesňování relaxace pomocí řezů. (Python Software Foundation, n.d.; PuLP Documentation team, n.d.)

Postup při sestavování modelu v tomto prostředí začíná zadáním nebo načtením vstupních dat modelu, poté je vytvořen objekt optimalizační úlohy, následně jsou zavedeny rozhodovací proměnné, zapsána účelová funkce a omezení, spuštěn řešitel, a nakonec vypsány výsledky. Tento sled kroků odpovídá běžné logice práce s knihovnou PuLP.

Níže uvedené obecné sestavení modelu bylo zpracováno na základě dokumentace PuLP a oficiální dokumentace jazyka Python (PuLP documentation team, n.d.; Python Software Foundation, n.d.).

V úvodním řádku ukázky kódu (obrázek 2) programu je načtena knihovna PuLP, která bude v dalším textu používána pod zkráceným označením `pl`, takže všechny její funkce a objekty lze zapisovat stručněji a přehledněji. Tento krok je nezbytný, protože právě prostřednictvím knihovny PuLP jsou v programu vytvářeny objekty modelu, rozhodovací proměnné, účelová funkce i omezení.

Obrázek 2: Ukázka kódu 1

```
1 import pulp as pl
```

V další části kódu (obrázek 3) jsou zadána vstupní data modelu v obecné podobě, kdy seznam *c* obsahuje koeficienty účelové funkce, matice *A* obsahuje koeficienty jednotlivých omezení a seznam *b* představuje pravé strany těchto omezení. Současně jsou pomocí funkcí `len(c)` a `len(b)` automaticky určeny počty rozhodovacích proměnných a omezení, což umožňuje, aby se další části programu přizpůsobily rozsahu zadaných dat bez nutnosti ruční změny počtu indexů v kódu.

Obrázek 3: Ukázka kódu 2

```
3 # 1. VSTUPNÍ DATA
4 # koeficienty účelové funkce
5 c = [12, 8, 15]
6 # matice koeficientů omezení
7 A = [
8     [2, 1, 3],
9     [1, 2, 1],
10    [0, 1, 2]
11 ]
12 # pravé strany omezení
13 b = [100, 80, 40]
14 # počet proměnných a počet omezení
15 n = len(c)
16 m = len(b)
```

Příkazem uvedeným v obrázku 4 je vytvořen samotný optimalizační model jako objekt třídy `LpProblem`, přičemž první argument určuje jeho název a druhý argument stanovuje, že cílem úlohy bude maximalizace. V programu tak vzniká základní objekt, do něhož budou v dalších krocích postupně přidávány rozhodovací proměnné, účelová funkce a omezení.

Obrázek 4: Ukázka kódu 3

```
18 # 2. VYTVORENÍ MODELU
19 model = pl.LpProblem("Obecny_model", pl.LpMaximize)
```

V další části kódu (obrázek 5) jsou zavedeny rozhodovací proměnné modelu prostřednictvím funkce `LpVariable.dicts`, která vytvoří celou množinu proměnných najednou a uloží je do slovníkové struktury. Základní název proměnných je určen řetězcem „x“, indexy proměnných jsou vytvořeny pomocí `range(1, n + 1)`, takže program pracuje s proměnnými  $x_1, x_2, \dots, x_n$ , parametr `lowBound=0` zajišťuje jejich nezápornost a parametr `cat="Continuous"` určuje, že jde o spojité proměnné, tedy proměnné, které mohou nabývat i neceločíselných hodnot.

Obrázek 5: Ukázka kódu 4

```
22 x = pl.LpVariable.dicts(
23     "x",
24     range(1, n + 1),
25     lowBound=0,
26     cat="Continuous")
```

V ukázce kódu na obrázku 6 je do modelu přidána účelová funkce, která je zapsána jako součet součinů koeficientů účelové funkce a příslušných rozhodovacích proměnných. Funkce `lpSum(...)` vytváří lineární součet jednotlivých členů, zatímco výraz `c[j - 1] * x[j]` zajišťuje propojení koeficientu uloženého v seznamu `c` s odpovídající proměnnou `x[j]`. Použití `j - 1` je dáno tím, že Python indexuje seznamy od nuly, zatímco samotné proměnné jsou kvůli srozumitelnosti číslovány od jedničky. Druhý argument "Ucelova\_funkce" slouží pouze jako pojmenování této části modelu.

Obrázek 6: Ukázka kódu 5

```
29 # 4. ÚČELOVÁ FUNKCE
30 # max Z = c1*x1 + c2*x2 + ... + cn*xn
31 model += pl.lpSum(c[j - 1] * x[j] for j in range(1, n + 1)), "Ucelova_funkce"
```

V této části kódu (Obrázek 7) jsou do modelu postupně doplněna všechna omezení, přičemž smyčka `for i in range(m)` zajišťuje, že se stejný princip aplikace omezení provede pro každý řádek matice `A`. Uvnitř smyčky je levá strana každého omezení vytvořena jako lineární součet součinů koeficientů z příslušného řádku matice `A` a rozhodovacích proměnných `x[j]`, zatímco pravá strana omezení je převzata ze seznamu `b`. Výsledkem je tedy automatické vytvoření celé soustavy omezení ve tvaru  $Ax \leq b$ , přičemž každé omezení dostává i vlastní textové označení ve tvaru `Omezeni_1`, `Omezeni_2` a podobně.

Obrázek 7: Ukázka kódu 6

```
33 # 5. OMEZENÍ
34 # A*x <= b
35 for i in range(m):
36     model += (
37         pl.lpSum(A[i][j - 1] * x[j] for j in range(1, n + 1)) <= b[i],
38         f"Omezeni_{i + 1}"
39     )
```

Pro řešení je zde nejprve vytvořen objekt `resic` (obrázek 8), který představuje nastavení řešitele CBC, jenž bude použit k výpočtu optimálního řešení modelu, parametr `msg=False` současně určuje, že se při výpočtu nebudou do konzole vypisovat podrobné technické informace o průběhu řešení. Následný příkaz `model.solve(resic)` pak předá sestavený model solveru CBC, který provede vlastní optimalizační výpočet a uloží získané výsledky zpět do objektu modelu i do jednotlivých proměnných.

Obrázek 8: Ukázka kódu 7

```
40 # 6. ŘEŠENÍ
41 resic = pl.PULP_CBC_CMD(msg=False)
42 model.solve(resic)
```

Závěrečná část programu (obrázek 9) slouží k výpisu výsledků řešení, přičemž nejprve je zobrazen stav řešení modelu (například zda bylo nalezeno optimální řešení), dále je vypsána číselná hodnota účelové funkce a nakonec i konkrétní hodnoty jednotlivých rozhodovacích proměnných. Výraz `pl.LpStatus[model.status]` převádí interní číselný kód stavu modelu na textovou podobu, funkce `pl.value(model.objective)` vrací hodnotu účelové funkce po vyřešení modelu a atribut `x[j].varValue` obsahuje výslednou optimální hodnotu příslušné proměnné.

Obrázek 9: Ukázka kódu 8

```
44 # 7. VÝPIS
45 print("Status:", pl.LpStatus[model.status])
46 print("Hodnota účelové funkce:", pl.value(model.objective))
47 for j in range(1, n + 1):
48     print(f"x{j} =", x[j].varValue)
```

Knihovna PuLP představuje vhodný prostředek pro programovou implementaci optimalizačních modelů, protože umožňuje přehledně převést matematický zápis úlohy do kódu.

Teoretická východiska vymezila základní principy rozhodovacího procesu, matematického modelování i vybrané optimalizační metody, které lze při řešení podnikových problémů využít. Pro jejich praktické uplatnění je však nejprve nutné zasadit řešenou úlohu do konkrétního provozního prostředí. Následující kapitola proto představuje podnik Prefab Factory a.s. a zaměřuje se na ty souvislosti jeho fungování, které jsou podstatné pro formulaci optimalizačního modelu.

## 4 Podnik Prefab Factory a.s.

Společnost Prefab Factory a.s. je výrobní podnik zaměřený na výrobu prefabrikovaných koupelnových modulů. Sídlo společnosti je v Horní Bříze, ale výrobní závod byl v současné době přesunut do Přehýšova u Plzně. Výroba probíhá v závodě o ploše přes 6 000 m<sup>2</sup> a na výrobě se podílí více než 100 odborných řemeslníků. Podnik je postaven na sériové výrobě koupelnových modulů, které vznikají jako hotové celky a na stavbu se dopravují už dokončené. (Prefab Factory a.s., n.d.)

Výrobní program podniku lze zjednodušeně rozdělit do tří hlavních skupin modulů, a to na katalogové, typové a atypické. Toto členění je důležité nejen z hlediska samotné výroby, ale i z pohledu přípravy dokumentace, plánování kapacity a celkové náročnosti zakázky.

První skupinu tvoří katalogové moduly. U těchto modulů má podnik již hotové a ověřené technické řešení i výrobní dokumentaci. Výroba je proto jednodušší, rychlejší a celkově méně problematická, protože zde nevzniká potřeba znovu řešit základní konstrukční otázky nebo upravovat podklady pro výrobu. Katalogové moduly tedy představují nejvíce standardizovanou část výrobního programu.

Druhou skupinou jsou typové moduly. Ty sice nejsou zcela totožné jako katalogové moduly, ale lze je zařadit do skupin, v nichž si jsou moduly napříč různými zakázkami v řadě ohledů podobné. Podobnost se může týkat například dispozice, monolitické desky, základního konstrukčního řešení nebo technického uspořádání. Podnik tak může při jejich přípravě do určité míry využívat již existující dokumentaci a dřívější řešení, která se upraví pro konkrétní projekt. Výroba těchto modulů je tedy náročnější než u modulů katalogových, ale stále zde existuje určitý stupeň opakovatelnosti a možnost navázat na dříve zpracované podklady.

Poslední skupinu tvoří atypické moduly. U nich již podnik nemůže ve větší míře vycházet z hotových řešení a většina přípravy probíhá od začátku. Je nutné nově řešit konstrukční uspořádání, technické detaily i související dokumentaci, což tuto skupinu činí nejnáročnější z hlediska přípravy i výroby. Atypické moduly jsou tak spojeny s vyšší pracností, vyšší nejistotou a menší možností využití předchozích zkušeností v přímé podobě.

## **Podnikové procesy**

Celý proces začíná obdržením poptávky od zákazníka. Podkladem bývá studie nebo projektový návrh, podle kterého se posuzuje počet modulů, jejich rozmístění v objektu a technické vazby na stavbu. V této chvíli se řeší, do jaké míry lze projekt převést do podoby standardních typových nebo až katalogových modulů. Smyslem je dostat návrh co nejbližší typovému řešení, protože opakovatelná výroba je jednodušší na přípravu, lépe se plánuje a při vyšších objemech se zefektivňuje. Pokud dispozice projektu dovoluje využít typové moduly, upravuje se řešení tak, aby bylo možné co nejvíce koupelen nebo WC osadit standardními variantami. Součástí návrhu bývá i sjednocování technických detailů, například stoupačkových jader, revizních a montážních otvorů nebo dalších konstrukčních prvků. Když takové úpravy nejsou možné nebo je zákazník nechce přijmout, zůstává část projektu v atypickém řešení. V této fázi se tedy rozhoduje o tom, jaký podíl zakázky bude založen na katalogových a typových modulech a jaký podíl bude řešen atypicky.

V jednom projektu se zpravidla pracuje se čtyřmi až osmi variantami modulů. Jeden z modulů je pak vyroben jakožto vzorový, jehož výroba trvá šest až osm týdnů. Vzorový modul následně slouží jako odsouhlasený standard provedení pro zbytek projektu. Tím se sjednocuje kvalita, technické řešení a snižuje se riziko, že se jednotlivé kusy budou v průběhu výroby rozcházet.

Ve chvíli, kdy je způsob řešení zakázky odsouhlasen se přechází ke zpracování technické dokumentace. Vznikají výrobní výkresy, instalační plány a další podklady, podle nichž pak běží samotná výroba. Tato fáze je důležitá, protože se zde projekt převádí z dispozičního a architektonického návrhu do podoby, podle které lze modul skutečně vyrobit. Bez dokončení celé technické dokumentace nelze navázat dalšími výrobními kroky.

Vlastní výroba koupelen začíná externím objednááním nebo vlastním zhotovením monolitické desky podle projektové dokumentace. Tato dokumentace obsahuje také vymodelovaný objekt ve speciálním programu Arkitech Software, který je přímo vytvořen pro CNC (computer numerical control) stroj Arkitech framer. Jedná se o integrované řešení návrhu a výroby produktů z lehké lisované oceli. CNC stroj podle předlohy sestaví jednotlivé vertikální a horizontální U profily, které následně slouží jako stavební prvky pro montáž stěn jako celků. Stěny se následně lepí a kotví na monolit.

Hotový skelet prochází kontrolou rozměrové přesnosti a kvality, protože právě přesnost konstrukce ovlivňuje všechny další navazující práce. Když je základní nosná část připravena, vzniká prostorový celek, na který se navazuje dalšími instalacemi.

Po dokončení nosné části se do modulu vedou rozvody elektřiny, vody a odpadu. Následují kontroly a zkoušky, aby bylo jisté, že je technická část správně připravená. Potom přichází vyplnění pěnou jakožto tepelnou a zvukovou izolací a následují další uzavírání modulu. Výroba se tak postupně posouvá od základní konstrukce k vnitřnímu technickému vybavení.

Další fází tvoří obklady, dlažby a osazení sanitárního vybavení. Do modulu se montují umyvadla, baterie, sprchové kouty nebo vany, radiátory, zrcadla a další prvky podle konkrétní varianty. U některých řešení se přidává i kuchyňská příprava nebo další technické části navázané na provoz bytu.

Obrázek 10: Hotový prefabrikovaný modul



Zdroj: vlastní zpracování 2026

Na konci výroby se modul znovu zkontroluje, vyčistí a připraví k expedici. Každý modul je provázán s elektronickou dokumentací a výsledkem je hotový výrobek připravený k transportu a osazení na stavbě. Dodávku a instalaci zajišťuje podnik sám. Moduly se osazují podle typu stavby buď po patrech vertikálně, nebo přes otvory ve fasádě s horizontálním přesunem. Celý proces tedy nekončí dokončením výroby na hale, ale až ve chvíli, kdy je modul dopraven a správně osazen montážním týmem.

Provoz podniku je omezen kapacitou výroby i montáže. Výrobní kapacita v průměru činí 10 modulů za pracovní den. Optimální montážní kapacita je 6 modulů za den při jednom montážním týmu nebo 12 modulů za den při dvou týmech, které pracují paralelně. Současně může v jednom období probíhat více projektů, takže je nutné plánovat výrobu s předstihem a sladit ji s harmonogramy staveb. Kapacitní tlak proto nevzniká jen ve výrobní hale, ale už při projektové přípravě, při koordinaci variant, při samotné výrobě i při návaznosti na dopravu a montáž.

Na zatížení podniku má přímý vliv skladba zakázek. Typové moduly jsou opakovatelnější, lépe se sjednocují a při větších sériích se jejich výroba zefektivňuje. Atypické moduly naopak zvyšují nároky na přípravu, koordinaci i samotné provedení. Vedle počtu zakázek je proto rozhodující i jejich struktura, tedy poměr katalogových, typových a atypických modulů, počet variant v projektu a technická náročnost jednotlivých řešení.

### **Doprava modulů na stavbu**

Expedice prefabrikovaných modulů představuje v podniku samostatný logistický problém, který je úzce svázán s postupem realizace stavby. Hotové moduly nejsou na místo určení dopravovány jednorázově pro celý objekt, ale postupně podle montážní připravenosti nosné konstrukce a aktuální fáze výstavby. Dodávka proto obvykle směřuje pouze do té části objektu, kterou je možné po příjezdu modulů bezprostředně osadit.

Doprava modulů je pevně provázána se způsobem jejich montáže. V podniku probíhá instalace buď po jednotlivých patrech vertikálně, nebo po dokončení skeletu stavby horizontálně skrz připravené otvory ve fasádě. Logistika proto nezahrnuje pouze samotný převoz z výrobního závodu na stavbu, ale také přesné časování dodávky, návaznost na stavební připravenost objektu, dostupnost manipulační techniky a koordinaci s montážní etapou. U horizontální instalace je navíc nutné respektovat technické řešení objektu a

přípravenost fasádních otvorů, protože moduly jsou osazovány přímo do jednotlivých pater budovy.

Krátkodobé plánování dopravy je proto vhodné chápat jako operativní rozhodování v horizontu několika dnů až jednoho či dvou týdnů. V tomto časovém úseku se neřeší zásobení celé stavby, ale pouze expedice takového množství modulů, které odpovídá právě realizované montážní etapě. Celkový projekt může po dokončení zahrnovat desítky nebo i nižší stovky modulů, v krátkodobém horizontu je však relevantní jen dílčí objem odpovídající konkrétní části objektu, například jednomu podlaží nebo jeho ucelené sekci.

Význam dopravy podtrhuje i územní rozsah realizací podniku. Prefab Factory v současnosti zajišťuje dodávky modulů pro stavby například v Plzni, Kladně a v několika pražských lokalitách jako jsou Roztyly, Vackov, Pitkovice nebo Kamýk. Současně realizuje také projekt Nuppu Magnolia 4 a 5 v Bratislavě. Z toho je zřejmé, že podniková logistika není omezena pouze na místní nebo regionální rozvoz, ale zahrnuje i delší přepravní relace, u nichž roste význam přesného plánování návaznosti mezi výrobou, dopravou a montáží.

## 5 Model optimalizace výběru zakázek při omezené výrobní kapacitě

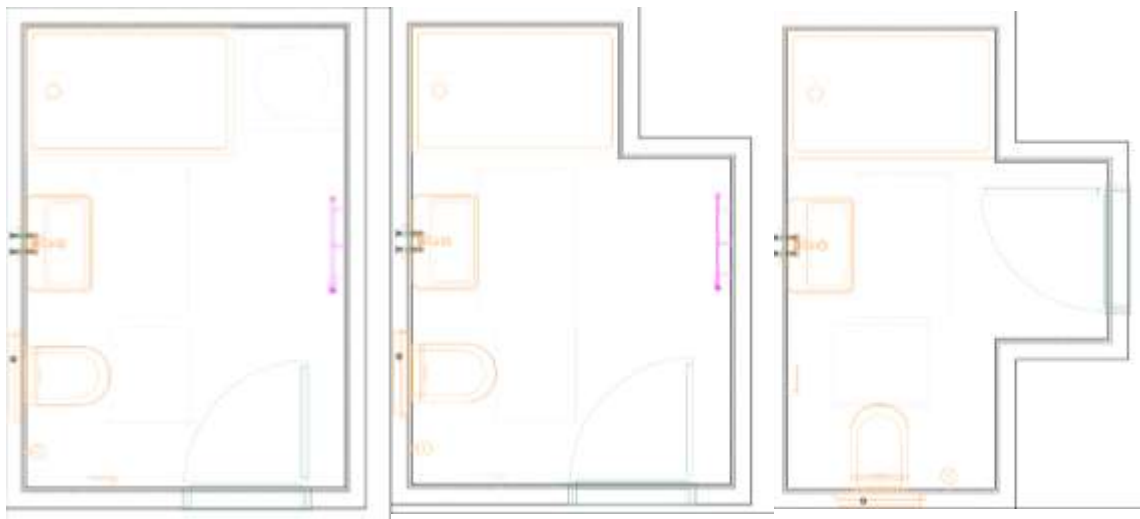
Z předchozího popisu podnikových procesů je patrné, že podnik musí v každém plánovacím období vyvažovat omezené výrobní možnosti s aktuální poptávkou. Zakázky se přitom liší nejen očekávaným ekonomickým přínosem, ale i svojí strukturou a náročností z hlediska realizace. Rozhodování proto nelze redukovat na posouzení jednotlivých zakázek odděleně, ale je třeba uvažovat jejich vzájemnou kombinaci v rámci celkových kapacitních možností podniku. Právě tato rozhodovací situace je vhodná pro zachycení pomocí matematického modelu.

Podnik v každém plánovacím období rozhoduje, které zakázky přijme k realizaci. Rozhodnutí se musí řídit hlavně faktorem omezené výrobní kapacity, která se však nemůže vyjádřit pomocí vyrobených kusů, tedy jeden vyrobený modul není rovný jedné spotřebované kapacitní jednotce. Kapacita se odvíjí od faktu, že je podnik schopný vyrobit 10 průměrných modulů denně, kde se průměrným myslí určitý střed obtížnosti, z pohledu výroby. Za průměrně obtížný modul je zde považován koupelnový modul střední velikosti, který se vyrábí poprvé a jeho zhotovení nedoprovází komplikace v podobě obtížného půdorysného tvaru či složitých doplňkových prací (např. typový modul B8). Časové období, které je v tomto případě sledováno, pracuje s disponibilní kapacitou 520 jednotek, což odpovídá 52 pracovním dnům, které při šestidenním pracovním týdnem odpovídají zhruba dvěma kalendářním měsícům. Každý jednotlivý typ modulu má součinitel kapacitního vytížení  $\lambda$ , který vyjadřuje náročnost vzhledem k referenčnímu modulu viz Tabulka 3.

Moduly jsou rozřazeny do tří kategorií, a to na katalogové typu A, typové s označením B a atypické C. Podnik nabízí 4 druhy katalogových modulů, typové moduly rozřazuje do 17 typů a atypické do 4 druhů. Členění modulů do těchto skupin přitom vychází především z jejich velikosti a funkce. Katalogové moduly představují čtyři základní koupelnové varianty odlišných velikostí, které tvoří nejvíce standardizovanou část výrobního programu. U atypických modulů jsou v modelu sledovány čtyři typy, které vznikají kombinací velikosti a obtížnosti, a zachycují tak rozdílnou náročnost těchto řešení z hlediska přípravy i výroby. Typové moduly pak představují různé kombinace

velikosti, dispozičního uspořádání a dalších konstrukčních či technických prvků, které mají vliv na jejich výrobní náročnost.

Obrázek 11: Schématické zobrazení půdorysu katalogového, typového a atypického modulu



Zdroj: vlastní zpracování 2026

Součinitel  $\lambda$  je pro moduly typu A konstantní, jelikož se jedná o identickou výrobu stejných modulů, kde odpadá pracnost zejména v odvětví projektování a také je zde minimální riziko vzniku chyb, které brzdí plynulost provozu. (Součinitel  $\lambda$  pro moduly A lze pomyslně chápat, jakožto automatické umístění do šestého objemového pásma pro moduly B viz. další text). Pro typové moduly B je díky jejich obdobným prvkům zaveden pásmový systém (šest pásem), který zvýhodňuje vyšší objem určitého modulu napříč všemi zakázkami. Součinitel  $\lambda$  pro moduly B je pak základ, který může být další výrobou snižován, dostane-li se objem do vyšších pásem viz Tabulka 5. Atypické moduly s označením C jsou pak díky využití vyšší hodnoty součinitele  $\lambda$  znevýhodněny, protože u nich odpadá jakákoliv sériovost a objevují se u nich konstrukčně hůře zvládnutelné prvky.

Z tabulky 3 lze dále vyčíst jednotkovou marži každého modulu, tedy jakým dílem přispívá k celkovému zisku podniku. Rozhodování o přijetí zakázek však není možné založit pouze na marži očekávané z výroby daných modulů. Jednotlivé zakázky se liší nejen počtem požadovaných modulů, ale také jejich strukturou, tedy zastoupením katalogových, typových a atypických modulů. Přitom neplatí, že vyšší pracnost modulu automaticky znamená i vyšší marži, ani že modul s vyšší marží musí být pro podnik při

omezené kapacitě vždy výhodnější. Ekonomický přínos a kapacitní náročnost proto musí být posuzovány současně.

Do rozhodování vstupují další parametry, které proces ovlivňují. Základem je samozřejmě matice požadavků jednotlivých zákazníků (Tabulka 2). Poptávka je zpravidla z největší části tvořena typovými moduly, které jsou doplněny o katalogové v místech, kde to zákazníkovi vyhovuje a atypickými v místech, kde má zákazník speciální požadavek, nebo to konstrukční řešení stavby nedovolí jinak. Na tuto skutečnost navazuje fakt, že se atypické řešení často využívá pro první či poslední patro stavby, které je často konstrukčně a typologicky řešeno jinak. V určitých případech je pak možné rozdělit zakázku na opakovatelnou část, tedy na katalogovou a typovou, a na část atypickou, která se poté nemusí dodat. Výsledkem je přijetí pouze té části zakázky, která neobsahuje atypické moduly. Toto řešení je však ojedinělé a ve velkém závisí na komunikaci podniku a zákazníka. V tabulce 4 je patrné, k jakým zakázkám je takto možné přistupovat. Ve stejné tabulce je pak také dán parametr určující prioritu zakázek. Taková zakázka musí být splněna. Tento vztah vychází z obchodních vztahů podniku a zákazníka, například díky dlouhodobé spolupráci, nebo již smluvně uzavřené rámcové dohodě.

S ohledem na ochranu citlivých obchodních údajů byly některé vstupní hodnoty prezentované v ilustrativním příkladu aplikace modelu anonymizovány a částečně upraveny (zejména názvy zákazníků a marže jednotlivých modulů). Cílem těchto úprav bylo zamezit možnosti identifikace zákazníků a současně zabránit tomu, aby zveřejněné údaje mohly některý ze subjektů jakýmkoli způsobem ovlivnit nebo znevýhodnit.

### **Stávající praxe**

V podniku je přijetí zakázek v současnosti posuzováno bez využití formalizovaného optimalizačního postupu, a to především na základě zkušenosti a odborného úsudku odpovědných pracovníků. Prioritní zakázky jsou přijímány automaticky, zatímco zbylé zakázky jsou hodnoceny zejména podle očekávané celkové marže podniku a orientační výrobní náročnosti ve vztahu k disponibilní kapacitě plánovacího období. Takový postup je v praxi rychlý a snadno použitelný, avšak zůstává založen na zjednodušeném posouzení zakázek a nezohledňuje v plném rozsahu všechny souvislosti, které ovlivňují využití výrobní kapacity i celkový ekonomický výsledek.

### 5.1.1 Formulace zadání problému

Rozhodování o výběru zakázek lze jako příklad formulovat následujícím způsobem:

Je dána množina 30 zakázek, o jejichž přijetí podnik rozhoduje v rámci plánovacího období o délce 52 pracovních dnů. Při uvažovaném provozu v režimu šesti pracovních dnů týdně odpovídá tento horizont přibližně dvěma kalendářním měsícům. Každá zakázka obsahuje požadavky na určitý počet prefabrikovaných modulů jednotlivých typů A, B a C. Požadavky zakázek na konkrétní moduly jsou uvedeny v tabulce 2, ekonomické a kapacitní parametry modulů v tabulce 3, priorita zakázek a možnost neplnění atypické části v tabulce 4 a definice objemových pásem pro moduly typu B v tabulce 5. Výroba je v tomto období omezena disponibilní kapacitou  $K = 520$ , což při průměrné výrobní kapacitě 10 průměrně náročných modulů denně odpovídá zvolenému plánovacímu horizontu. Kapacitní náročnost modulů přitom závisí na jejich typu a u modulů typu B také na objemu výroby. Úkolem je rozhodnout, které zakázky budou přijaty, případně zda bude realizována jejich atypická část, tak aby nebyla překročena disponibilní kapacita a současně byla maximalizována celková marže podniku.

#### Neřiditelné vstupy modelu

$m_{a,b,c}$  ... jednotková marže modulů ze skupin A, B, C

$\lambda_{a,b,c}$  ... jednotkové kapacitní vytížení modulů ze skupin A, B, C

$n_{ia}$  ... poptávka po modulu  $A_a$  v  $i$ . zakázce

$n_{ib}$  ... poptávka po modulu  $B_b$  v  $i$ . zakázce

$n_{ic}$  ... poptávka po modulu  $C_c$  v  $i$ . zakázce

$d_i = \begin{cases} 1, & \text{atypickou část } i. \text{ zakázky lze nedodat} \\ 0, & \text{atypická část } i. \text{ zakázky musí být dodána} \end{cases}$

$g_i = \begin{cases} 1, & i. \text{ zakázka má prioritu a je nutné ji plnit} \\ 0, & i. \text{ zakázka může být odmítnuta} \end{cases}$

Tabulka 2: Poptávané množství modulů (zakázky jsou v řádcích a moduly ve sloupcích)

		typ modulu																									
		A1	A2	A3	A4	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	C1	C2	C3	C4	
zakázka	Z_1	11	11	0	0	0	0	0	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	5	0
	Z_2	0	0	0	0	0	1	11	0	0	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Z_3	2	2	4	4	0	0	0	0	5	6	0	0	0	5	0	0	5	0	0	0	0	0	0	1	0	0
	Z_4	0	0	0	0	0	0	1	0	1	0	0	0	0	0	3	5	1	7	0	5	1	1	0	0	0	1
	Z_5	0	0	0	0	0	0	1	1	0	0	3	6	2	0	0	0	1	2	0	5	0	4	0	0	1	0
	Z_6	4	0	0	0	4	0	0	0	8	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	0	0
	Z_7	6	0	0	2	0	0	0	0	0	0	0	0	0	2	1	0	0	1	0	6	0	3	1	2	0	0
	Z_8	2	0	0	1	0	5	0	5	0	8	0	1	0	0	4	0	0	0	0	0	0	1	1	0	0	0
	Z_9	0	0	0	0	2	8	4	7	0	0	0	1	0	0	0	0	0	0	0	3	1	0	1	0	0	0
	Z_10	0	0	0	0	0	7	0	0	0	0	0	0	0	0	1	0	0	7	0	0	0	0	7	4	0	0
	Z_11	0	0	0	0	0	1	0	1	6	5	1	0	0	2	0	0	1	5	0	0	1	0	0	1	0	0
	Z_12	6	0	0	3	0	0	0	3	0	8	0	0	3	0	0	0	0	2	0	0	0	0	0	2	0	0
	Z_13	0	0	0	0	3	0	2	0	0	0	1	0	0	2	1	1	6	0	6	0	2	0	1	0	0	0
	Z_14	0	0	0	0	5	2	6	0	0	0	0	5	1	0	1	5	0	0	0	3	0	0	1	0	0	0
	Z_15	0	0	0	0	0	0	0	7	0	0	0	0	0	0	1	0	8	4	1	0	0	0	2	0	0	0
	Z_16	0	0	0	0	0	2	0	0	8	0	0	0	5	0	1	0	4	0	0	0	4	0	3	0	1	0
	Z_17	0	0	0	0	0	0	0	0	0	3	8	0	0	0	7	0	0	0	0	0	0	0	0	3	0	1
	Z_18	0	4	2	0	0	2	1	0	1	2	1	0	0	0	0	0	0	0	0	6	0	3	0	0	2	0
	Z_19	0	2	0	3	0	0	2	0	0	6	6	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
	Z_20	0	0	0	0	0	0	0	0	0	0	0	2	0	0	7	0	0	6	0	0	0	0	2	0	0	4
	Z_21	0	0	0	0	0	0	0	0	5	0	11	0	0	0	7	0	0	0	0	0	0	0	0	6	0	0
	Z_22	0	1	0	0	0	0	2	0	4	2	0	1	1	0	0	0	6	3	0	0	3	1	0	0	0	0
	Z_23	0	0	0	0	1	0	0	1	0	0	0	4	0	0	11	0	0	3	5	0	0	0	3	0	0	0
	Z_24	0	0	0	0	0	11	0	10	9	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0
	Z_25	0	0	0	0	0	0	0	5	3	0	0	7	0	0	0	0	0	2	0	0	4	0	0	5	0	0
	Z_26	0	0	0	0	0	10	0	0	1	0	0	0	0	0	5	0	0	0	0	0	0	0	5	0	7	0
	Z_27	0	0	0	0	5	0	0	0	2	0	0	3	0	0	2	6	0	5	0	0	1	1	0	0	0	0
	Z_28	6	0	3	0	0	0	0	0	0	2	0	0	0	4	0	5	0	0	0	0	0	0	0	0	0	4
	Z_29	0	0	0	0	1	0	0	5	0	7	0	0	0	0	0	0	5	6	0	2	6	0	0	1	0	0
	Z_30	0	0	0	0	0	0	3	3	1	0	0	4	0	1	0	3	0	0	0	7	0	3	3	0	0	0

Zdroj: vlastní zpracování 2026

Tabulka 3 - přehled ekonomických a kapacitních parametrů pro jednotlivé moduly

	<i>A1</i>	<i>A2</i>	<i>A3</i>	<i>A4</i>	<i>B1</i>	<i>B2</i>	<i>B3</i>	<i>B4</i>	<i>B5</i>	<i>B6</i>	<i>B7</i>	<i>B8</i>	<i>B9</i>
Marže $m_{a,b,c}$ (Kč)	155000	190000	222500	272500	119000	124100	129200	134300	141100	146200	151300	156400	161500
pásma úspor	0	0	0	0	1	1	1	1	1	1	1	1	1
kapacitní vytížení $\lambda$	0,525	0,5625	0,75	0,9	0,7	0,7	0,7	0,75	0,75	0,75	0,75	1	1
	<i>B10</i>	<i>B11</i>	<i>B12</i>	<i>B13</i>	<i>B14</i>	<i>B15</i>	<i>B16</i>	<i>B17</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	
Marže $m_{a,b,c}$ (Kč)	166600	173400	183600	187000	195500	202300	209100	215900	139500	157500	177000	198000	
pásma úspor	1	1	1	1	1	1	1	1	0	0	0	0	
kapacitní vytížení $\lambda$	1	1	1,15	1,15	1,2	1,2	1,2	1,2	0,75	1	1,15	1,2	

Zdroj: vlastní zpracování 2026

V tabulce 4 lze pozorovat parametr  $g_i$ , který určuje prioritu zakázky, kde  $g_i = 1$  znamená, že se zakázka musí přijmout. Dále tabulka 4 obsahuje parametr  $d_i$ , kde  $d_i = 1$  umožňuje rozložení zakázky a separaci atypické části, která se pak nemusí přijmout k výrobě.

Tabulka 4 - přehled priorit zakázek a možnosti neplnění atypické části zakázky

	<i>Z_1</i>	<i>Z_2</i>	<i>Z_3</i>	<i>Z_4</i>	<i>Z_5</i>	<i>Z_6</i>	<i>Z_7</i>	<i>Z_8</i>	<i>Z_9</i>	<i>Z_10</i>	<i>Z_11</i>	<i>Z_12</i>	<i>Z_13</i>	<i>Z_14</i>	<i>Z_15</i>
priorita zakázek $g_i$	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0
možnost neplnění atypické části $d_i$	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0
	<i>Z_16</i>	<i>Z_17</i>	<i>Z_18</i>	<i>Z_19</i>	<i>Z_20</i>	<i>Z_21</i>	<i>Z_22</i>	<i>Z_23</i>	<i>Z_24</i>	<i>Z_25</i>	<i>Z_26</i>	<i>Z_27</i>	<i>Z_28</i>	<i>Z_29</i>	<i>Z_30</i>
priorita zakázek $g_i$	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
možnost neplnění atypické části $d_i$	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Zdroj: vlastní zpracování 2026

Tabulka 5: Přehled kapacitních pásem pro moduly typu B

pásmo	rozsah v ks	kapacitní součinitel
1	1 až 10	1,00
2	11 až 20	0,95
3	21 až 30	0,90
4	31 až 40	0,85
5	41 až 50	0,80
6	51 a více	0,75

Zdroj: vlastní zpracování 2026

### Příklad využití kapacitních pásem

Uvažujme typový modul, jehož celkový objem výroby dosáhne 27 kusů. Bez použití pásem by se celý objem posuzoval jednotně, takže by výsledná hodnota byla rovna  $27 \lambda$ . Při využití kapacitních pásem však prvních 10 kusů spadá do 1. pásma se součinitelem 1,00, dalších 10 kusů do 2. pásma se součinitelem 0,95 a zbývajících 7 kusů do 3. pásma se součinitelem 0,90. Výsledná hodnota po zohlednění pásem je tedy  $10 \cdot 1,00 + 10 \cdot 0,95 + 7 \cdot 0,90 = 25,8$ . Konečné kapacitní využití daného modulu by pak bylo dáno až součinem této výsledné hodnoty a příslušného součinitele  $\lambda$ , tedy  $25,8 \cdot \lambda$ . Z příkladu je patrné, že zavedení pásem snižuje výslednou kapacitní náročnost.

### Indexy modelu

$i \in \{1; 2; 3; \dots; 30\}$  kde  $i$  označuje zakázku

$a \in \{1; 2; 3; 4\}$  kde  $a$  označuje katalogový modul skupiny A

$b \in \{1; 2; 3; 4; \dots; 17\}$  kde  $b$  označuje typový modul skupiny B

$c \in \{1; 2; 3; 4\}$  kde  $c$  označuje atypický modul skupiny C

$k \in \{1; 2; 3; 4; 5; 6\}$  kde  $k$  označuje typ pásma

### Hlavní rozhodovací proměnné

$x_i$  ... binární proměnná vyjadřující, zda bude zakázka  $i$  přijata k realizaci

$$x_i = \begin{cases} 1, & \text{pokud je zakázka } i \text{ přijata} \\ 0, & \text{pokud je zakázka } i \text{ odmítnuta} \end{cases}$$

$u_i$  ... binární proměnná vyjadřující, zda bude atypická část zakázky  $i$  realizována

$$u_i = \begin{cases} 1, & \text{pokud se atypická část } i \text{ zakázky realizuje} \\ 0, & \text{pokud je atypická část } i \text{ odmítnuta} \end{cases}$$

### Účelová funkce modelu

Účelová funkce sčítá marži přes tři druhy modulů, první výraz popisuje marži z výroby katalogových modulů typu A, druhý výraz marži z typových modulů s označením B a poslední výraz udává marži z výroby atypických modulů C.

$$\max Z = \sum_{i=1}^{30} \sum_{a=1}^4 m_a n_{ia} x_i + \sum_{i=1}^{30} \sum_{b=1}^{17} m_b n_{ib} x_i + \sum_{i=1}^{30} \sum_{c=1}^4 m_c n_{ic} u_i \quad (20)$$

Zatímco u typů A a B se marže přičítá při přijetí zakázky  $x_i = 1$ . U atypických modulů se využívá proměnná  $u_i$ , která určuje, jestli je atypická část skutečně realizována.

### Omezující podmínky modelu

Podmínky modelu zahrnují vazbu mezi přijetím zakázky a realizací příslušné atypické části, výpočet celkových objemů jednotlivých modulů, rozdělení typových modulů do objemových pásem a příslušnou logiku návaznosti pásem.

### Vazba mezi přijetím typové a atypické části

Zde musí platit, že přijetí atypické části může být nejvýše rovno přijetí hlavní části zakázky. Atypickou část tedy nelze realizovat bez přijetí katalogové a typové části zakázky.

$$u_i \leq x_i \quad \forall i = 1, 2, \dots, 30 \quad (21)$$

### Možnost neplnění atypické části

Pro šest zakázek je povolena možnost neplnění atypické části poptávky pomocí vstupu  $d_i$ . Vztahu (22) sleduje, že pokud  $d_i=0$  (nutnost dodání atypické části), vznikne díky vztahu (21) rovnost mezi  $x_i$  a  $u_i$ . Pokud je  $d_i=1$ , pak vzniká možnost, kde  $u_i = 0$ , ačkoliv  $x_i = 1$ .

$$u_i \geq x_i - d_i \quad \forall i = 1, 2, \dots, 30 \quad (22)$$

### Prioritní zakázky musí být splněny

Mezi zakázkami se nachází čtyři prioritní, které je nutné splnit. Pokud je binární vstup  $g_i$  roven jedné, musí být rovna jedné také proměnná  $x_i$  určující plnění zakázky.

$$x_i \geq g_i \quad \forall i = 1, 2, \dots, 30 \quad (23)$$

### Celkový počet modulů

Moduly jsou rozdělené do tří kategorií (A, B, C), kde každá kategorie podléhá podmínce, která vyjadřuje, jak se z požadavků zakázky určí výsledný objem výroby. Parametry  $n_{ia}$ ,  $n_{ib}$  a  $n_{ic}$  přitom označují počty požadovaných modulů v zakázce  $i$ , a to vždy pro příslušnou skupinu modulů. Ve vztazích (24) a (25) je zřejmé, že se objem odvíjí od proměnné  $x_i$ . U atypických modulů ve vztahu (26) naopak vidíme použití proměnné  $u_i$  a to z důvodu možného neplnění této části.

$$Q_a = \sum_{i=1}^{30} n_{ia} x_i \quad \forall a = 1, 2, 3, 4 \quad (24)$$

$$Q_b = \sum_{i=1}^{30} n_{ib} x_i \quad \forall b = 1, 2, \dots, 17 \quad (25)$$

$$Q_c = \sum_{i=1}^{30} n_{ic} u_i \quad \forall c = 1, 2, 3, 4 \quad (26)$$

### Omezení rozdělení výroby do objemových pásem

Vyrobený počet (ks) typových modulů typu  $Q_b$  musí být rozdělen do jednotlivých objemových pásem  $y_{bk}$ . Součet kusů zařazených do všech pásem se proto musí rovnat celkovému objemu výroby  $Q_b$ .

$$Q_b = \sum_{k=1}^6 y_{bk} \quad \forall b = 1, 2, \dots, 17 \quad (27)$$

### Omezení objemových pásem

Model pracuje se šesti objemovými pásmy, přičemž za běžná pásma jsou uvažována pásma 1 až 5, protože mají stejnou šířku 10 kusů (jsou uzavřena stejnou horní mezí). Podmínka (28) zajišťuje, že pásmo obsahuje maximálně 10 kusů, pokud je pomocí binární proměnné  $z_{bk}=1$  aktivní. Pokud aktivní není, musí být logicky objem  $y_{bk}$  roven nule, protože  $z_{bk}=0$ .

Například pro modul  $B_3$  ležící v druhém pásmu platí podmínka (29), kde je-li  $z_{3,2}=1$ , může  $y_{32}$  nabývat hodnot od 0 do 10. Je-li  $z_{3,2}=0$ , musí být i  $y_{3,2}=0$ .

Poslední šesté pásmo (30) je shora omezeno maximálním možným objemem typového modulu  $B_b$ . Šesté pásmo by bylo možné omezit také pomocí velké konstanty např. 10 000, místo součtu požadavků pro daný modul napříč zakázkami. Využití konstanty však není pro model ideální, jelikož neexistuje přímá vazba s maximálním reálným objemem pásma. Z toho důvodu je zde využita přesnější vazba, vycházející přímo ze vstupních dat.

$$y_{bk} \leq 10z_{bk} \quad (28)$$

$$\forall b = 1,2, \dots, 17, \forall k = 1,2,3,4,5$$

$$y_{3,2} \leq 10z_{3,2} \quad (29)$$

$$y_{b6} \leq \left( \sum_{i=1}^{30} n_{ib} \right) z_{b6} \quad \forall b = 1,2, \dots, 17 \quad (30)$$

### Kaskádová aktivace pásem

Vyšší objemové pásmo může být využito pouze tehdy, pokud je aktivní také pásmo bezprostředně předcházející (31). Tím je zajištěno postupné, kaskádové plnění pásem. Ve vztahu (32) lze na příkladu vidět, že druhé pásmo nelze využít bez aktivace pásma prvního.

$$z_{bk} \leq z_{b,k-1} \quad (31)$$

$$\forall b = 1,2, \dots, 17, \forall k = 2,3,4,5,6$$

$$\rightarrow z_{b2} \leq z_{b,1} \quad (32)$$

Aby nebylo možné přeskočit nižší pásmo a současně využívat pásmo vyšší, musí být nižší pásmo před aktivací vyššího plně naplněno, a proto platí vztah (33). Podmínka je formulována pro všechna pásma od druhého po šesté, protože u nich je nutné, aby bylo předchozí pásmo plně naplněno. Poslední šesté pásmo může být tedy využito až tehdy, kdy je páté pásmo zaplněno v rozsahu 10 kusů.

$$y_{b,k-1} \geq 10z_{bk} \quad (33)$$

$$\forall b = 1,2, \dots, 17, \forall k = 2,3,4,5,6$$

### Kapacitní omezení výroby

Vztah (34) vyjadřuje, že součet kapacitní náročnosti všech vyráběných modulů nesmí ve sledovaném období překročit disponibilní kapacitu podniku. Kapacita je v modelu vyjádřena v kapacitních jednotkách, přičemž jedna kapacitní jednotka odpovídá výrobě jednoho průměrně náročného modulu. Výrazy na levé straně proto představují přepočtenou kapacitní spotřebu katalogových, typových a atypických modulů, zatímco pravá strana 520 udává maximální objem disponibilní kapacity v daném období.

$$\sum_{a=1}^4 \lambda_a Q_a + \sum_{b=1}^{17} \sum_{k=1}^6 \alpha_k \lambda_b y_{bk} + \sum_{c=1}^4 \lambda_c Q_c \leq 520 \quad (34)$$

### Obligátní podmínky

Tyto podmínky vymezují obor hodnot pro jednotlivé proměnné modelu, jinými slovy určují interval, jehož hodnot mohou výsledky nabývat. Rozhodovací proměnné  $x_i$ ,  $u_i$  a pomocná proměnná  $z_{bk}$  jsou binární, protože vyjadřují rozhodnutí typu ano/ne, a mohou tedy nabývat pouze hodnot 0 nebo 1.

$$\begin{aligned} x_i &\in \{0,1\} \forall i = 1,2, \dots, 30 \\ z_{bk} &\in \{0,1\} \forall b = 1,2, \dots, 17, \forall k = 1,2,3,4,5,6 \\ u_i &\in \{0,1\} \forall i = 1,2, \dots, 30 \end{aligned} \quad (35)$$

### 5.1.2 Výsledný matematický model

Maximalizuj:

$$Z = \sum_{i=1}^{30} \sum_{a=1}^4 m_a n_{ia} x_i + \sum_{i=1}^{30} \sum_{b=1}^{17} m_b n_{ib} x_i + \sum_{i=1}^{30} \sum_{c=1}^4 m_c n_{ic} u_i$$

Za podmínek:

$$\begin{aligned} u_i &\leq x_i \quad \forall i = 1,2, \dots, 30 \\ u_i &\geq x_i - d_i \quad \forall i = 1,2, \dots, 30 \\ x_i &\geq g_i \quad \forall i = 1,2, \dots, 30 \end{aligned}$$

$$Q_a = \sum_{i=1}^{30} n_{ia} x_i \quad \forall a = 1,2,3,4$$

$$Q_b = \sum_{i=1}^{30} n_{ib} x_i \quad \forall b = 1,2, \dots, 17$$

$$Q_c = \sum_{i=1}^{30} n_{ic} u_i \quad \forall c = 1,2,3,4$$

$$Q_b = \sum_{k=1}^6 y_{bk} \quad \forall b = 1,2, \dots, 17$$

$$y_{bk} \leq 10z_{bk}$$

$$\forall b = 1,2, \dots, 17, \forall k = 1,2,3,4,5,6$$

$$y_{b6} \leq \left( \sum_{i=1}^{30} n_{ib} \right) z_{b6} \quad \forall b = 1,2, \dots, 17$$

$$z_{bk} \leq z_{b,k-1}$$

$$\forall b = 1,2, \dots, 17, \forall k = 2,3,4,5,6$$

$$y_{b,k-1} \geq 10z_{bk}$$

$$\forall b = 1,2, \dots, 17, \forall k = 2,3,4,5,6$$

$$\sum_{a=1}^4 \lambda_a Q_a + \sum_{b=1}^{17} \sum_{k=1}^6 \alpha_k \lambda_b y_{bk} + \sum_{c=1}^4 \lambda_c Q_c \leq 520$$

$$x_i \in \{0,1\} \quad \forall i = 1,2, \dots, 30$$

$$z_{bk} \in \{0,1\} \quad \forall b = 1,2, \dots, 17, \forall k = 1,2,3,4,5,6$$

$$u_i \in \{0,1\} \quad \forall i = 1,2, \dots, 30$$

### 5.1.3 Řešení matematického modelu pomocí jazyka Python

Pro řešení modelu byl zvolen jazyk Python, jelikož je úloha už poměrně rozsáhlá a obsahuje množství vzájemně propojených proměnných a omezení. Nejde jen o samotné rozhodnutí, které zakázky přijmout a které ne, ale také o atypickou část zakázky, o celkové objemy jednotlivých modulů, jejich rozdělení do objemových pásem a o kontrolu

kapacitního omezení. Při takto rozsáhlé struktuře by bylo řešení v tabulkovém procesoru nepřehledné a obtížně upravitelné při změnách vstupních dat.

Python umožňuje rozdělit celé řešení do několika navazujících kroků. Nejprve jsou načtena vstupní data z excelového souboru, poté je sestaven samotný optimalizační model, následně je úloha vyřešena pomocí CBC (COIN-OR Branch and Cut) řešitele, a nakonec jsou výsledky automaticky zapsány do připraveného výstupu ve formě excelovských listů. Díky tomu je celý postup přehlednější a lépe se s ním pracuje i ve chvíli, kdy je potřeba model upravit, doplnit nebo znovu spustit s jinými vstupními hodnotami.

Další výhodou využití Pythonu je, že umožňuje přirozeně převádět matematický model do programové podoby. V této práci je využita knihovna PuLP, pomocí které lze zapisovat rozhodovací proměnné, účelovou funkci i omezující podmínky způsobem, který přímo navazuje na matematickou formulaci modelu.

### Převod matematického zápisu do prostředí Python

Obrázek 12 přibližuje část kódu, ve které jsou vytvořeny hlavní rozhodovací proměnné modelu. Nejprve jsou zavedeny binární rozhodovací proměnné  $x$  a  $u$ , kde  $x$  určuje přijetí zakázky a  $u$  realizaci její atypické části. Dále kód na obrázku 12 vytváří pomocné celočíselné proměnné  $Q_A$ ,  $Q_B$  a  $Q_C$ , které vyjadřují celkové počty vyrobených modulů v jednotlivých skupinách. Současně jsou zde definovány i proměnné  $y$  a  $z$  pro moduly skupiny B a objemová pásma, pro rozdělení výroby do pásem a jejich aktivaci.

Obrázek 12: Proměnné modelu

```
x = pl.LpVariable.dicts("x", data.investori, lowBound=0, upBound=1, cat="Binary")
u = pl.LpVariable.dicts("u", data.investori, lowBound=0, upBound=1, cat="Binary")

Q_A = pl.LpVariable.dicts("Q_A", data.moduly_A, lowBound=0, cat="Integer")
Q_B = pl.LpVariable.dicts("Q_B", data.moduly_B, lowBound=0, cat="Integer")
Q_C = pl.LpVariable.dicts("Q_C", data.moduly_C, lowBound=0, cat="Integer")

y = pl.LpVariable.dicts("y", (data.moduly_B, data.pasma), lowBound=0, cat="Integer")
z = pl.LpVariable.dicts("z", (data.moduly_B, data.pasma), lowBound=0, upBound=1, cat="Binary")

big_m_B = {
    b: sum(data.pozadavky[i][b] for i in data.investori)
    for b in data.moduly_B
}
```

Zdroj: vlastní zpracování 2026 dle PuLP documentation team. (n.d.)

V dolní části kódu na obrázku 12 je ještě vytvořen pomocný parametr  $big\_M\_B$ . Ten pro každý modul skupiny B určuje maximální možný objem výroby jako součet všech

požadavků napříč zakázkami. V programu se pak využívá při formulaci omezení pro poslední otevřené pásmo.

Na obrázku 13 je zachycen zápis účelové funkce. Kód nejprve samostatně sestavuje tři dílčí části marže podle skupin modulů A, B a C. U skupin A a B je marže vázána na přijetí zakázky pomocí proměnné  $x[i]$ , zatímco u skupiny C je navázána na realizaci atypické části prostřednictvím proměnné  $u[i]$ . Na konci jsou tyto tři části sečteny a přidány do modelu jako celková marže, kterou se model snaží maximalizovat.

Obrázek 13: Účelová funkce modelu

```
marze_A = pl.lpSum(  
    data.marze[a] * data.pozadavky[i][a] * x[i]  
    for i in data.investori  
    for a in data.moduly_A  
)  
marze_B = pl.lpSum(  
    data.marze[b] * data.pozadavky[i][b] * x[i]  
    for i in data.investori  
    for b in data.moduly_B  
)  
marze_C = pl.lpSum(  
    data.marze[c] * data.pozadavky[i][c] * u[i]  
    for i in data.investori  
    for c in data.moduly_C  
)  
  
model += marze_A + marze_B + marze_C, "Celkova_marze"
```

Zdroj: vlastní zpracování 2026 dle PuLP documentation team. (n.d.)

Na obrázku 14 je zachycen úsek kódu, kde se do programové logiky převádí základní vztah mezi přijetím zakázky, atypickou částí a možnou prioritou zakázky. Kód zde pro každou zakázku vytváří vazbu mezi přijetím zakázky  $x[i]$  a realizací atypické části  $u[i]$ , takže atypickou část nelze plnit bez přijetí zakázky. Současně je zde zohledněna možnost neplnění atypické části u vybraných zakázek a také požadavek, aby byly prioritní zakázky vždy přijaty.

Obrázek 14: Logika mezi typovou a atypickou částí zakázky

```
for i in data.investori:  
    model += u[i] <= x[i], f"Atypicka_cast_muze_byt_jen_prijeti_{i}"  
    model += u[i] >= x[i] - data.neplneni_atyp[i], f"Neplneni_atypicke_casti_{i}"  
    model += x[i] >= data.priorita[i], f"Prioritni_zakazka_{i}"
```

Zdroj: vlastní zpracování 2026 dle PuLP documentation team. (n.d.)

Na obrázku 15 lze pozorovat výpočet celkových objemů jednotlivých modulů. Kód zde pro každou skupinu modulů A, B a C vytváří podmínky, které určují, že celkový počet vyrobených kusů odpovídá součtu požadavků ze všech zakázek. Tento úsek tedy převádí do programu podmínky pro výpočet celkových objemů výroby jednotlivých modulů.

Obrázek 15: Celkový počet modulů

```
for a in data.moduly_A:
    model += Q_A[a] == pl.lpSum(data.pozadavky[i][a] * x[i] for i in data.investori), f"Q_A_{a}"

for b in data.moduly_B:
    model += Q_B[b] == pl.lpSum(data.pozadavky[i][b] * x[i] for i in data.investori), f"Q_B_{b}"

for c in data.moduly_C:
    model += Q_C[c] == pl.lpSum(data.pozadavky[i][c] * u[i] for i in data.investori), f"Q_C_{c}"
```

Zdroj: vlastní zpracování 2026 dle PuLP documentation team. (n.d.)

Ukázka kódu na obrázku 16 ilustruje rozdělení modulů skupiny B do objemových pásem. Nejprve program stanoví, že celkový počet kusů modulu  $Q_B[b]$  musí odpovídat součtu kusů zařazených do všech pásem  $y[b][k]$ . Následně jsou pro jednotlivá pásma vytvořeny horní meze, aby do nich nebylo možné zařadit více kusů, než kolik dané pásmo připouští.

Kód na obrázku 16 současně rozlišuje běžná pásma a poslední otevřené pásmo. U běžných pásem je horní mez dána jejich šířkou, zatímco u posledního pásma je použita hodnota  $big\_M\_B$ , která vychází z maximálního možného objemu daného modulu. Tím je v kódu převedena logika rozdělení výroby do pásem i omezení jejich kapacity.

Obrázek 16: Objemová pásma

```
for b in data.moduly_B:
    model += Q_B[b] == pl.lpSum(y[b][k] for k in data.pasma), f"Rozklad_do_pasem_{b}"

posledni_pasmo = max(data.pasma)
for b in data.moduly_B:
    for k in data.pasma:
        if k == posledni_pasmo:
            model += y[b][k] <= big_m_B[b] * z[b][k], f"Horni_mez_posledni_pasmo_{b}_{k}"
        else:
            sirka = data.sirky_pasem[k]
            if sirka is None:
                raise ValueError(f"Pásma {k} musí mít definovanou šířku.")
            model += y[b][k] <= sirka * z[b][k], f"Horni_mez_pasma_{b}_{k}"
```

Zdroj: vlastní zpracování 2026 dle PuLP documentation team. (n.d.)

Na obrázku 17 je zachycena kaskádová logika objemových pásem pro typové moduly skupiny B. Kód zde zajišťuje, že vyšší pásmo může být aktivní pouze tehdy, pokud je aktivní i pásmo předchozí. Současně je doplněna podmínka, podle níž musí být při využití vyššího pásma předchozí pásmo nejprve plně zaplněno.

Obrázek 17: Kaskádové plnění

```
for b in data.moduly_B:
    for idx in range(1, len(data.pasma)):
        k = data.pasma[idx]
        k_pred = data.pasma[idx - 1]
        model += z[b][k] <- z[b][k_pred], f"Kaskadova_aktivace_{b}_{k}"

        sirka_pred = data.sirky_pasem[k_pred]
        if sirka_pred is None:
            raise ValueError(f"Předchozí pásmo {k_pred} musí mít definovanou šířku.")
        model += y[b][k_pred] >= sirka_pred * z[b][k], f"Plne_zaplneni_predchoziho_pasma_{b}_{k}"
```

Zdroj: vlastní zpracování 2026 dle PuLP documentation team. (n.d.)

Na obrázku 18 je pak zachyceno kapacitní omezení modelu. Kód zde nejprve samostatně vyjadřuje kapacitní spotřebu modulů skupin A, B a C. U skupin A a C se kapacita počítá jako součin kapacitního vytížení a celkového počtu vyrobených kusů, zatímco u skupiny B je navíc zohledněn i koeficient příslušného objemového pásma.

V posledním řádku jsou tyto tři části sečteny a porovnány s celkovou disponibilní kapacitou podniku.

Obrázek 18: Omezení kapacity

```
kapacita_A = pl.lpSum(data.kapacitni_vytizeni[a] * Q_A[a] for a in data.moduly_A)
kapacita_B = pl.lpSum(
    data.koeficienty_pasem[k] * data.kapacitni_vytizeni[b] * y[b][k]
    for b in data.moduly_B
    for k in data.pasma
)
kapacita_C = pl.lpSum(data.kapacitni_vytizeni[c] * Q_C[c] for c in data.moduly_C)

model += kapacita_A + kapacita_B + kapacita_C <= data.kapacita_celkem, "Kapacitni_omezeni"
```

Zdroj: vlastní zpracování 2026 dle PuLP documentation team. (n.d.)

Po zapsání rozhodovacích proměnných, účelové funkce a omezujících podmínek je model v prostředí Python za pomoci knihovny Pulp sestaven ve finální podobě. Ukázky kódu z minulé podkapitoly zachycují právě jen ty části, kde se matematický zápis převáděl do programové formy. Další části už nespočívají v doplnění logiky matematického modelu, ale v jeho předání řešiteli a následném exportu výsledků.

Finální matematický model je předán CBC řešiteli, který je vhodný pro práci s lineárními a smíšenými celočíselnými modely. Řešitel po spuštění skriptu načte celý matematický model a vyhledá takové řešení, jenž splní veškeré podmínky a zároveň maximalizuje hodnotu účelové funkce.

Následně je model vyřešen a hodnoty rozhodovacích proměnných jsou automaticky exportovány do výstupního souboru. Výhodou popsaného postupu je přehledná manipulace se vstupními a výstupními daty. Poptávka se samozřejmě mění v čase a model musí mít aktuální data, což je v tomto případě jednoduše proveditelné, jelikož stačí upravit vstupní parametry v excelovém souboru a spustit model znovu.

#### **5.1.4 Intepretace výsledků**

Z dosažených výsledků je patrné, že model při zadaných podmínkách dosáhl optimálního řešení v podobě celkové marže 102 130 300 Kč. Tato hodnota však vychází z anonymizovaných a částečně upravených vstupních dat, která byla v modelu použita z důvodu ochrany citlivých obchodních informací. Výslednou marži proto nelze chápat jako reálně dosažitelný výsledek podniku, ale jako modelový ukazatel umožňující posoudit chování modelu. Jak ukazuje tabulka 6, rozhodující část celkové marže vytváří typová skupina B, a to 80 025 800 Kč. Skupina A katalogových modulů se na výsledku podílí částkou 15 080 000 Kč a atypická skupina C částkou 7 024 500 Kč. Z rozložení výsledků je tedy vidět, že hlavní ekonomický význam mají v nalezeném řešení právě typové moduly skupiny B, což bylo ale zřejmé již před vyřešením modelu, jelikož poptávka je zpravidla tvořena většinovým zastoupením typových modulů.

V tabulce 7 je uvedeno, že celkový počet vyrobených modulů činí 623. Tento počet je nutné interpretovat ve vztahu k výrobní kapacitě, která byla stanovena na 520 kapacitních jednotek, přičemž jedna jednotka odpovídá výrobě jednoho průměrně náročného modulu. Z toho vyplývá, že skutečná výroba přesahuje pomyslnou nominální kapacitu, což je dáno rozdílnou náročností jednotlivých typů modulů. Průměrné kapacitní vytížení na jeden modul se v tomto případě pohybuje přibližně mezi 83–84 % výrobní kapacity. Z uvedených hodnot lze usuzovat, že výroba byla ve větší míře orientována na méně náročné typy modulů, jako jsou malé koupelny či samostatná WC, které vyžadují nižší kapacitní náročnost. Tento výsledek přitom není primárně důsledkem optimalizačního modelu, ale ve vyšší míře odráží strukturu samotných zakázek, které obsahují vyšší podíl méně kapacitně náročných modulů.

Do celkového průměru se současně promítá vliv objemových pásem. S rostoucím využitím těchto pásem dochází u typových modulů k postupné konvergenci kapacitního vytížení k hodnotě 75 %. Z celkového počtu bylo vyrobeno 500 kusů modulů B, 79 kusů A a 44 kusů C.

Při podrobnějším zkoumání výsledků (uvedených v tabulce 9) je patrné, že ani uvnitř jednotlivých skupin (A, B, C) není výroba rozdělena rovnoměrně. Ve skupině A má nejvyšší objem modul A1 s 37 kusy, zatímco ve skupině C převažuje modul C2 s 24 kusy (pro atypické moduly se však nejedná o 24 stejných kusů typu C2, ale 24 modulů, kde je nutné každý řešit samostatně, což je přesným opakem katalogových modulů typu A). Ještě výraznější rozdíly jsou ale u skupiny B, kde nejvyšší objem vykazují moduly B6 s 59 kusy, B5 s 52 kusy a B4 se 47 kusy. Naopak jiné moduly zůstávají na podstatně nižších hodnotách. To ukazuje, že model nevolí výrobu rovnoměrně napříč portfoliem, ale snaží se kvůli úsporám z rozsahu soustředit výrobu do co nejmenšího počtu celkových skupin.

Tabulka 6: Výsledná marže

marže celkem	marže A	marže B	marže C
102 130 300,00 Kč	15 080 000,00 Kč	80 025 800,00 Kč	7 024 500,00 Kč

Zdroj: vlastní zpracování 2026

Tabulka 7: Počet vyrobených kusů modulů jednotlivých tříd

Celkové Q	Q_A	Q_B	Q_C
623	79	500	44

Zdroj: vlastní zpracování 2026

Tabulka 8 - Rozpuštění typových modulů do objemových pásem

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17
pásma_1	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10
pásma_2	5	10	10	10	10	10	10	8	4	5	10	6	10	10	10	9	10
pásma_3	0	10	10	10	10	10	10	0	0	0	3	0	10	7	7	0	7
pásma_4	0	9	2	10	10	10	2	0	0	0	0	0	8	0	0	0	0
pásma_5	0	0	0	7	10	10	0	0	0	0	0	0	0	0	0	0	0
pásma_6	0	0	0	0	2	9	0	0	0	0	0	0	0	0	0	0	0

Zdroj: vlastní zpracování 2026

Z tabulek 8 a 10 je dobře patrné, jak se ve výsledku projevuje pásmová logika modelu. U modulů skupiny B nejsou důležité jen jejich celkové objemy, ale i to, do jakých objemových pásem se výroba dostala. Vyšší pásma se aktivují pouze tehdy, když jsou zaplněna nižší pásma, takže model respektuje zadanou kaskádovou strukturu. Nejvýrazněji je to patrné u modulů B5 a B6, které se dostaly až do šestého pásma. U těchto modulů se tedy nejvíce projevil efekt úspor z rozsahu. Je ale potřeba doplnit, že úspory z rozsahu nejsou v modelu zachyceny jako plynulá změna, ale pomocí lineární aproximace se skokovými změnami po pásmech. S přechodem do vyššího pásma se tak

mění koeficient kapacitní náročnosti ( $1 \rightarrow 0,95 \rightarrow \dots \rightarrow 0,75$ ), čímž model zjednodušeně vystihuje efektivnější výrobu při vyšších objemech. Naopak u modulů s nižší výrobou zůstává produkce pouze v prvních pásmech a tento efekt se zde téměř neprojeví.

Důležitou část výsledku ukazuje tabulka 11, kde jsou zachyceny hodnoty proměnných  $x_i$  a  $u_i$ . Z tabulky 11 lze vyčíst, že model nepřijal všechny zakázky, ale šest z nich vyřadil úplně, konkrétně zakázky  $Z_4$ ,  $Z_{20}$ ,  $Z_{23}$ ,  $Z_{25}$ ,  $Z_{26}$  a  $Z_{27}$ , u nichž je  $x_i = 0$ . Ostatní zakázky byly přijaty.

Z tabulky 11 je současně vidět, že model preferuje neplnění atypické části modelu v případech, kdy takovou možnost má. Konkrétně se jedná o zakázky  $Z_3$ ,  $Z_5$ ,  $Z_{10}$ ,  $Z_{28}$  a  $Z_{30}$ , u nichž vychází  $x_i = 1$ , ale současně  $u_i = 0$ . To znamená, že základ zakázky přijat byl, avšak atypická část realizována nebyla. V případě zakázky  $Z_4$  model nešel ani cestou částečného plnění, ale zakázku odmítl úplně, tedy  $x_i = 0$ .

Z tohoto výsledku plyne, že atypická část těchto zakázek je z pohledu modelu nevýhodná. Jinými slovy, pokud má model možnost atypickou část vynechat, dává ve všech přijatých případech přednost právě této variantě. Ve výsledném řešení se tedy neobjevil ani jeden případ, kdy by byla atypická část realizována, přestože to nebylo nutné. To naznačuje, že při zadaných maržích, kapacitní náročnosti a celkové kapacitě podniku nepřináší atypická část dostatečný přínos vzhledem k tomu, jak zatěžuje výrobní kapacitu.

Tabulka 9 - Počet vyrobených kusů jednotlivých modulů

A1	A2	A3	A4	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17	C1	C2	C3	C4
37	20	9	13	15	39	32	47	52	59	32	18	14	15	23	16	38	27	27	19	27	9	24	9	2

Zdroj: vlastní zpracování 2026

Tabulka 10 - Vizualizace využití objemových pásem

	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17
aktivace pásma_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
aktivace pásma_2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
aktivace pásma_3	0	1	1	1	1	1	1	0	0	0	1	0	1	1	1	0	1
aktivace pásma_4	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0
aktivace pásma_5	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
aktivace pásma_6	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0

Zdroj: vlastní zpracování 2026

Tabulka 11 - Výsledné přijetí zakázek a přijetí příslušných atypických částí

	Z_1	Z_2	Z_3	Z_4	Z_5	Z_6	Z_7	Z_8	Z_9	Z_10	Z_11	Z_12	Z_13	Z_14	Z_15
x <sub>i</sub>	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
u <sub>i</sub>	1	1	0	0	0	1	1	1	1	0	1	1	1	1	1
	Z_16	Z_17	Z_18	Z_19	Z_20	Z_21	Z_22	Z_23	Z_24	Z_25	Z_26	Z_27	Z_28	Z_29	Z_30
x <sub>i</sub>	1	1	1	1	0	1	1	0	1	0	0	0	1	1	1
u <sub>i</sub>	1	1	1	1	0	1	1	0	1	0	0	0	0	1	0

Zdroj: vlastní zpracování 2026

## **5.2 Analýza výsledků a možné doporučení pro podnik**

Výsledky optimalizačního modelu nepředstavují pouze jednorázové řešení konkrétní ilustrativní úlohy, ale slouží také jako podklad pro formulaci praktických doporučení pro podnik Prefab Factory. Získané poznatky ukazují, že ekonomická výhodnost zakázek není dána jen jejich celkovou hodnotou, ale také strukturou požadovaných modulů, jejich kapacitní náročností a mírou opakovatelnosti výroby. Právě z tohoto důvodu je vhodné navázat na výsledky modelu doporučeními, která mohou podniku pomoci nejen při samotném výběru zakázek, ale také při obchodním jednání se zákazníky a při hledání takových úprav zadání, které povedou k efektivnějšímu využití kapacity i k lepším ekonomickým výsledkům.

### **Srovnání modelu s dosavadním postupem využívaným v praxi**

Přínos navrženého modelu lze posoudit porovnáním se způsobem rozhodování, který je v podniku v současnosti využíván při přijímání zakázek. V běžné praxi jsou zakázky posuzovány především podle očekávané marže a orientačního vyhodnocení jejich výrobní náročnosti ve vztahu k disponibilní kapacitě sledovaného období. Tento postup přitom nemá vždy zcela jednotnou podobu, protože se odvíjí od konkrétní situace a je ve značné míře založen na zkušenosti a odborném úsudku pracovníků podniku. Pro účely srovnání byl tento praktický přístup přiblížen tak, že se u každé zakázky stanovila její celková marže jako součet marží všech požadovaných modulů, zakázky byly následně seřazeny od nejvyšší celkové marže po nejnižší a postupně zařazovány do plánu až do okamžiku, kdy by přijetí další zakázky vedlo k překročení disponibilní kapacity.

Při takto pojatém postupu odpovídajícím současné podnikové praxi by bylo přijato 21 zakázek a výsledná celková marže by činila 99 094 500 Kč. Optimalizační model oproti tomu vedl k přijetí 24 zakázek a k dosažení celkové marže 102 130 300 Kč. Modelové řešení tedy vychází lépe jak z hlediska počtu přijatých zakázek, tak z hlediska celkového ekonomického výsledku. Rozdíl mezi oběma přístupy ukazuje, že rozhodování založené převážně na celkové marži zakázky a hrubém posouzení její náročnosti nemusí vést k nevhodnějšímu výsledku.

Současně je však třeba uvést, že přímé propojení modelu s běžnou praxí podniku není bez obtíží. Model předpokládá přesná, aktuální a dostatečně strukturovaná vstupní data, která musí být k dispozici již ve chvíli rozhodování. V reálném provozu však konečná podoba

zakázky nemusí být vždy známa s úplnou přesností, některé parametry se mohou v průběhu jednání dále měnit a do rozhodování vstupují i další okolnosti obchodního nebo provozního charakteru. Z tohoto důvodu nelze předpokládat, že by při přímém nasazení do praxe model automaticky vedl k lepším výsledkům, než postup založený na zkušenosti pracovníků podniku, protože právě jejich úsudek dokáže na tyto změny a neúplnosti pružně reagovat. Přínos modelu proto v současné podobě spočívá především v systematizaci rozhodování a v kvantitativní podpoře posouzení zakázek, nikoli v tom, že by bez dalších úprav jednoznačně nahrazoval stávající praxi. V rámci konzultací byla současně zmíněna možnost ověřit využitelnost modelu i na datech z dalších plánovacích období. Do budoucna tak lze uvažovat o jeho dalším zpřesnění a o postupném přiblížení modelu reálným podmínkám rozhodování v podniku.

### **Vyšší standardizace a sériovost zakázek**

V této části je uvažována situace, kdy se podnik nesoustředí pouze na pasivní výběr zakázek, ale snaží se aktivně ovlivnit jejich výslednou podobu tak, aby byla výroba co nejvíce standardizovaná a sériová. Předpokládá se, že zákazníkům může být nabídnuta sleva jako motivace k úpravě původního zadání. Jedná se zejména o možnost převodu typových modulů na katalogové. Možnosti převodu jsou podrobněji rozepsány dále v textu. Cílem takové úpravy je zvýšit podíl opakovatelných řešení, snížit kapacitní náročnost výroby a u typových modulů současně lépe využít objemová pásma, v nichž se projevuje efekt úspor ze sériovosti.

Postup je založen na porovnání dvou variant řešení. Optimální řešení odpovídající původní struktuře zakázek bude porovnáno s novým scénářem, kde u vybraných zakázek dojde k navýšení standardizace. Upravená data budou znovu dosazena do optimalizačního modelu a výsledky obou scénářů budou porovnány z hlediska celkové marže, využití kapacity a skladby přijatých zakázek. Rozdíl mezi původním a upraveným řešením následně ukáže, zda by taková sleva byla pro podnik ekonomicky přijatelná a do jaké výše by ji bylo možné poskytnout, aniž by došlo ke zhoršení celkového výsledku.

### **Scénář standardizace: převod na katalogové moduly A**

Scénář vychází z předpokladu, že podnik dokáže ústupkem v ceně motivovat k drobným úpravám architektonické studie tak, že je možné místo typových řešení B využít přímo katalogová A. Úpravy podkladů zahrnují například malé změny dispozice bytu, posun příček a instalačních šachet, kde rozsah úprav probíhá v řádu centimetrů. Nový scénář

proto vychází z úpravy poptávky pouze u technicky realistických případů, tedy není možné přejít z velkého typového modulu na malý katalogový. Převody jsou voleny pouze u modulů, kde odpovídá jejich vzájemná funkce a půdorysná plocha (možný rozdíl v řádu cm). V tabulce 12 jsou vyznačeny možnosti převodu typových modulů na katalogové. Moduly B12 a B13 nelze převést na žádný katalogový modul, jelikož zaujímají prostor jako A4, ale jsou pouze koupelnou (jedna místnost), kdežto A4 je dvoj modulem (dvě místnosti, jeden modul). Funkce modulu je pak rozdílná a převod nelze provést.

Tabulka 12 - Možné převedení typových modulů na katalogové

A1	B1, B2, B3
A2	B4, B5, B6, B7, B17
A3	B8, B9, B10, B11
A4	B14, B15, B16

Zdroj: vlastní zpracování 2026

V tabulce 13 lze vidět, jak vypadá zadání zakázky, převedly by se veškeré typové moduly na katalogové. Spuštění modelu s tímto vstupem poskytne řešení, u kterého lze očekávat vyšší hodnotu účelové funkce.

Po dosazení nových vstupů a spuštění našel model nové optimum. Toto řešení je však zatím počítáno při zachování původních jednotkových marží, tedy bez zohlednění případné slevy, kterou by podnik mohl zákazníkovi nabídnout jako motivaci ke standardizaci řešení. V základním optimálním řešení byla celková marže 102 130 300 Kč. Ve scénáři maximální standardizace vzrostla na 144 558 200 Kč. Zlepšení tedy činí 42 427 900 Kč, což je přibližně 41,5 % navíc. Současně se zvýšil i počet přijatých zakázek. V původním řešení bylo přijato 24 ze 30 zakázek, zatímco po převodu typových modulů na katalogové bylo přijato 28 ze 30 zakázek. Nově byly přijaty zakázky Z\_4, Z\_20, Z\_23, Z\_25 a Z\_27, naopak byla odmítnuta zakázka Z\_15. Zakázka Z\_26 zůstala nepřijata v obou variantách.

V původním řešení podnik vyráběl celkem 623 modulů. Po maximální standardizaci výroba vzrostla na 725 modulů. To znamená, že v tomto konkrétním případě přinesla maximální standardizace dva pozitivní efekty zároveň. Jednak zvýšila celkovou marži a jednak umožnila přijmout více zakázek. Výsledek ukazuje, že dává smysl, aby podnik aktivně motivoval k úpravám, které umožní katalogové řešení.

Tabulka 13 - Nové zadání s úplnou standardizací

	A1	A2	A3	A4	B12	B13	C1	C2	C3	C4
Z_1	11	21	0	0	0	0	0	6	5	0
Z_2	12	11	0	0	0	0	0	0	0	0
Z_3	2	13	9	4	0	5	0	1	0	0
Z_4	1	2	3	12	5	1	1	0	0	1
Z_5	1	8	8	5	1	2	0	0	1	0
Z_6	8	8	0	0	1	0	3	0	0	0
Z_7	6	3	3	8	0	1	1	2	0	0
Z_8	7	14	5	1	0	0	1	0	0	0
Z_9	14	7	1	4	0	0	1	0	0	0
Z_10	7	0	1	7	0	0	7	4	0	0
Z_11	1	14	2	5	0	1	0	0	1	0
Z_12	6	11	3	5	0	0	0	2	0	0
Z_13	5	3	3	6	1	6	0	1	0	0
Z_14	13	0	7	3	5	0	0	1	0	0
Z_15	0	7	1	5	0	8	2	0	0	0
Z_16	2	12	6	0	0	4	0	3	0	1
Z_17	0	11	7	0	0	0	0	3	0	1
Z_18	3	11	2	6	0	0	0	0	2	0
Z_19	2	14	0	6	0	0	0	0	0	0
Z_20	0	0	9	6	0	0	2	0	0	4
Z_21	0	16	7	0	0	0	0	6	0	0
Z_22	2	10	2	3	0	6	1	0	0	0
Z_23	1	1	15	8	0	0	3	0	0	0
Z_24	11	19	0	3	0	0	0	0	0	0
Z_25	0	12	7	2	0	0	0	0	5	0
Z_26	10	1	5	0	0	0	0	5	0	7
Z_27	5	3	5	5	6	0	1	0	0	0
Z_28	6	2	7	0	5	0	0	0	0	4
Z_29	1	18	0	8	0	5	0	0	1	0
Z_30	3	4	5	7	3	0	3	3	0	0

Zdroj: vlastní zpracování 2026

Maximální přípustná sleva je pak odvozována právě prostřednictvím marže, nikoliv změny celkové ceny zakázky. Důvodem je skutečnost, že při převodu typového modulu na katalogový nedochází pouze ke změně ceny pro zákazníka, ale také ke změně nákladové náročnosti řešení. Samotná změna ceny by proto nevystihovala skutečný ekonomický dopad standardizace pro podnik. Současně tento přístup odpovídá i formulaci optimalizačního modelu, jehož účelová funkce je založena na maximalizaci celkové marže. Rozdíl hodnot účelové funkce mezi původním a upraveným scénářem tak

vyjadřuje ekonomický přínos standardizace a zároveň prostor, který může podnik využít pro případné poskytnutí slevy zákazníkovi.

Rozdíl hodnot účelové funkce mezi základním řešením a scénářem maximální standardizace činil 42 427 900 Kč. Vzhledem k tomu, že ve scénáři standardizace bylo přijato 28 zakázek, odpovídá tento rozdíl průměrnému prostoru pro slevu ve výši 1 515 282 Kč na jednu přijatou zakázku. Tato hodnota nepředstavuje skutečnou slevu, kterou by bylo vhodné poskytnout každé zakázce ve stejné výši, ale orientační průměrnou část dodatečně vytvořeného zisku, kterou může podnik využít jako obchodní motivaci zákazníka ke standardizaci řešení. Současně je třeba připomenout, že se i v tomto případě jedná o ilustrativní modelový výsledek vycházející ze záměrně upravených vstupních dat, a nikoli o přímý návrh reálné obchodní politiky podniku.

Přesnější vyjádření je však nutné provést na úrovni konkrétní zakázky. U jednotlivé zakázky je maximální přípustná sleva dána rozdílem mezi její marží ve standardizované variantě a marží v původní variantě. Pro zakázku  $i$  tedy platí vztah (36):

$$sleva_i^{\max} = M_i^{\text{převod}} - M_i^{\text{původní}} \quad (36)$$

kde  $M_i^{\text{převod}}$  označuje marži zakázky po převodu vybraných typových modulů na katalogové a  $M_i^{\text{původní}}$  marži zakázky v původním stavu. Pokud je tento rozdíl kladný, může podnik poskytnout slevu až do této výše, aniž by si u dané zakázky ekonomicky pohoršil.

Pro názornou ukázkou lze dosadit do vztahu (36) údaje například ze zakázky 1 ( $Z_1$ ), kde byla skladba modulů následující: 11 ks modulu  $A_1$ , 11  $A_2$ , 5  $B_4$ , 5  $B_5$ , 6  $C_2$  a 5  $C_3$ . Po převodu se první zakázka změnila na: 11 $A_1$ , 21 $A_2$ , 6 $C_2$ , 5 $C_3$ .

$$M_{Z_1}^{\text{původní}} = 7\,002\,000 \text{ Kč}$$

$$M_{Z_1}^{\text{převod}} = 7\,525\,000 \text{ Kč}$$

$$\Delta M_{Z_1} = 7\,525\,000 - 7\,002\,000 = 523\,000 \text{ Kč}$$

U zakázky  $Z_1$  tedy vzniká pro podnik díky standardizaci dodatečný zisk 523 000 Kč, který lze interpretovat jako maximální přípustnou slevu, kterou může podnik zákazníkovi nabídnout, aniž by si oproti původní variantě ekonomicky pohoršil.

Jinými slovy, podnik dokáže prvnímu zákazníkovi nabídnout slevu maximálně 523 000 Kč v případě upravení objednávky, kde se standardizuje právě 10 typových modulů na moduly katalogové. Jedná se tedy o maximální slevu 52 300 Kč na převod jednoho modulu.

Výsledky scénáře maximální standardizace ukázaly, že převod typových modulů na katalogové může pro podnik představovat významný ekonomický přínos. V porovnání se základním řešením došlo ke zvýšení hodnoty účelové funkce, a tím i k vytvoření prostoru pro obchodní motivaci zákazníků ke standardizaci řešení. Platí ale, že tento prostor nelze vhodně posuzovat souhrnně za celý soubor zakázek, ale je nutné jej vyhodnocovat na úrovni jednotlivých zakázek podle jejich konkrétní modulové skladby. Scénář maximální standardizace proto nepředstavuje přímý návrh plošného postupu v praxi, ale extrémní referenční variantu, na jejímž základě lze odhadnout potenciál standardizace a vymezit rámec pro individuální obchodní vyjednávání se zákazníkem. Scénářová analýza tak potvrdila, že standardizace řešení je pro podnik nejen výrobně, ale i ekonomicky přínosná, a že optimalizační model může sloužit nejen pro rozhodování o přijetí zakázek, ale také jako podpora při jednání o jejich úpravě.

## Závěr

Cílem této bakalářské práce bylo představit vybraný podnikový problém společnosti Prefab Factory a.s. a navrhnout jeho řešení s využitím metod matematického programování. V teoretické části byly vymezeny základní principy rozhodovacího procesu, operačního výzkumu, matematického programování a lineárního programování jako nástrojů využitelných při podpoře podnikového rozhodování. Dále byl představen samotný podnik, jeho výrobní program a vybrané souvislosti týkající se výroby, kapacitního zatížení a návaznosti na dopravu a montáž.

Praktická část práce byla zaměřena na problém výběru zakázek při omezené výrobní kapacitě. S ohledem na charakter výroby prefabrikovaných koupelnových modulů byl sestaven optimalizační model, jehož cílem bylo maximalizovat celkovou marži podniku při respektování kapacitních omezení. Navržený matematický model byl následně implementován a vyřešen v prostředí Python. Výsledky ukázaly, že matematické programování je pro daný typ rozhodovací úlohy vhodným přístupem a umožňuje získat přehledné a ekonomicky odůvodněné řešení. Přínos modelu spočívá zejména v tom, že neposuzuje zakázky pouze podle jejich objemu nebo očekávané marže, ale zohledňuje také jejich kapacitní náročnost, strukturu modulů, míru standardizace a možnost neplnění atypické části u vybraných zakázek.

Přínos navrženého modelu lze posoudit i porovnáním se způsobem rozhodování, který je v podniku v současnosti využíván při přijímání zakázek. Tento postup přijal 21 zakázek a výsledná celková marže činila 99 094 500 Kč, zatímco optimalizační model umožnil přijmout 24 zakázek a dosáhnout celkové marže 102 130 300 Kč. Z porovnání je tedy patrné, že model dokáže při daných vstupních podmínkách nalézt výhodnější skladbu zakázek než postup založený převážně na zkušenostním posouzení.

Součástí praktického vyhodnocení byla také scénářová analýza zaměřená na zvýšení sériovosti výroby prostřednictvím převodu typových modulů na katalogové. Modelový scénář ukázal, že vyšší míra standardizace může mít pro podnik významný ekonomický přínos, a to jak ve formě zvýšení hodnoty účelové funkce, tak ve formě vyššího počtu přijatých zakázek. Podniku se proto doporučuje zvážit motivování zákazníků k drobným úpravám jejich studií nebo poptávek tak, aby bylo možné ve větší míře využívat katalogová řešení. Optimalizační model tak může sloužit nejen jako nástroj pro

rozhodování o přijetí zakázek, ale také jako podpora při obchodním jednání o jejich výsledné podobě.

Přes uvedené přínosy je však třeba zdůraznit, že navržený model představuje účelové zjednodušení reálného podnikového prostředí. Jeho cílem nebylo postihnout všechny provozní, obchodní a organizační souvislosti v celé jejich šíři, ale soustředit se na jednu konkrétní rozhodovací úroveň, a to na volbu zakázek při omezené výrobní kapacitě a při zohlednění rozdílné náročnosti modulů. Model proto pracuje se staticky vymezeným plánovacím obdobím, s předem známou strukturou poptávky a s takovými vstupními údaji, které je možné kvantitativně zachytit. V reálné podnikové praxi se však konečná podoba zakázek může dále měnit, některé parametry nejsou v okamžiku rozhodování známy s úplnou přesností a do výsledného posouzení vstupují i další okolnosti, například obchodní vztahy, vývoj jednání se zákazníkem nebo aktuální provozní situace podniku. Z tohoto důvodu nelze model v současné podobě považovat za přímou náhradu stávajícího způsobu rozhodování.

Omezením modelu je dále skutečnost, že zachycuje především kapacitní stránku výběru zakázek, nikoli celé navazující řízení výroby. Není zde detailně řešeno časové rozvržení výroby v rámci jednotlivých dnů, konkrétní návaznost výrobních operací, doprava modulů na stavbu ani omezení montážních týmů. Stejně tak není model v současné podobě koncipován jako nástroj, který by pružně reagoval na průběžné změny vstupních dat v reálném čase. Scénářová část práce navíc pracuje s ilustrativními a částečně upravenými modelovými daty, takže její výsledky nelze bezprostředně převádět na konkrétní obchodní rozhodnutí podniku. Právě schopnost pracovníků pružně reagovat na průběžné změny a na neúplnost některých informací je zároveň důvodem, proč nelze automaticky předpokládat, že by model při okamžitém praktickém nasazení v každé situaci přinášel lepší výsledek, než postup založený na zkušenosti. Přínos modelu tak v současné podobě spočívá především ve zpřehlednění rozhodovací situace, v kvantitativní podpoře posouzení zakázek a v možnosti systematicky ověřovat, zda intuitivně přijaté rozhodnutí odpovídá i ekonomicky výhodnému řešení.

Navazující rozvoj modelu by proto mohl směřovat několika směry. V první řadě by bylo vhodné ověřit jeho využitelnost na datech z dalších plánovacích období a sledovat, nakolik se jeho doporučení přibližují reálným podnikovým rozhodnutím a jejich výsledkům. Dalším krokem by mohlo být rozšíření modelu o detailnější časovou složku, návaznost výroby, dopravy a montáže. Přínosné by bylo také doplnění modelu o možnost

průběžné aktualizace vstupních dat, aby lépe odpovídal situaci, kdy se parametry zakázek v čase mění. Smysluplným směrem dalšího vývoje je tedy zejména postupné přibližování modelu reálným podmínkám podniku při zachování jeho praktické použitelnosti pro rozhodování.

Přes uvedená omezení zvolený model splnil svůj účel a poskytl smysluplné výstupy pro danou rozhodovací úlohu. Stanovené cíle práce tak byly splněny. Byly představeny teoretické základy matematického programování, popsána vybraná problematika podniku Prefab Factory a.s., navržen a vyřešen vhodný matematický model a na jeho základě formulována doporučení pro podnik. Hlavním přínosem práce je ukázka toho, že metody matematického programování mohou být ve výrobním podniku prakticky využitelné nejen pro samotnou optimalizaci, ale také jako podpora manažerského a obchodního rozhodování.

## Seznam obrázků

Obrázek 1: Parametry řešitele.....	29
Obrázek 2: Ukázka kódu 1.....	30
Obrázek 3: Ukázka kódu 2.....	31
Obrázek 4: Ukázka kódu 3.....	31
Obrázek 5: Ukázka kódu 4.....	31
Obrázek 6: Ukázka kódu 5.....	32
Obrázek 7: Ukázka kódu 6.....	32
Obrázek 8: Ukázka kódu 7.....	32
Obrázek 9: Ukázka kódu 8.....	33
Obrázek 10: Hotový prefabrikovaný modul .....	36
Obrázek 11: Schématické zobrazení půdorysu katalogového, typového a atypického modulu .....	40
Obrázek 12: Proměnné modelu.....	51
Obrázek 13: Účelová funkce modelu.....	52
Obrázek 14: Logika mezi typovou a atypickou částí zakázky.....	52
Obrázek 15: Celkový počet modulů .....	53
Obrázek 16: Objemová pásma .....	53
Obrázek 17: Kaskádové plnění .....	54
Obrázek 18: Omezení kapacity .....	54

## Seznam tabulek

Tabulka 1: Struktura matematického modelu v MS excel.....	28
Tabulka 2: Poptávané množství modulů (zakázky v řádcích a moduly ve sloupcích) ...	43
Tabulka 3 - přehled ekonomických a kapacitních parametrů pro jednotlivé moduly ....	44
Tabulka 4 - přehled priorit zakázek a možnosti neplnění atypické části zakázky .....	44
Tabulka 5: Přehled kapacitních pásem pro moduly typu B .....	45
Tabulka 6: Výsledná marže .....	56
Tabulka 7: Počet vyrobených kusů modulů jednotlivých tříd .....	56
Tabulka 8 - Rozpuštění typových modulů do objemových pásem.....	56
Tabulka 9 - Počet vyrobených kusů jednotlivých modulů .....	58
Tabulka 10 - Vizualizace využití objemových pásem .....	58
Tabulka 11 - Výsledné přijetí zakázek a přijetí příslušných atypických částí.....	58
Tabulka 12 - Možné převedení typových modulů na katalogové.....	61
Tabulka 13 - Nové zadání s úplnou standardizací .....	62

## Zdroje a literatura

*Během přípravy této práce autor použil nástroj ChatGPT (OpenAI) ke stylistickým úpravám textu, jazykovým korekcím a dílčím konzultacím při kontrole a úpravách programového kódu. Po použití tohoto nástroje autor obsah pečlivě zkontroloval, upravil podle potřeby a přebírá plnou odpovědnost za výsledný obsah práce.*

Aggarwal, S., & Gupta, N. (2014). Applications of mathematical programming models for product mix optimization in world steel industry: Challenges and directions. *Managing in Recovering Markets*, 133–142. [https://doi.org/10.1007/978-81-322-1979-8\\_10](https://doi.org/10.1007/978-81-322-1979-8_10)

Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to linear optimization*. Athena Scientific / MIT Press. <https://doi.org/10.1080/07408179808966531>

COIN-OR CBC documentation team. (n.d.). *Introduction*. COIN-OR CBC User's Guide. <https://coin-or.github.io/Cbc/intro.html>

Cornuéjols, G., & Tütüncü, R. (2006). *Optimization methods in finance*. Cambridge University Press.

Deshpande, M. V. (2021). A study of applications of mathematics in industry area. *International Journal of Modern Trends in Technology and Management*, 67(1), 136–140. <https://doi.org/10.14445/22315373/ijmtt-v67i1p518>

Frontline Systems. (n.d.). *Step by step easy to use guide for Excel's Solver*. Frontline Systems.

Gros, I. (2003). *Kvantitativní metody v manažerském rozhodování*. Grada Publishing.

Jablonský, J. (2007). *Operační výzkum: Kvantitativní modely pro ekonomické rozhodování* (3. vyd.). Professional Publishing.

Luňáček, O., & Heralecký, P. (2009). *Optimalizace podnikových aktivit*. Key Publishing.

Mallia, B., Das, M., & Das, C. (2021). Fundamentals of transportation problem. *International Journal of Engineering and Advanced Technology*, 10(5), 90–103. <https://doi.org/10.35940/ijeat.E2654.0610521>

Marin, A., & Castro, J. (2020). A review of industrial applications of optimization models. *Processes*, 8(6), 745.

Microsoft. (n.d.). Define and solve a problem by using Solver. <https://support.microsoft.com/en-gb/office/define-and-solve-a-problem-by-using-solver-5d1a388f-079d-43ac-a7eb-f63e45925040>

Mula, J. (2010). Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204(3), 377–390. <https://doi.org/10.1016/j.ejor.2009.09.008>

Nayak, J., Varshney, S., & Shekhar, C. (2023). *Modeling and applications in operations research*. CRC Press.

Öztürk, A. (2013). Historical development of operations research and its features. *Alphanumeric Journal*, 1(1), 1–11.

- Plevný, M., & Žižka, M. (2010). *Modelování a optimalizace v manažerském rozhodování*. Západočeská univerzita v Plzni.
- Prefab Factory a.s. (n.d.). O nás. Dostupné 2. 4. 2026 z <https://www.prefabfactory.cz/o-nas/>
- pulp documentation team. (n.d.). Optimization with PuLP. <https://coin-or.github.io/pulp/>
- Python Software Foundation. (n.d.). The Python tutorial. <https://docs.python.org/3/tutorial/>
- Schulz, J. (2021). Applying mathematical optimization in practice. *SN Operations Research*. Springer. <https://doi.org/10.1007/s43069-020-00046-9>
- Shrader, C. R. (2006). *History of operations research in the United States Army: Volume 1, 1942–1962*. Center of Military History, United States Army.
- Šubrt, T., Bartoška, J., Brožová, H., Domeová, L., Houška, M., & Kučera, P. (2011). *Ekonomicko-matematické metody*. Aleš Čeněk.
- Yinusa, A., & Faezipour, M. (2023). Optimizing healthcare delivery: A model for staffing, patient assignment, and resource allocation. *Applied System Innovation*, 6(5), 78. <https://doi.org/10.3390/asi6050078>

# Abstrakt

Bílek, J. (2026). *Aplikace matematického programování pro řešení problému podnikové praxe* [Bakalářská práce, Západočeská univerzita v Plzni].

**Klíčová slova:** matematické programování, operační výzkum, optimalizace, výrobní kapacita, výběr zakázek, prefabrikované moduly

Tato bakalářská práce se zaměřuje na využití matematického programování při řešení vybraného problému podnikové praxe na příkladu společnosti Prefab Factory a.s. V teoretické části jsou vymezeny základní principy rozhodovacího procesu, operačního výzkumu, matematického programování a lineárního programování jako nástroje pro podporu manažerského rozhodování. Praktická část práce představuje vybraný podnik, jeho výrobní program a souvislosti spojené s výrobou prefabrikovaných koupelnových modulů. Pozornost je následně soustředěna na rozhodování o přijetí zakázek při omezené výrobní kapacitě. Za tímto účelem je navržen matematický model, který zohledňuje ekonomický přínos zakázek, jejich kapacitní náročnost, strukturu požadovaných modulů a míru standardizace výroby. Model je implementován v programovacím jazyce Python a využit pro nalezení řešení, které vede k co nejlepšímu využití disponibilní kapacity podniku při maximalizaci celkové marže. Součástí práce je také porovnání výsledků modelu se současnou podnikovou praxí a scénářová analýza zaměřená na zvýšení standardizace výroby. V závěru práce jsou zhodnoceny přínosy i omezení navrženého řešení a formulována doporučení pro podnik.

## Abstract

Bílek, J. (2026). *The use of mathematical programming for solving problems in business practice* [Bachelor's thesis, University of West Bohemia].

**Keywords:** mathematical programming, operations research, optimization, production capacity, order selection, prefabricated modules

This bachelor's thesis deals with the application of mathematical programming to a selected business problem in Prefab Factory a.s. The theoretical part outlines the main principles of decision-making, operations research, mathematical programming, and linear programming as instruments for managerial support. The practical part presents the company, its production portfolio, and the main aspects connected with the manufacture of prefabricated bathroom modules. The attention is then directed to the issue of selecting orders under limited production capacity. For this purpose, a mathematical model is developed that reflects the economic contribution of individual orders, their capacity demands, the structure of required modules, and the level of production standardization. The model is implemented in Python and applied to identify a solution that ensures the most effective use of the company's available capacity while maximizing the total profit margin. The thesis also contains a comparison between the model results and the company's current decision-making practice, as well as a scenario analysis focused on a higher degree of production standardization. The final part evaluates the main benefits and limitations of the proposed approach and formulates recommendations for the company.