

**EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY**

Evidenčné číslo: 103003/I/2023/36122163741409796

**NÁSTROJE OPTIMALIZÁCIE VÝROBNÝCH DÁVOK A
PLÁNOVANIA PRODUKCIE**

Aplikácia v podniku Schaeffler Skalica, spol. s. r. o.

Diplomová práca

**EKONOMICKÁ UNIVERZITA V BRATISLAVE
FAKULTA HOSPODÁRSKEJ INFORMATIKY**

**NÁSTROJE OPTIMALIZÁCIE VÝROBNÝCH DÁVOK A
PLÁNOVANIA PRODUKCIE**

Aplikácia v podniku Schaeffler Skalica, spol. s. r. o.

Diplomová práca

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



Ekonomická univerzita v Bratislave
Fakulta hospodárskej informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Michal Ďurčík
Študijný program: data science v ekonómii (Jednoodborové štúdium,
inžiniersky II. st., denná forma)
Študijný odbor: ekonómia a manažment
Typ záverečnej práce: Inžinierska záverečná práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Nástroje optimalizácie výrobných dávok a plánovania produkcie

Anotácia: Podniky vo výrobnej sfére musia denne realizovať rozhodnutia, aké výrobky má podnik produkovať, v akých výrobných dávkach ich má produkovať, kedy má tieto výrobné dávky produkovať, na ktorých strojoch a v ktorých výrobných fabrikách ich má produkovať či dokonca ako tieto výrobné dávky usporiadať za sebou. Cieľom diplomovej práce je preskúmať a navrhnúť nástroje podpory rozhodovania o produkčných procesoch podniku.

Vedúci: doc. Ing. Marian Reiff, PhD.
Katedra: KOVE FHI - Katedra operačného výskumu a ekonometrie
Vedúci katedry: prof. Mgr. Juraj Pekár, PhD.
Dátum zadania: 25.10.2021

Dátum schválenia: 13.04.2023

prof. Mgr. Juraj Pekár, PhD.
osoba zodpovedná za realizáciu študijného programu

Čestné vyhlásenie

Čestne vyhlasujem, že záverečnú prácu som vypracoval samostatne a že som uviedol všetku použitú literatúru.

Dátum:

.....

Pod'akovanie

Touto cestou by som sa chcel pod'akovať všetkým, ktorí mi akýmkoľvek spôsobom pomohli pri spracúvaní tejto diplomovej práce. Osobitné pod'akovanie by som chcel venovať môjmu vedúcemu práce doc. Ing. Marianovi Reiffovi, PhD. za užitočné rady, podnety a usmernenia pri písaní diplomovej práce. Ďalej by som sa chcel pod'akovať spoločnosti Schaeffler Skalica, spol. s. r. o. a jej zamestnancom za ústretovosť, trpezlivosť, ochotu pri získavaní a poskytovaní dát potrebných pri písaní mojej záverečnej práce. V neposlednom rade by som chcel venovať pod'akovanie zamestnancom spoločnosti z oddelenia MOVE, od ktorých som získal pri písaní práce veľa nových poznatkov a informácií.

Abstrakt

ĎURČÍK, Michal: *Nástroje optimalizácie výrobných dávok a plánovania produkcie - Aplikácia v podniku Schaeffler Skalica, spol. s. r. o.* - Ekonomická univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra operačného výskumu a ekonometrie. - Vedúci záverečnej práce: doc. Ing., Marian Reiff, PhD. - Bratislava: FHI EU, 2023, 90 s.

Cieľom záverečnej práce je priblížiť čitateľovi rôzne modifikácie modelov matematického programovania na optimalizáciu výrobných dávok a plánovania produkcie, poukázať na ich nedostatky a vytvorenie takého modelu matematického programovania, ktorý vystihuje skúmaný výrobný proces. Cieľom diplomovej práce je taktiež vyvinúť nástroj na riešenie vytvoreného modelu matematického programovania, ktorý je aplikovateľný aj v takých výrobných procesoch, ktoré sú v súčasnosti neriešiteľné tradičnými optimalizačnými technikami vzhľadom na ich vysokú komplexnosť a výkonnosť súčasnej výpočtovej technológie. Výsledkom záverečnej práce je aplikácia vytvoreného nástroja v skúmanom výrobnom procese.

Práca je rozdelená do štyroch kapitol, pričom prvú kapitolu venujeme úvodu do problematiky optimalizácie výrobných dávok a plánovania produkcie, historickým štádiám jej vývoja a taktiež jej teoretickému zadefinovaniu. V ďalšej kapitole je zadefinovaný hlavný cieľ tejto práce. Tretia kapitola obsahuje teoretické vysvetlenie metód použitých pri riešení nami skúmanej problematiky. Záverečná kapitola sa zaoberá podrobným opisom skúmaného objektu, modifikáciou úlohy matematického programovania a opisom dát, ktoré sú nevyhnutné na jej riešenie. Ďalej je tu opísaný vývoj genetického algoritmu na riešenie optimalizačnej úlohy. Táto kapitola taktiež obsahuje riešenie úlohy matematického programovania exaktným a genetickým algoritmom, analýzu získaných výsledkov a aplikáciu genetického algoritmu v reálnom výrobnom procese. Záver štvrtej kapitoly pojednáva o možnostiach integrácie vytvoreného genetického algoritmu do reálnych výrobných procesov.

Kľúčové slová

optimalizácia, plánovanie produkcie, zmiešané celočíselné programovanie, genetický algoritmus

Abstract

ĎURČÍK, Michal: *Lotsizing and production scheduling optimization methods - Application in the company Schaeffler Skalica, spol. s. r. o.* - University of Economics in Bratislava. Faculty of Economic Informatics; Department of Operations research and Econometrics. - Supervisor of the Diploma thesis: doc. Ing., Marian Reiff, PhD. - Bratislava: FHI EU, 2023, number of pages 90.

The aim of this thesis is to explain to the reader various modifications of mathematical programming models for optimizing production batches and production planning, to show their deficiencies and to create a mathematical programming model that represents the examined production process. Furthermore, the purpose of this thesis is to develop a tool for solving a created mathematical programming model which is applicable in production processes which are, at present, irresolvable by traditional optimization techniques due to their high complexity and efficiency of present information technology, and subsequent application of the tool in the examined production process.

The thesis is divided into four chapters. The first chapter is dedicated to the introduction of the topic of the optimization of production batches in production planning; historic stages of its development as well as its theoretical definition. In the next chapter, the main goal of the thesis is defined. The third chapter includes a theoretical explanation of methods used in solving the examined problem. The final chapter deals with a detailed description of the examined subject, modification of the problem of mathematical programming and the description of the data that is crucial for solving the problem. Next, the development of the genetic algorithm for solving the optimization problem is described. This chapter also includes solving the mathematical programming problem by means of an exact and genetic algorithm, analysis of obtained results and application of the genetic algorithm in a real production process. The conclusion of Chapter four discusses the possibilities of integration of the created genetic algorithm into real production processes.

Key words

optimization, production planning, mixed integer programming, genetic algorithm

Obsah

Zoznam ilustrácií a zoznam tabuliek	8
Úvod	10
1 Súčasný stav riešenej problematiky	12
1.1 Výrobné koncepcie	12
1.2 Plánovanie produkcie	15
1.3 Teória výpočtovej zložitosti	16
1.4 Vývoj nástrojov na optimalizáciu výrobných dávok a plánovania	17
2 Cieľ práce	21
3 Metodika práce a metódy skúmania	22
3.1 Modely matematického programovania	22
3.1.1 <i>CLSP model</i>	23
3.1.2 <i>CLSD-SM model</i>	25
3.1.3 <i>CLSD-PM model</i>	28
3.2 Genetický algoritmus	31
3.3 Softvérové nástroje na riešenie optimalizačných úloh	36
4 Výsledky práce a diskusia	38
4.1 Charakter spoločnosti	38
4.2 Charakteristika skúmaného objektu	38
4.3 Modifikácia modelu CLSD-PM pre nami skúmaný objekt	42
4.4 Vstupné dáta	45
4.5 Exaktné riešenie optimalizačnej úlohy	48
4.6 Riešenie optimalizačnej úlohy genetickým algoritmom	58
4.6.1 <i>Konštrukcia genetického algoritmu</i>	58
4.6.2 <i>Aplikácia genetického algoritmu na redukovanom súbore údajov</i>	65
4.6.3 <i>Porovnanie výsledkov genetického a exaktného algoritmu</i>	75
4.6.4 <i>Aplikácia genetického algoritmu na kompletnom súbore údajov</i>	78
4.7 Diskusia	83
Záver	85
Zoznam použitej literatúry	87
Prílohy	90

Zoznam ilustrácií a zoznam tabuliek

Zoznam grafov

1	Percentuálne využitie linky EQ-85025917	53
2	Percentuálne využitie linky EQ-85025921	54
3	Rozloženie nákladov	55
4	Celkové rozloženie nákladov	55
5	Priebeh riešenia redukovaného problému genetickým algoritmom	67
6	Percentuálne využitie linky EQ-85025917	70
7	Percentuálne využitie linky EQ-85025921	71
8	Rozloženie nákladov	72
9	Celkové rozloženie nákladov	73
10	Porovnanie nákladových metrík	76
11	Porovnanie percentuálneho využitia strojov	77
12	Priebeh riešenia kompletného problému genetickým algoritmom	79
13	Percentuálne využitie linky EQ-85025917	80
14	Percentuálne využitie linky EQ-85025921	81
15	Rozloženie nákladov	82
16	Celkové rozloženie nákladov	83

Zoznam obrázkov

1	Genetický algoritmus	36
2	Ilustratívny príklad výrobného procesu ložísk	39
3	Montážna úroveň produkcie	41
4	Výstup z Gurobi solvera	49
5	Návrh chromozómu	59
6	Algoritmus inicializácie	60
7	Algoritmus rekombinácie	61
8	Operátor rekombinácie	62
9	Algoritmus operátora mutácie	63

Zoznam tabuliek

1	Porovnanie výrobných koncepcií	14
2	Rozhodovacie premenné v modeli CLSP	24
3	Parametre v modeli CLSP	24
4	Rozhodovacie premenné v modeli CLSD-SM	27
5	Parametre v modeli CLSD-SM	27
6	Rozhodovacie premenné v modeli CLSD-PM	30
7	Parametre v modeli CLSD-PM	30
8	Rozhodovacie premenné v modeli CLSD-PM so sklzom	44
9	Parametre v modeli CLSD-PM so sklzom	44
10	Výrobný plán pre linku EQ-85025917	50
11	Výrobný plán pre linku EQ-85025921	51
12	Kapacitná analýza linky EQ-85025917	52
13	Kapacitná analýza linky EQ-85025921	52
14	Celkové náklady	54
15	Nákladová analýza	57
16	Výrobný plán pre linku EQ-85025917	68
17	Výrobný plán pre linku EQ-85025921	69
18	Kapacitná analýza linky EQ-85025917	70
19	Kapacitná analýza linky EQ-85025921	71
20	Celkové náklady	72
21	Nákladová analýza	74
22	Kapacitná analýza linky EQ-85025917	80
23	Kapacitná analýza linky EQ-85025921	81
24	Celkové náklady	82

Úvod

V súčasnej dobe, sme svedkami rôznych neočakávaných udalostí, ktoré priamo či nepriamo ovplyvňujú život každého z nás, ľudí žijúcich na planéte. Príkladom takýchto udalostí je pandémia COVID-19, či vojna na Ukrajine. Tieto, často nepredpokladateľné udalosti spôsobujú vysokú volatilitu v ekonomike. Pandémia COVID-19 spôsobila sektorový šok v cestovnom ruchu, vojna na Ukrajine zase enormný nárast cien energií. Firmy pôsobiace v podmienkach trhových ekonomík musia byť schopné prispôbiť sa takýmto šokom. V opačnom prípade by mohli prísť o svoje postavenie na trhu a následne by im mohol hroziť zánik. Z tohto dôvodu je pre firmy dôležité, hľadať rôzne nástroje, ako sa v týchto neustále meniacich podmienkach prispôbiť trhu a vývoju cien.

Aby sa podniky dokázali efektívne prispôbovať trhovým podmienkam, musia byť schopné efektívne a pružne alokovať svoje vzácne a veľakrát obmedzené výrobné zdroje. V rámci priemyselnej výroby ide najmä o alokáciu kapitálu. Pre priemyselné podniky je cieľom vyhnúť sa vytváraniu nadbytočných a nepotrebných zásob. Vytváranie zásob totiž zadržiava finančný kapitál, ktorý by v opačnom prípade mohla firma využiť na produkciu výrobkov pre iných zákazníkov. Ďalším problémom, ktoré nadbytočné zásoby spôsobujú, je možné znehodnotenie zásob. Mnohé výrobky sú náchylné na skazu spôsobenú prírodnými vplyvmi. Ako príklad by sme mohli uviesť skazu produkcie vplyvom korózie, skazu potravinových výrobkov vplyvom prekročenia záručnej lehoty či starnutie plastov, ktoré výrobky môžu obsahovať. V neposlednom rade zásoby alokujú plochy, ktoré by podnik v opačnom prípade mohol použiť na umiestnenie výrobných zariadení, ktoré by podniku mohli priniesť dodatočný výnos. Nízke zásoby znamená vyrábať iba vtedy, keď to zákazník požaduje. Z tohto uhla pohľadu je pre podnik teda výhodné vyrábať iba v takých výrobných dávkach, ktoré sa rovnajú potrebám zákazníkov. Tento uhol pohľadu je ale často spochybňovaný podnikovými ekonómami.

Ak by podnik vyrábal iba v minimálnych výrobných dávkach, vznikli by podniku iné náklady spôsobené častou prestavbou strojov. Pred začatím výroby určitého výrobku je potrebné stroj nastaviť tak, aby daný výrobok mohol vyrábať. Takúto prestavbu nazývame pretypovanie stroja. V prípade, ak sú potreby zákazníkov veľmi nízke a čas prestavby stroja vysoký, môže nastať situácia, že vytvorenie adekvátnej zásoby bude pre firmu výhodnejšie, ako vyrábať veľmi malé výrobné dávky. Rovnaké platí aj v prípade, že trhovú rovnováhu

ohrozujú ponukové šoky. V rámci realizácie správneho rozhodnutia musí podnik voliť istý kompromis. Tu sa nám naskytá otázka: Ako zaujať ten správny postoj a zvoliť ideálny kompromis medzi nízkymi zásobami a nákladmi na pretypovanie stroja?

1 Súčasný stav riešenej problematiky

1.1 Výrobné koncepcie

Mnohé podniky v uplynulých rokoch prechádzali zo zaužívaných koncepcií hromadnej výroby na koncepciu štíhlej výroby, ktorá kladie dôraz na kvalitu a rýchlosť reakcie prispôsobenia sa neustále sa meniacim požiadavkám zákazníkov.¹

Štíhla výroba je výrobná filozofia, ktorá sa snaží odstrániť plytvanie a zvýšiť efektívnosť výroby. Snaží sa maximalizovať hodnotu pre zákazníka tým, že dodáva vysokokvalitné výrobky pri čo najnižších možných nákladoch a v čo najkratšom čase. V štíhlej výrobe sa zdôrazňuje neustále zlepšovanie, účasť zamestnancov a dôraz na minimalizáciu plytvania vo všetkých formách vrátane nadvýroby, čakania, chýb, nadmerného spracovania, nadmerných zásob či zbytočného pohybu. Cieľom štíhlej výroby je vybudovať vysoko efektívny, prispôsobivý a pohotovo reagujúci výrobný systém, ktorý sa dokáže rýchlo prispôbiť meniacim sa potrebám spotrebiteľov.²

Zrod koncepcie štíhlej výroby sa odohral v Japonsku v spoločnosti Toyota v štyridsiatych rokoch dvadsiateho storočia. Výrobný systém Toyota bol založený na myšlienke, že len malá časť celkového produkčného času potrebného na spracovanie výrobku pridáva hodnotu konečnému zákazníkovi. Tento systém nehl'adal efektívnosť výroby vo veľkých výrobných dávkach. Bol to presný opak toho, v čom videl efektívnosť výroby západný svet. V tomto období sa na Západe vyvíjal koncept hromadnej výroby Henryho Forda.³

Hromadná výroba je výrobný systém založený na koncepte veľkoobjemovej produkcie s minimálnym množstvom prestavieb strojov. Hromadná výroba je opačným prístupom k filozofii štíhlej výroby, ktorá je založená na myšlienke, vyrábať len vtedy, keď to zákazník vyžaduje. Hromadná výroba sa vyznačuje tzv. princípom tlaku, kde produkcia je plánovaná na dlhšie časové obdobie na základe predikcie dopytu

¹KLIER, Thomas H. How lean manufacturing changes the way we understand the manufacturing sector [elektronický zdroj]. 1993. 9 s. Dostupné na: <https://www.chicagofed.org/-/media/publications/economic-perspectives/1993/ep-may-jun1993-part1-klier-pdf>

²AKERS, Paul A. 2 Second Lean. United States of America : FastCap Press, 2019. 154 s. ISBN 978-0-984-66228-9.

³MELTON, T. Benefits of lean manufacturing : What lean thinking has to offer the process industries [elektronický zdroj]. 2005. 9 s. Dostupné na: <http://mimesolutions.com/PDFs/WEB\%20Trish\%20Melton\%20Lean\%20Manufacturing\%20July\%202005.pdf>

do budúcnosti. Výroba sa realizuje podľa predpokladaných požiadaviek zákazníkov aj v prípade, že títo zákazníci ešte nezadali objednávky. Štíhla výroba sa vyznačuje tzv. princípom ťahu, ktorý je založený na aktuálnych objednávkach zákazníkov. Hneď ako príde objednávka, začne sa s výrobou daného výrobku. Neexistuje žiadne predvídanie ani prognózovanie dopytu. Pri princípe ťahu sa žiadny výrobok nevyrába, kým naň nie je vyhotovená objednávka.

Hlavné rozdiely medzi koncepciou štíhlej výroby a tradičnou koncepciou hromadnej výroby možno zhrnúť v nasledujúcich bodoch.

- Systém hromadnej produkcie sa zameriava na výrobu vo veľkých výrobných dávkach. Cieľom tejto koncepcie je vyrobiť čo najviac výrobkov v jednej výrobnej sérii. Naopak, koncepcia štíhlej výroby sa zameriava na výrobu len takého počtu výrobkov, aby pokryla požiadavky zákazníkov.
- Systém hromadnej výroby sa vyznačuje vysokými skladovými zásobami. Položky z vytvorených zásob sa následne dodávajú distribútorom, ktorí distribuujú tieto položky do obchodov. Metóda štíhlej výroby často poskytuje tovar priamo zákazníkovi. Snaha je vytvárať iba minimálne nevyhnutné množstvo zásob. Štíhla výroba sa zameriava na schodnosť rýchlej reakcie na zmeny požiadavky zákazníka. Z toho dôvodu sú zásoby v systéme štíhlej výroby nežiadúce.
- Plánovanie hromadnej výroby závisí od množstva zložitých aspektov, ako sú napríklad: konkurencia, čas distribúcie, úroveň zásob, ceny na trhu alebo dodatočná výroba potrebná z dôvodu propagácie výrobkov. Takéto plánovanie je komplikované a vyžaduje si použitie pokročilých nástrojov na úrovni podniku. Koncepcia štíhlej výroby je naopak založená na aktuálnom dopyte zákazníkov a plánuje sa preto oveľa jednoduchšie. Vyrába sa iba vtedy, keď zákazník potvrdí objednávku.
- Pri koncepcii hromadnej výroby vzniká veľké množstvo odpadu, napríklad vplyvom prekročenia záručných lehôt zásob či skazy spôsobenej skladovaním tovaru. Štíhla výroba sa zase vyznačuje snahou o minimalizáciu akéhokoľvek odpadu.
- Koncepcia hromadnej výroby je vo všeobecnosti zameraná na ponuku, pričom koncepcia štíhlej výroby sa zameriava na zákazníka a jeho aktuálne požiadavky.

- V systémoch hromadnej výroby je výroba prispôsobená na produkciu rovnakého výrobku vo veľkých výrobných dávkach a z toho dôvodu, je častokrát vybavená ťažkými strojmi a výrobnými linkami, ktoré sú časovo náročné na prestavbu. V systémoch štíhlej výroby sa však vyskytujú skôr ľahké stroje, ktoré sú časovo nenáročné na prestavbu.⁴

Tabuľka 1: Porovnanie výrobných koncepcií

Hromadná výroba	Štíhla výroba
Veľké výrobné dávky	Malé výrobné dávky
Zriedkavá prestavba strojov	Častá prestavba strojov
Veľké zásoby	Minimálne zásoby
Zložité plánovacie nástroje	Plánovanie na základe objednávok zákazníkov
Veľké množstvo odpadu	Minimalizácia odpadu
Zameraná na ponuku	Zameraná na zákazníka
Ťažké výrobné zariadenia	Jednoduché výrobné zariadenia

Zdroj: vlastné spracovanie, podľa⁴

Výkonnosť a vhodnosť konkrétnej výrobnéj koncepcie vo veľkej miere závisí od situácie na trhu a správneho fungovania trhového mechanizmu. Vplyvom ponukových alebo dopytových šokov vzniká trhová nerovnováha. Dopytové a ponukové šoky sú náhle, neočakávané zmeny v dopyte po tovaroch a službách, resp. v ponuke tovarov a služieb v ekonomike. Ak dôjde k rýchlej a neočakávanej zmene v množstve výrobkov a služieb, ktoré sú spotrebitelia ochotní si zaobstarať, krivka dopytu sa posunie, čím vzniká dopytový šok. Môžu k tomu prispieť zmeny v príjmoch spotrebiteľov, zmeny v ich preferenciách alebo zmeny v dôvere spotrebiteľov. Naopak, ak dôjde k náhlejšej a neočakávanej zmene množstva výrobkov a služieb, ktoré sú výrobcovia ochotní a schopní vyprodukovať, krivka ponuky sa posunie, čím vzniká ponukový šok. Môže to byť spôsobené najmä prírodnými katastrofami, narušením dodávateľských reťazcov, technickými zlepšeniami, zmenami výrobných nákladov a podobne.

V posledných rokoch sme svedkami rôznych šokov na strane ponuky a rovnako aj na strane dopytu. Pandémia COVID-19 prispela k ponukovým šokom spôsobených narušením dodávateľských reťazcov. Vojna na Ukrajine spôsobila enormný nárast cien energií. V týchto spomenutých prípadoch profitovali najmä spoločnosti, ktoré disponovali

⁴ MIČIETOVÁ, Mária. LEAN PRODUCTION, LEAN VS. MASS PRODUCTION, TPM AS A TOOL OF LEAN PRODUCTION [elektronický zdroj]. 2011. 7 s. Dostupné na : <https://pernerscontacts.upce.cz/index.php/perner/article/view/918/752>

väčšími zásobami a mohli z nich náhle výkyvy kompenzovať. Z historických pozorovaní možno konštatovať, že koncepcie hromadnej výroby sú vhodnejšie v prípade ponukových šokov. V prípade dopytových šokov sú efektívnejšie naopak koncepcie štíhlej výroby. Vďaka nízkym zásobám a jednoduchým prispôsobiteľným výrobným zariadeniam dokážu pružnejšie reagovať na zmeny v potrebách zákazníkov či zmeny v trendoch a inováciách. Z uvedeného konštatujeme, že v súčasných ekonomických podmienkach je vhodné voliť určitý kompromis.

1.2 Plánovanie produkcie

„Plánovaním produkcie sa chápe plánovanie obstarávania zdrojov a surovín, a plánovanie výrobných činností potrebných na transformáciu surovín na hotové výrobky tak, aby boli naplnené potreby zákazníkov čo najefektívnejším a najekonomickejším spôsobom.“⁵

Podniky vo výrobnjej sfére musia denne realizovať dôležité rozhodnutia. Rozhodnutia o tom, aké výrobky má podnik produkovať, v akých výrobných dávkach ich má produkovať, kedy má tieto výrobné dávky produkovať, na ktorých strojoch a v ktorých výrobných fabrikách ich má produkovať, či dokonca, ako tieto výrobné dávky usporiadať za sebou. Úlohou je prijať také rozhodnutia, aby boli splnené ciele podniku v čo najvyššej možnej miere. Ako sme už spomenuli, každá z výrobných koncepcii si kladie za cieľ iné kritériá. Keď, cieľom systému hromadnej výroby je minimalizácia počtu prestavieb strojov, koncepcia štíhlej výroby sa usiluje o minimalizáciu zásob.

Môže však nastať situácia, keď sú podnikové ciele do istej miery protichodné. Taká situácia nastáva v prípade, že podnik volí určitý kompromis medzi jednotlivými výrobnými koncepciami. Ako možný konflikt optimalizačných kritérií je minimalizácia zásob, čo vidíme najmä z koncepcie štíhlej výroby, pri súčasnej minimalizácii nákladov na prestavbu stroja, čím sa vyznačuje zase koncepcia hromadnej výroby. V takomto prípade hľadáme optimálny kompromis medzi stanovenými ekonomickými cieľmi, pričom je však potrebné uspokojiť potreby zákazníkov.

⁵POCHET, Yves - WOLSEY, Laurence A. Production Planning by Mixed Integer Programming. New York : Springer Science + Business Media, Inc., 2006. s. 3. ISBN 978-1-4419-2132-1.

1.3 Teória výpočtovej zložitosti

Teória výpočtovej zložitosti je interdisciplinárny študijný odbor, ktorý sa zaoberá skúmaním náročnosti riešenia výpočtových problémov z hľadiska zdrojov (čas či pamäť) potrebných na ich riešenie. Teória zložitosti stanovuje existenciu rozhodnuteľných problémov, ktoré sú síce principiálne riešiteľné, no nemusia byť riešiteľné v praxi, a to v prípade, ak si riešenie vyžaduje obrovské množstvo času a pamäte.⁶

Aby sme boli schopní nájsť riešenie vyhovujúce všetkým stanoveným kritériám, nástroje plánovania produkcie sa stávajú čoraz komplikovanejšie a sofistikovanejšie. Vo všeobecnosti problémy plánovania produkcie a optimalizácie výrobných dávok sú opísateľné matematickými modelmi, ktoré sú rozhodnuteľné a sú v princípe riešiteľné počítačom, avšak v závislosti od náročnosti problému na čas či pamäť nemusia byť takéto problémy riešiteľné v praxi.⁷

V závislosti od zložitosti výrobného procesu sú na optimalizáciu výrobných dávok a plánovania využívané rôzne metódy. Pri triviálnych výrobných procesoch sa často využívajú jednoduché analytické modely či modely lineárneho programovania, ktoré sú v praxi ľahko riešiteľné. Avšak v reálnom svete sa zväčša stretávame so zložitými výrobnými procesmi, ktorých špecifiká nie je možné opísať jednoduchým modelom. V takých prípadoch sa využívajú modely zmiešaného celočíselného programovania, ktoré sú často vzhľadom na výpočtovú zložitosť veľmi ťažko riešiteľné či dokonca pri väčšom počte rozhodovacích premenných neriešiteľné. V snahe doručiť určité riešenie aj pre neriešiteľné úlohy matematického programovania sa v súčasnosti vyvíjajú rôzne heuristické metódy.⁸

Ako miery náročnosti riešenia problému používame čas a priestor (pamäť) potrebné na ich riešenie. Náročnosť problému sa vyjadruje ako nárast potrebného času alebo priestoru v závislosti od dĺžky n vstupného slova, ktoré reprezentuje inštanciu problému. Vždy uvažujeme o najhoršom prípade zložitosti, čo je, pre každé n , najvyšší potrebný čas alebo priestor spomedzi všetkých vstupov dĺžky n .⁹

⁶ALLENDER, Eric W. - LOUI, Michael C. - REGAN, Kenneth W. Complexity Theory [elektronický zdroj]. 2012. 35 s. Dostupné na: <https://people.cs.rutgers.edu/~allender/papers/ALR12.pdf>

⁷SIPSER, Michael. Introduction to the Theory of Computation. 2. vyd. United States of America : Thomson Learning, Inc., 2006. 431 s. ISBN 0-534-95097-3.

⁸POCHET, Yves - WOLSEY, Laurence A. Production Planning by Mixed Integer Programming. New York : Springer Science + Business Media, Inc., 2006. 481 s. ISBN 978-1-4419-2132-1.

⁹ALLENDER, Eric W. - LOUI, Michael C. - REGAN, Kenneth W. Complexity Theory [elektronický zdroj]. 2012. 35 s. Dostupné na: <https://people.cs.rutgers.edu/~allender/papers/ALR12.pdf>

V závislosti od zdrojov potrebných na riešenie problému, delíme výpočtové problémy do viacerých zložitostných tried. Najskloňovanejšie sú však v súčasnosti triedy P a NP. Do triedy P zaradíme také problémy, na ktorých riešenie existujú algoritmy, ktoré ich dokážu riešiť v polynomiálnom čase. Ak má algoritmus polynomiálnu časovú zložitosť, čas potrebný na vyriešenie problému je priamoúmerný polynómu veľkosti vstupných údajov. Inak povedané, počet operácií potrebných na vyriešenie problému narastá polynomiálnou funkciou vstupu. Vo všeobecnosti sa problémy spadajúce do tejto triedy so súčasnou výpočtovou technikou, považujú za pomerne jednoducho riešiteľné aj pre väčšie rozmery úloh. Do zložitostnej triedy NP zaradíme také problémy, ktorých riešenie dokážeme overiť v polynomiálnom čase, avšak neexistujú algoritmy, ktoré by také problémy dokázali riešiť v polynomiálnom čase. Problémy, ktoré sa vyznačujú zložitou NP, sú často len ťažko riešiteľné už pri malých rozmeroch úloh. Žiaľ, veľké množstvo problémov reálneho sveta patrí práve do tejto skupiny úloh. Rovnako to je aj v prípade mnohých nástrojov optimalizácie výrobných dávok a plánovania.¹⁰

1.4 Vývoj nástrojov na optimalizáciu výrobných dávok a plánovania

Výskum v oblasti nástrojov na optimalizáciu výrobných dávok a plánovania sa začal klasickým deterministickým modelom zásob EOQ (Economic order quantity), ktorý bol prvýkrát publikovaný Fordom W. Harrisom v roku 1913. Tento model stanovuje optimálnu veľkosť objednávky surovín alebo komponentov vzhľadom na náklady spojené s ich skladovaním a náklady spojené s ich dodaním. Model EOQ bol publikovaný s cieľom poskytnúť manažérom nástroj umožňujúci zjednodušenie rozhodovania pri objednávaní surovín a komponentov od dodávateľov.

Neskôr bol v roku 1918 E. W. Taftom publikovaný model EPQ (Economic production quantity), ktorý je rozšírením modelu EOQ. Hlavným rozdielom týchto dvoch modelov je, že model EOQ predpokladá, že objednané množstvo je k dispozícii ihneď po objednaní v celej dávke. Naopak, EPQ model predpokladá, že spoločnosť vyrába svoje vlastné výrobky, ktoré sú k dispozícii pre ďalšie spracovanie či predaj jeden po druhom a nie v dávkach. Z týchto dôvodov je model EPQ vhodnejší na aplikáciu vo výrobných podnikoch, ktoré realizujú svoju vlastnú produkciu.

¹⁰ARORA, Sanjeev. - BARAK, Boaz. COMPUTATIONAL COMPLEXITY : A Modern Approach. New York : Cambridge University Press, 2009. 579 s. ISBN 978-0-521-42426-4.

Predpoklady modelov typu EOQ sú ale značne limitujúce. Jedným z predpokladov je jednoúrovňový produkčný proces. V súčasnosti sú ale produkčné procesy natoľko zložité, že dodržať tento predpoklad je takmer nemožné. Dnešné výrobky sú často veľmi sofistikované a vyžadujú náročné procesy zložené z viacerých operácií. Ďalším nedostatkom je, že modely typu EOQ nezahŕňajú kapacitné ohraničenia. V praxi to znamená, že plánovač musí kapacitné ohraničenia zohľadniť ex post, čím sa môže odkloniť od optimálneho riešenia. Ďalším nedostatkom je predpoklad stacionarity potrieb zákazníkov. Model predpokladá, že zákazník dopytuje produkciu pravidelne v rovnakých výškach.

Avšak aj napriek týmto obmedzeniam nemôžeme podceňovať vplyv spomenutých modelov na neskorší výskum v oblasti nástrojov optimalizácie výrobných dávok. Práve z dôvodu ich reštriktívnosti boli neskôr vyvinuté iné typy modelov, ktoré nám umožňujú presnejšie opísať realitu. Mnohé z týchto modelov sú založené na základných poznatkoch pochádzajúcich z modelov EOQ či EPQ. Model EOQ je taktiež dôležitou súčasťou histórie operačného výskumu, nakoľko predstavuje jednu z prvých publikovaných aplikácií matematického modelu na rozhodovanie v podnikaní.¹¹

Prvý z modelov, kde boli zohľadnené kapacity výroby, bol ELSP model (Economic lot scheduling problem) publikovaný v roku 1958 Jackom Rogersom v práci A Computational Approach to the Economic Lot Scheduling Problem. Cieľom modelu je nájsť taký výrobný plán, ktorý je prípustný vzhľadom na kapacitné ohraničenie, pričom viaceré výrobky sú vyrábané v dávkach na jednom stroji a súčasne minimalizuje náklady. Tento model však predpokladal iba stacionárny dopyt. Napriek reštriktívnym predpokladom je ELSP kvôli svojej nelineárnosti, kombinatorickým vlastnostiam a zložitosti všeobecne známy ako NP-tiažký problém.¹²

Ďalším modelom, ktorý už predpokladal variabilný dopyt v čase (ide o dynamický model), bol Wagner-Whitin. Tento model je pomenovaný po jeho tvorcach, ktorí ho prvýkrát publikovali v roku 1958 v práci „Dynamic Version of the Economic Lot Size Model“.¹³ Harvey M. Wagner a Thomson M. Whitin ho definovali ako dynamickú verziu modelu EOQ. Je to model, ktorý predpokladá konečný plánovací horizont, ktorý je rozdelený na niekoľko

¹¹CHOI, Tsan-Ming. Handbook of EOQ Inventory Problems : Stochastic and Deterministic Models and Applications. New York : Springer Science + Business Media, Inc., 2014. 279 s. ISBN 978-1-4614-7638-2.

¹²ROGERS, Jack. A Computational Approach to the Economic Lot Scheduling Problem [elektronický zdroj]. 1958. 28 s. Dostupné na: <https://doi.org/10.1287/mnsc.4.3.264>

¹³WAGNER, Harvey M. - WHITIN, Thomson M. Dynamic Version of the Economic Lot Size Model [elektronický zdroj]. 1958. 9 s. Dostupné na: <http://www.jstor.org/stable/2626974>

podobdobí. Dopyt je stanovený pre každé podobdobje zvlášť a môže sa líšiť v čase. Wagner-Whitin model však nezahrňa obmedzenie kapacít.¹⁴

V snahe čo najpresnejšie opísať výrobný proces bolo následne vyvíjané nespočetné množstvo najmä modelov zmiešaného celočíselného programovania. Výskumníci sa zameriavali na adaptáciu týchto modelov na rôzne produkčné systémy. Vznikalo tak veľké množstvo modelov, ktoré sa vyznačovali špecifikami výrobných systémov, na ktoré boli adaptované. Vzhľadom na extenzívny výskum v rámci modelov je nemožné v tejto práci zhrnúť celý ich vývoj, avšak v nasledujúcej kapitole si predstavíme modely zmiešaného celočíselného programovania, z ktorých sme vychádzali pri konštrukcii modelu pre nami skúmaný výrobný systém.

Je dôležité poznamenať, že napriek tomu, že pomerne sofistikované modely boli vytvorené už v 90. rokoch 20. storočia, stále sú vzhľadom na výkonnosť súčasnej výpočtovej techniky veľmi náročné na riešenie. Zväčša sú to úlohy NP-t'ázké. V súčasnosti nedisponujeme metódami, ktoré by nám umožňovali nájsť optimálne riešenie pre takéto náročné problémy. V snahe doručiť riešenie aj pre úlohy, ktoré sú dodnes neriešiteľné štandardnými exaktnými metódami, boli v posledných desaťročiach vyvinuté viaceré heuristické nástroje, ktoré uľahčujú ich riešenie.

Takéto metódy však veľakrát neumožňujú nájsť exaktné optimálne riešenie, avšak umožňujú nájsť relatívne dobré riešenia vzhľadom na čas riešenia daného problému. Algoritmy poskytujúce približné riešenia, avšak nie exaktné, radíme do triedy heuristických algoritmov. Heuristické algoritmy možno ďalej rozdeliť na špecifické heuristické algoritmy a na metaheuristické algoritmy. Ich hlavným rozdielom je, že špecifické heuristické algoritmy sú závislé od problému, na ktorého riešenie sú navrhnuté, pričom metaheuristické metódy sú všeobecne použiteľné na riešenie veľkého množstva optimalizačných problémov.

Metaheuristiky sú vhodné na riešenie takých problémov, ktorých množina riešení je veľmi veľká, avšak na rozdiel od exaktných optimalizačných algoritmov metaheuristiky negarantujú optimálnosť získaných riešení. Na rozdiel od štandardných exaktných optimalizačných techník, ktoré sú často založené na prehl'adávaní „hrubou silou“, metaheuristické algoritmy sú založené na efektívnom zmenšení priestoru

¹⁴BURTSEVA, Larysa et al. Production Planning and Scheduling for Lot Processing. Newcastle upon Tyne : Cambridge Scholars Publishing, 2022. 397 s. ISBN 978-1-5275-8502-7.

prehľadávania. Hlavnými výhodami metaheuristického prístupu k riešeniu optimalizačných úloh sú: rýchlejšie riešenie problému, možnosť riešiť aj úlohy veľkých rozmerov, sú jednoducho navrhnutelné a implementovateľné. Väčšina metaheuristických nástrojov je založená na napodobňovaní prirodzených procesov z prírody.¹⁵ Medzi metaheuristické algoritmy patria napríklad:

- genetický algoritmus,
- simulované chladenie,
- algoritmus roja častíc,
- zakázané prehľadávanie,
- mravčie algoritmy.

¹⁵EL-GHAZALI, Talbi. METAHEURISTICS FROM DESIGN TO IMPLEMENTATION. New Jersey : John Wiley & Sons, Inc., 2009. 618 s. ISBN 978-0-470-27858-1.

2 Cieľ práce

Hlavným cieľom diplomovej práce je vytvorenie takého nástroja, ktorý umožňuje priemyselnému podniku prijímať pravidelné rozhodnutia o tom, kedy a aké výrobky má produkovať, v akých výrobných dávkach ich má produkovať a ako produkciu jednotlivých výrobkov usporiadať za sebou tak, aby boli náklady na skladovanie produkcie, náklady na prestavbu strojov a náklady spojené s oneskoreným dodaním produkcie zákazníkovi minimálne. Nástroj má ponúknuť podniku výrobný plán, ktorý tvorí optimálny kompromis medzi jednotlivými nákladovými metrikami.

Na splnenie hlavného cieľa je však potrebné splniť skupinu čiastkových cieľov. Prvý z čiastkových cieľov je dokonalé spoznanie skúmaného výrobného procesu. Aby sme sa vyhli klamu kompozície, budeme skúmať výrobný proces ako previazaný systém, pretože čo je dobré pre časť, ešte neznamená, že bude dobré aj pre celok. Na získanie optimálneho výrobného plánu budeme formulovať optimalizačný model matematického programovania. Každý optimalizačný model však poskytuje iba také dobré výsledky, ako dobre vystihuje realitu. Úlohu matematického programovania budeme formulovať tak, aby v čo najväčšej možnej miere vystihovala špecifiká nami skúmaného objektu.

Ďalším čiastkovým cieľom bude získavanie údajov. Keďže optimalizáciu výrobných plánov budeme realizovať v podniku Schaeffler Skalica, spol. s r. o., údaje potrebné na jej vykonanie budeme získavať z jej podnikových databáz, prostredníctvom konzultácií s pracovníkmi spoločnosti či iných relevantných zdrojov poskytnutých spoločnosťou.

Následne budeme formulovaný optimalizačný model riešiť. Tradičné optimalizačné techniky však vzhľadom na výkonnosť súčasnej výpočtovej technológie pri riešení komplexných optimalizačných úloh často zlyhávajú. Cieľom tejto práce je vytvorenie takého nástroja na riešenie formulovaného modelu matematického programovania, ktorý je aplikovateľný aj v takých výrobných procesoch, ktoré sú v súčasnosti neriešiteľné tradičnými optimalizačnými technikami vzhľadom na ich vysokú komplexnosť a výkonnosť súčasnej výpočtovej technológie. Vytvorený nástroj budeme porovnávať s exaktným prístupom riešenia na redukovanom súbore údajov, zhodnotíme jeho efektívnosť a následne optimalizačný nástroj aplikujeme v reálnom výrobnom procese.

3 Metodika práce a metódy skúmania

3.1 Modely matematického programovania

„Modely a metódy matematického programovania umožňujú transformovať reálne ekonomické procesy do matematických modelov predovšetkým v súvislosti s riešením efektívneho využívania ohraničených disponibilných zdrojov (surovín, materiálu, časového fondu strojov a zariadení, plôch, počtu pracovníkov, investičných a finančných prostriedkov a podobne), ktoré sú potrebné na dosiahnutie stanovaných cieľov. Ide teda o relatívne presne kvantitatívne hodnotenie výberu určitého riešenia z veľkého množstva možných riešení, ktoré je z hľadiska matematicky formulovaného cieľa najlepšie.“¹⁶

Matematické programovanie je metóda operačného výskumu a súčasť vedného odboru aplikovanej matematiky, ktorá nám umožňuje nájsť najlepšie možné riešenie spomedzi všetkých prípustných riešení. Najlepším riešením chápeme také riešenie, ktoré extremalizuje účelovú funkciu pri splnených ohraničujúcich podmienkach. Účelová funkcia je funkcia, prostredníctvom ktorej vyjadrujeme optimalizačné kritériá. Optimalizačné kritérium je cieľ, ktorý sa usilujeme splniť v čo najlepšej možnej miere. V prípade, ak sa usilujeme o dosiahnutie viacerých cieľov, hovoríme o úlohách viackritériálnej optimalizácie. Riešenia, ktoré vyhovujú všetkým štruktúrnym ohraničeniam, nazývame prípustné riešenia. Prostredníctvom štruktúrnych ohraničení formulujeme podmienky, ktorým majú jednotlivé riešenia podliehať. Premenné, prostredníctvom ktorých dosahujeme požadovaný cieľ, sa nazývajú rozhodovacie premenné.¹⁷ Účelová funkcia, ako aj štruktúrne ohraničenia, môžu mať rôzne formy. V závislosti od charakteru funkcií obsiahnutých v úlohách matematického programovania a podmienok kladených na rozhodovacie premenné môžeme úlohy matematického programovania klasifikovať do viacerých tried¹⁸:

- lineárne programovanie,
- nelineárne programovanie,

¹⁶BREZINA, Ivan - PEKÁR, Juraj. Úvod do operačného výskumu I. Bratislava : Letra Edu, 2018. s. 52. ISBN 978-80-89962-17-4.

¹⁷BREZINA, Ivan - IVANIČOVÁ, Zlatica - PEKÁR, Juraj. Operačná analýza. Bratislava : Jura Edition, 2007. 243 s. ISBN 978-80-8078-176-7.

¹⁸WILIAMS, Paul H. Model Building in Mathematical Programming. 5. vyd. United Kingdom : John Wiley & Sons, Ltd., 2013. 411 s. ISBN 978-1-118-44333-0.

- celočíselné programovanie.

Každá forma úlohy matematického programovania má svoje charakteristiky a je vhodná na riešenie určitých typov problémov. V našej práci sa však vzhľadom na špecifiká skúmaného objektu budeme zaoberať modelmi zmiešaného celočíselného programovania. Viac o modeloch matematického programovania možno nájsť v knihe: Úvod do operačného výskumu I. (BREZINA, I. a kol. 2018)

V reálnom svete existuje veľké množstvo problémov, ktoré je možné opísať určitou kombináciou lineárneho a celočíselného programovania. V prípade, ak úloha pozostáva iba z celočíselných rozhodovacích premenných, považujeme ju za úlohu celočíselného programovania. Ak je však na časť rozhodovacích premenných kladená podmienka celočíselnosti a ostatné rozhodovacie premenné sú spojité, hovoríme o zmiešanom celočíselnom programovaní.

Modely zmiešaného celočíselného programovania na optimalizáciu výrobných dávok a plánovania boli doposiaľ pod extenzívnym vývojom. Operační výskumníci v snahe čo najpresnejšie opísať špecifiká výrobného procesu, vyvinuli množstvo modelov zmiešaného celočíselného programovania. Z dôvodu veľkého množstva vyvinutých modelov si v tejto časti predstavíme iba tie modely, ktoré sú základom vývoja modelu, ktorý použijeme na optimalizáciu výrobných dávok a plánovania v nami skúmanom výrobnom procese.

3.1.1 *CLSP model*

Základ modelu, s ktorým sa budeme v našej práci zaoberať je, CLSP (Capacitated Lot Sizing Problem) model. Ide svojím spôsobom o rozšírenie už spomenutého Wagner-Whitin modelu o obmedzenia kapacít. Podobne ako ELSP, ide o viac výrobkový („multi item“) model. Popisuje teda produkčný proces viacerých výrobkov na jednej úrovni produkcie. Je to model dynamický, kedy pripúšťaťame rôzne sa vyvíjajúci dopyt zákazníkov v čase. Nakoľko model pripúšťaťa produkciu viacerých výrobkov v čase t , radíme ho medzi úlohy NP-t'ážké. Pre veľké rozmery je pomerne t'ážko riešiteľný.

Model formulujeme ako úlohu zmiešaného celočíselného programovania v nasledovnom tvare¹⁹:

$$\min \sum_{j=1}^J \sum_{t=1}^T (US_j X_{jt} + H_j I_{jt}) \quad (1)$$

$$I_{jt} = I_{j(t-1)} + Q_{jt} - D_{jt} \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (2)$$

$$I_{j0} = I0_j$$

$$P_j Q_{jt} \leq C_t X_{jt} \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (3)$$

$$\sum_{j=1}^J P_j Q_{jt} + \sum_{j=1}^J S_j X_{jt} \leq C_t \quad (t = 1, 2, \dots, T) \quad (4)$$

$$X_{jt} \in \{0, 1\} \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (5)$$

$$I_{jt}, Q_{jt} \geq 0 \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (6)$$

Tabuľka 2: Rozhodovacie premenné v modeli CLSP

Symbol	Definícia
I_{jt}	Zásoba výrobku j na konci času t .
Q_{jt}	Vyprodukované množstvo výrobku j v čase t .
X_{jt}	Binárna premenná, ktorá označuje, či sa realizuje pretypovanie na výrobok j v čase t .

Zdroj: vlastné spracovanie, podľa¹⁹

Tabuľka 3: Parametre v modeli CLSP

Symbol	Definícia
C_t	Disponibilná kapacita stroja v čase t .
D_{jt}	Dopyt po výrobku j v čase t .
H_j	Nazáporné náklady skladovania výrobku j .
$I0_j$	Počiatočná zásoba výrobku j .
J	Počet výrobkov.
P_j	Cyklový čas výrobku j .
S_j	Čas pretypovania na výrobok j .
U	Minutový faktor stroja.
T	Počet periód.

Zdroj: vlastné spracovanie, podľa¹⁹

¹⁹KIMMS, Alf - DREXL, Andreas. Lot Sizing and Scheduling: Survey and Extensions [elektronický zdroj]. 1996. 28 s. Dostupné na: https://www.econstor.eu/bitstream/10419/175395/1/manuskript_421.pdf

Kde rovnica (1) je účelová funkcia. Kritériom optimálnosti je minimalizácia nákladov na skladovanie a súčasne nákladov na pretypovania stroja. Ohraničenie (2) je bilančná rovnica zásob. Zásoby j -teho výrobku v čase t sa musia rovnať zásobám z predošlého obdobia navýšeným o produkciu v čase t a zníženým o dopyt zákazníkov v čase t . Ohraničenie (3) je ohraničenie, ktoré zabezpečuje, že produkcia výrobku j v čase t môže nastať iba v prípade, že stroj je na výrobok j v čase t pretypovaný. Vtedy rozhodovacia premenná X_{jt} nadobúda hodnotu 1, a tým uvoľňuje kapacitu C_t pre produkciu výrobku j . V prípade, že rozhodovacia premenná X_{jt} nadobúda hodnotu 0, ohraničenie zabezpečuje, že produkcia výrobku j v čase t nastať nemôže, nakoľko je kapacita $C_t = 0$. Ohraničenie (4) je ohraničenie kapacít stroja. Zabezpečuje, aby bolo možné naplánovať iba taký objem produkcie, ktorý neprekročí kapacity stroja v čase t . Táto kapacita je ponížená o časy pretypovania realizovaných výrobkov. Nakoniec formulujeme dodatočné podmienky v tvare (5),(6), kde na premennú X_{jt} je kladená podmienka binárnosti a I_{jt}, Q_{jt} nadobúdajú hodnoty z množiny nezáporných čísel.

3.1.2 CLSD-SM model

V praxi sa často stretávame s výrobnými procesmi, ktoré sa vyznačujú určitými špecifikami. V už spomenutom modeli CLSP predpokladáme, že čas pretypovania stroja na určitý výrobok nezávisí od toho, v akom stave sa stroj nachádza. To znamená, že bez ohľadu na to, na aký výrobok je stroj nastavený, čas pretypovania na nasledujúci výrobok bude vždy rovnaký. Tento predpoklad je značne limitujúci. V praxi sa stretávame s takými výrobnými procesmi, kde čas pretypovania stroja závisí od toho, na aký výrobok bol stroj doposiaľ nastavený. Hovoríme o sekvenčnej závislosti časov pretypovania. Na optimalizáciu výrobných dávok pre produkčné systémy vyznačujúce sa týmto špecifikom je vhodný model CLSD-SM (Single Machine Capacitated Lotsizing problem with Sequence Dependent setup costs). Tento model môžeme považovať o rozšírený model CLSP. Rovnako ako model CLSP, model CLSD-SM obsahuje kapacitné ohraničenia stroja. Stále je to model zahŕňajúci iba jeden stroj. Ide o model jednoúrovňový, t. j. model abstrahuje od predošlých výrobných operácií. V prípade aplikácií jednoúrovňových modelov na viacúrovňové výrobné procesy je vhodné optimalizáciu dávok vykonať na najvyššom stupni produkcie (montáž), kde dochádza k najvyššej pridanej hodnote. Ide o model dynamický. To znamená, že potreby

zákazníka môžu byť rôzne v čase. Nakoľko model pripúšťa produkciu viacerých výrobkov v čase t , zaradíme ho do tzv. „big bucekt“ NP-t ťažkých úloh.

Model formulujeme ako úlohu zmiešaného celočíselného programovania v nasledovnom tvare²⁰:

$$\min \sum_{i=1}^J \sum_{j=1}^J \sum_{t=1}^T US_{ij}X_{ijt} + \sum_{j=1}^J \sum_{t=1}^T H_j I_{jt} \quad (7)$$

$$I_{jt} = I_{j(t-1)} + Q_{jt} - D_{jt} \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (8)$$

$$I_{j0} = I0_j$$

$$\sum_{j=1}^J P_j Q_{jt} + \sum_{i=1}^J \sum_{j=1}^J S_{ij} X_{ijt} \leq C_t \quad (t = 1, 2, \dots, T) \quad (9)$$

$$C_t \left(\sum_{i=1}^J X_{ijt} + Z_{j(t-1)} \right) - P_j Q_{jt} \geq 0 \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (10)$$

$$Z_{j0} = Z0_j$$

$$\sum_{j=1}^J Z_{jt} = 1 \quad (t = 1, 2, \dots, T) \quad (11)$$

$$\sum_{i=1}^J X_{ikt} + Z_{k(t-1)} = \sum_{j=1}^J X_{kjt} + Z_{kt} \quad (k = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (12)$$

$$F_{jt} \geq F_{it} + 1 - J(1 - X_{ijt}) \quad (i = 1, 2, \dots, J; j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (13)$$

$$F_{jt} \geq 0; Q_{jt} \geq 0; I_{jt} \geq 0; \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (14)$$

$$X_{ijt} \in \{0, 1\}; \quad (i = 1, 2, \dots, J; j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (15)$$

$$Z_{jt} \in \{0, 1\}; \quad (j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (16)$$

²⁰HAASE, Knut. Capacitated Lot-Sizing with Sequence Dependent Setup Costs [elektronický zdroj]. 1994. 14 s. Dostupné na: https://www.econstor.eu/bitstream/10419/155415/1/manuskript_340.pdf

Tabuľka 4: Rozhodovacie premenné v modeli CLSD-SM

Symbol	Definícia
I_{jt}	Zásoba výrobku j na konci času t .
Q_{jt}	Vyprodukované množstvo výrobku j v čase t .
F_{jt}	Umelá premenná, ktorá sa využíva na elimináciu podcyklov.
Z_{jt}	Binárna premenná. Ak nadobúda hodnotu 1, stroj je nastavený na výrobu výrobku j na konci času t a na začiatku času $t + 1$. V opačnom prípade $Z_{jt} = 0$.
X_{ijt}	Binárna premenná. Ak $X_{ijt} = 1$, stroj je prestavený na výrobok j po výrobku i v čase t . V opačnom prípade $X_{ijt} = 0$.

Zdroj: vlastné spracovanie, podľa²⁰

Tabuľka 5: Parametre v modeli CLSD-SM

Symbol	Definícia
C_t	Disponibilná kapacita stroja v čase t .
D_{jt}	Dopyt po výrobku j v čase t .
H_j	Nazáporné náklady skladovania výrobku j .
IO_j	Počiatočná zásoba výrobku j .
$Z0_j$	Reprezentuje, na aký výrobok je stroj nastavený na konci obdobia 0 a na začiatku obdobia 1.
J	Počet výrobkov.
P_j	Cyklový čas výrobku j .
S_{ij}	Čas pretypovania z výrobku i na výrobok j .
U	Minutový faktor stroja.
T	Počet periód.

Zdroj: vlastné spracovanie, podľa²⁰

Kde rovnica č. (7) je účelová funkcia. Naším kritériom optimálnosti je opäť minimalizácia nákladov na pretypovanie a súčasne minimalizácia nákladov na skladovanie produkcie. Ohraničenie č. (8) je bilančná rovnica zásob, ktorá zabezpečuje, že zásoby j -teho výrobku v čase t sa musia rovnať zásobám výrobku j z predošlého obdobia navýšeným o produkciu výrobku j v čase t a zníženým o dopyt zákazníkov po výrobku j v čase t . Ohraničenie (9) je ohraničenie kapacít stroja. Zabezpečuje, aby bolo možné naplánovať iba taký objem produkcie, ktorý neprekročí kapacity stroja v čase t . Táto kapacita je ponížená o časy pretypovania realizovaných výrobkov. Ohraničenie (10) zabezpečuje, aby náklady na pretypovanie stroja na výrobok j boli pridané k celkovým nákladom v prípade, že sa pretypovanie na výrobok j v čase t realizuje. Ohraničenia (11) a (12) zabezpečujú, že stroj môže byť pri prechode z jedného obdobia do nasledujúceho

nastavený na výrobu iba jedného výrobku. Ohraničenie (13) eliminuje podcykly. Nakoniec formulujeme dodatočné podmienky v tvare (14), (15) a (16), kde na premenné X_{ijt} a Z_{jt} je kladená podmienka binárnosti a I_{jt}, Q_{jt} a F_{jt} nadobúdajú hodnoty z množiny nezáporných čísel.

3.1.3 CLSD-PM model

Doposiaľ sme sa zaoberali iba modelmi opisujúcimi jednoúrovňový výrobný proces obsahujúci jeden stroj. V praxi sa však stretávame s takými výrobnými procesmi, ktoré obsahujú viaceré výrobné stroje, ktoré sú do určitej miery medzi sebou zastupiteľné. V jednoduchosti povedané, výrobky môžu byť vyrábané na viacerých strojoch. Takýto výrobný proces obsahuje viaceré výrobné zariadenia, ktoré sú paralelne do výrobného procesu zapojené. Vykonávajú teda do veľkej miery podobné operácie, avšak v závislosti od kompatibility výrobného zariadenia sa medzi sebou dokážu zastúpiť. Spomenuté špecifiká obsahuje model CLSP-PM (Parallel Machine Capacitated Lotsizing problem with Sequence Dependent setup costs). Model predpokladá N výrobkov, ktoré je v závislosti od zastupiteľnosti možné produkovať na M rôznych strojoch počas T období, pričom pretypovanie strojov je sekvenčne závislé a dopyt zákazníka po výrobkoch sa v čase môže rôzne vyvíjať. Ide o rozšírenie modelu CLSD-SM o paralelné zapojenie strojov. Stále však hovoríme o modeli popisujúcom iba jednoúrovňový výrobný proces. Ako sme spomenuli, zahrnutím každého ďalšieho špecifika do modelu matematického programovania, sa takýto model stáva výpočtovo náročnejší. CLSD-PM model preto zaradíme medzi úlohy NP-t'ážké.

Model formulujeme ako úlohu zmiešaného celočíselného programovania v nasledovnom tvare²¹:

$$\min \sum_{m=1}^M \sum_{i=1}^J \sum_{j=1}^J \sum_{t=1}^T U_m S_{mij} X_{mijt} + \sum_{i=1}^J \sum_{t=1}^T H_i I_{it} \quad (17)$$

$$I_{i(t-1)} + \sum_{m=1}^M Q_{mit} - D_{it} = I_{it} \quad (i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (18)$$

$$I_{i0} = I0_i$$

$$\sum_{i=1}^J P_{mi} Q_{mit} + \sum_{i=1}^J \sum_{j=1}^J S_{mij} X_{mijt} \leq C_{mt} \quad (m = 1, 2, \dots, M; t = 1, 2, \dots, T) \quad (19)$$

$$Q_{mit} \leq G_{mit} \left(\sum_{j=1}^J X_{mijt} Z_{mit} \right) \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (20)$$

$$Z_{mi(t+1)} + \sum_{j=1}^J X_{mijt} = Z_{mit} + \sum_{j=1}^J X_{mjit} \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (21)$$

$$Z_{mi1} = Z1_{mi}$$

$$\sum_{i=1}^J Z_{mit} = 1 \quad (m = 1, 2, \dots, M; t = 1, 2, \dots, T + 1) \quad (22)$$

$$Q_{mit} \leq G_{mit} A_{mi} \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (23)$$

$$F_{mit} + N X_{mijt} - (N - 1) - N Z_{mit} \leq F_{mit} \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; \\ j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (24)$$

$$Q_{mit} \geq 0; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (25)$$

$$I_{it} \geq 0; \quad (i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (26)$$

$$X_{mijt} \in \{0, 1\}; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; \\ j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (27)$$

$$Z_{mit} \in \{0, 1\}; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T + 1) \quad (28)$$

$$Q_{mit} \in \mathbb{Z}; F_{mit} \in \mathbb{R}; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (29)$$

²¹JAMES, Ross J. W. - ALMADA-LOBO, Bernardo. Single and Parallel Machine Capacitated Lotsizing and Scheduling : New Iterative MIP-based Neighborhood Search Heuristics [elektronický zdroj]. 2011. 10 s. Dostupné na: <https://www.sciencedirect.com/science/article/pii/S0305054811000463>

Tabuľka 6: Rozhodovacie premenné v modeli CLSD-PM

Symbol	Definícia
I_{it}	Zásoba výrobku i na konci času t .
Q_{mit}	Vyprodukované množstvo výrobku i na stroji m v čase t .
F_{mit}	Umelá premenná, ktorá sa využíva na elimináciu podcyklov.
Z_{mit}	Binárna premenná. Ak nadobúda hodnotu 1, stroj m je nastavený na výrobu výrobku i na začiatku času t . V opačnom prípade $Z_{mit} = 0$.
X_{mijt}	Binárna premenná. Ak $X_{mijt} = 1$, stroj je prestavený na výrobok j po výrobku i na stroji m v čase t . V opačnom prípade $X_{mijt} = 0$.

Zdroj: vlastné spracovanie, podľa²¹

Tabuľka 7: Parametre v modeli CLSD-PM

Symbol	Definícia
C_{mt}	Disponibilná kapacita stroja m v čase t .
D_{it}	Dopyt po výrobku i v čase t .
H_i	Nazáporné náklady skladovania výrobku i .
IO_i	Počiatočná zásoba výrobku i .
$Z1_{mi}$	Reprezentuje, na aký výrobok i je stroj m nastavený na začiatku obdobia 1 a na konci obdobia 0.
J	Počet výrobkov.
P_{mi}	Cyklový čas výrobku i na stroji m .
S_{mij}	Čas pretypovania z výrobku i na výrobok j na stroji m .
U_m	Minutový faktor stroja m .
G_{mit}	Maximálne množstvo výrobku i , ktoré sa môže vyprodukovať na stroji m v čase t .
A_{mi}	Zastupiteľnosť strojov.
M	Počet strojov.
T	Počet periód.

Zdroj: vlastné spracovanie, podľa²¹

Kde rovnica č. (17) je účelová funkcia. Naším kritériom optimálnosti je opäť minimalizácia nákladov na pretypovanie a súčasne minimalizácia nákladov na skladovanie produkcie. Ohraničenie č. (18) je bilančná rovnica zásob, ktorá zabezpečuje, že zásoby i -teho výrobku v čase t sa musia rovnať zásobám výrobku i z predošlého obdobia navýšeným o produkciu výrobku i v čase t a zníženým o dopyt zákazníkov po výrobku i v čase t . Ohraničenie (19) je ohraničenie kapacít stroja. Zabezpečuje, aby bolo možné naplánovať iba taký objem produkcie, ktorý neprekročí kapacity stroja m v čase t . Táto kapacita je ponížená o časy pretypovania realizovaných výrobkov. Ohraničenie (20) zabezpečuje,

aby náklady na pretypovanie stroja m na výrobok i boli pridané k celkovým nákladom v prípade, že sa pretypovanie na výrobok i v čase t realizuje. Ohraničenia (21) a (22) zabezpečujú, že každý stroj môže byť pri prechode z jedného obdobia do nasledujúceho nastavený na výrobu iba jedného výrobku. Ohraničenie (23) zabezpečuje, že produkcia i -teho výrobku môže nastať iba v prípade, že je stroj na to prispôsobený. Ohraničenie (24) eliminuje podcykly. Nakoniec formulujeme dodatočné podmienky v tvare (25), (26), (27), (28) a (29), kde na premenné X_{mit} a Z_{mit} je kladená podmienka binárnosti a I_{it} a Q_{mit} nadobúdajú hodnoty z množiny nezáporných čísel, pričom Q_{mit} patrí do množiny celých čísel a F_{mit} patrí do množiny reálnych čísel.

3.2 Genetický algoritmus

Genetický algoritmus bol vyvinutý J. Hollandom v 70. rokoch 20. storočia (University of Michigan, USA) na pochopenie adaptačných procesov prírodných systémov. Neskôr, v osemdesiatych rokoch 19. storočia, bol genetický algoritmus použitý na riešenie optimalizačných problémov. Genetický algoritmus radíme do triedy evolučných algoritmov, ktoré sú založené na princípoch Darwinovej teórie biologickej evolúcie. Genetický algoritmus je optimalizačná technika, ktorá sa používa na nájdenie najlepšieho riešenia alebo súboru riešení daného problému napodobňovaním procesu prirodzeného výberu a genetickej rekombinácie.²²

Nakoľko genetický algoritmus je inšpirovaný biologickými procesmi, väčšina príslušnej terminológie je taktiež prevzatá z biológie. Základnými prvkami každého genetického algoritmu sú chromozóm, gén, populácia, generácia a fitness funkcia. Každé riešenie v terminológii genetického algoritmu je reprezentované chromozómom. Riešením rozumieme súbor hodnôt všetkých rozhodovacích premenných v riešenej úlohe. V závislosti od špecifika skúmaného objektu môže byť chromozóm reprezentovaný rôznymi spôsobmi.

Vo väčšine prípadov sa využíva binárna reprezentácia chromozómu, pretože je to jednoduchý a efektívny spôsob jeho reprezentácie. Pri binárnom kódovaní je každý z chromozómov reprezentovaný číselným reťazcom fixnej dĺžky pozostávajúcom z núl a jednotiek. Každý člen reťazca nazývame gén. Gén je najmenšia jednotka chromozómu.

²²EL-GHAZALI, Talbi. METAHEURISTICS FROM DESIGN TO IMPLEMENTATION. New Jersey : John Wiley & Sons, Inc., 2009. 618 s. ISBN 978-0-470-27858-1.

Každá unikátna obmena, ktorú gén môže nadobudnúť, sa označuje ako alela. V prípade binárnej reprezentácie chromozómu má každý gén dve alely, pretože môže nadobúdať iba hodnoty 0 alebo 1. To znamená, že jeden gén vystačí na zakódovanie iba dvoch rôznych hodnôt. Z toho dôvodu je na zakódovanie hodnôt rozhodovacích premenných zväčša potrebných niekoľko génov. Základnou myšlienkou binárneho kódovania chromozómu je fakt, že veľakrát rozhodovacie premenné musia spĺňať určité podmienky. Často sa na rozhodovacie premenné kladie podmienka, z akého intervalu hodnôt môžu pochádzať. V prípade, ak rozhodovacia premenná môže nadobúdať len veľmi malé hodnoty, je nežiadúce vytvárať zbytočne dlhý chromozóm. Tento spôsob reprezentácie chromozómu nám teda umožňuje minimalizovať množstvo nepotrebných informácií pre prácu algoritmu, čím zabezpečuje vysokú efektívnosť riešenia. Rovnako aj aplikácia operátorov genetického algoritmu je na takto reprezentovanom chromozóme pomerne jednoduchá.²³

V súčasnosti je však čoraz populárnejšia reprezentácia chromozómu reálnymi hodnotami. V tomto prípade je každé riešenie reprezentované chromozómom, ktorý pozostáva z reťazca reálnych čísel, ktoré znamenajú konkrétne hodnoty rozhodovacích premenných. Z toho vyplýva, že na zakódovanie hodnoty jednej rozhodovacej premennej nám postačuje jeden gén. Jeden gén v prípade tejto reprezentácie môže nadobúdať veľké množstvo alel. Na rozdiel od binárneho kódovania, reprezentácia chromozómu reálnymi hodnotami nám umožňuje riešiť optimalizačné problémy v množine reálnych čísel a súčasne nie je potrebné prevádzať hodnoty rozhodovacích premenných medzi jednotlivými sústavami. V závislosti od špecifik skúmaného problému má každý spôsob reprezentácie chromozómu svoje výhody, ale aj nevýhody, ktoré významne vplyvajú na výkonnosť algoritmu ako celku.²³

Pod pojmom populácia v rámci genetického algoritmu rozumieme súbor všetkých riešení v jednej generácii. Veľkosť populácie je v genetických algoritmoch veľmi dôležitým parametrom, nakoľko do značnej miery ovplyvňuje jej rôznorodosť. Veľkosť populácie by však taktiež nemala byť veľmi veľká, pretože môže spôsobiť zníženie efektivity genetického algoritmu. Efektívnosť genetického algoritmu a jeho schodnosť hľadať nové a lepšie riešenia je do veľkej miery závislá od optimálnej veľkosti populácie. Generáciou rozumieme jeden cyklus evolučného procesu. Každá generácia spočíva vo vytvorení novej

²³MITCHELL, Melanie. An Introduction to Genetic Algorithms. 5. vyd. Cambridge : Massachusetts Institute of Technology, 1999. 158 s. ISBN 0-262-13316-4.

populácie riešení použitím genetických operátorov, ako je selekcia, rekombinácia či mutácia. Počet generácií potrebných na konvergenciu k optimálnemu riešeniu závisí od zložitosti riešeného problému a stanovenej veľkosti populácie. Vo všeobecnosti však platí, že väčšie populácie potrebujú na konvergenciu viac generácií ako menšie populácie. Často je však počet generácií či populácií stanovovaný metódou pokusu a omylu.

Fitness funkcia v rámci genetického algoritmu predstavuje svojím spôsobom účelovú funkciu. Je to funkcia, ktorá každému riešeniu reprezentovanému chromozómom priraduje určitú hodnotu. V závislosti od optimalizačného kritéria sa usilujeme nájsť minimálnu alebo maximálnu hodnotu tejto funkcie.

Základy genetického algoritmu počínajú v myšlienke neustáleho zlepšovania inicializačných riešení. Genetický algoritmus začína s populáciou inicializačných riešení a pokračuje opakovaným uplatňovaním princípov selekcie, rekombinácie a mutácie, ktoré slúžia na generovanie nových populácií kandidátskych riešení. V každej generácii sa vyberú najvhodnejší jedinci, ktorí sa spárujú a vytvoria potomstvo, ktoré zdedí priaznivé vlastnosti svojich rodičov. Títo potomkovia sa potom náhodne mutujú, aby sa vniesla rozmanitosť do generácie a zabezpečilo sa, že sa preskúmajú nové a potenciálne lepšie riešenia. Tento proces pokračuje počas viacerých generácií, kým sa nenájde uspokojivé riešenie alebo kým sa nesplní kritérium ukončenia algoritmu. Genetické algoritmy sa používajú na riešenie širokej škály optimalizačných problémov v oblastiach, ako sú financie, ekonómia, strojárstvo, informatika a iné.²⁴ Väčšina genetických algoritmov sa skladá zo spomenutých základných operátorov, ktoré uvádzame na obrázku č. 1.

(a) Inicializácia

Inicializácia je proces vytvárania počiatočných (štartovacích) riešení, ktoré vstupujú do cyklu evolúcie. Tento krok je pre genetický algoritmus nevyhnutný, nakoľko genetický algoritmus je založený na myšlienke postupného zlepšovania týchto riešení. Existuje viacero spôsobov, ako je inicializáciu možné vykonať. Častým prístupom je náhodné generovanie riešení. Tento spôsob je však problematický pri úlohách s viazaným extrémom, pretože náhodné generovanie môže spôsobiť, že inicializačné riešenia nie sú prípustnými vzhľadom na ohraničujúce podmienky.

²⁴EL-GHAZALI, Talbi. METAHEURISTICS FROM DESIGN TO IMPLEMENTATION. New Jersey : John Wiley & Sons, Inc., 2009. 618 s. ISBN 978-0-470-27858-1.

V tomto prípade je následne nutné aplikovať určitý algoritmus, ktorý je schopný také riešenie opraviť. Ďalším problémom je, že takéto náhodné riešenia sú vzhľadom na optimalizačné kritérium často pomerne nekvalitné. Nakoľko však kvalita inicializačného riešenia do veľkej miery vplýva na efektivitu genetického algoritmu, je vhodné využiť sofistikovanejšie algoritmy, ako napríklad rôzne heuristické metódy. Dobrá inicializačná stratégia môže výrazne zlepšiť výkon algoritmu tým, že zabezpečí, aby počiatočná populácia pokrývala široký rozsah prehládavacieho priestoru a súčasne obsahovala dobré riešenia.²⁵

(b) Selekcia

Selekcia je kľúčovým prvkom genetického algoritmu. Cieľom selekcie je výber jednotlivcov z pôvodnej populácie, ktorí budú slúžiť ako rodičia pre nasledujúcu populáciu. Základným princípom selekcie je uprednostňovanie jednotlivcov (riešení), ktoré dosahujú vyššiu fitness hodnotu (hodnotu účelovej funkcie). Celý proces je inšpirovaný prírodnou selekciou. V prírode majú najsilnejší jedinci väčšiu šancu získať potravu a rozmnožovať sa, následkom čoho sa ich gény viac podieľajú na tvorbe ďalšej generácie. Existuje viacero spôsobov, ktoré možno použiť na selekciiu jedincov z populácie na základe ich fitness hodnoty, avšak najpoužívanejšie sú:

- Ruleta

Táto metóda je založená na priradení počtov dielikov ruletového kola priamo úmerne k fitness hodnotám jednotlivcov. Je zrejmé, že najlepší jedinec má najväčší počet dielikov na ruletovom kolese, a preto má aj najväčšiu pravdepodobnosť byť vybraný ako rodič pre ďalšiu populáciu. Najhorší jedinec má dielikov najmenej, a preto pravdepodobnosť výberu takého jedinca je najmenšia.

- Turnajový výber

Pri tejto metóde sa jedinci vyberajú náhodne z pôvodnej populácie a súťažia medzi sebou v turnaji. Jedinec s najvyššou fitness hodnotou sa vyberie ako rodič pre ďalšiu populáciu. Tento proces sa opakuje, kým početnosť novej populácie sa nerovná početnosti populácie pôvodnej.

²⁵MIRJALILI, Seyedali. *Evolutionary Algorithms and Neural Networks : Theory and Applications*. Brisbane : Springer International Publishing AG, 2019. 159 s. ISBN 978-3-319-93024-4.

- Výber na základe poradia

Táto metóda zahŕňa zoradenie jedincov z pôvodnej populácie na základe ich fitness hodnôt a následný výber jedincov s najvyššou fitness hodnotou ako rodičov pre ďalšiu populáciu.

V rámci selekcie veľmi dôležitú úlohu zohráva elitizmus. Elitizmus je stratégia používaná v genetických algoritmoch na zachovanie najlepších jedincov z jednej generácie do druhej. Táto metóda spočíva vo výbere najlepších jedincov z pôvodnej populácie a ich priame kopírovanie do novej generácie bez aplikácie operátora selekcie. Tento prístup môže významne ovplyvniť efektívnosť genetického algoritmu, pretože zabraňuje možnej strate najlepšieho jedinca pri selekcii.²⁵

(c) Rekombinácia

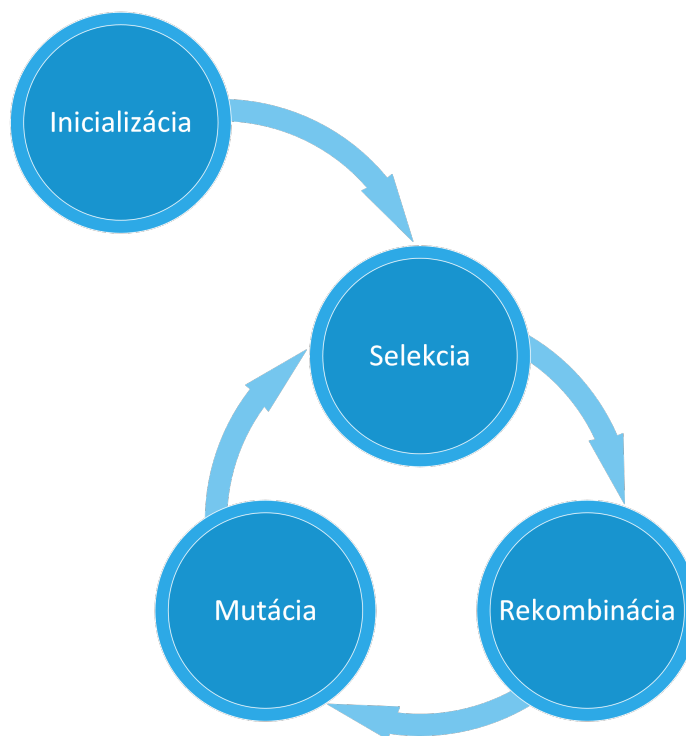
Základnou myšlienkou rekombinácie je simulovať prirodzený proces genetickej rekombinácie, pri ktorej dochádza k výmene genetického materiálu medzi rodičmi s cieľom vytvoriť potomstvo s novými vlastnosťami. Operátor rekombinácie funguje tak, že sa náhodne vyberú dvaja rodičia selektovaní z pôvodnej populácie a vymení sa časť ich genetického materiálu, čím vzniká nový potomok. Tento proces je veľmi špecifický a do veľkej miery závisí od reprezentácie chromozómu. Vo všeobecnosti je však založený na výmene genetického materiálu medzi rodičmi okolo jedného alebo viacerých náhodne vybraných bodov. Proces rekombinácie sa opakuje, kým sa nevytvorí požadovaný počet potomkov.²⁵

(d) Mutácia

Mutácia je genetický operátor, ktorý náhodne mení genetickú informáciu (chromozóm) jedného alebo viacerých jedincov v populácii. Účelom operátora mutácie je vnieť do populácie genetickú rozmanitosť a zabrániť tomu, aby algoritmus konvergoval k suboptimálnemu riešeniu. Operátor mutácie funguje tak, že náhodne vyberie niekoľko jedincov z populácie a následne zmení hodnotu jedného alebo viacerých génov v ich chromozóme. Konkrétny spôsob mutácie závisí od návrhu chromozómu a špecifika riešeného problému. V binárnej reprezentácii môže operátor mutácie prevrátiť jeden bit v chromozóme, zatiaľ čo pri reprezentácii chromozómu reálnymi hodnotami môže operátor mutácie pridať náhodnú hodnotu jednému z génov. Operátor mutácie sa zvyčajne aplikuje s nízkou pravdepodobnosťou, aby sa zabránilo

príliš veľkému narušeniu existujúcej genetickej informácie. Konkrétna použitá pravdepodobnosť závisí od špecifik riešeného problému a veľkosti populácie.²⁵

Obr. 1: Genetický algoritmus



Zdroj: vlastné spracovanie

3.3 Softvérové nástroje na riešenie optimalizačných úloh

Softvér na riešenie optimalizačných problémov je typ počítačového programu určeného na riešenie zložitých matematických problémov, ktoré zahŕňajú hľadanie najlepšieho možného riešenia z množiny alternatív.

Na riešenie takýchto problémov poskytuje softvér na optimalizáciu veľké množstvo algoritmov a techník vrátane lineárneho programovania, celočíselného programovania, nelineárneho programovania, dynamického programovania a genetických algoritmov. Tieto nástroje umožňujú používateľom modelovať a analyzovať zložité systémy, nájsť optimálne riešenie a prijímať lepšie rozhodnutia.

Používanie optimalizačného softvéru sa stáva čoraz populárnejším, pretože organizácie sa snažia zlepšiť svoju činnosť, znížiť náklady a zvýšiť efektívnosť. S rýchlym rozvojom počítačových technológií sa optimalizačný softvér stal výkonnejším,

používateľsky prívetivejším a dostupnejším, čo podnikom a jednotlivcom uľahčuje riešenie zložitých optimalizačných problémov. To viedlo k vývoju mnohých optimalizačných softvérových balíkov, z ktorých každý má svoje silné a slabé stránky, čo umožňuje používateľom vybrať si nástroj, ktorý najlepšie vyhovuje ich špecifickým potrebám. Tento software je zvyčajne dostupný v podobe knižníc. Medzi najpoužívanejšie v súčasnosti patria napríklad:

- Gurobi,
- CPLEX,
- SciPy,
- geneticalgorithm.

4 Výsledky práce a diskusia

4.1 Charakter spoločnosti

Schaeffler je vysoko inovatívna a technologicky vyspelá nemecká spoločnosť, ktorá sa zaoberá výrobou komponentov s vysokou presnosťou. Spoločnosť je známa najmä vďaka výrobe vysoko presných ložísk, ktoré sa používajú v širokom spektre priemyselných aplikácií. Do portfólia spoločnosti však patria taktiež výrobky, ako sú: komponenty a systémy pre prevodovky, komponenty pre spojky, riadenia pre automatické prevodovky, torzné tlmiče, diferenciály, rozvody, mechatronické komponenty, lineárne vedenia či komponenty pre elektrické pohony.

Spoločnosť založili v roku 1946 bratia Wilhelm a Georg Schaeffler v nemeckom Herzogenaurachu, kde spoločnosť dodnes sídli. V súčasnosti spoločnosť Schaeffler pôsobí vo viac ako 50 krajinách sveta a zamestnáva približne 83 000 pracovníkov v približne 200 pobočkách sveta. V roku 2022 spoločnosť Schaeffler vygenerovala tržby vo výške 15,8 miliardy eur, čo ju robí jednou z najväčších rodinných spoločností na svete. S viac ako 1 800 patentovými prihláškami v roku 2021 je Schaeffler podľa DPMA (Nemecký úrad pre patenty a ochranné známky) tretou najinovatívnejšou spoločnosťou v Nemecku.²⁶

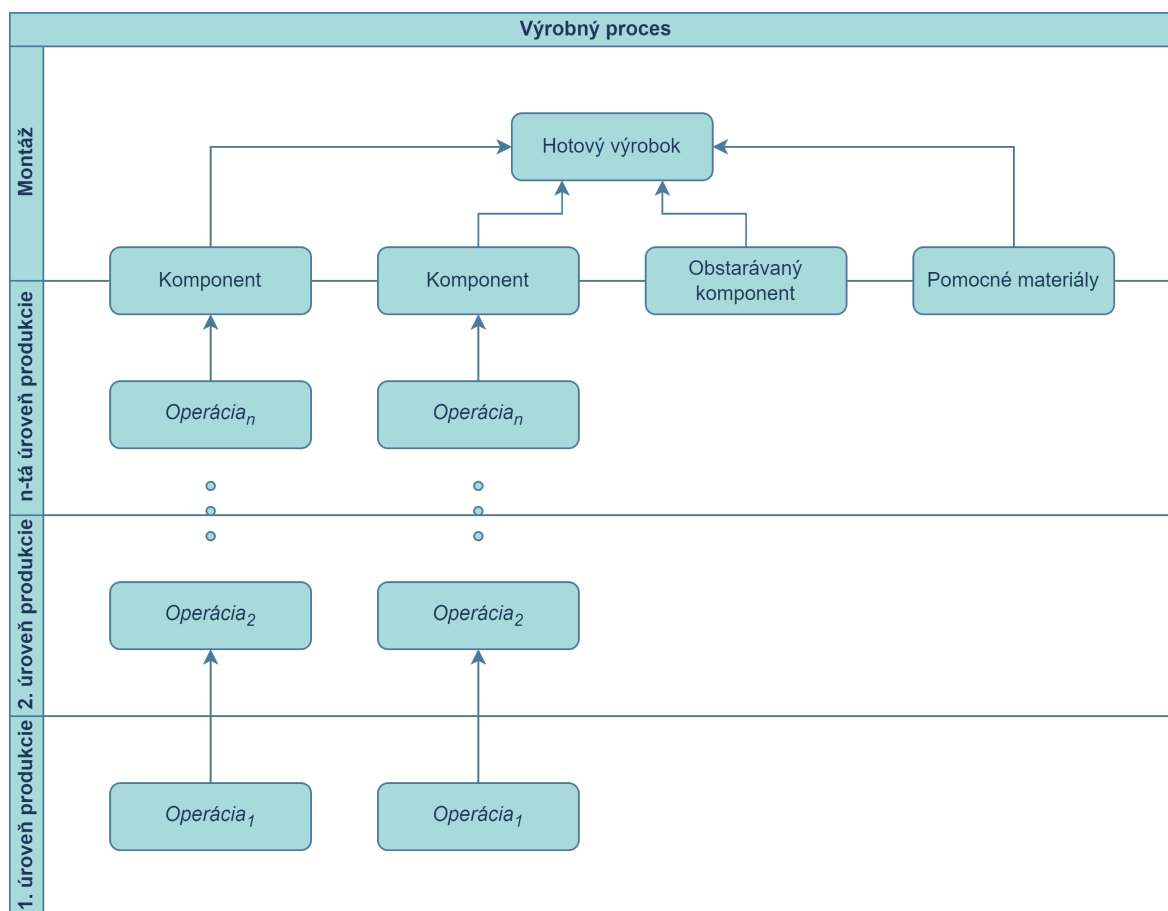
4.2 Charakteristika skúmaného objektu

Nakoľko primárnym zameraním spoločnosti Schaeffler je výroba ložísk, predmetom skúmania bude práve plánovanie ich produkcie. Výroba vysokokvalitných ložísk je pomerne náročný výrobný proces, ktorý sa skladá z viacerých na seba naväzujúcich operácií. Ilustratívny zovšeobecnený príklad výrobného procesu ložísk uvádzame na obrázku č. 2.

Výroba ložísk začína obstaraním materiálov potrebných na výrobu komponentov, z ktorých sa skladajú. Následne sa tieto komponenty opracovávajú pomocou na seba naväzujúcich operácií. Hotové komponenty sa na montážnej úrovni produkcie skladajú, čím vytvoria hotový výrobok. Každá úroveň produkcie sa podieľa na hodnote hotového výrobku. Spravidla však najvyššiu pridanú hodnotu vytvára montáž, nakoľko na tejto úrovni

²⁶<https://www.schaeffler.com/en/>

Obr. 2: Ilustratívny príklad výrobného procesu ložísk



Zdroj: vlastné spracovanie

produkcie vstupujú do výrobku všetky komponenty. Súčasne tak na montážnej úrovni vznikajú podniku najvyššie náklady na skladovanie produkcie. Každú operáciu výrobného procesu ložísk vykonávajú špecifické zariadenia. Tieto zariadenia sa značne líšia časom a náročnosťou prestavby na výrobu iného výrobku. Z toho vyplýva, že na nižších úrovniach produkcie síce nevznikajú podniku vysoké náklady skladovania, no môžu vzniknúť vysoké náklady pretypovania. Na nižších úrovniach produkcie je preto vhodné vyrábať vo vyšších výrobných dávkach pri nižšej frekvencii pretypovania strojov. Na vyšších úrovniach produkcie je to spravidla naopak. Náklady na skladovanie produkcie hotových výrobkov sú síce vysoké, no náklady na pretypovanie montážnych liniek sú nižšie. V tomto prípade je vhodné vyrábať v nižších výrobných dávkach pri vyššej frekvencii pretypovania strojov.

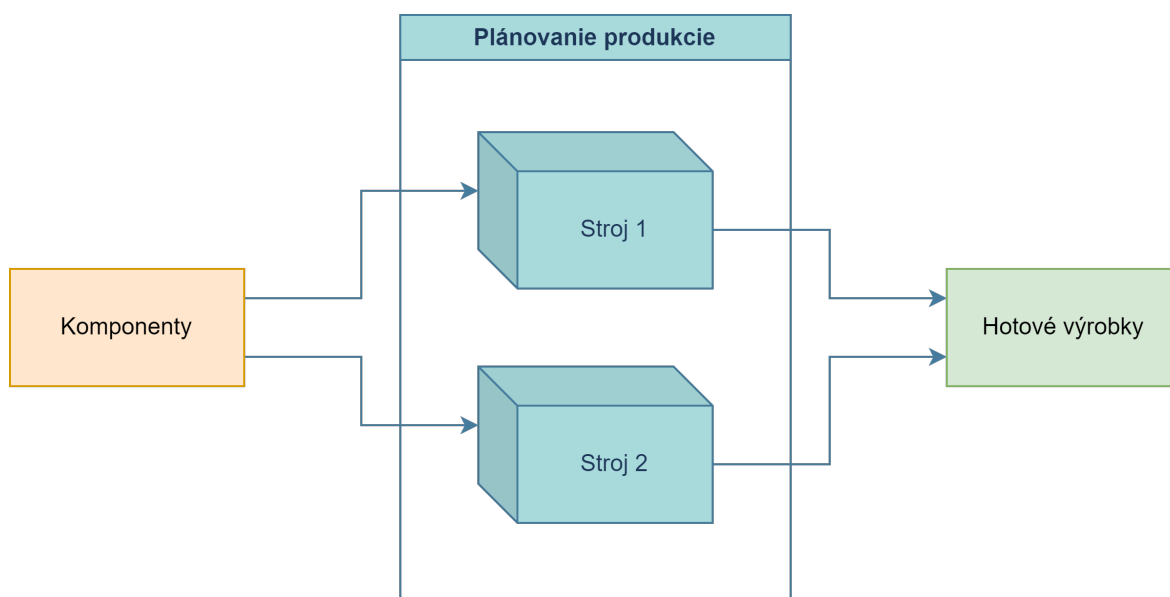
Ideálny prístup k optimalizácii je teda pozerat' sa na výrobný proces ako na previazaný systém. V opačnom prípade sa dopúšťame klamu kompozície. Čo je dobré pre časť, neznamená, že bude dobré pre celok. Každý optimalizačný model je len

taký dobrý, ako dobre vystihuje realitu. Aj keď už v 90. rokoch minulého storočia existovali modely, ktoré opisujú viacúrovňový výrobný proces, stále sú takéto úlohy pre reálne výrobné procesy so súčasnou výpočtovou technikou neriešiteľné. Existujú preto viaceré heuristické metódy, ako k takému problému možno pristúpiť. Jednou z alternatív je aplikácia optimalizačných techník na každej úrovni produkcie zvlášť, pričom ako prvé sa optimalizácia aplikuje na najvyššej úrovni produkcie. Následne sa optimalizujú výrobné dávky a plány na nižších úrovniach produkcie, pričom ako dopyt sa považuje vnútropodnikový dopyt tvorený nasledujúcou výrobnou operáciou. Pre zaistenie efektívnych plánov preto nie je vhodné aplikovať konštantné výrobné dávky pre všetky výrobné operácie. Vzhľadom na komplexnosť výrobného procesu, v tejto práci budeme prezentovať aplikáciu optimalizačných techník na montážnej úrovni produkcie, pričom postup riešenia je aplikovateľný na všetky výrobné úrovne.

Každá zo spomenutých výrobných operácií môže byť vykonávaná viacerými strojmi, ak si to výrobný proces vyžaduje. Často tak nastáva v prípade, ak takt výrobných operácií na nižších úrovniach produkcie je nižší, ako na vyšších úrovniach produkcie. V praxi to znamená, že prechádzajúce výrobné operácie sú rýchlejšie ako nasledujúce. Jedno z riešení je spomenuté paralelné zapojenie strojov pri operáciách, ktoré sa vyznačujú vyššími cyklovými časmi. V prípade nami skúmaného objektu (montáž ložísk) sú vo výrobnom procese paralelne zapojené dve montážne linky. Montážnu úroveň produkcie reprezentujeme na obrázku č. 3.

Každá z montážnych liniek sa v praxi môže vyznačovať určitými špecifikami, ako napríklad: čas prestavby linky, cyklový čas linky či ich zastupiteľnosť. V nami skúmanom objekte sa však montážne linky nelíšia cyklovými časmi či časmi potrebnými na prestavbu linky z i-teho výrobku na j-ty výrobok. Budeme preto abstrahovať od týchto špecifik. Časy pretypovania strojov a cyklové časy strojov v nami skúmanom objekte, závisia iba od výrobkov, nie však od montážnej linky, na ktorej sú výrobky zmontované. Dôležitým špecifikom, ktorým sa montážne linky v nami skúmanom objekte vyznačujú, je ich zastupiteľnosť. Zastupiteľnosť linky chápeme ako schopnosť liniek zmontovať rovnaký výrobok. Táto schopnosť závisí najmä od náradia, akým sú linky vybavené. V nami skúmanom objekte nie sú obe linky vybavené rovnakým náradím, a preto je ich zastupiteľnosť limitovaná. Existujú výrobky, ktoré dokážu zmontovať obe linky, no niektoré výrobky dokáže zmontovať iba jedna z nich.

Obr. 3: Montážna úroveň produkcie



Zdroj: vlastné spracovanie

V tejto práci sa budeme usilovať nájsť taký výrobný plán pre montážnu úroveň produkcie, aby boli splnené všetky špecifiká jednotlivých výrobných liniek a súčasne, aby náklady boli minimálne.

4.3 Modifikácia modelu CLSD-PM pre nami skúmaný objekt

Pri konštrukcii optimalizačného modelu matematického programovania pre nami skúmaný objekt vychádzame z modelov prezentovaných v časti 3.1 Modely matematického programovania. Nakoľko však spomenuté modely v nedostatočnej miere vystihujú náš skúmaný objekt, bolo potrebné ich istým spôsobom prispôbiť.

V praxi sa často stretávame s neočakávanými udalosťami, ktoré významne vplyvajú na prípustnosť výrobných plánov. Prípustnosť výrobného plánu môže byť ohrozená napríklad v prípade narušenia dodávateľského reťazca, neočakávaného navýšenia dopytu zo strany zákazníka alebo oneskorenia výrobného procesu vplyvom porúch na stroji. Dôsledkom tohto môže dôjsť k oneskoreniu dodávky zákazníkovi a v konečnom dôsledku k zníženiu spokojnosti zákazníka. Nahromadenú produkciu, ktorú už firma nedokáže dodať načas zákazníkovi, budeme označovať ako sklz. Vznik sklzu je pre producenta a aj pre zákazníka neželaný stav, a preto sa usilujú o ich minimalizáciu.

Zo spomenutých dôvodov je potrebné vyvinúť taký optimalizačný model, ktorý okrem minimalizácie nákladov na tvorenie zásob a nákladov na pretypovanie minimalizuje aj náklady na vytvorený sklz. Rovnako je však dôležité upraviť aj bilančnú rovnicu zásob. Bilančná rovnica definuje objem produkcie, ktorý je potrebné v danom časovom horizonte vyprodukovať, aby bol splnený dopyt zákazníkov. Bilančná rovnica je úzko spätá s kapacitnými ohraničeniami. Ak bilančná rovnica stanoví, že je potrebné vyprodukovať vyšší objem produkcie (vplyvom nahromadených objednávok), ako je možné vzhľadom na kapacitu strojov vyprodukovať, nastáva situácia, kedy úloha nemá žiadne prípustné riešenie. Pri formulácii úlohy matematického programovania budeme formulovať sklz ako odchýlkovú premennú, pričom ak odchýlka nadobúda kladnú hodnotu, jej výška sa prejaví v hodnote účelovej funkcie. V nami skúmanom objekte je minútový faktor oboch montážnych liniek rovnaký. Rovnako platí, že čas prestavby linky a taktiež cyklové časy výrobkov nezávisia od linky, na ktorej sú výrobky zhotovované, ale iba od jednotlivých výrobkov, prípadne od ich výrobných sekvencií.

Model formulujeme ako úlohu zmiešaného celočíselného programovania v nasledovnom tvare:

$$\min \sum_{m=1}^M \sum_{i=1}^J \sum_{j=1}^J \sum_{t=1}^T US_{ij}X_{mijt} + \sum_{i=1}^J \sum_{t=1}^T H_i I_{it} + \sum_{i=1}^J \sum_{t=1}^T V_i B_{it} \quad (30)$$

$$I_{i(t-1)} - B_{i(t-1)} + \sum_{m=1}^M Q_{mit} - D_{it} = I_{it} - B_{it} \quad (i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (31)$$

$$I_{i0} = I0_i$$

$$\sum_{i=1}^J P_i Q_{mit} + \sum_{i=1}^J \sum_{j=1}^J S_{ij} X_{mijt} \leq C_{mt} \quad (m = 1, 2, \dots, M; t = 1, 2, \dots, T) \quad (32)$$

$$Q_{mit} \leq G \left(\sum_{j=1}^J X_{mijt} Z_{mit} \right) \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (33)$$

$$Z_{mi(t+1)} + \sum_{j=1}^J X_{mijt} = Z_{mit} + \sum_{j=1}^J X_{mjit} \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (34)$$

$$Z_{mi1} = Z1_{mi}$$

$$\sum_{i=1}^J Z_{mit} = 1 \quad (m = 1, 2, \dots, M; t = 1, 2, \dots, T + 1) \quad (35)$$

$$Q_{mit} \leq GA_{mi} \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (36)$$

$$F_{mit} + NX_{mijt} - (N - 1) - NZ_{mit} \leq F_{mit} \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (37)$$

$$Q_{mit} \geq 0; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (38)$$

$$I_{it} \geq 0; B_{it} \geq 0; \quad (i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (39)$$

$$X_{mijt} \in \{0, 1\}; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; j = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (40)$$

$$Z_{mit} \in \{0, 1\}; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T + 1) \quad (41)$$

$$Q_{mit} \in \mathbb{Z}; F_{mit} \in \mathbb{R}; \quad (m = 1, 2, \dots, M; i = 1, 2, \dots, J; t = 1, 2, \dots, T) \quad (42)$$

Tabuľka 8: Rozhodovacie premenné v modeli CLSD-PM so sklzom

Symbol	Definícia
I_{it}	Zásoba výrobku i na konci času t .
B_{it}	Sklz výrobku i na konci času t .
Q_{mit}	Vyprodukované množstvo výrobku i na stroji m v čase t .
F_{mit}	Umelá premenná, ktorá sa využíva na elimináciu podcyklov.
Z_{mit}	Binárna premenná. Ak nadobúda hodnotu 1, stroj m je nastavený na výrobu výrobku i na začiatku času t . V opačnom prípade $Z_{mit} = 0$.
$X_{mi jt}$	Binárna premenná. Ak $X_{mi jt} = 1$, stroj je prestavený na výrobok j po výrobku i na stroji m v čase t . V opačnom prípade $X_{mi jt} = 0$.

Zdroj: vlastné spracovanie

Tabuľka 9: Parametre v modeli CLSD-PM so sklzom

Symbol	Definícia
C_{mt}	Disponibilná kapacita stroja m v čase t .
D_{it}	Dopyt po výrobku i v čase t .
H_i	Nezáporné náklady skladovania výrobku i .
V_i	Nezáporné náklady spôsobené sklzom jednotky výrobku i .
IO_i	Počiatočná zásoba výrobku i .
$Z1_{mi}$	Reprezentuje, na aký výrobok je stroj nastavený na začiatku obdobia 1 a na konci obdobia 0.
J	Počet výrobkov.
P_i	Cyklový čas výrobku i .
S_{ij}	Čas pretypovania z výrobku i na výrobok j .
U	Minútový faktor.
A_{mi}	Zastupiteľnosť strojov.
G	Veľké kladné číslo.
M	Počet strojov.
T	Počet periód.

Zdroj: vlastné spracovanie

Kde rovnica č. (30) je účelová funkcia. Naším kritériom optimálnosti je minimalizácia nákladov na pretypovanie, minimalizácia nákladov na skladovanie produkcie a súčasne minimalizácia nákladov spôsobených sklzom. Ohraničenie č. (31) je bilančná rovnica zásob, ktorá zabezpečuje, že zásoby i -teho výrobku v čase t ponížené o sklz i -teho výrobku v čase t sa musia rovnať zásobám výrobku i z predošlého obdobia poníženým o sklz výrobku i z predošlého obdobia navýšeným o produkciu výrobku i v čase t a poníženým o dopyt zákazníkov po výrobku i v čase t . Ohraničenie (32) je ohraničenie kapacít stroja. Zabezpečuje, aby bolo možné naplánovať iba taký objem produkcie, ktorý neprekročí

kapacity stroja m v čase t . Táto kapacita je ponížená o časy pretypovania realizovaných výrobkov. Ohraničenie (33) zabezpečuje, aby náklady na pretypovanie stroja m na výrobok i boli pridané k celkovým nákladom v prípade, že sa pretypovanie na výrobok i v čase t realizuje. Ohraničenia (34) a (35) zabezpečujú, že každý stroj môže byť pri prechode z jedného obdobia do nasledujúceho, nastavený na výrobu iba jedného výrobku. Ohraničenie (36) zabezpečuje, že produkcia i – *teho* výrobku môže nastať iba v prípade, že je stroj na to prispôsobený. Ohraničenie (37) eliminuje podcykly. Nakoniec formulujeme dodatočné podmienky v tvare (38), (39), (40), (41) a (42), kde na premenné X_{mit} a Z_{mit} je kladená podmienka binárnosti a I_{it} , B_{it} a Q_{mit} nadobúdajú hodnoty z množiny nezáporných čísel, pričom Q_{mit} patrí do množiny celých čísel a F_{mit} patrí do množiny reálnych čísel.

4.4 Vstupné dáta

Optimalizačné modely sú nástroje, ktoré pomáhajú organizáciám prijímať dôležité rozhodnutia. Každý optimalizačný model môže však poskytovať iba také kvalitné výsledky, ako kvalitné sú dáta, ktoré do neho vstupujú. Z tohto dôvodu dáta, ktoré vstupujú do optimalizačného modelu, zohrávajú rozhodujúcu úlohu pri presnosti a účinnosti výstupov z modelu. V tejto časti sa preto zameriame na dáta, ktoré do optimalizačného modelu vstupujú, ako tieto dáta získavame, spracúvame a transformujeme, aby boli s modelom kompatibilné a súčasne, aby model poskytoval najlepšie možné výsledky.

Prezentovaný model je model dynamický, čo znamená, že vstupné dáta sa v čase menia. Z toho dôvodu je dôležité upresniť obdobie, za ktoré je optimalizácia vykonávaná. V našom prípade sme stanovili plánovací horizont na tri kalendárne mesiace. Z dôvodu pomerne vysokej volatility a neistoty na trhoch v súčasnosti je zbytočné uvažovať o dlhšom plánovacom horizonte. Analyzovaný plánovací horizont začína v 26. týždni kalendárneho roka 2022 a končí 37. týždňom kalendárneho roka 2022. Plánovací horizont teda obsahuje 12 kalendárnych týždňov, pričom počas tohto obdobia bolo potrebné na základe objednávok zákazníkov produkovať 232 rôznych výrobkov. Pri zahrnutí všetkých výrobkov je však optimalizačný model priveľký na to, aby sme ho vedeli riešiť v rozumnom čase. Kompletný model má 1 311 316 ohraničení a 1 314 512 premenných, z ktorých je 11 136 spojitých a 1 303 376 celočíselných. Optimalizačný model takých rozmerov je s nami disponovanou výpočtovou technikou neriešiteľný. Museli sme preto pristúpiť k jeho redukcii. Vytvorili

sme si dva súbory údajov. Jeden súbor údajov obsahuje údaje o všetkých výrobkoch a druhý (redukovaný súbor) obsahuje iba ich časť. Niektoré údaje v redukovanom súbore sú náhodne generované a nemusia korešpondovať s realitou. Redukovaný súbor údajov využijeme na testovanie a porovnanie výkonnosti optimalizačných techník, ktoré budeme v práci využívať. Redukovaný súbor údajov spolu s vysvetlivkami jednotlivých atribútov uvádzame v prílohách č. 1 a 2. Kompletný súbor údajov uvádzame v prílohách č. 3 a 4. Z dôvodu veľkých rozmerov kompletného súboru údajov, uvádzame iba jeho začiatok a koniec. Nakoľko má identickú štruktúru s redukovaným súborom, nie je potrebné uvádzať vysvetlivky.

Dáta potrebné na optimalizáciu plánovania v nami skúmanom objekte sme získavali dôkladným zisťovaním a pozorovaním. Dáta sme čerpali zo zdrojov, ako sú vnútro podnikové databázy či expertné odhady zamestnancov a technológov. Pre náš optimalizačný model sú nevyhnutné nasledujúce vstupné dáta:

(a) Dopyt zákazníkov

Dopyt zákazníkov je determinovaný na základe potvrdených objednávok od zákazníkov. Objem produkcie každého z výrobkov a čas jeho dodania zákazníkovi závisí od objednávky zákazníka. V nami skúmanom objekte sa vyskytujú výrobky, ktoré sú dodávané zákazníkovi v rovnakých množstvách a v pravidelných časových intervaloch, avšak vyskytujú sa aj jednorázové objednávky zákazníka. Dopyt zákazníkov je reprezentovaný dvojrozmerným polom, ktoré udáva koľko kusov konkrétneho výrobku je potrebné dodať zákazníkovi v konkrétnom kalendárnom týždni.

(b) Zastupiteľnosť strojov

Zastupiteľnosť strojov je ich schopnosť zastúpiť sa pri produkcii určitého výrobku. V našom prípade ju reprezentuje dvojrozmerné pole, ktorého prvky sú binárne hodnoty. V prípade, ak je výrobok možné produkovať na danom stroji, nadobúda hodnotu 1, v opačnom prípade nadobúda hodnotu 0.

(c) Náklady na skladovanie produkcie

Ak spoločnosť vyprodukuje výrobky, no nepredá ich, je to pre spoločnosť stratená príležitosť. Ak by spoločnosť produkciu predala, mohla by finančné prostriedky

z jej predaja investovať do iných aktív, prípadne produkovať pre iného zákazníka. Za náklady skladovania jedného kusu výrobku považujeme hodnotu, ktorú spoločnosť musela investovať do jeho produkcie, kým nadobudol skladovaný stav.

(d) Počiatočné zásoby produkcie

Počiatočná zásoba je objem produkcie konkrétneho výrobku, ktorý má spoločnosť na sklade v období $t = 0$.

(e) Cyklový čas výrobkov

Cyklový čas sa vzťahuje na celkový čas potrebný na výrobu jednej jednotky výrobku od začiatku výrobných operácií až po jej koniec. V našom prípade tento čas uvádzame v minútach.

(f) Náklady na sklz

Keďže penalizácie za omeškanie dodávky zákazníkovi majú zväčša paušálnu formu, je pomerne náročné ich prepočítať na jednotku produkcie. V nami skúmanom objekte majú tieto náklady skôr podobu váh. Čím sú tieto náklady vyššie, tým viac bude model uprednostňovať ich znižovanie. Čím sú tieto náklady nižšie, tým bude model náchylnejší na ich akceptáciu. Keďže tieto náklady sú stanovované pre každý výrobok zvlášť, umožňujú nám do modelu zaviesť dôležitosť včasnej dodávky konkrétnych výrobkov.

(g) Počiatočný sklz

Je objem produkcie konkrétneho výrobku v čase $t = 0$, ktorý ešte nebol vyprodukovaný, aj keď už mal byť dodaný zákazníkovi.

(h) Sekvenčne závislé časy pretypovania

Pod sekvenčnou závislosťou časov pretypovania rozumieme časy potrebné na prestavbu stroja z i -teho výrobku na j -ty výrobok. Tieto časy reprezentuje matica pretypovania, ktorú uvádzame v prílohách č. 2 a 4. Matica pretypovania obsahuje časy potrebné na prestavbu stroja (v minútach) z každého výrobku na každý.

(i) Minútový faktor liniek

Čas, kedy je montážna linka v prestavbe, znamená pre spoločnosť stratenú príležitosť, nakoľko v tomto momente nie je linka schopná zhotovovať žiadne výrobky.

V tomto prípade je stratená príležitosť v podobe pokračujúcej produkcie výrobku, na ktorý je stroj nastavený. Náklady na prestavbu montážnej linky budeme definovať ako náklady obetovanej príležitosti. Keďže sú obe linky z dlhodobého hľadiska prispôsobiteľné, náklady na prestavbu oboch liniek budeme považovať za rovnaké. Náklady obetovanej príležitosti sme vypočítali ako prevrátenú hodnotu cyklového času výrobku vynásobenú jeho hodnotou. Následne sme tieto hodnoty pre všetky výrobky spriemerovali. Získali sme tak priemernú hodnotu výrobku, ktorú je linka schopná vyprodukovať za minútu.

(j) Disponibilná kapacita montážnych liniek

Disponibilná kapacita linky je celkový disponibilný čas (v minútach), ktorý môže montážna linka produkovať za jeden týždeň. Tento čas determinuje počet zmien, v ktorých sa pracuje na danom pracovisku. V nami skúmanom objekte sa pracuje v trojzmennej prevádzke. Maximálny disponibilný týždenný čas je 7200 minút, avšak tento čas môže byť krátený vplyvom dovoleniek, štátnych sviatkov, odstávok strojov a podobne.

(k) Počiatočné nastavenie montážnych liniek

Tento údaj nám hovorí o tom, na produkciu akých výrobkov je m -tá linka nastavená v čase $t = 0$.

4.5 Exaktné riešenie optimalizačnej úlohy

Úlohu zmiešaného celočíselného programovania sme riešili v prostredí programovacieho jazyka python s pomocou optimalizačného solvera Gurobi²⁷. Zdrojový kód uvádzame v prílohe č. 5. Riešenie sme realizovali na osobnom počítači s procesorom Intel Core i7 10. generácie so štyrmi fyzickými jadrami a ôsmimi logickými procesormi. Redukovaný model obsahuje 3 292 ohraničení a 3 380 premenných, z čoho je 480 spojitých a 2 900 celočíselných premenných. Riešenie úlohy trvalo 386,34 sekúnd, pričom získaný výstup uvádzame na obrázku č. 4.

²⁷<https://www.gurobi.com/documentation/>

Obr. 4: Výstup z Gurobi solvera

Explored 369103 nodes (12564823 simplex iterations) in 386.34 seconds (327.80 work units)
Thread count was 8 (of 8 available processors)

Solution count 10: 240278 240278 240424 ... 241320

Optimal solution found (tolerance 1.00e-04)

Best objective 2.402776981239e+05, best bound 2.402776981239e+05, gap 0.0000%

Hodnota účelovej funkcie

Zdroj: vlastné spracovanie

Pri rešpektovaní optimálneho výrobného plánu bude spoločnosť vytvárať náklady vo výške 240 277,69 €. Optimálny výrobný plán pre obe montážne linky uvádzame v tabuľkách č. 10 a 11. V tabuľkách uvádzame poradie, v ktorom má spoločnosť výroby produkovať a súčasne výrobné dávky, v akých má spoločnosť tieto výrobky produkovať, aby zabezpečila minimálne náklady ich produkcie. Rovnako uvádzame aj čistý výrobný čas, čistý čas pretypovania a taktiež súčet týchto časov pre každý výrobok v každom období zvlášť. Pre ilustráciu interpretujeme výrobný plán v 26. kalendárnom týždni na montážnej linke EQ-85025917. Na zabezpečenie optimálneho výrobného plánu má spoločnosť v 26. kalendárnom týždni na výrobnnej linke EQ-85025917 zhotovovať nasledujúce 3 výrobky. Ako prvé bude produkovať výrobok č. 000038580-6063-16 vo výrobnnej dávke 3 999 kusov. Čistý výrobný čas výrobnnej dávky je 359,91 minút. Čas pretypovania je rovný 0, nakoľko stroj bol počiatočne nastavený na produkciu tohto výrobku. Nasledujúcim výrobkom je výrobok č. 000073229-0063-17 vo výrobnnej dávke 7 470 kusov. Čistý výrobný čas produkcie výrobnnej dávky je 485,55 minút. Čas pretypovania z výrobku 000038580-6063-16 na výrobok 000073229-0063-17 je 120 minút, čo je spolu s produkčným časom 605,55 minút. Ďalším výrobkom v poradí je výrobok č. 000242462-6063-13. Tento výrobok je na zabezpečenie optimálneho výrobného plánu potrebné produkovať vo výrobnnej dávke 5 922 kusov. Čistý výrobný čas sa rovná 414,54 minút a čas prestavby stroja z predošlého výrobku je 120 minút, čo spolu tvorí 534,54 minút. Analogicky možno interpretovať výrobný plán aj pre nasledujúce kalendárne týždne pre obe výrobné linky.

Tabuľka 10: Výrobný plán pre linku EQ-85025917

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
064886085-2061-43	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
000073229-0063-17	Poradie	2	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	7470	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	485,55	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	120	0	0	0	0	0	0	0	0	0	0
	Čas spolu	605,55	0	0	0	0	0	0	0	0	0	0
000199095-0063-18	Poradie	0	0	0	0	0	0	0	0	3	1	0
	Výrobná dávka	0	0	0	0	0	0	0	0	1368	7600	0
	Výrobný čas	0	0	0	0	0	0	0	0	136,8	760	0
	Čas pretypovania	0	0	0	0	0	0	0	0	150	0	0
	Čas spolu	0	0	0	0	0	0	0	0	286,8	760	0
000075353-6063-43	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
000239232-6063-13	Poradie	0	0	0	0	0	0	0	0	2	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	5270	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	326,74	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	120	0	0
	Čas spolu	0	0	0	0	0	0	0	0	446,74	0	0
000038580-6063-16	Poradie	1	0	0	0	0	2	1	1	1	0	2
	Výrobná dávka	3999	0	0	0	0	1944	1156	5100	0	4000	4100
	Výrobný čas	359,91	0	0	0	0	174,96	104,04	459	0	360	369
	Čas pretypovania	0	0	0	0	0	120	0	0	0	150	0
	Čas spolu	359,91	0	0	0	0	294,96	104,04	459	0	510	369
000071986-6063-17	Poradie	0	2	0	0	3	1	0	0	2	0	0
	Výrobná dávka	0	5500	0	0	5000	5000	0	0	5000	0	0
	Výrobný čas	0	335,5	0	0	305	305	0	0	305	0	0
	Čas pretypovania	0	120	0	0	120	0	0	0	120	0	0
	Čas spolu	0	455,5	0	0	425	305	0	0	425	0	0
000242462-6063-13	Poradie	3	1	0	0	2	0	0	0	3	1	0
	Výrobná dávka	5922	7098	0	0	12810	0	0	0	9605	0	0
	Výrobný čas	414,54	496,86	0	0	896,7	0	0	0	672,35	0	0
	Čas pretypovania	120	0	0	0	120	0	0	0	120	0	0
	Čas spolu	534,54	496,86	0	0	1016,7	0	0	0	120	672,35	0
000040606-6063-45	Poradie	0	3	1	1	1	0	0	0	0	0	0
	Výrobná dávka	0	1724	20270	3160	0	0	0	0	0	0	0
	Výrobný čas	0	127,576	1499,98	233,84	0	0	0	0	0	0	0
	Čas pretypovania	0	120	0	0	0	0	0	0	0	0	0
	Čas spolu	0	247,576	1499,98	233,84	0	0	0	0	0	0	0

Zdroj: vlastné spracovanie

Tabuľka 11: Výrobný plán pre linku EQ-85025921

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	Poradie	2	1	0	0	0	0	0	2	1	1	0
	Výrobná dávka	5400	3700	0	0	0	0	0	4800	200	4800	0
	Výrobný čas	324	222	0	0	0	0	0	288	12	288	0
	Čas pretypovania	120	0	0	0	0	0	0	120	0	0	0
	Čas spolu	444	222	0	0	0	0	0	408	12	288	0
064886085-2061-43	Poradie	0	2	0	0	2	0	2	0	0	4	1
	Výrobná dávka	0	12700	0	0	3500	0	3500	0	0	3500	0
	Výrobný čas	0	889	0	0	245	0	245	0	0	245	0
	Čas pretypovania	0	120	0	0	120	0	150	0	0	120	0
	Čas spolu	0	1009	0	0	365	0	395	0	0	365	0
000073229-0063-17	Poradie	0	0	0	3	1	0	0	0	0	0	0
	Výrobná dávka	0	0	0	11520	2520	0	0	0	0	0	0
	Výrobný čas	0	0	0	748,8	163,8	0	0	0	0	0	0
	Čas pretypovania	0	0	0	120	0	0	0	0	0	0	0
	Čas spolu	0	0	0	868,8	163,8	0	0	0	0	0	0
000199095-0063-18	Poradie	0	0	0	0	3	1	1	0	0	0	0
	Výrobná dávka	0	0	0	0	5016	4712	0	0	0	0	0
	Výrobný čas	0	0	0	0	501,6	471,2	0	0	0	0	0
	Čas pretypovania	0	0	0	0	150	0	0	0	0	0	0
	Čas spolu	0	0	0	0	651,6	471,2	0	0	0	0	0
000075353-6063-43	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
000239232-6063-13	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
000038580-6063-16	Poradie	0	0	0	2	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	4401	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	396,09	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	120	0	0	0	0	0	0	0
	Čas spolu	0	0	0	516,09	0	0	0	0	0	0	0
000071986-6063-17	Poradie	0	0	0	0	0	0	0	0	0	3	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	5000	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	305	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	120	0
	Čas spolu	0	0	0	0	0	0	0	0	0	425	0
000242462-6063-13	Poradie	1	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	10800	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	756	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	756	0	0	0	0	0	0	0	0	0	0
000040606-6063-45	Poradie	0	3	1	1	0	0	3	1	1	0	2
	Výrobná dávka	0	2013	13513	0	0	0	5760	0	4560	0	4560
	Výrobný čas	0	148,962	999,962	0	0	0	426,24	0	337,44	0	337,44
	Čas pretypovania	0	120	0	0	0	0	120	0	0	0	120
	Čas spolu	0	268,962	999,962	0	0	0	546,24	0	337,44	0	457,44

Zdroj: vlastné spracovanie

V tabuľkách č. 12 a 13 uvádzame kapacitnú analýzu optimálnych výrobných plánov montážnych liniek. Kapacitná analýza nám poskytuje veľa hodnotných poznatkov. Za povšimnutie stojí, že výrobné linky zďaleka nie sú naplno využité. V optimálnych výrobných plánoch sa vyskytujú obdobia, kedy je montážna linka dokonca využitá na menej ako 50 %. Môže to byť spôsobené aplikáciou neoptimálnych výrobných plánov v minulosti, vplyvom čoho boli montážne linky síce kapacitne využité, no neefektívnym spôsobom.

Tabuľka 12: Kapacitná analýza linky EQ-85025917

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
064886085-2061-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000073229-0063-17	605,55	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000199095-0063-18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	286,80	760,00	0,00
000075353-6063-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000239232-6063-13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	446,74	0,00	0,00
000038580-6063-16	359,91	0,00	0,00	0,00	0,00	294,96	104,04	459,00	0,00	0,00	510,00	369,00
000071986-6063-17	0,00	455,50	0,00	0,00	425,00	305,00	0,00	0,00	425,00	0,00	0,00	0,00
000242462-6063-13	534,54	496,86	0,00	0,00	1016,70	0,00	0,00	0,00	120,00	672,35	0,00	0,00
000040606-6063-45	0,00	247,58	1499,98	233,84	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Využitá kapacita	1500,00	1199,94	1499,98	233,84	1441,70	599,96	104,04	459,00	545,00	1405,89	1270,00	369,00
Disponibilná kapacita	1500,00	1200,00	1500,00	1500,00	1500,00	600,00	1500,00	1500,00	1800,00	1500,00	1300,00	1500,00
Zaťaženosť stroja (%)	100,00	99,99	100,00	15,59	96,11	99,99	6,94	30,60	30,28	93,73	97,69	24,60
Z toho pretypovanie	240,00	240,00	0,00	0,00	240,00	120,00	0,00	0,00	240,00	270,00	150,00	0,00
Z toho výroba	1260,00	959,94	1499,98	233,84	1201,70	479,96	104,04	459,00	305,00	1135,89	1120,00	369,00
Pomer pretypovania (%)	16,00	20,00	0,00	0,00	16,65	20,00	0,00	0,00	44,04	19,20	11,81	0,00

Zdroj: vlastné spracovanie

Tabuľka 13: Kapacitná analýza linky EQ-85025921

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	444,00	222,00	0,00	0,00	0,00	0,00	0,00	0,00	408,00	12,00	288,00	0,00
064886085-2061-43	0,00	1009,00	0,00	0,00	365,00	0,00	395,00	0,00	0,00	0,00	365,00	0,00
000073229-0063-17	0,00	0,00	0,00	868,80	163,80	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000199095-0063-18	0,00	0,00	0,00	0,00	651,60	471,20	0,00	0,00	0,00	0,00	0,00	0,00
000075353-6063-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000239232-6063-13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000038580-6063-16	0,00	0,00	0,00	516,09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000071986-6063-17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	425,00	0,00
000242462-6063-13	756,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000040606-6063-45	0,00	268,96	999,96	0,00	0,00	0,00	546,24	0,00	337,44	0,00	457,44	0,00
Využitá kapacita	1200,00	1499,96	999,96	1384,89	1180,40	471,20	941,24	0,00	745,44	12,00	1535,44	0,00
Disponibilná kapacita	1200,00	1500,00	1000,00	1500,00	1500,00	600,00	1500,00	1500,00	900,00	1500,00	1700,00	1500,00
Zaťaženosť stroja (%)	100,00	100,00	100,00	92,33	78,69	78,53	62,75	0	82,83	0,80	90,32	0
Z toho pretyp	120,00	240,00	0,00	240,00	270,00	0,00	270,00	0,00	120,00	0,00	360,00	0,00
Z toho výroba	1080,00	1259,96	999,96	1144,89	910,40	471,20	671,24	0,00	625,44	12,00	1175,44	0,00
Pomer pretypovania (%)	10,00	16,00	0,00	17,33	22,87	0,00	28,69	0,00	16,10	0,00	23,45	0,00

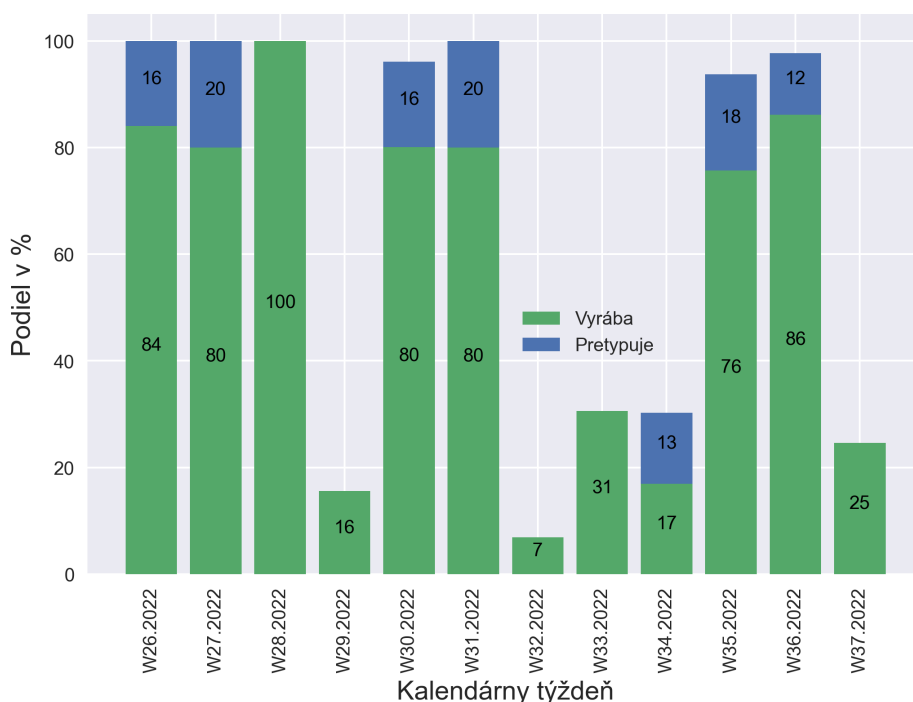
Zdroj: vlastné spracovanie

Optimalizáciou sme získali také výrobné plány, ktoré sú časovo efektívnejšie najmä vplyvom minimalizácie nákladov na pretypovanie montážnych liniek. Spoločnosť tak môže prijať dodatočné objednávky na produkciu výrobkov. Rovnako za zmienku

stojí pomer pretypovania z využitej kapacity montážnej linky. Konceptia štíhlej výroby považuje iba čistý produkčný čas za čas, ktorý pridáva výrobku hodnotu. Všetko ostatné považuje za plytvanie. Cieľom štíhlej výroby je maximalizovať pridanú hodnotu a zároveň minimalizovať plytvanie. Informácia o pomere pretypovania zohráva pre nás dôležitú úlohu najmä pri porovnávaní časovej efektívnosti výrobných plánov. Rovnako dôležité je poznamenať, že cenotvorba výrobkov je stanovená na základe určitého pomeru pretypovania, ktorý je zákazník ochotný zaplatiť. Ak výrobný plán tento pomer prekročí, podnik sa odkloní od plánovaného rozpočtu, čo má negatívny vplyv na ekonomiku podniku. Optimálny výrobný plán však reprezentuje najlepší možný spôsob, ako produkciu výrobkov naplánovať, vzhľadom na náklady s ňou spojené. Nakoľko neexistuje lepší spôsob, táto metrika má pre nás najmä informatívny charakter. Keďže cieľom tejto práce nie je cenotvorba výrobkov, ale optimalizácia výrobných dávok a plánov spojených s ich produkciou, nebudeme sa jej ďalej venovať.

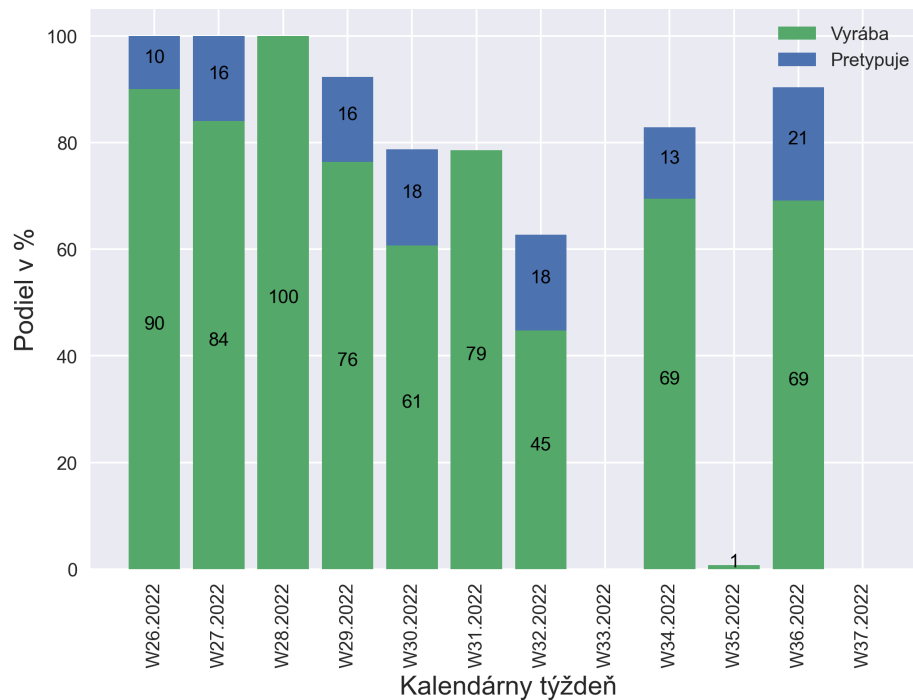
Na grafoch č. 1 a 2. vizualizujeme percentuálne rozloženie čistého výrobného času a času potrebného na prestavbu linky vzhľadom na celkovú disponibilnú kapacitu montážnej linky. Bližšie sa týmto výstupom budeme zaoberať pri hodnotení efektívnosti výrobného plánu získaného pomocou genetického algoritmu.

Graf 1: Percentuálne využitie linky EQ-85025917



Zdroj: vlastné spracovanie

Graf 2: Percentuálne využitie linky EQ-85025921



Zdroj: vlastné spracovanie

Najdôležitejším kritériom na porovnanie efektívnosti výrobných plánov sú jednotlivé nákladové metriky. Ide o hodnoty rozhodovacích premenných optimálneho riešenia zahrnutých v účelovej funkcii nami formulovaného modelu a ich agregácie. V tabuľke č. 14 uvádzame hodnoty nákladových metrick optimálneho výrobného plánu pre všetky montážne linky a výrobky spolu, avšak pre každé obdobie zvlášť.

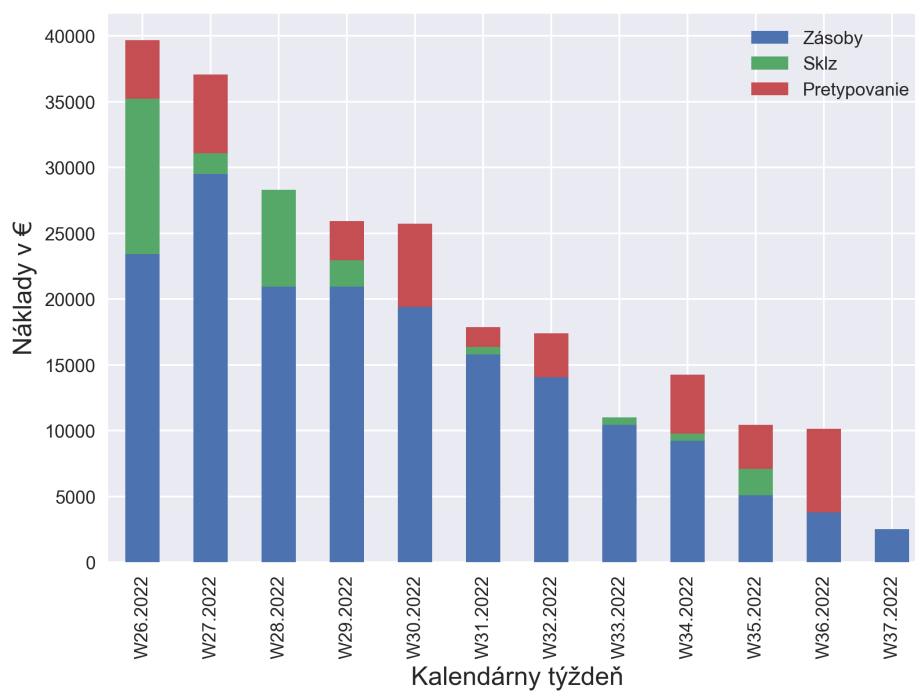
Tabuľka 14: Celkové náklady

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022	Σ
Celkom zásoby €	23420,69	29506,64	20930,36	20930,36	19390,44	15795,46	14039,08	10444,10	9221,52	5080,76	3792,68	2504,60	175056,69
Celkom skiz €	11789,30	1581,30	7366,90	2006,40	0,00	578,00	0,00	547,20	547,20	2000,00	0,00	0,00	26416,30
Celkom pretypovanie €	4477,47	5969,96	0,00	2984,98	6343,08	1492,49	3358,10	0,00	4477,47	3358,10	6343,08	0,00	38804,71
Spolu €	39687,46	37057,90	28297,26	25921,73	25733,52	17865,95	17397,18	10991,30	14246,19	10438,86	10135,76	2504,60	240277,70

Zdroj: vlastné spracovanie

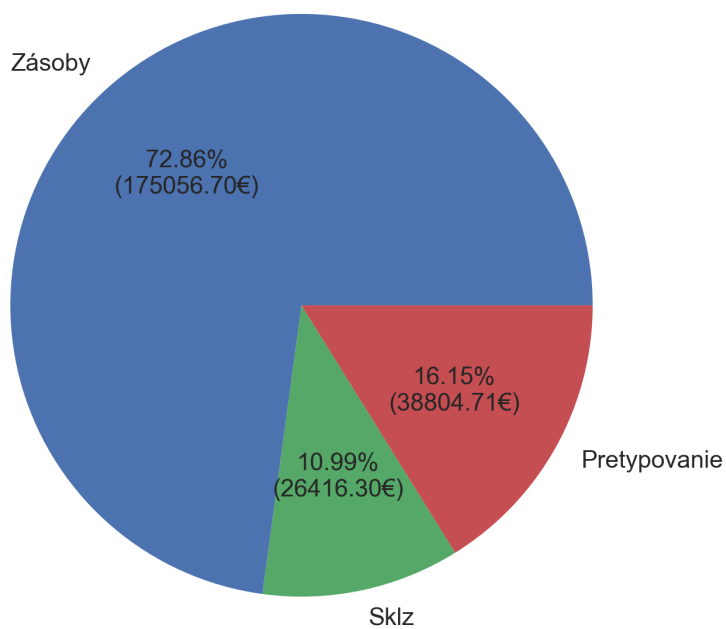
Rozloženie nákladov v rámci jednotlivých období graficky vizualizujeme na grafe č. 3. Za povšimnutie stojí postupný pokles nákladov v priebehu plánovacieho horizontu. Ide najmä o náklady na skladovanie produkcie. Tento jav je spôsobený ich príliš vysokým počiatočným stavom. Optimálny výrobný plán sa usiluje o postupné odbúranie týchto nákladov v priebehu celého plánovacieho horizontu.

Graf 3: Rozloženie nákladov



Zdroj: vlastné spracovanie

Graf 4: Celkové rozloženie nákladov



Zdroj: vlastné spracovanie

Na grafe č. 4 prezentujeme rozloženie nákladov za všetky obdobia, linky a výrobky spolu, pričom najväčšou mierou k celkovým nákladom prispievajú náklady na skladovanie produkcie a najmenšou mierou sa na celkových nákladoch podieľajú náklady spôsobené sklzom.

Detailnejší pohľad na nákladové metriky nám poskytuje nákladová analýza, ktorú uvádzame v tabuľke č. 15. Táto analýza nám umožňuje odhaliť, akou mierou konkrétne výrobky prispievajú k hodnote účelovej funkcie. Rovnako nám umožňuje zistiť informácie o tom, akým spôsobom sú náklady v rámci konkrétneho výrobku rozložené, a to ako v peňažných, tak aj v naturálnych jednotkách.

Ako ilustratívny príklad interpretácie nákladovej analýzy použijeme 26. kalendárny týždeň. Celkové náklady produkcie v stanovenom plánovacom horizonte tvoria 240 277,69 €. Tieto náklady sme ďalej rozložili na náklady spôsobené skladovaním produkcie, náklady na prestavbu montážnych liniek a na náklady spôsobené oneskorením dodávky objednanej produkcie. Nakoľko náklady spôsobené skladovaním produkcie sa na celkových nákladoch podieľajú najväčšou mierou, znamenajú pre podnik najväčší potenciál pre vytvorenie úspory. Z tohto dôvodu je pre podnik dôležité zistiť informácie o tom, ktoré výrobky sa na vytváraní nadbytočných zásob najviac podieľajú. Z nákladovej analýzy je zrejmé, že v 26. kalendárnom týždni ide o výrobok č. 000075353-6063-43. Analogicky možno interpretovať aj ostatné nákladové metriky.

Tabuľka 15: Nákladová analýza

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	Zásoby (ks)	0	3700	200	200	0	0	0	0	0	0	0
	Zásoby (€)	0,00	3225,66	174,36	174,36	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	1492,49	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1492,49	0,00	0,00
064886085-2061-43	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	9200	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	9200,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	1492,49	0,00	0,00	1492,49	0,00	1865,61	0,00	0,00	0,00	1492,49
000073229-0063-17	Zásoby (ks)	0	0	0	0	540	540	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	330,05	330,05	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	1492,49	0,00	0,00	1492,49	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000199095-0063-18	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	3952	3952	5016	5016	0	0	0	1368	1368	0	0
	Sklz (€)	1580,80	1580,80	2006,40	2006,40	0,00	0,00	0,00	547,20	547,20	0,00	0,00
	Pretypovanie (€)	0,00	0,00	0,00	0,00	1865,61	0,00	0,00	0,00	0,00	1865,61	0,00
000073553-6063-43	Zásoby (ks)	20200	20200	17900	17900	16100	12500	12500	8900	8900	7100	5300
	Zásoby (€)	14455,12	14455,12	12809,24	12809,24	11521,16	8945,00	8945,00	6368,84	6368,84	5080,76	3792,68
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000239232-6063-13	Zásoby (ks)	13640	12400	12090	12090	11470	9920	7750	6200	4340	0	0
	Zásoby (€)	8965,57	8150,52	7946,76	7946,76	7539,23	6520,42	5094,08	4075,26	2852,68	0,00	0,00
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1492,49	0,00
000038580-6063-16	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	1	1	4401	0	0	1156	0	0	0	4000	0
	Sklz (€)	0,50	0,50	2200,50	0,00	0,00	578,00	0,00	0,00	0,00	2000,00	0,00
	Pretypovanie (€)	0,00	0,00	0,00	1492,49	0,00	1492,49	0,00	0,00	0,00	0,00	1865,61
000071986-6063-17	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	500	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	400,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	1492,49	0,00	0,00	1492,49	0,00	0,00	0,00	1492,49	0,00	1492,49
000242462-6063-13	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	608	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	608,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	1492,49	0,00	0,00	0,00	1492,49	0,00	0,00	0,00	1492,49	0,00	0,00
000040606-6063-45	Zásoby (ks)	0	3737	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	3675,34	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	0	0	3160	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	3160,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	2984,98	0,00	0,00	0,00	0,00	1492,49	0,00	0,00	0,00	1492,49

Zdroj: vlastné spracovanie

4.6 Riešenie optimalizačnej úlohy genetickým algoritmom

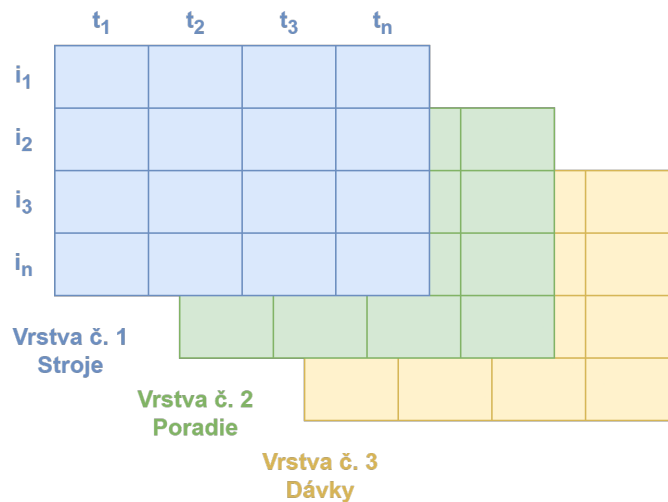
Genetický algoritmus na optimalizáciu výrobných dávok a plánovania pre nami skúmaní objekt sme programovali v jazyku python. Nepoužívali sme žiadne špecifické knižnice určené na konštrukciu genetických algoritmov. Všetky operátory genetického algoritmu sme programovali svojpomocne. Zdrojový kód uvádzame v prílohe č. 6.

4.6.1 Konštrukcia genetického algoritmu

Jedným z najdôležitejších aspektov konštrukcie genetického algoritmu je návrh chromozómu, ktorý má reprezentovať riešenie. Od jeho návrhu závisí výkonnosť algoritmu a taktiež jeho schopnosť nájsť nové potenciálne efektívne riešenia. Efektívny návrh chromozómu je taký, ktorý dokáže uchovávať všetky potrebné informácie a súčasne si zachová jednoduchosť.

Pre nami skúmaný objekt bude každé riešenie reprezentovať chromozóm rozdelený do troch vzájomne prepojených vrstiev. Každú z vrstiev reprezentuje dvojrozmerné pole obsahujúce špecifické informácie o konkrétnom výrobku v konkrétnom období. Prvá vrstva bude uchovávať informácie o priradení produkcie i -teho výrobku ku konkrétnemu stroju v t -tom období. Nakoľko nami skúmaný objekt obsahuje 2 montážne linky, hodnoty v prvej vrstve môžu nadobúdať iba hodnoty 0 (keď i -ty výrobok nebude vyrábaný v t -tom období na žiadnom stroji), 1 (ak produkcia i -teho výrobku v t -tom období je priradená stroju EQ-85025917) alebo 2 (ak produkcia i -teho výrobku v t -tom období je priradená stroju EQ-85025921). Druhá vrstva bude uchovávať informácie o tom, v akom poradí bude produkcia i -teho výrobku v t -tom období, na stroji ku ktorému bola v prvej vrstve priradená, vykonávaná. Tretia vrstva bude uchovávať informácie o veľkosti výrobných dávok i -teho výrobku v t -tom období. Táto forma návrhu chromozómu umožňuje jednoduchú aplikáciu operátorov genetického algoritmu a súčasne uchováva všetky potrebné informácie, ktoré vystihujú nami skúmaný objekt. Jediným špecifikom, čím sa táto forma reprezentácie riešenia odlišuje od prezentovaného modelu matematického programovania je, že umožňuje priradiť produkciu i -teho výrobku v t -tom období iba jednému stroju. V praxi to znamená, že nie je možné produkovať ten istý výrobok v tom istom kalendárnom týždni na oboch montážnych linkách. Ilustráciu návrhu chromozómu uvádzame na obrázku č. 5.

Obr. 5: Návrh chromozómu

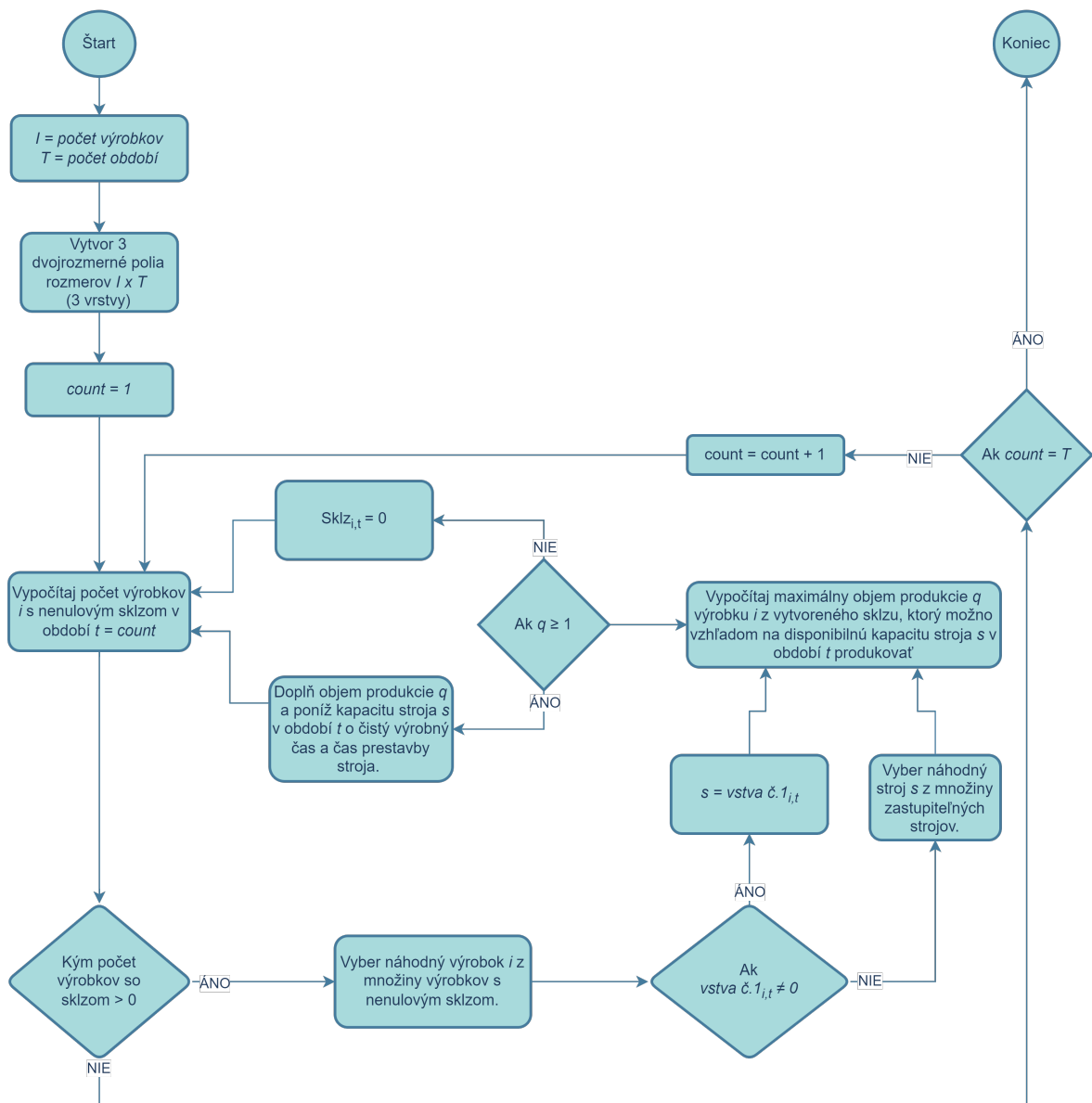


Zdroj: vlastné spracovanie

Hlavnou myšlienkou genetického algoritmu je postupné zlepšovanie počiatkovej populácie riešení. Kvalita počiatkových riešení významne ovplyvňuje efektívnosť algoritmu ako celku, a preto je dôležité venovať dostatočnú pozornosť spôsobu ich inicializácie. Na obrázku č. 6 uvádzame vývojový diagram algoritmu inicializácie počiatkového riešenia pre nami skúmaný objekt. V prvom kroku sme vytvorili 3 dvojrozmerné polia rozmerov $I * T$, ktoré majú slúžiť ako chromozóm pre riešenie. Ide o vrstvy, ktoré sme opísali vyššie. Všetky hodnoty vo všetkých vrstvách sú v počiatkovej fáze rovné 0. Z toho vyplýva, že podnik zatiaľ neprodukuje žiadne výrobky. Je to stav, kedy bude sklz dodávok rovný súčtu počiatkového sklzu a dopytu po produkcii za celý plánovací horizont. V takomto prípade bude sklz maximálny. Celý algoritmus je založený na postupnom odbúravaní tohto sklzu. Algoritmus vstúpi do cyklu, kedy overuje existenciu výrobkov s nenulovým sklzom. Ak také výrobky existujú, algoritmus vyberie náhodný výrobok z výrobkov s nenulovým sklzom a zistí, či sa už na niektorom zo strojov daný výrobok v danom období vyrába. Ak áno, algoritmus vypočíta maximálny objem produkcie z vytvoreného sklzu, ktorý je možné produkovať na danom stroji vzhľadom na jeho kapacitu. Ak sa však daný výrobok v danom období ešte nevyrába, algoritmus náhodne vyberie stroj z množiny zastupiteľných strojov a opäť vypočíta maximálny objem produkcie z vytvoreného sklzu, ktorý možno vzhľadom na kapacitu stroja produkovať. V ďalšom kroku algoritmus overí, či stanovené množstvo produkcie je väčšie, nanajvýš rovné jednej (overenie voľnej kapacity). Ak áno, tak algoritmus vypočítaný objem produkcie priradí stroju a poníži jeho voľnú kapacitu

o čistý výrobný čas a o čas prestavby stroja. Ak nie, tak algoritmus vynuluje sklz pre daný výrobok v danom období a opäť vypočíta počet výrobkov s nenulovým sklzom. Tento proces pokračuje, kým počet výrobkov s nenulovým sklzom je väčší ako 0. Ak sa táto podmienka poruší, algoritmus z cyklu vystúpi a overí, či je odbúraný sklz pre všetky obdobia plánovacieho horizontu. Ak nie, algoritmus začne odbúrať sklz pre ďalšie obdobie. Ak áno, algoritmus končí. Algoritmus opakujeme v závislosti od stanoveného počtu jedincov v populácii.

Obr. 6: Algoritmus inicializácie

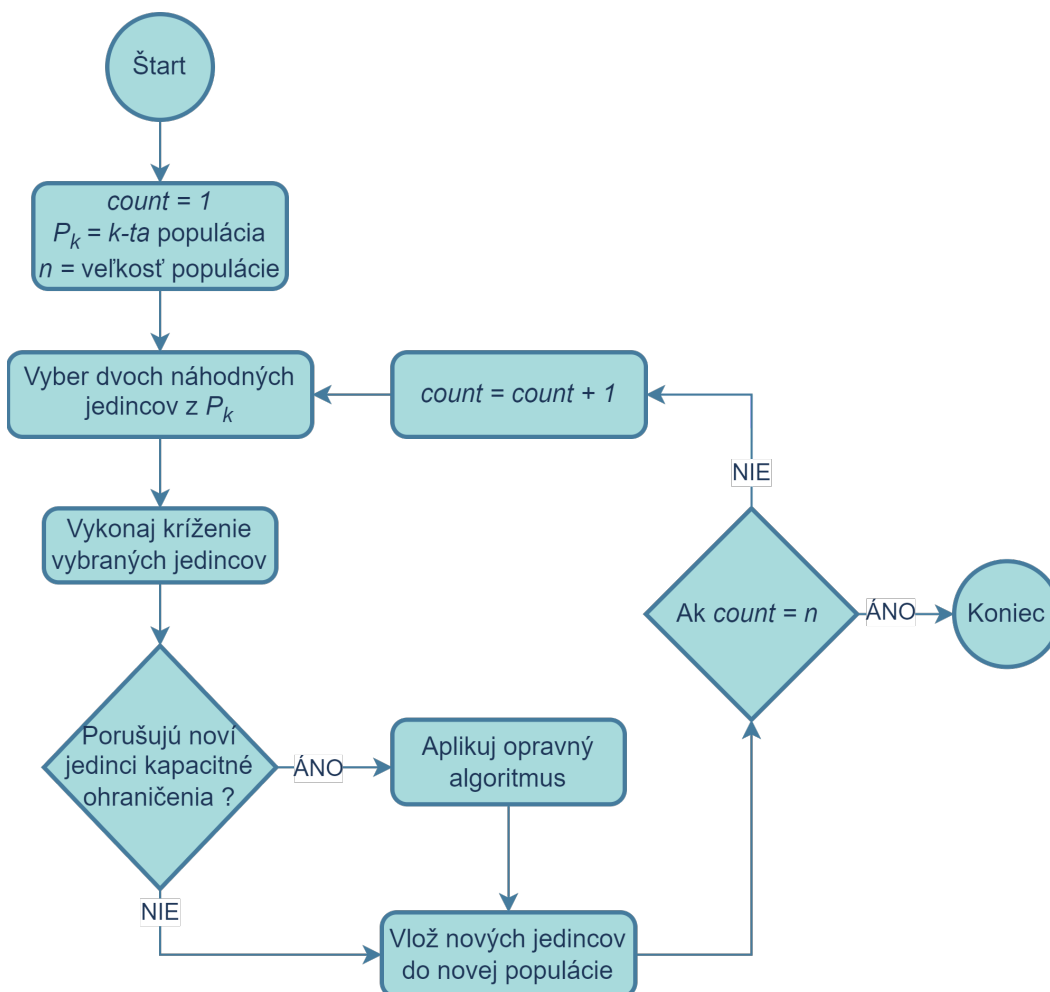


Zdroj: vlastné spracovanie

Nasledujúcim operátorom genetického algoritmu je selekcia. V tejto práci sme na realizáciu selekcie zvolili algoritmus turnajový výber s veľkosťou turnaja = 2. Turnajový výber je založený na náhodnom výbere dvoch jedincov z pôvodnej populácie. Títo jedinci následne medzi sebou hrajú turnaj. Jediniec s vyššou fitness hodnotou vyhráva a postupuje do novej populácie. Tento proces sa opakuje, kým veľkosť novej populácie sa nerovná veľkosti populácie pôvodnej. V rámci operátora selekcie v tejto práci aplikujeme stratégiu elitizmu.

Ďalším genetickým operátorom v poradí je rekombinácia. Rekombinácia je jeden z najdôležitejších operátorov genetického algoritmu. Ak operátor rekombinácie nedokáže generovať nových jedincov rozdielných od ich rodičov, genetický algoritmus s veľkou pravdepodobnosťou zlyhá. Na obrázku č. 7 uvádzame vývojový diagram algoritmu rekombinácie pre nami skúmaný objekt.

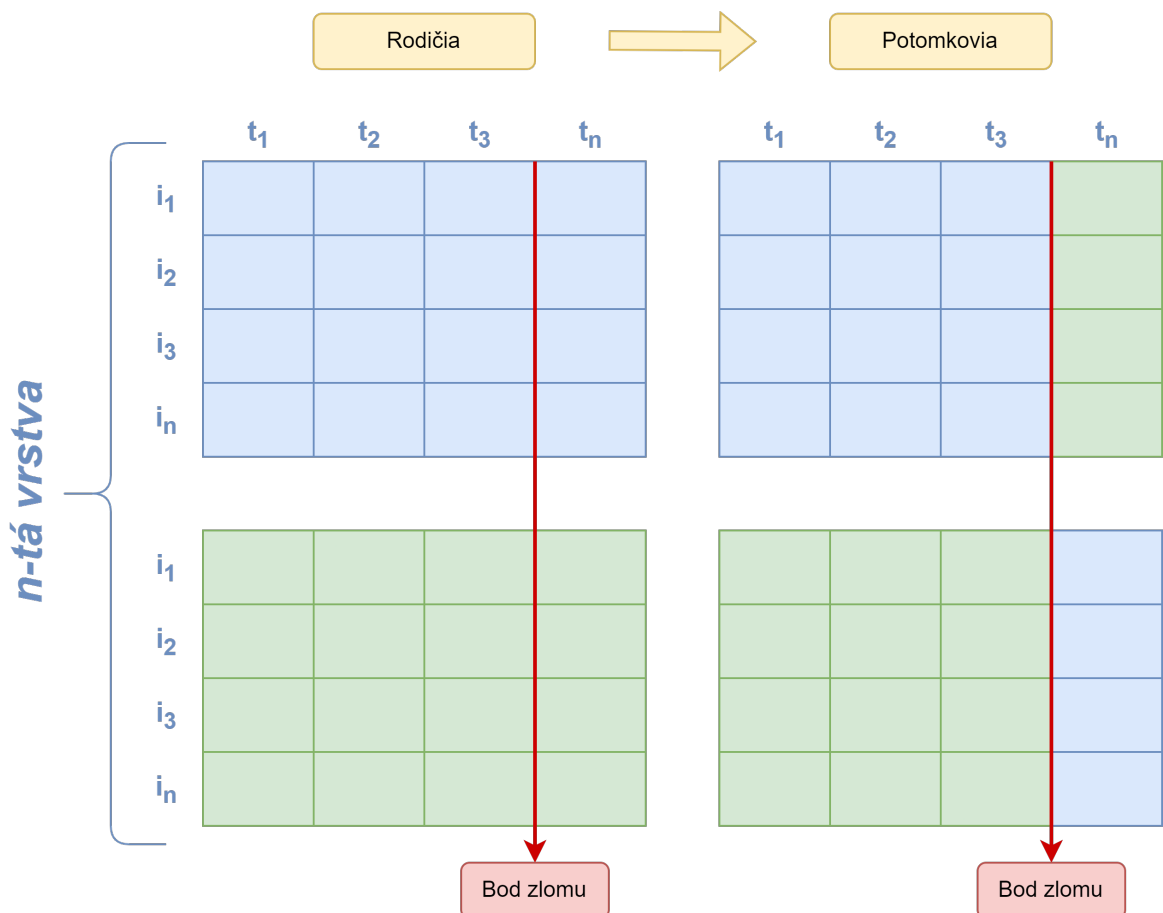
Obr. 7: Algoritmus rekombinácie



Zdroj: vlastné spracovanie

Najskôr sa náhodne vyberú dvaja jedinci z pôvodnej populácie, ktorí budú slúžiť ako rodičia pre novú populáciu. Následne sa vykoná ich rekombinácia. Operátor rekombinácie detailne ilustrujeme na obrázku č. 8. V našej práci sme využili jednoduchý spôsob jej realizácie. Najskôr sa náhodne vygeneruje bod zlomu a následne sa okolo tohoto bodu vymení genetická informácia medzi rodičmi. Tento proces je aplikovaný na všetkých troch vrstvách chromozómu. Vznikajú tak noví jedinci. Pred ich pridaním do novej populácie je však potrebné skontrolovať, či neporušujú kapacitné ohraničenia. Z toho dôvodu je aplikovaný opravný algoritmus. Tento algoritmus najskôr skontroluje prípustnosť riešení. Ak riešenia neporušujú kapacitné ohraničenia, sú pridané do novej generácie. Ak áno, opravný algoritmus v každom období uberie nevyhnutné množstvo produkcie, aby boli splnené kapacitné ohraničenia počas celého plánovacieho horizontu a následne pridá nových potomkov do novej populácie. Tento cyklus sa opakuje až do momentu, kým počet jedincov v novej populácii sa nerovná počtu jedincov v pôvodnej populácii.

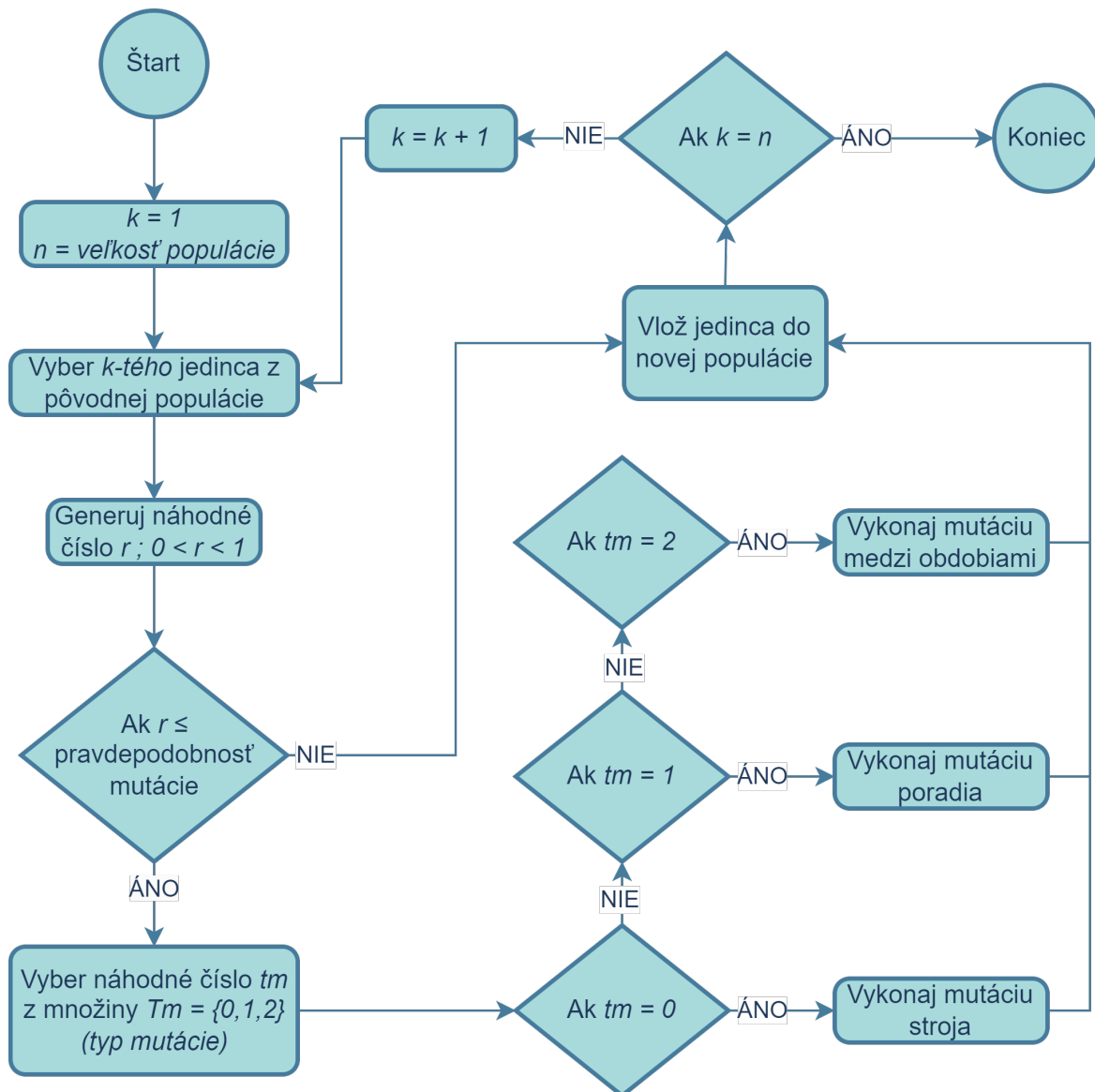
Obr. 8: Operátor rekombinácie



Zdroj: vlastné spracovanie

Posledným operátorom genetického algoritmu v poradí je mutácia. Hlavnou úlohou operátora mutácie je vniesť do populácie genetickú rozmanitosť a zabrániť tomu, aby algoritmus konvergoval k suboptimálnemu riešeniu. Algoritmus operátora mutácie pre nami riešený problém uvádzame na vývojovom diagrame na obrázku č. 9.

Obr. 9: Algoritmus operátora mutácie



Zdroj: vlastné spracovanie

Algoritmus postupne prechádza všetkých jedincov z populácie. Každému jedincovi priradí číslo z intervalu od 0 po 1 a toto číslo porovná s explicitne stanovenou pravdepodobnosťou mutácie. Ak je pravdepodobnosť mutácie vyššia ako priradené číslo, algoritmus náhodne vyberie jeden z operátorov mutácie, aplikuje ju na jedinca a zapíše

ho do novej populácie. Ak je však pravdepodobnosť mutácie nižšia ako priradené číslo, algoritmus priradí jedinca do novej populácie bez aplikácie akejkoľvek mutácie. Tento proces pokračuje, kým veľkosť novej populácie sa nerovná veľkosti populácie pôvodnej. V tejto práci sme využili nasledujúce spôsoby vykonania operátora mutácie:

(a) Mutácia stroja

Tento spôsob výkonu mutácie je založený na zmene stroja, na ktorom sa produkcia náhodne vybraného výrobku v náhodne vybranom období realizuje, prípadne úplnému zamedzeniu jeho produkcie. Mutácia stroja sa vykonáva na prvej vrstve, ale vzhľadom na kapacitné ohraňovania strojov, ovplyvňuje aj ostatné. V prvom kroku sa vygeneruje náhodné obdobie t a náhodný výrobok i . Následne algoritmus na prvej vrstve priradí výrobku náhodné číslo z množiny prípustných hodnôt (iné ako doposiaľ priradené bolo). Nakoľko sa mutácia vykonáva na prvej vrstve, množina prípustných hodnôt obsahuje hodnoty reprezentujúce zastupiteľné stroje pre náhodne vybraný výrobok i a nulu (pre prípad, že sa výrobok vyrábať v náhodne vybranom období t nebude). Ak algoritmus vyberie nulu, vymaže produkciu i -teho výrobku v t -tom období a upraví všetky vrstvy. Ak algoritmus vyberie iné číslo, produkciu i -teho výrobku umiestní na začiatok t -teho obdobia a priradí mu maximálny možný objem produkcie zo súčtu sklzu a dopytu po t -te obdobie, vzhľadom na disponibilnú kapacitu vybraného stroja.

(b) Mutácia poradia

Tento spôsob mutácie je vykonávaný na druhej vrstve chromozómu, avšak vzhľadom na disponibilnú kapacitu strojov ovplyvňuje aj tretiu vrstvu chromozómu. Najskôr sa vygeneruje náhodné obdobie t a následne sa náhodne vyberú dva výrobky i_1 a i_2 z t -teho obdobia. Ak majú oba výrobky rovnakú hodnotu v prvej vrstve (sú priradené rovnakému stroju v rovnakom období), ich poradie sa vymení. Nakoniec sa upraví tretia vrstva prostredníctvom opravného algoritmu, aby sa zabezpečila prípustnosť nového riešenia.

(c) Mutácia medzi obdobia

Tento spôsob realizácie operátora mutácie je aplikovaný na tretej vrstve chromozómu. Spočíva v náhodnom výbere výrobku i a dvoch období t_1 a t_2 (rovnakého výrobku v dvoch rôznych obdobiach). Následne je určitý objem produkcie i -teho výrobku

medzi obdobiami t_1 a t_2 zamenený. Tento objem produkcie je stanovený so zreteľom na disponibilnú kapacitu strojov a dopyt či sklz i -teho výrobku.

Po ukončení každého cyklu evolučného procesu vzniká nová generácia. Ak sa fitness hodnota najlepšieho jedinca z generácie nezlepší počas vopred stanovenej hranice počtu generácií, reinitializuje sa 80 % populácie v aktuálnej generácii. Cyklus evolučného procesu sa opakuje, až kým nedovíši vopred stanovený celkový počet generácií.

4.6.2 Aplikácia genetického algoritmu na redukovanom súbore údajov

Schopnosť genetického algoritmu konvergovať k optimálnemu riešeniu do značnej miery ovplyvňujú vstupné parametre, ktoré sme opísali v časti Konštrukcia genetického algoritmu. Ide o nasledujúce parametre:

- veľkosť populácie,
- počet generácií,
- pravdepodobnosť mutácie,
- hranica nezlepšenia.

Pre správne nastavenie parametrov je často potrebné použiť metódu pokusu a omylu. Pri riešení problému na redukovanom súbore údajov sme skúšali viaceré kombinácie vstupných parametrov. Pri každom navýšení vstupných parametrov sa sice šanca, že algoritmus začne konvergovať k optimálnemu riešeniu zvyšuje, avšak zvyšuje sa aj čas potrebný na riešenie problému. Volili sme preto istý kompromis medzi časovou náročnosťou a nárokmi na kvalitu výsledného riešenia. Najlepšie výsledky vzhľadom na čas riešenia úlohy sme získali pri stanovení veľkosti populácie na 300 jedincov, počtu generácií na 3 000, pravdepodobnosti mutácie na hodnotu 0,4 a maximálnej hranice nezlepšenia na 50 generácií.

Na grafe č. 5 vizualizujeme hodnoty nákladových metrík v priebehu riešenia problému nami vytvoreným genetickým algoritmom. Z grafu možno vidieť postupnú konvergenciu k určitému suboptimálnemu riešeniu. Dôležité je poznamenať, že sice

genetický algoritmus pri nami stanovených parametroch nedokázal nájsť optimálne riešenie, no efektívnym spôsobom dokázal počiatočné riešenia zlepšovať. Algoritmus skončil po 3 000 generáciách, pričom najlepšie riešenie našiel už v 2 548. generácii. Hodnota účelovej funkcie pre toto riešenie je 256 486,07 €. Dôležitým aspektom, aby genetický algoritmus dokázal nájsť nové potenciálne efektívne riešenia, je jeho schopnosť prijať v určitej generácii aj horšie riešenie. Tento aspekt možno vidieť aj na grafe priebehu riešenia problému. Hneď na začiatku sa fitness hodnota najlepšieho jedinca z generácie počas viacerých generácií zhoršovala, avšak v konečnom dôsledku algoritmus začal prudko konvergovať k lepšiemu a lepšiemu riešeniu. Následne algoritmus vstúpil do určitého suboptimálneho extrém, z ktorého sa však po niekoľkých desiatkach generáciách dostal a opäť začal konvergovať k lepšiemu riešeniu. Rovnako je z grafu možné vidieť, že inicializačné riešenie obsahovalo pomerne vysoké náklady na sklz a naopak, nízke náklady skladovania. V priebehu riešenia však za účelom zlepšenia agregovaného kritéria (celkové náklady) algoritmus prijímal riešenia s vyššími nákladmi na skladovanie produkcie a s nižšími nákladmi na vytvorený sklz. Algoritmus taktiež dokázal pomerne efektívnym spôsobom odbúrať náklady na pretypovanie montážnych liniek.

V tabuľkách č. 16 a 17 uvádzame výrobné plány riešenia poskytnutého genetickým algoritmom pre obe výrobné linky. Nakoľko však ilustratívny príklad interpretácie jednotlivých hodnôt tabuľky sme rozobrali pri exaktnom riešení úlohy zmiešaného celočíselného programovania, nebudeme sa interpretácii týchto hodnôt už venovať. Interpretácia hodnôt obsiahnutých vo výrobných plánoch získaných prostredníctvom genetického algoritmu je identická s tými získanými exaktným algoritmom.

V tabuľkách č. 18 a 19 uvádzame kapacitnú analýzu výrobných plánov montážnych liniek získaných genetickým algoritmom. Interpretácia hodnôt obsiahnutých v týchto tabuľkách je opäť analogická k tým, spomenutým pri exaktnom riešení. Rovnako ako pri exaktnom riešení, dochádzame k rovnakému záveru, a to k takému, že obe výrobné linky zd'aleka nie sú naplno využité.

Na grafoch č. 6 a 7 vizualizujeme percentuálne rozloženie čistého výrobného času a času potrebného na prestavbu linky, vzhľadom na celkovú disponibilnú kapacitu montážnej linky. Ich agregáciám sa budeme zaoberať pri hodnotení efektívnosti výrobného plánu získaného pomocou genetického algoritmu.

Graf 5: Priebeh riešenia redukovaného problému genetickým algoritmom



Zdroj: vlastné spracovanie

Tabuľka 16: Výrobný plán pre linku EQ-85025917

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
064886085-2061-43	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
000073229-0063-17	Poradie	0	0	0	2	1	0	0	0	0	0	0
	Výrobná dávka	0	0	0	11520	1980	0	0	0	0	0	0
	Výrobný čas	0	0	0	748,8	128,7	0	0	0	0	0	0
	Čas pretypovania	0	0	0	120	0	0	0	0	0	0	0
	Čas spolu	0	0	0	868,8	128,7	0	0	0	0	0	0
000199095-0063-18	Poradie	0	0	0	0	0	0	0	0	0	1	1
	Výrobná dávka	0	0	0	0	0	0	0	0	0	7600	1
	Výrobný čas	0	0	0	0	0	0	0	0	0	760	0,1
	Čas pretypovania	0	0	0	0	0	0	0	0	0	150	0
	Čas spolu	0	0	0	0	0	0	0	0	0	910	0,1
000075353-6063-43	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
000239232-6063-13	Poradie	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0
000038580-6063-16	Poradie	1	1	0	1	0	0	0	0	1	0	0
	Výrobná dávka	1854	2146	0	4400	0	0	0	0	4000	0	0
	Výrobný čas	166,86	193,14	0	396	0	0	0	0	360	0	0
	Čas pretypovania	0	120	0	120	0	0	0	0	120	0	0
	Čas spolu	166,86	313,14	0	516	0	0	0	0	480	0	0
000071986-6063-17	Poradie	0	2	0	0	2	1	0	0	2	0	0
	Výrobná dávka	0	5500	0	0	5000	5000	0	0	5000	0	0
	Výrobný čas	0	335,5	0	0	305	305	0	0	305	0	0
	Čas pretypovania	0	120	0	0	120	0	0	0	120	0	0
	Čas spolu	0	455,5	0	0	425	305	0	0	425	0	0
000242462-6063-13	Poradie	2	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	17330	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	1213,1	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	120	0	0	0	0	0	0	0	0	0	0
	Čas spolu	1333,1	0	0	0	0	0	0	0	0	0	0
000040606-6063-45	Poradie	0	3	1	0	0	0	1	1	1	0	0
	Výrobná dávka	0	4207	20270	0	0	0	5757	1	4560	0	0
	Výrobný čas	0	311,318	1499,98	0	0	0	426,018	0,074	337,44	0	0
	Čas pretypovania	0	120	0	0	0	0	120	0	0	0	0
	Čas spolu	0	431,318	1499,98	0	0	0	546,018	0,074	337,44	0	0

Zdroj: vlastné spracovanie

Tabuľka 17: Výrobný plán pre linku EQ-85025921

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022	
053345657-3062-43	Poradie	0	3	1	0	0	2	1	0	1	3	1	0
	Výrobná dávka	0	5400	3501	0	0	198	1	0	4800	200	4800	0
	Výrobný čas	0	324	210,06	0	0	11,88	0,06	0	288	12	288	0
	Čas pretypovania	0	120	0	0	0	120	0	0	120	120	0	0
	Čas spolu	0	444	210,06	0	0	131,88	0,06	0	408	132	288	0
064886085-2061-43	Poradie	2	1	0	0	1	0	3	0	0	0	4	0
	Výrobná dávka	6777	5923	0	0	3500	0	3500	0	0	0	3500	0
	Výrobný čas	474,39	414,61	0	0	245	0	245	0	0	0	245	0
	Čas pretypovania	120	0	0	0	120	0	120	0	0	0	120	0
	Čas spolu	594,39	414,61	0	0	365	0	365	0	0	0	365	0
000073229-0063-17	Poradie	1	0	0	0	0	0	2	0	0	0	0	0
	Výrobná dávka	7470	0	0	0	0	0	540	0	0	0	0	0
	Výrobný čas	485,55	0	0	0	0	0	35,1	0	0	0	0	0
	Čas pretypovania	120	0	0	0	0	0	120	0	0	0	0	0
	Čas spolu	605,55	0	0	0	0	0	155,1	0	0	0	0	0
000199095-0063-18	Poradie	0	0	2	1	0	0	4	1	0	0	0	0
	Výrobná dávka	0	0	5018	0	0	0	4707	1370	0	0	0	0
	Výrobný čas	0	0	501,8	0	0	0	470,7	137	0	0	0	0
	Čas pretypovania	0	0	150	0	0	0	150	0	0	0	0	0
	Čas spolu	0	0	651,8	0	0	0	620,7	137	0	0	0	0
000075353-6063-43	Poradie	0	0	0	0	0	0	0	0	0	0	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	0	0
000239232-6063-13	Poradie	0	0	0	0	0	0	0	0	0	2	0	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	5270	0	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	326,74	0	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	120	0	0
	Čas spolu	0	0	0	0	0	0	0	0	0	446,74	0	0
000038580-6063-16	Poradie	0	0	0	0	0	1	0	2	0	0	0	1
	Výrobná dávka	0	0	0	0	0	3100	0	5100	0	0	0	4100
	Výrobný čas	0	0	0	0	0	279	0	459	0	0	0	369
	Čas pretypovania	0	0	0	0	0	120	0	150	0	0	0	120
	Čas spolu	0	0	0	0	0	399	0	609	0	0	0	489
000071986-6063-17	Poradie	0	0	0	0	0	0	0	0	0	0	2	0
	Výrobná dávka	0	0	0	0	0	0	0	0	0	0	5000	0
	Výrobný čas	0	0	0	0	0	0	0	0	0	0	305	0
	Čas pretypovania	0	0	0	0	0	0	0	0	0	0	120	0
	Čas spolu	0	0	0	0	0	0	0	0	0	0	425	0
000242462-6063-13	Poradie	0	2	0	0	2	0	0	0	0	1	0	0
	Výrobná dávka	0	6490	0	0	12810	0	0	0	0	9605	0	0
	Výrobný čas	0	454,3	0	0	896,7	0	0	0	0	672,35	0	0
	Čas pretypovania	0	120	0	0	120	0	0	0	0	120	0	0
	Čas spolu	0	574,3	0	0	1016,7	0	0	0	0	792,35	0	0
000040606-6063-45	Poradie	0	0	0	2	0	0	0	0	0	0	3	0
	Výrobná dávka	0	0	0	16205	0	0	0	0	0	0	4560	0
	Výrobný čas	0	0	0	1199,17	0	0	0	0	0	0	337,44	0
	Čas pretypovania	0	0	0	150	0	0	0	0	0	0	120	0
	Čas spolu	0	0	0	1349,17	0	0	0	0	0	0	457,44	0

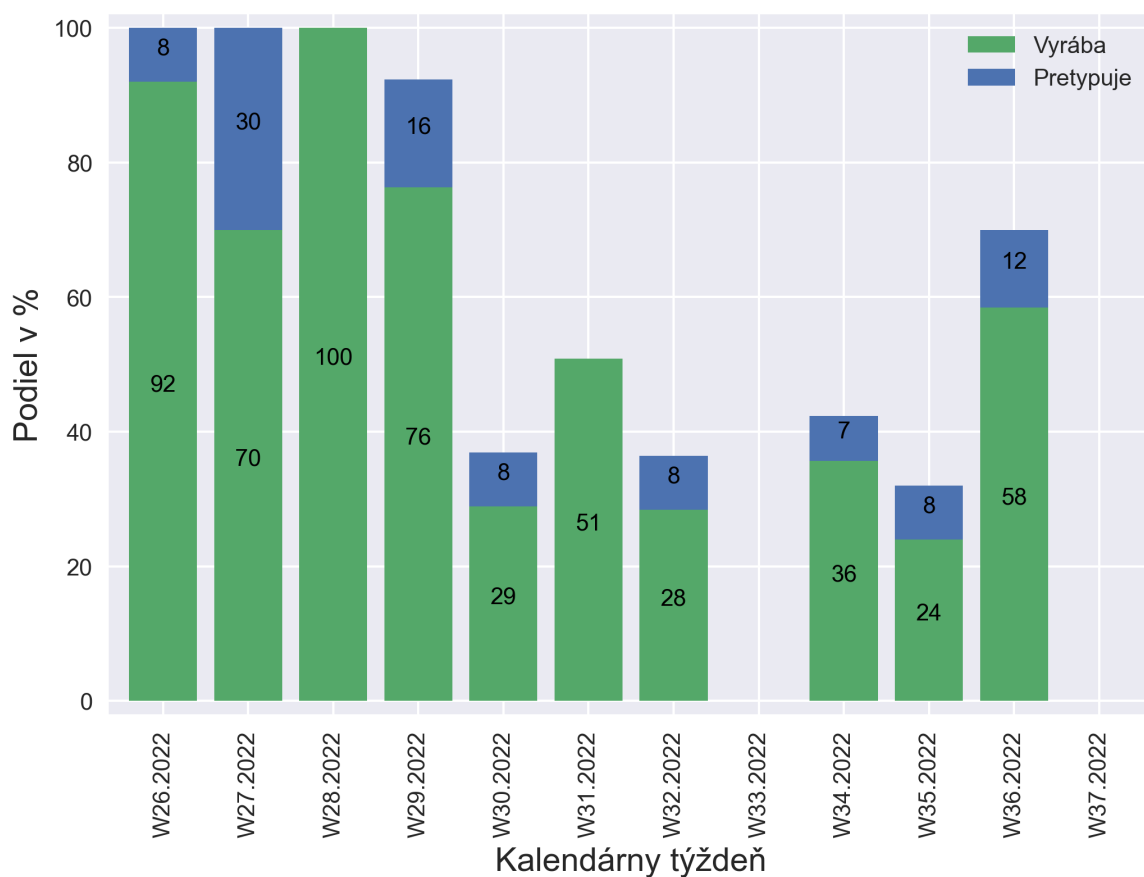
Zdroj: vlastné spracovanie

Tabuľka 18: Kapacitná analýza linky EQ-85025917

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
064886085-2061-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000073229-0063-17	0,00	0,00	0,00	868,80	128,70	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000199095-0063-18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	910,00	0,10
000075353-6063-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000239232-6063-13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000038580-6063-16	166,86	313,14	0,00	516,00	0,00	0,00	0,00	0,00	0,00	480,00	0,00	0,00
000071986-6063-17	0,00	455,50	0,00	0,00	425,00	305,00	0,00	0,00	425,00	0,00	0,00	0,00
000242462-6063-13	1333,10	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000040606-6063-45	0,00	431,32	1499,98	0,00	0,00	0,00	546,02	0,07	337,44	0,00	0,00	0,00
Využitá kapacita	1499,96	1199,96	1499,98	1384,80	553,70	305,00	546,02	0,07	762,44	480,00	910,00	0,10
Disponibilná kapacita	1500,00	1200,00	1500,00	1500,00	1500,00	600,00	1500,00	1500,00	1800,00	1500,00	1300,00	1500,00
Zaťaženosť stroja (%)	100,00	100,00	100,00	92,32	36,91	50,83	36,40	0,00	42,36	32,00	70,00	0,01
Z toho pretypovanie	120,00	360,00	0,00	240,00	120,00	0,00	120,00	0,00	120,00	120,00	150,00	0,00
Z toho výroba	1379,96	839,96	1499,98	1144,80	433,70	305,00	426,02	0,07	642,44	360,00	760,00	0,10
Pomer pretypovania (%)	8,00	30,00	0,00	17,33	21,67	0,00	21,98	0,00	15,74	25,00	16,48	0,00

Zdroj: vlastné spracovanie

Graf 6: Percentuálne využitie linky EQ-85025917



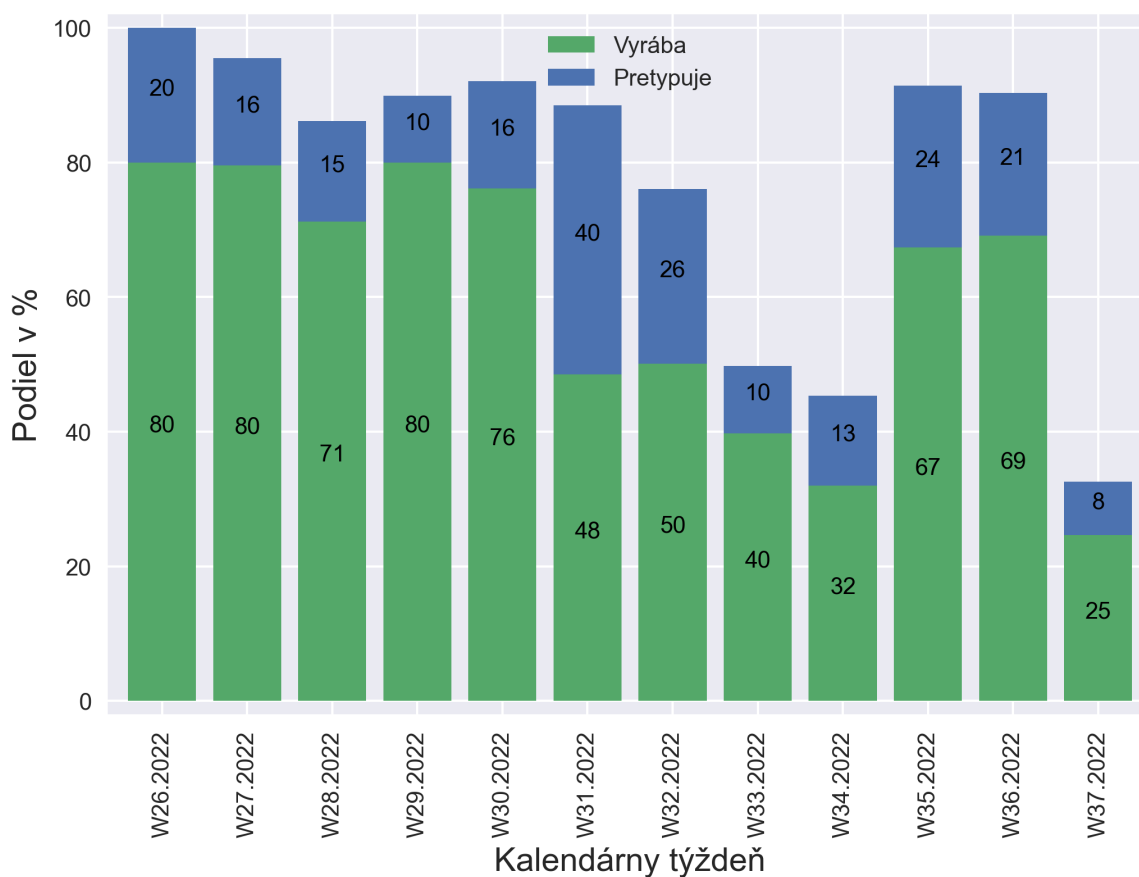
Zdroj: vlastné spracovanie

Tabuľka 19: Kapacitná analýza linky EQ-85025921

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	0,00	444,00	210,06	0,00	0,00	131,88	0,06	0,00	408,00	132,00	288,00	0,00
064886085-2061-43	594,39	414,61	0,00	0,00	365,00	0,00	365,00	0,00	0,00	0,00	365,00	0,00
000073229-0063-17	605,55	0,00	0,00	0,00	0,00	0,00	155,10	0,00	0,00	0,00	0,00	0,00
000199095-0063-18	0,00	0,00	651,80	0,00	0,00	0,00	620,70	137,00	0,00	0,00	0,00	0,00
000075353-6063-43	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000239232-6063-13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	446,74	0,00	0,00
000038580-6063-16	0,00	0,00	0,00	0,00	0,00	399,00	0,00	609,00	0,00	0,00	0,00	489,00
000071986-6063-17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	425,00	0,00
000242462-6063-13	0,00	574,30	0,00	0,00	1016,70	0,00	0,00	0,00	0,00	792,35	0,00	0,00
000040606-6063-45	0,00	0,00	0,00	1349,17	0,00	0,00	0,00	0,00	0,00	0,00	457,44	0,00
Využitá kapacita	1199,94	1432,91	861,86	1349,17	1381,70	530,88	1140,86	746,00	408,00	1371,09	1535,44	489,00
Disponibilná kapacita	1200,00	1500,00	1000,00	1500,00	1500,00	600,00	1500,00	1500,00	900,00	1500,00	1700,00	1500,00
Zaťaženie stroja (%)	100,00	95,53	86,19	89,94	92,11	88,48	76,06	49,73	45,33	91,41	90,32	32,60
Z toho pretypovanie	240,00	240,00	150,00	150,00	240,00	240,00	390,00	150,00	120,00	360,00	360,00	120,00
Z toho výroba	959,94	1192,91	711,86	1199,17	1141,70	290,88	750,86	596,00	288,00	1011,09	1175,44	369,00
Pomer pretypovania (%)	20,00	16,75	17,40	11,12	17,37	45,21	34,18	20,11	29,41	26,26	23,45	24,54

Zdroj: vlastné spracovanie

Graf 7: Percentuálne využitie linky EQ-85025921



Zdroj: vlastné spracovanie

Rozloženie nákladov výrobného plánu získaného genetickým algoritmom pre obe montážne linky a pre každé obdobie zvlášť uvádzame v tabuľke č. 20.

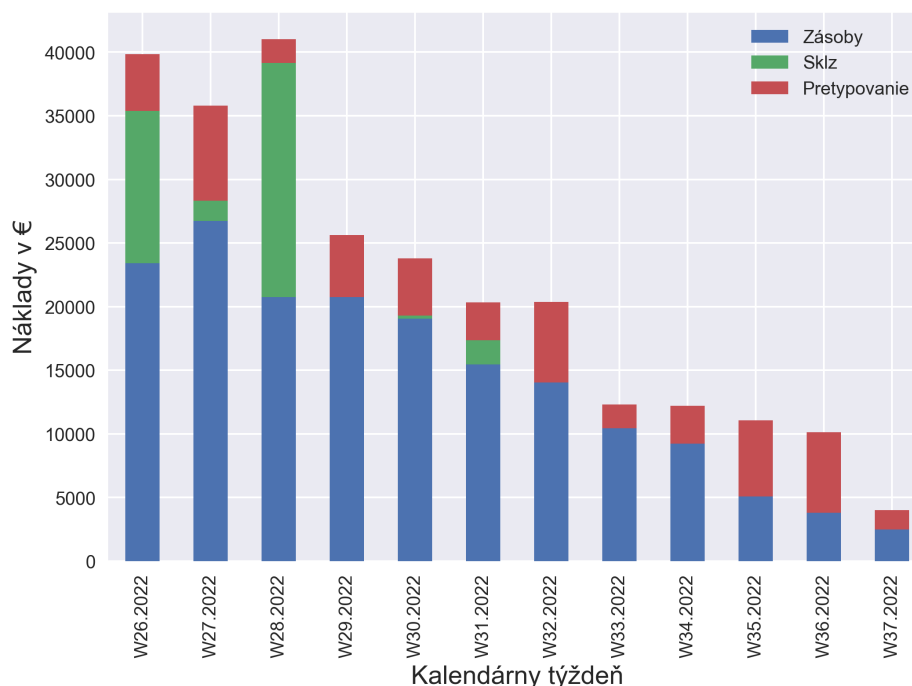
Tabuľka 20: Celkové náklady

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022	Σ
Celkom zásoby €	23420,69	26743,22	20759,48	20761,44	19064,97	15467,38	14039,08	10444,10	9221,52	5080,76	3792,68	2504,60	171299,92
Celkom sklz €	11956,80	1580,80	18403,00	0,00	238,80	1885,20	2,20	0,40	0,40	0,40	0,40	0,00	34068,40
Celkom pretypovanie €	4477,47	7462,44	1865,61	4850,59	4477,47	2984,98	6343,08	1865,61	2984,98	5969,96	6343,08	1492,49	51117,75
Spolu €	39854,96	35786,47	41028,09	25612,03	23781,23	20337,56	20384,35	12310,11	12206,90	11051,12	10136,16	3997,09	256486,07

Zdroj: vlastné spracovanie

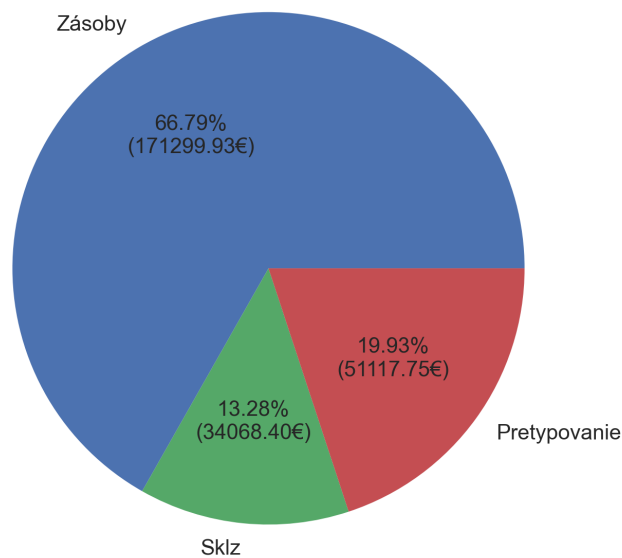
Hodnoty nákladových metrik pre každé obdobie zvlášť a pre obe linky spolu vizualizujeme na grafe č. 8. Rovnako ako pri exaktnom prístupe riešenia môžeme konštatovať, že počiatočný stav zásob a sklzu je privysoký. Môžeme tak vidieť, že výrobný plán získaný pomocou genetického algoritmu sa rovnako ako optimálny výrobný plán usiluje o postupné odbúranie týchto nákladov počas celého plánovacieho horizontu.

Graf 8: Rozloženie nákladov



Zdroj: vlastné spracovanie

Graf 9: Celkové rozloženie nákladov



Zdroj: vlastné spracovanie

Na grafe č. 9 prezentujeme agregáciu nákladových metrík výrobného plánu získaného genetickým algoritmom pre obe montážne linky za všetky obdobia spolu. Tento pohľad na nákladové metriky nám umožňuje zistiť, akou mierou sa podieľajú na celkových nákladoch.

Detailnejší pohľad na jednotlivé nákladové metriky uvádzame v nákladovej analýze v tabuľke č. 21. Prezentujeme tu, akou mierou jednotlivé výrobky za každé obdobie prispievajú k hodnote účelovej funkcie ako v hodnotovom vyjadrení, tak aj v naturálnych jednotkách.

Tabuľka 21: Nákladová analýza

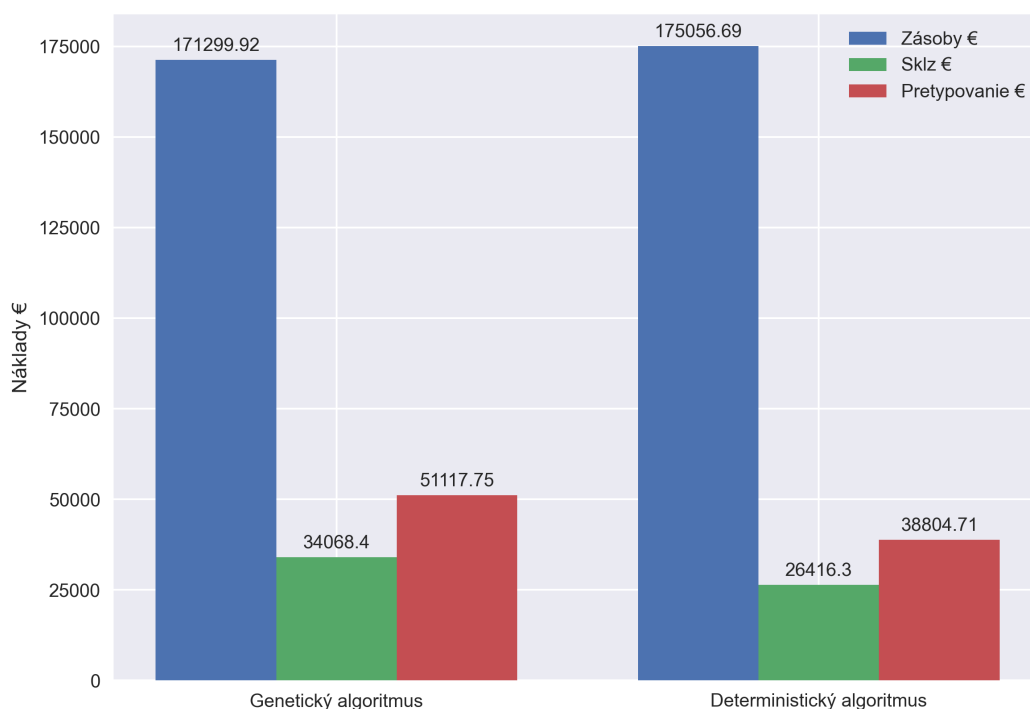
	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
053345657-3062-43	Zásoby (ks)	0	0	1	1	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,87	0,87	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	5400	0	0	0	199	1	0	0	0	0	0
	Sklz (€)	6480,00	0,00	0,00	0,00	238,80	1,20	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	1492,49	0,00	0,00	0,00	1492,49	0,00	0,00	1492,49	1492,49	0,00
064886085-2061-43	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	2423	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	2423,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	1492,49	0,00	0,00	0,00	1492,49	0,00	1492,49	0,00	0,00	0,00	1492,49
000073229-0063-17	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	1492,49	0,00	0,00	1492,49	0,00	0,00	1492,49	0,00	0,00	0,00	0,00
000199095-0063-18	Zásoby (ks)	0	0	2	2	2	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	2,61	2,61	2,61	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	3952	3952	0	0	0	4710	3	1	1	1	0
	Sklz (€)	1580,80	1580,80	0,00	0,00	0,00	1884,00	1,20	0,40	0,40	0,40	0,00
	Pretypovanie (€)	0,00	0,00	1865,61	0,00	0,00	0,00	1865,61	0,00	0,00	0,00	1865,61
000073533-6063-43	Zásoby (ks)	20200	20200	17900	17900	16100	12500	12500	8900	8900	7100	5300
	Zásoby (€)	14455,12	14455,12	12809,24	12809,24	11521,16	8945,00	8945,00	6368,84	6368,84	5080,76	3792,68
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
000239232-6063-13	Zásoby (ks)	13640	12400	12090	12090	11470	9920	7750	6200	4340	0	0
	Zásoby (€)	8965,57	8150,52	7946,76	7946,76	7539,23	6520,42	5094,08	4075,26	2852,68	0,00	0,00
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1492,49	0,00
000038580-6063-16	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	2146	0	4400	0	0	0	0	0	0	0	0
	Sklz (€)	1073,00	0,00	2200,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	1492,49	0,00	1492,49	0,00	1492,49	0,00	1865,61	0,00	1492,49	0,00
000071986-6063-17	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	500	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	400,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	1492,49	0,00	0,00	1492,49	0,00	0,00	0,00	1492,49	0,00	1492,49
000242462-6063-13	Zásoby (ks)	0	0	0	0	0	0	0	0	0	0	0
	Zásoby (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	0	0	0	0	0	0	0	0	0	0	0
	Sklz (€)	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	1492,49	1492,49	0,00	0,00	1492,49	0,00	0,00	0,00	0,00	1492,49	0,00
000040606-6063-45	Zásoby (ks)	0	4207	0	2	2	2	0	0	0	0	0
	Zásoby (€)	0,00	4137,58	0,00	1,97	1,97	1,97	0,00	0,00	0,00	0,00	0,00
	Sklz (ks)	0	0	16203	0	0	0	1	0	0	0	0
	Sklz (€)	0,00	0,00	16203,00	0,00	0,00	0,00	1,00	0,00	0,00	0,00	0,00
	Pretypovanie (€)	0,00	1492,49	0,00	1865,61	0,00	0,00	1492,49	0,00	0,00	0,00	1492,49

Zdroj: vlastné spracovanie

4.6.3 Porovnanie výsledkov genetického a exaktného algoritmu

Optimalizačné kritéria pre nami skúmaný problém sú jednotlivé nákladové metriky. Nezáleží na metóde, akou je úloha optimalizovaná, stále je hlavným zámerom minimalizovať agregáciu nákladových metrík. Z tohto dôvodu sú tieto ukazovatele taktiež najkredibilnejšie pri porovnávaní efektívnosti výrobných plánov. Na grafe č. 10 uvádzame porovnanie nákladových metrík pre výrobné plány získané genetickým algoritmom a exaktným algoritmom. Keďže výrobný plán získaný exaktným spôsobom je optimálny, hlavnou úlohou tejto analýzy je zhodnotiť, akou mierou sa efektivita výrobného plánu získaného genetickým algoritmom líši od optimálneho výrobného plánu. Ako možno na grafe vidieť, hodnoty nákladových metrík výrobného plánu získaného genetickým algoritmom sa približujú hodnotám nákladových metrík optimálneho výrobného plánu. Genetický algoritmus dokázal nájsť riešenie generujúce nižšie náklady na skladovanie produkcie ako exaktný prístup. Na druhej strane pri hodnotení nákladov na sklz a prestavbu montážnych liniek je riešenie získané genetickým algoritmom menej efektívne. Pri agregácii nákladových metrík zistíme, že rozdiel medzi nákladmi, ktoré generuje optimálny výrobný plán a nákladmi, ktoré generuje výrobný plán získaný genetickým algoritmom, je 16 208,37 €, čo tvorí približne 7 %-né zhoršenie v porovnaní s optimálnym výrobným plánom. Je potrebné poznamenať, že tento výsledok sme získali s explicitne stanovenými vstupnými parametrami. V prípade, ak by sme genetický algoritmus nechali riešiť dlhší čas, mohli by sme získať výrobný plán s nižšou odchýlkou od optimálneho výrobného plánu. Riešenie získané genetickým algoritmom je síce o 7 % horšie, no v prípade zložitých výrobných systémov, kedy nie je možné získať optimálne riešenie vzhľadom na výkonnosť výpočtových technológií, môže byť takáto odchýlka atraktívna. Pri zložitých výrobných systémoch nie je síce možné vyrábať optimálne, no je možné vyrábať efektívne tým, že spoločnosť volí najlepší možný prístup.

Graf 10: Porovnanie nákladových metrick

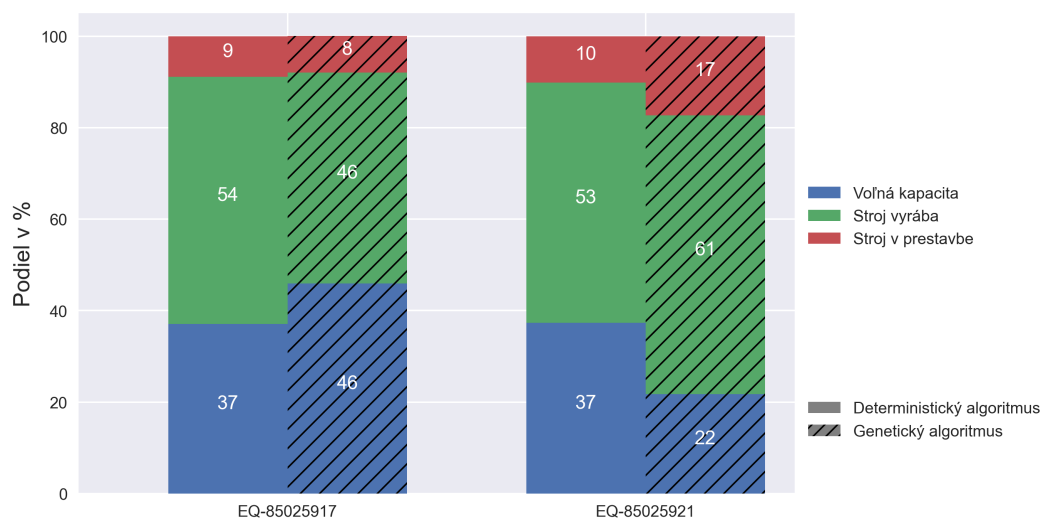


Zdroj: vlastné spracovanie

Doposiaľ sme hodnotili efektívnosť výrobných plánov iba vzhľadom na hodnoty nákladových metrick, ktoré generujú. Fakt, že výrobný plán je efektívnejší vzhľadom na náklady, ktoré generuje, ešte neznamená, že bude aj časovo efektívnejší. Na grafe č. 11 vizualizujeme percentuálne využitie oboch výrobných liniek pri optimálnom výrobnom pláne a výrobnom pláne získanom genetickým algoritmom. Pri porovnaní získaných výrobných plánov však platí, že optimálny výrobný plán je nielen nákladovo, ale aj časovo efektívnejší. Ako možno vidieť na grafe, optimálny výrobný plán je takmer perfektne vyvážený. Znamená to, že obe výrobné linky sú približne rovnako zaťažené (63 %) a taktiež rozloženie využitého času medzi výrobu a prestavbu je takmer identické. Rovnaké sa nedá povedať pri výrobných plánoch získaných genetickým algoritmom. Genetický algoritmus priradil viac produkcie stroju EQ-85025921 ako stroju EQ-85025917. Ak jednotlivé pomery pre obe linky spriemerujeme, zistíme, že optimálny výrobný plán dosahuje 63 %-né priemerné zaťaženie výrobných liniek, pričom výrobný plán získaný genetickým algoritmom dosahuje 66 %-né priemerné zaťaženie výrobných liniek. Znamená to, že optimálny výrobný plán dosahuje pri nižšom zaťažení výrobných liniek nižšie náklady. Ak sa pozrieme bližšie na priemerné rozloženie zaťaženia montážnych liniek, zistíme, že pri oboch výrobných plánoch stroje v priemere vyrábajú 53,5 % z disponibilného času.

Výrobné plány sa líšia iba v priemernom percentuálnom pretypovaní, kedy pri optimálnom výrobnom pláne montážne linky strávia v priemere 9,5 % z disponibilného času v prestavbe a pri výrobnom pláne získanom genetickým algoritmom tento čas tvorí priemerne 12,5 % z celkového disponibilného času. Tento jav prisudzujeme práve nedostatočnému rozloženiu produkcie medzi obe montážne linky v nami vytvorenom genetickom algoritme. Evidujeme tu priestor pre zlepšenie algoritmu v schopnostiach mutácie strojov.

Graf 11: Porovnanie percentuálneho využitia strojov



Zdroj: vlastné spracovanie

Z uvedeného možno konštatovať, že sa nám podarilo vyvinúť efektívny nástroj na optimalizáciu výrobných dávok a plánovania a možno ho tak aplikovať aj na výrobné systémy, ktoré v súčasnej dobe nie sú riešiteľné exaktným prístupom vzhľadom na výkonnosť súčasnej výpočtovej technológie.

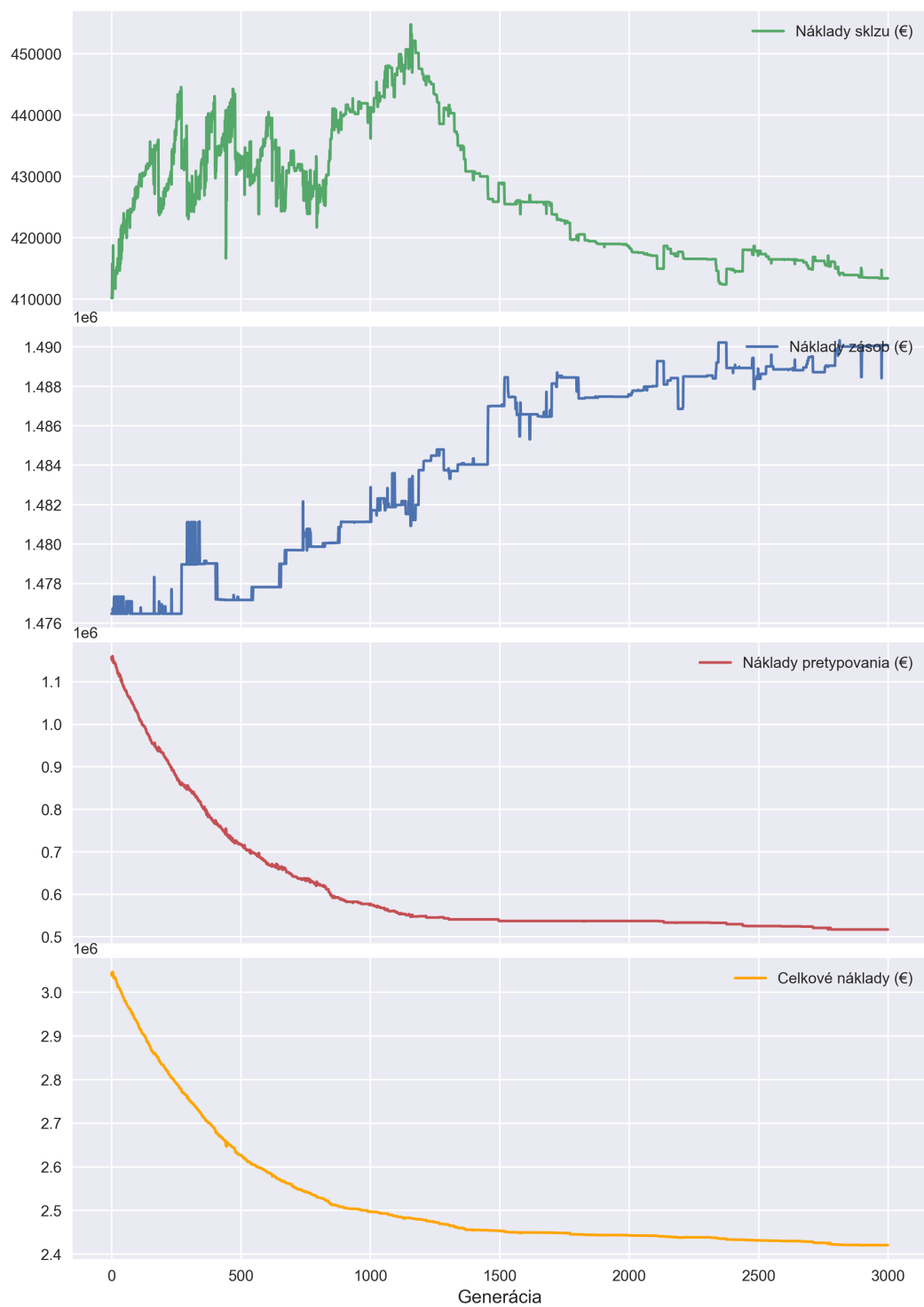
4.6.4 Aplikácia genetického algoritmu na kompletnom súbore údajov

Po dôkladnom otestovaní vytvoreného genetického algoritmu na redukovanom súbore údajov, sme algoritmus aplikovali na kompletnom súbore údajov. Ako vstupné parametre sme stanovili veľkosť populácie na 500 jedincov, za počet generácií sme zvolili 3 000 generácií, pravdepodobnosť mutácie sme stanovili na hodnotu 0,4 a ako maximálnu hranicu nezlepšenia sme zvolili 50 generácií. Najlepšie riešenie algoritmus našiel v 2 975. generácii s hodnotou účelovej funkcie rovnej 2 420 475,77 €.

Hodnoty nákladových metrick v priebehu riešenia uvádzame na grafe č. 12. Na grafe možno vidieť postupnú konvergenciu k určitému riešeniu. Nakoľko úloha takých rozmerov nie je exaktne riešiteľná, nevieme určiť, či ide o globálny alebo lokálny extrém. Kvalitu riešenia môžeme hodnotiť iba na základe porovnania riešení redukovanej úlohy. Ako možno na grafe vidieť, najväčšiu úsporu nákladov dokázal algoritmus získať odbúraním nákladov na pretypovanie. Na druhej strane, náklady spôsobené skladovaním produkcie sa v priebehu celého riešenia uberali stúpajúcim trendom. Tento jav prisudzujeme schopnosti genetického algoritmu efektívne zoskupovať malé výrobné dávky. Náklady na sklz spočiatku prudko rástli, no nakoniec sa dostali na úroveň inicializačného riešenia.

Z dôvodu veľkých rozmerov úlohy riešenej na kompletnom súbore údajov, budeme výsledky riešenia uvádzať iba v agregovaných veličinách za všetky výrobky spolu. V tabuľkách č. 22 a 23 uvádzame kapacitnú analýzu výrobných plánov pre obe montážne linky. Tu by sme chceli podotknúť nízke zaťaženie oboch výrobných liniek a taktiež pomerne vysoký pomer pretypovania, vyskytujúci sa vo viacerých obdobiach pri oboch montážnych linkách. Tento fakt môže mať za následok zhoršenie ekonomickej situácie, nakoľko sa produkcia výrobkov pre zákazníka vplyvom častej prestavby stroja predražuje. Tento problém je však riešiteľný väčším zaťažením výrobného systému (navýšením požiadaviek zákazníkov). Na grafoch č. 13 a 14 graficky vizualizujeme percentuálne využitie výrobných liniek, vzhľadom na ich celkový disponibilný čas.

Graf 12: Priebek riešenia kompletného problému genetickým algoritmom



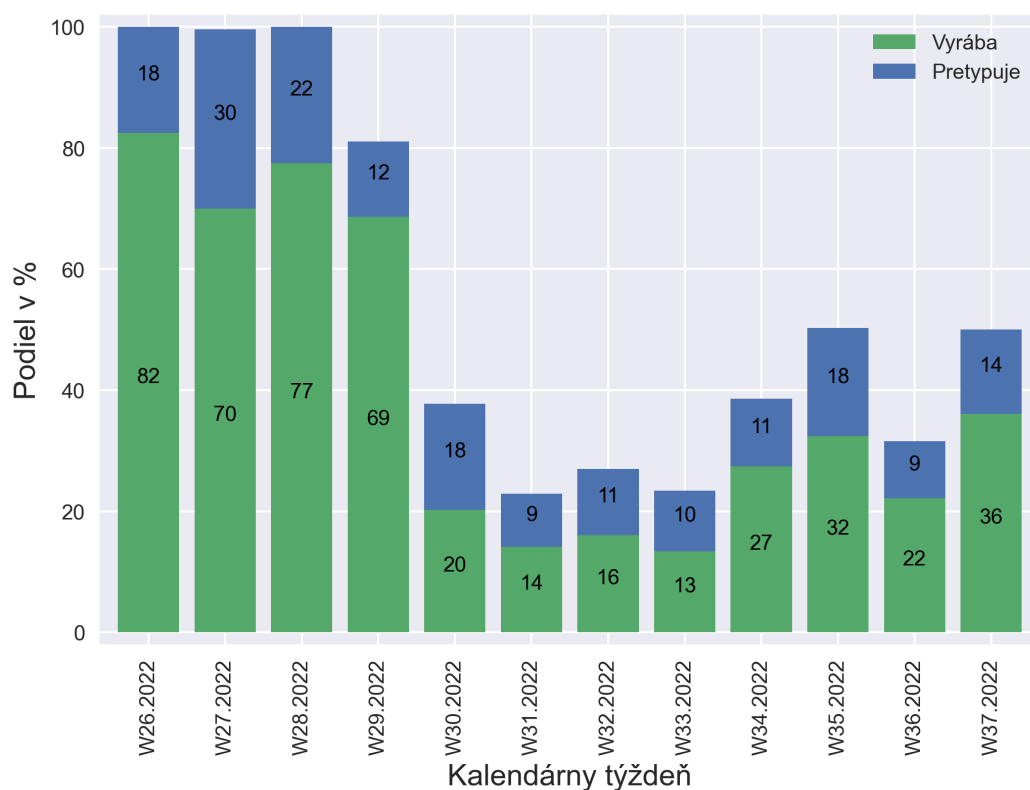
Zdroj: vlastné spracovanie

Tabuľka 22: Kapacitná analýza linky EQ-85025917

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
Využitá kapacita	7199,59	5738,36	7199,73	5449,44	2716,24	1647,90	1944,05	1681,73	2779,90	3375,71	2270,55	3120,60
Disponibilná kapacita	7200,00	5760,00	7200,00	6720,00	7200,00	7200,00	7200,00	7200,00	7200,00	6720,00	7200,00	6240,00
Zaťaženosť stroja (%)	99,99	99,62	100,00	81,09	37,73	22,89	27,00	23,36	38,61	50,23	31,54	50,01
Z toho pretypovanie	1260,00	1710,00	1620,00	840,00	1260,00	630,00	790,00	720,00	810,00	1200,00	680,00	870,00
Z toho výroba	5939,60	4028,36	5579,73	4609,44	1456,24	1017,90	1154,05	961,73	1969,90	2175,71	1590,55	2250,60
Pomer pretypovania (%)	17,50	29,80	22,50	15,41	46,39	38,23	40,64	42,81	29,14	35,55	29,95	27,88

Zdroj: vlastné spracovanie

Graf 13: Percentuálne využitie linky EQ-85025917



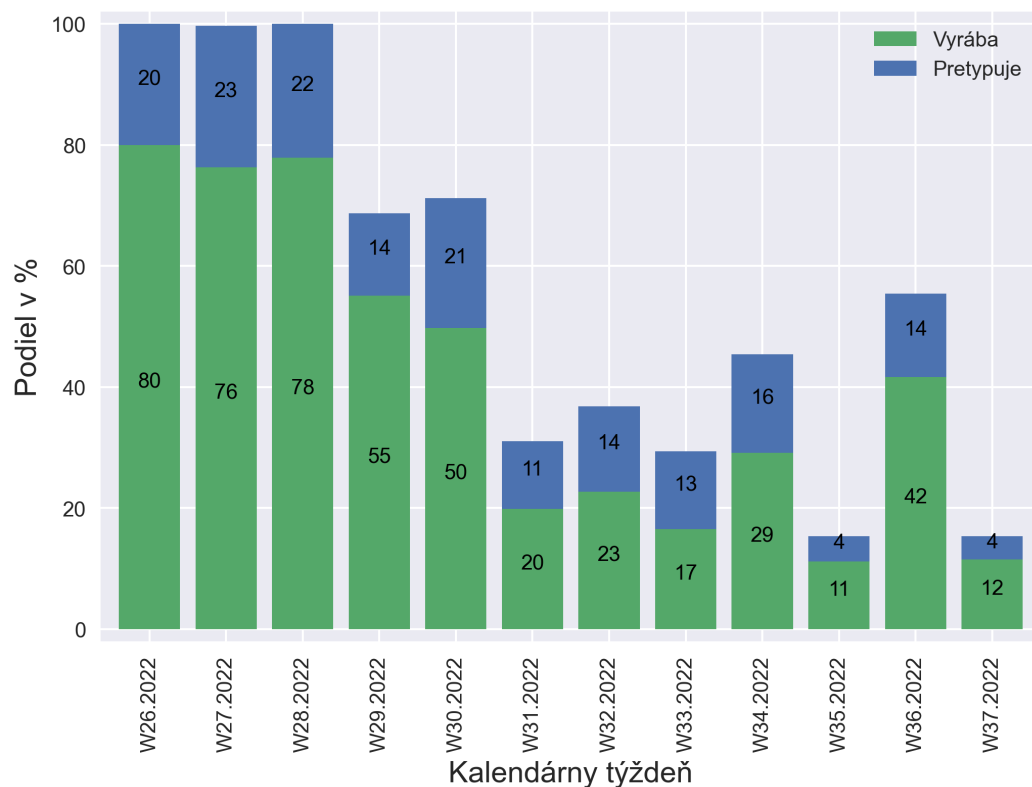
Zdroj: vlastné spracovanie

Tabuľka 23: Kapacitná analýza linky EQ-85025921

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022
Využitá kapacita	7199,47	7174,76	7198,89	4946,79	4784,59	2238,82	2652,91	1693,40	3270,72	1108,85	3988,84	960,00
Disponibilná kapacita	7200,00	7200,00	7200,00	7200,00	6720,00	7200,00	7200,00	5760,00	7200,00	7200,00	7200,00	6240,00
Zaťaženosť stroja (%)	99,99	99,65	99,98	68,71	71,20	31,09	36,85	29,40	45,43	15,40	55,40	15,38
Z toho pretypovanie	1440,00	1680,00	1590,00	980,00	1440,00	810,00	1020,00	740,00	1170,00	300,00	990,00	240,00
Z toho výroba	5759,47	5494,76	5608,89	3966,79	3344,59	1428,82	1632,91	953,40	2100,72	808,85	2998,84	720,00
Pomer pretypovania (%)	20,00	23,42	22,09	19,81	30,10	36,18	38,45	43,70	35,77	27,06	24,82	25,00

Zdroj: vlastné spracovanie

Graf 14: Percentuálne využitie linky EQ-85025921



Zdroj: vlastné spracovanie

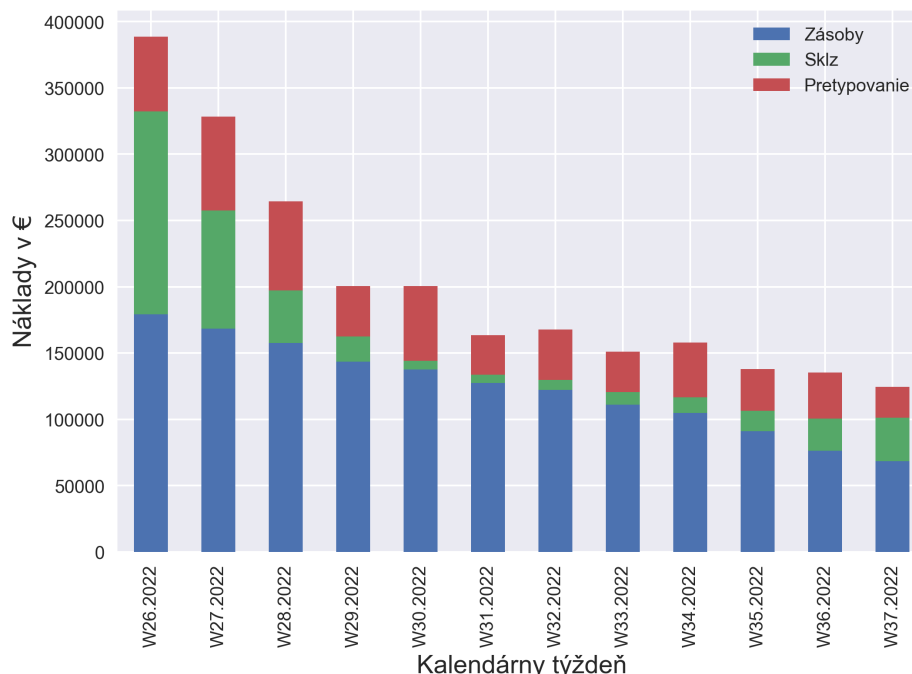
Tabuľka 24: Celkové náklady

	W26.2022	W27.2022	W28.2022	W29.2022	W30.2022	W31.2022	W32.2022	W33.2022	W34.2022	W35.2022	W36.2022	W37.2022	Σ
Celkom zásoby €	179207,48	168489,08	157623,34	143433,80	137779,41	127556,70	122097,28	111075,41	104876,38	91200,91	76512,32	68558,21	1488410,33
Celkom sklz €	153205,43	89013,67	39713,41	19145,13	6304,08	5993,22	7828,64	9609,42	11699,67	15410,38	24028,70	32817,64	414769,39
Celkom pretypovanie €	56341,24	70739,56	66983,47	37978,17	56341,24	30048,66	37769,50	30466,00	41316,91	31300,69	34848,10	23162,51	517296,05
Spolu €	388754,15	328242,31	264320,22	200557,10	200424,73	163598,58	167695,42	151150,83	157892,96	137911,98	135389,12	124538,36	2420475,77

Zdroj: vlastné spracovanie

V tabuľke č. 24 uvádzame rozloženie celkových nákladov pre riešenie kompletnej úlohy medzi jednotlivé nákladové metriky pre každé obdobie zvlášť. Na grafe č. 15 tieto nákladové metriky graficky vizualizujeme. Na grafe si možno všimnúť vysoké počiatočné náklady na skladovanie produkcie a taktiež vysoké náklady spôsobené sklzom. Výrobný plán získaný genetickým algoritmom tieto náklady v priebehu plánovacieho horizontu efektívne odbúrava. Usiluje sa minúť nadbytočne vytvorené zásoby a znížiť počiatočný sklz produkcie. Tieto poznatky sú úzko späté s kapacitnou analýzou. Možno si všimnúť, že prvé tri obdobia, kedy sú náklady na sklz vysoké, sú obe výrobné linky využité naplno. Je to spôsobené najmä vysokým sklzom, ktorý sa algoritmus snaží odbúrať v čo najkratšom časovom horizonte.

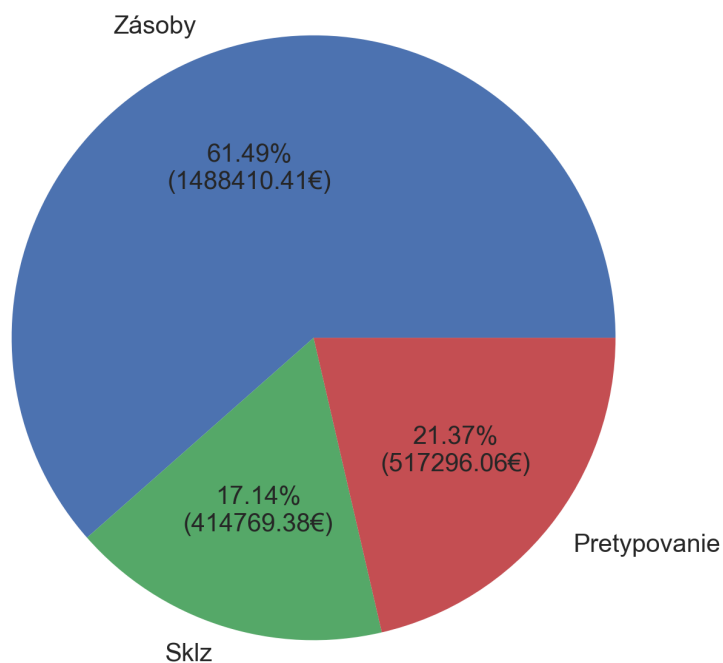
Graf 15: Rozloženie nákladov



Zdroj: vlastné spracovanie

Na grafe č. 16 uvádzame rozloženie celkových nákladov medzi jednotlivé nákladové metriky spolu za všetky obdobia.

Graf 16: Celkové rozloženie nákladov



Zdroj: vlastné spracovanie

4.7 Diskusia

V predkladanej práci sme sa zaoberali iba statickou aplikáciou nástrojov na optimalizáciu výrobných dávok a plánovania. Vyvinuli sme genetický algoritmus, ktorý efektívnym spôsobom optimalizuje výrobné dávky a plány ako také. Avšak, bez správnej a efektívnej integrácie takýchto nástrojov do reálnych výrobných procesov sú takéto nástroje zbytočné. Keďže požiadavky zákazníkov sa v čase menia, je potrebné, aby sa podnik zaoberal ich dynamickou integráciou. Dynamická integrácia sofistikovaných nástrojov optimalizácie výrobných plánov umožňuje plánovacie procesy automatizovať.

Jednou z možností dynamickej aplikácie takých nástrojov je tzv. prístup rolovaného plánovacieho horizontu. Ide o spôsob, kedy pevne stanovený plánovací horizont sa postupom času posúva vpred. Napríklad, každý týždeň sa plánovací horizont posunie o týždeň vpred. Uvedený prístup umožňuje, aby genetický algoritmus bežal stále na pozadí. Akonáhle sa posunie plánovací horizont o obdobie vpred, aktualizujú sa požiadavky zákazníkov na ďalší týždeň. Týmto krokom sa aktuálny výrobný plán genetického algoritmu dostáva

mimo efektívneho riešenia a algoritmus začne opäť riešenie zlepšovať. Pri tomto prístupe je však potrebné pevne stanoviť obdobie, kedy už výrobný plán nemôže byť menený vzhľadom na čas potrebný na obstaranie komponentov a čas produkcie výrobkov.

Integráciou nástrojov optimalizácie výrobných dávok a plánovania do reálnych procesov a ich kontinuálnym a automatizovaným využitím môže podnik získať veľkú konkurenčnú výhodu. Automatizované plánovanie umožňuje podniku eliminovať chyby spôsobené personálom a súčasne znížiť podniku personálne náklady. Na druhej strane s integráciou automatizovaného plánovania prichádzajú aj prekážky, ktoré musí podnik riešiť, a to najmä v podobe ich integrácie ako takej.

Záver

Optimalizácia výrobných dávok a plánovania je už mnoho rokov predmetom veľkého záujmu v oblasti operačného výskumu. Cieľom tejto optimalizácie je zvýšiť efektívnosť a ziskovosť podnikov, znížením výrobných nákladov a zvýšením efektívnosti využitia disponibilných kapacít. Pri optimalizácii výrobných dávok a plánovania je však potrebné zohľadniť vplyv dopytových a ponukových šokov na ekonomiku. Dopytové šoky, ako napríklad zmeny v preferenciách spotrebiteľov alebo hospodársky pokles, môžu viesť k výkyvom v dopyte po tovaroch a službách. Ponukové šoky, ako napríklad prírodné katastrofy alebo narušenie dodávateľských reťazcov, môžu viesť k nedostatku alebo prebytku zásob. Tieto šoky môžu mať výrazný vplyv na ziskovosť a konkurencieschopnosť podnikov a vyžadujú si primerané úpravy výrobných koncepcií. V posledných rokoch sme sa presvedčili, že tak ako koncepcia štíhlej výroby, tak ani koncepcia hromadnej výroby nie je efektívna v prípade identifikácie optimálnych výrobných dávok. V diplomovej práci sme sa zamerali na určitý kompromis týchto dvoch koncepcií.

Hlavným cieľom tejto práce bolo preto vytvoriť nástroj schopný identifikácie takého výrobného plánu, ktorý minimalizuje agregáciu nákladov na skladovanie produkcie, nákladov na prestavbu stroja a taktiež nákladov spojených s oneskoreným dodaním produkcie zákazníčkovi. V diplomovej práci sme preskúmali viaceré modely matematického programovania určené na optimalizáciu výrobných dávok a plánovania a diskutovali sme o ich výhodách a obmedzeniach. Modifikáciou prezentovaných modelov matematického programovania sme získali model, ktorý spĺňa všetky kritériá a vystihuje špecifiká nami skúmaného výrobného procesu. Avšak z dôvodu veľkej komplexnosti skúmaného výrobného procesu sme nedokázali model riešiť exaktným spôsobom. Pristúpili sme preto k alternatívnym metódam jeho riešenia. Na riešenie formulovaného optimalizačného modelu sme využili metaheuristickú metódu optimalizácie genetický algoritmus. Algoritmus sme programovali v jazyku python bez použitia špecifických knižníc určených na konštrukciu genetického algoritmu. Aby sme boli schopní zhodnotiť efektívnosť nami vytvoreného genetického algoritmu, vytvorili sme redukovaný súbor údajov, na ktorom sme vykonali optimalizáciu ako exaktným, tak aj nami vytvoreným genetickým algoritmom. Získané výsledky sme následne analyzovali.

Zistili sme, že riešenie získané genetickým algoritmom pri explicitne stanovených parametroch sa v agregovaných nákladoch odchyľuje iba o 7 % od riešenia získaného exaktným spôsobom. V prípade hodnotenia časovej efektívnosti výrobných plánov sme zistili, že produkcia v optimálnom výrobnom pláne bola takmer identicky rozložená medzi obe výrobné linky, pričom genetický algoritmus uprednostňoval priradenie produkcie jednej z výrobných liniek. Po hlbšej analýze priemerného zaťaženia montážnych liniek sme však prišli na to, že pri oboch výrobných plánoch stroje v priemere vyrábajú 53,5 % z disponibilného času. Výrobné plány sa líšili iba v priemernom percentuálnom pretypovaní. Evidujeme tu priestor na zlepšenie algoritmu v operátore mutácie strojov, nakoľko lepším vyvážením produkcie medzi oba stroje možno získať výrobný plán s nižším priemerným zaťažením strojov. Nakoniec sme genetický algoritmus aplikovali na kompletný súbor údajov, tzn. reálny výrobný proces.

Dospeli sme k záveru, že sa nám podarilo vyvinúť efektívny nástroj na optimalizáciu výrobných dávok a plánovania a možno ho tak aplikovať aj na výrobné systémy, ktoré v súčasnej dobe nie sú riešiteľné exaktným prístupom vzhľadom na výkonnosť súčasnej výpočtovej technológie.

Zoznam použitej literatúry

Knižné zdroje:

- [1] AKERS, Paul A. 2 Second Lean. United States of America : FastCap Press, 2019. 154 s. ISBN 978-0-984-66228-9.
- [2] ARORA, Sanjeev - BARAK, Boaz. COMPUTATIONAL COMPLEXITY : A Modern Approach. New York : Cambridge University Press, 2009. 579 s. ISBN 978-0-521-42426-4.
- [3] BREZINA, Ivan - IVANIČOVÁ, Zlatica - PEKÁR, Juraj. Operačná analýza. Bratislava : Jura Edition, 2007. 243 s. ISBN 978-80-8078-176-7.
- [4] BREZINA, Ivan - PEKÁR, Juraj. Úvod do operačného výskumu I. Bratislava : Letra Edu, 2018. 170 s. ISBN 978-80-89962-17-4.
- [5] BURTSEVA, Larysa et al. Production Planning and Scheduling for Lot Processing. Newcastle upon Tyne : Cambridge Scholars Publishing, 2022. 397 s. ISBN 978-1-5275-8502-7.
- [6] CHOI, Tsan-Ming. Handbook of EOQ Inventory Problems : Stochastic and Deterministic Models and Applications. New York : Springer Science + Business Media, Inc., 2014. 279 s. ISBN 978-1-4614-7638-2.
- [7] EL-GHAZALI, Talbi. METAHEURISTICS FROM DESIGN TO IMPLEMENTATION. New Jersey : John Wiley & Sons, Inc., 2009. 618 s. ISBN 978-0-470-27858-1.
- [8] LEE, Kwang Y. - EI-SHARKAWI, Mohamed A. MODERN HEURISTIC OPTIMIZATION TECHNIQUES : THEORY AND APPLICATIONS TO POWER SYSTEMS. New Jersey : John Wiley & Sons, Inc., 2008. 563 s. ISBN 978-0471-45711-4.
- [9] MIRJALILI, Seyedali. Evolutionary Algorithms and Neural Networks : Theory and Applications. Brisbane : Springer International Publishing AG, 2019. 159 s. ISBN 978-3-319-93024-4.

- [10] MITCHELL, Melanie. An Introduction to Genetic Algorithms. 5. vyd. Cambridge : Massachusetts Institute of Technology, 1999. 158 s. ISBN 0-262-13316-4.
- [11] POCHET, Yves - WOLSEY, Laurence A. Production Planning by Mixed Integer Programming. New York : Springer Science + Business Media, Inc., 2006. 481 s. ISBN 978-1-4419-2132-1.
- [12] SIPSER, Michael. Introduction to the Theory of Computation. 2. vyd. United States of Amerika : Thomson Learning, Inc., 2006. 431 s. ISBN 0-534-95097-3.
- [13] TAHA, Hamdy A. OPERATIONS RESEARCH : AN INTRODUCTION. 8. vyd. New Jersey : Pearson Education, Inc., 2007. 813 s. ISBN 0-13-188923-0.
- [14] WILIAMS, Paul H. Model Building in Mathematical Programming. 5. vyd. United Kingdom : John Wiley & Sons, Ltd., 2013. 411 s. ISBN 978-1-118-44333-0.

Elektronické zdroje:

- [15] ALLENDER, Eric W. - LOUI, Michael C. - REGAN, Kenneth W. Complexity Theory [elektronický zdroj]. 2012. 35 s. Dostupné na: <https://people.cs.rutgers.edu/~allender/papers/ALR12.pdf>
- [16] HAASE, Knut. Capacitated Lot-Sizing with Sequence Dependent Setup Costs [elektronický zdroj]. 1994. 14 s. Dostupné na: https://www.econstor.eu/bitstream/10419/155415/1/manuskript_340.pdf
- [17] JAMES, Ross J. W. - ALMADA-LOBO, Bernardo. Single and Parallel Machine Capacitated Lotsizing and Scheduling : New Iterative MIP-based Neighborhood Search Heuristics [elektronický zdroj]. 2011. 10 s. Dostupné na: <https://www.sciencedirect.com/science/article/pii/S0305054811000463>
- [18] KIMMS, Alf - DREXL, Andreas. Lot Sizing and Scheduling: Survey and Extensions [elektronický zdroj]. 1996. 28 s. Dostupné na: https://www.econstor.eu/bitstream/10419/175395/1/manuskript_421.pdf
- [19] KLIER, Thomas H. How lean manufacturing changes the way we understand the manufacturing sector [elektronický zdroj]. 1993. 9 s. Dostupné na:

<https://www.chicagofed.org/-/media/publications/economic-perspectives/1993/ep-may-jun1993-part1-klier-pdf>

- [20] MELTON, T. Benefits of lean manufacturing : What lean thinking has to offer the process industries [elektronický zdroj]. 2005. 9 s. Dostupné na: <http://mimesolutions.com/PDFs/WEB\%20Trish\%20Melton\%20Lean\%20Manufacturing\%20July\%202005.pdf>
- [21] MIČIETOVÁ, Mária. LEAN PRODUCTION, LEAN VS. MASS PRODUCTION, TPM AS A TOOL OF LEAN PRODUCTION [elektronický zdroj]. 2011. 7 s. Dostupné na: <https://pernerscontacts.upce.cz/index.php/perner/article/view/918/752>
- [22] ROGERS, Jack. A Computational Approach to the Economic Lot Scheduling Problem [elektronický zdroj]. 1958. 28 s. Dostupné na: <https://doi.org/10.1287/mnsc.4.3.264>
- [23] WAGNER, Harvey M. - WHITIN, Thomson M. Dynamic Version of the Economic Lot Size Model [elektronický zdroj]. 1958. 9 s. Dostupné na: <http://www.jstor.org/stable/2626974>
- [24] <https://www.schaeffler.com/en/>
- [25] <https://www.gurobi.com/documentation/>

Príloha 3: Kompletný súbor údajov

SKLZ	SKLAD	W26.20	W27.20	W28.20	W29.20	W30.20	W31.20	W32.20	W33.20	W34.20	W35.20	W36.20	W37.20	EQ-850	EQ-850	KHK/ks	Tz/min	Nakl_skAlt_nakl	
000040606-0063-01		20	0	0	0	5	0	0	0	0	0	0	0	1	1	0,9835	0,074	0,89 20,867	
000242403-6063-17		5	0	5	0	0	0	0	0	0	0	0	0	1	1	1,3817	0,064	0,22	
000617717-6063-15		0	0	0	0	0	5	0	0	0	0	0	480	1	1	0,9137	0,065	0,66	
000668877-6063-10	1750	0	0	0	0	5	0	0	0	0	0	0	0	0	1	1,09	0,068	1,22	
000199095-6063-62		27	0	0	0	0	0	8	0	0	0	0	13	1	1	1,3035	0,1	1,64	
000239232-6063-13		14880	1240	1240	310	0	620	1550	2170	1860	9610	0	0	1	1	0,6573	0,062	1,87	
000038580-6063-16		0	4000	0	4400	0	0	3100	0	5100	0	4000	0	4100	1	1	0,9303	0,09	1,81
000071986-6063-17	500	0	0	5000	0	0	5000	0	0	5000	0	5000	0	1	1	0,6299	0,061	1,47	
000242462-6063-13	17330	0	0	6490	0	0	12810	0	0	0	9605	0	0	1	1	0,9447	0,07	0,56	
000040606-6063-45		0	0	0	40680	0	0	0	5760	0	4560	0	0	1	1	0,9835	0,074	0,17	
	EQ-850	7200	5760	7200	6720	7200	7200	7200	7200	7200	6720	7200	6240	000038580-6063-16					
	EQ-850	7200	7200	7200	7200	7200	6720	7200	7200	7200	7200	7200	6240	000242462-6063-13					

Zdroj: vlastné spracovanie

Príloha 4: Kompletný súbor údajov - pretypovacia matica

000040606-0063-01	0	120	120	120	150	000242403-6063-17	120	120	120	150	000199095-6063-62	120	120	150	000239232-6063-13	120	120	120	150	000040606-6063-45	30
000242403-6063-17	120	0	120	120	150	000617717-6063-15	0	120	120	150	000199095-6063-62	120	120	150	000239232-6063-13	120	120	120	150	000040606-6063-45	30
000617717-6063-15	120	120	0	120	150	000668877-6063-10	120	120	120	150	000199095-6063-62	120	120	150	000239232-6063-13	120	120	120	150	000040606-6063-45	150
000668877-6063-10	120	120	120	0	150	000199095-6063-62	150	150	150	0	000199095-6063-62	150	150	150	000239232-6063-13	150	150	150	150	000040606-6063-45	150
000199095-6063-62	150	150	150	150	0	000199095-6063-62	150	150	150	0	000199095-6063-62	150	150	150	000239232-6063-13	150	150	150	150	000040606-6063-45	150
000239232-6063-13	120	120	120	120	150	000239232-6063-13	120	120	120	150	000239232-6063-13	120	120	120	000239232-6063-13	120	120	120	150	000040606-6063-45	0
000038580-6063-16	120	120	120	120	150	000038580-6063-16	120	120	120	150	000038580-6063-16	120	120	120	000038580-6063-16	120	120	120	150	000040606-6063-45	0
000071986-6063-17	120	120	120	120	150	000071986-6063-17	120	120	120	150	000071986-6063-17	120	120	120	000071986-6063-17	120	120	120	150	000040606-6063-45	0
000242462-6063-13	120	120	120	120	150	000242462-6063-13	120	120	120	150	000242462-6063-13	120	120	120	000242462-6063-13	120	120	120	150	000040606-6063-45	0
000040606-6063-45	30	120	120	120	150	000040606-6063-45	30	120	120	150	000040606-6063-45	30	120	120	000040606-6063-45	30	120	120	150	000040606-6063-45	0

Zdroj: vlastné spracovanie

Príloha 5: Exaktné riešenie úlohy zmiešaného celočíselného programovania

```
#----Importujeme kniznice---
import gurobipy as gp
from gurobipy import GRB
import pandas as pd
import numpy as np
import sys
sys.setrecursionlimit(10**6)

#___Nacitame si vstup v podobe vzoroveho exceloveho suboru___
# Vsetky atributy musia byt umiestnene na svojom mieste ! (od ich umiestnenia sa odvíja program)
data = pd.read_excel(r'C:/Users/michd/Documents/GA/vstupne_data.xlsx',sheet_name="DATA")
pretyp_matica = pd.read_excel(r'C:/Users/michd/Documents/GA/vstupne_data.xlsx',sheet_name="PRETYP")

# Sets
vyrobky = pretyp_matica.iloc[:,0 ].tolist()
obdobia = data.columns.tolist()[3:-6]
obdobia.append("WN")
stroje = data.iloc[-2:,2 ].tolist()

# Scalars
faktor = float(data.iloc[0, -1])
BIG_M = 1000000
N = len(vyrobky)
T = len(obdobia)-1

#Parameters
khk=data.set_index([data.iloc[:,0]]) .iloc[: -2, -4].to_dict()
cykl_casy=data.set_index([data.iloc[:,0]]) .iloc[: -2, -3].to_dict()
naklady_sklz = data.set_index([data.iloc[:,0]]) .iloc[: -2, -2].to_dict()
sklad = data.set_index([data.iloc[:,0]]) .iloc[: -2, 2].to_dict()
sklz = data.set_index([data.iloc[:,0]]) .iloc[: -2, 1].to_dict()
start= data.set_index([data.iloc[:,2 ]]) .iloc[: -2, :-6].to_dict()
#Tables
zastupitelnost = data.set_index([data.iloc[:,0]]) .iloc[: -2, -6:-4].transpose().stack().to_dict()
kapacita=data.set_index([data["SKLAD"]]) .iloc[: -2, 3:-6].stack().to_dict()
dopyt=data.set_index([data.iloc[:,0]]) .iloc[: -2, 3:-6].stack().to_dict()
pretyp_cas=pretyp_matica.set_index([pretyp_matica.iloc[:,0]]) .iloc[:, 1:].stack().to_dict()

#----Vytvorenie objektu model----
CLSP = gp.Model('clsp')

#----Deklaracia rozhodovacich premennych---
X = CLSP.addVars(((stroj, vyrobok, obdobie) for stroj in stroje for vyrobok in vyrobky for obdobie in obdobia
[0:-1] ),vtype=GRB.INTEGER, lb = 0, name="Produkcia")
T = CLSP.addVars(((stroj, vyr, vyrobok, obdobie) for stroj in stroje for vyr in vyrobky for vyrobok in vyrobky
for obdobie in obdobia[0:-1] ),vtype=GRB.BINARY, name="Pretyp")
I = CLSP.addVars(((vyrobok, obdobie) for vyrobok in vyrobky for obdobie in obdobia[0:-1] ), vtype=GRB.CONTINUOUS
, lb = 0.0, name="Zasoby" )
B = CLSP.addVars(((vyrobok, obdobie) for vyrobok in vyrobky for obdobie in obdobia[0:-1] ), vtype=GRB.CONTINUOUS
, lb = 0.0, name="Sklz" )
Y = CLSP.addVars(((stroj, vyrobok, obdobie) for stroj in stroje for vyrobok in vyrobky for obdobie in obdobia)
,vtype=GRB.BINARY, name="Prelom")
V = CLSP.addVars(((stroj, vyrobok, obdobie) for stroj in stroje for vyrobok in vyrobky for obdobie in obdobia
[0:-1] ),vtype=GRB.CONTINUOUS, name="Pomocna")

# Toto je bilancna rovnica v období 0
Bil_rovnica_0 = CLSP.addConstrs((sklad[vyrobok] - sklz[vyrobok] + gp.quicksum(X[stroj, vyrobok, obdobie[0]] for
stroj in stroje) - dopyt[vyrobok,obdobia[0]] == I[vyrobok,obdobia[0]] - B[vyrobok,obdobia[0]] for vyrobok in
vyrobky), name = "Bilancna Rovnica t0")
# Toto je ohranicenie ktore stanovuje rovnovahu na sklade
Bil_rovnica = CLSP.addConstrs((I[vyrobok, obdobia[obdobia.index(obdobie) -1]] - B[vyrobok, obdobia[obdobia.
index(obdobie) -1]] + gp.quicksum(X[stroj, vyrobok, obdobie] for stroj in stroje) - dopyt[vyrobok,obdobie] ==
I[vyrobok,obdobie] - B[vyrobok,obdobie] for vyrobok in vyrobky for obdobie in obdobia[1:-1]), name="Bilancna
Rovnica" )
# Kapacitne ohranicenie
Kapa_ohr = CLSP.addConstrs((gp.quicksum(cykl_casy[vyrobok] * X[stroj, vyrobok, obdobie] for vyrobok in vyrobky)
+ gp.quicksum( pretyp_cas[vyrobok, sub_vyrobok] * T[stroj,vyrobok,sub_vyrobok,obdobie] for vyrobok in vyrobky
for sub_vyrobok in vyrobky ) <= kapacita[stroj, obdobie] for stroj in stroje for obdobie in obdobia[0:-1]),
name = "Kapacitne ohranicenia")
# BIG M ohranicenie
Big_M_ohr = CLSP.addConstrs((X[stroj, vyrobok, obdobie] <= BIG_M * (gp.quicksum(T[stroj,sub_vyrobok,vyrobok,
obdobie] for sub_vyrobok in vyrobky) + Y[stroj,vyrobok,obdobie]) for stroj in stroje for vyrobok in vyrobky
for obdobie in obdobia[0:-1] ), name = "Velke M ohranicenie")
# Informacia pre prelyn obdobi
Prenos_ohr = CLSP.addConstrs((Y[stroj, vyrobok, obdobia[obdobia.index(obdobie) +1]] + gp.quicksum(T[stroj,
vyrobok,sub_vyrobok,obdobie] for sub_vyrobok in vyrobky) == Y[stroj,vyrobok,obdobie] + gp.quicksum(T[stroj,
sub_vyrobok,vyrobok,obdobie] for sub_vyrobok in vyrobky) for stroj in stroje for vyrobok in vyrobky for
obdobie in obdobia[0:-1]), name = "Informacia o prelyne" )
```

```

# Ohranicenie ktore zabezpeci ze sucet prenosov na stroji == 1
Sucet_prenosov = CLSP.addConstrs((gp.quicksum(Y[stroj, vyrobok, obdobie] for vyrobok in vyrobky) == 1 for stroj
    in stroje for obdobie in obdobia), name = "Sucet prenosov")
# Toto ohranicenie stanovuje pociatcne nastavenie strojov
Start = CLSP.addConstrs((Y[stroj, start[stroj], obdobia[0]] == 1 for stroj in start), name = "start")
# Toto ohranicenie elininuje podcykly
Cykl_ohr = CLSP.addConstrs((V[stroj, vyrobok, obdobie] + N*T[stroj, vyrobok, sub_vyrobok, obdobie] - (N-1) - N*
    Y[stroj, sub_vyrobok, obdobie] <= V[stroj, sub_vyrobok, obdobie] for stroj in stroje for vyrobok in vyrobky
    for sub_vyrobok in vyrobky for obdobie in obdobia[0:-1]), name = "Eliminacia cyklov")
# Toto ohranicenie zabezpecuje zastupitelnost
Zastupitelnost_ohr = CLSP.addConstrs((X[stroj, vyrobok, obdobie] <= BIG_M*zastupitelnost[stroj, vyrobok] for
    stroj in stroje for vyrobok in vyrobky for obdobie in obdobia[0:-1] ), name = "Zastupitelnost ohranicenie")

#___Definovanie ucelovej funkcie___
obj = gp.quicksum(faktor*pretyp_cas[vyrobok, sub_vyrobok]*T[stroj, vyrobok, sub_vyrobok, obdobie] for stroj in
    stroje for vyrobok in vyrobky for sub_vyrobok in vyrobky for obdobie in obdobia[0:-1]) + gp.quicksum(khk[
    vyrobok]*I[vyrobok, obdobie] for vyrobok in vyrobky for obdobie in obdobia[0:-1]) + gp.quicksum(naklady_sklz[
    vyrobok]*B[vyrobok, obdobie] for vyrobok in vyrobky for obdobie in obdobia[0:-1])
CLSP.setObjective(obj, GRB.MINIMIZE)
CLSP.optimize()

```

Zdroj: vlastné spracovanie

Príloha 6: Genetický algoritmus

```
import numpy as np
import pandas as pd
from random import randint
from random import uniform
import math
import matplotlib.pyplot as plt
plt.style.use('seaborn')
#####_Vstupne data_#####
#####

# Nacitame si vstup v podobe vzoroveho exceloveho suboru
# Vsetky atributy musia byt umiestnene na svojom mieste ! (od ich umiestnenia sa odvíja program)
data = pd.read_excel(r'C:/Users/michd/Documents/GA/input_data.xlsx', sheet_name="DATA")
pretyp_matica = pd.read_excel(r'C:/Users/michd/Documents/GA/input_data.xlsx', sheet_name="PRETYP")

minutovy_factor = float(data.iloc[0, -1])
casy_pretypovania_strojov = pretyp_matica.iloc[:,1:].to_numpy()
naklady_pretypovania_strojov = casy_pretypovania_strojov*minutovy_factor
vyrobky = pretyp_matica.iloc[:,0]
dopyt = data.iloc[:2,3:-6].to_numpy()
zastupitelnost_strojov = data.iloc[:2,15:17].to_numpy()
naklady_skladovania = data.iloc[:2,17].to_numpy()
cyklove_casy = data.iloc[:2,18].to_numpy()
naklady_sklzu = data.iloc[:2,19].to_numpy()
pociatocny_sklz = data.iloc[:2,1].to_numpy()
pociatocne_zasoby = data.iloc[:2,2].to_numpy()
kapacita = data.iloc[:2,:3:15].to_numpy()
posledne_nastavenie_strojov = [np.where(vyrobky == data.iloc[:2,-6].tolist()[0])[0][0], np.where(vyrobky ==
data.iloc[:2,-6].tolist()[1])[0][0]]

# Tato funkcia vrati hodnoty rozhodovacich premennych Xmijt
def pretyp (stroj,riesenie):
    index = np.where(riesenie[0] == stroj)
    poradie = np.array([index[0],riesenie[1][index],index[1]])
    if poradie.size != 0:
        index_poradia = np.lexsort((poradie[1],poradie[2]))
        vyroba = np.array([poradie[0][index_poradia],poradie[2][index_poradia]])
        z = np.insert(poradie[0][index_poradia][0:-1],0,posledne_nastavenie_strojov[stroj-1])
        return(np.transpose(np.array((z,vyroba[0],vyroba[1])))
    else:
        return(np.transpose(poradie))

# Tato funkcia vrati hodnoty rozhodovacich premennych pre zasoby a sklz
def vypocitaj_sklz (davky_lok):
    odchylky = dopyt - davky_lok
    odchylky = np.insert(odchylky,0,pociatocny_sklz-pociatocne_zasoby,1)
    sklz = np.cumsum(odchylky,axis=1)
    sklz[sklz<0] = 0
    return(sklz)

def vypocitaj_zasoby (davky_lok):
    odchylky = davky_lok-dopyt
    odchylky = np.insert(odchylky,0,pociatocne_zasoby-pociatocny_sklz,1)
    zasoby = np.cumsum(odchylky,axis=1)
    zasoby[zasoby<0] = 0
    return(np.array(zasoby))

# Ucelova funkcia
def HUF (riesenie):
    pretyp_index_s1 = pretyp(1,riesenie)
    pretyp_index_s2 = pretyp(2,riesenie)
    naklady_pretypovania = 0

    for x in pretyp_index_s1:
        naklady_pretypovania+=naklady_pretypovania_strojov[x[0],x[1]]
    for x in pretyp_index_s2:
        naklady_pretypovania+=naklady_pretypovania_strojov[x[0],x[1]]

    sklz = vypocitaj_sklz(riesenie[2])
    sklz_naklady = sum(naklady_sklzu @ sklz[:,1:])
    zasoby = vypocitaj_zasoby(riesenie[2])
    zasoby_naklady = sum(naklady_skladovania @ zasoby[:,1:])
    naklady_spolu = naklady_pretypovania + zasoby_naklady + sklz_naklady

    return(naklady_spolu)

# Funkcia ktora vracia zastupitelnost strojov
def zastupitelnost(vyrobok):
    pripustny_set = []
    for y in range(zastupitelnost_strojov.shape[1]):
        if zastupitelnost_strojov[vyrobok, y] != 0:
            pripustny_set.append(y+1)
    return(pripustny_set)
```

```

# Funkcia ktora vypocita cas potrebny na vyrobnu produkcie v case
def vypocitaj_potrebnu_kapacitu(stroj, obdobie, riesenie):
    poradie = pretyp(stroj, riesenie)
    casova_sпотреba = 0
    for x in poradie:
        if x[2] == obdobie:
            casova_sпотреba += casy_pretypovania_strojov[x[0],x[1]]+cyklove_casy[x[1]]*riesenie[2][x[1],x[2]]
        else:
            casova_sпотреba += 0
    return(casova_sпотреba)

# Funkcia opravi riesenie aby bolo pripustne vzhľadom na kapacitu
def oprav_riesenie(riesenie):
    vrstva_1 = np.copy(riesenie[0])
    vrstva_2 = np.copy(riesenie[1])
    vrstva_3 = np.copy(riesenie[2])
    for stroj in range(len(posledne_nastavenie_strojov)):
        for obdobie in range(dopyt.shape[1]):
            potrebna_kapacita = vypocitaj_potrebnu_kapacitu(stroj+1, obdobie, [vrstva_1, vrstva_2, vrstva_3])
            dispo_kapacita = kapacita[stroj,obdobie]
            while potrebna_kapacita > dispo_kapacita:
                nenulove_prvky = np.where(vrstva_1[:,obdobie] == stroj+1)[0]
                nahodny_vyrobok = nenulove_prvky[randint(0, len(nenulove_prvky)-1)]
                if potrebna_kapacita - dispo_kapacita >= cyklove_casy[nahodny_vyrobok]*vrstva_3[nahodny_vyrobok
,obdobie]:
                    pamat = vrstva_1[nahodny_vyrobok,obdobie]
                    vrstva_2[nahodny_vyrobok,obdobie] = 0
                    for vyrobok in range(vrstva_2.shape[0]):
                        if vrstva_2[vyrobok,obdobie] > pamat and vrstva_1[vyrobok,obdobie] == stroj+1:
                            vrstva_2[vyrobok,obdobie] -= 1
                    vrstva_1[nahodny_vyrobok,obdobie] = 0
                    vrstva_3[nahodny_vyrobok,obdobie] = 0
                    potrebna_kapacita = vypocitaj_potrebnu_kapacitu(stroj+1, obdobie, [vrstva_1, vrstva_2,
vrstva_3])
                else:
                    vrstva_3[nahodny_vyrobok,obdobie]-= (potrebna_kapacita - dispo_kapacita)/cyklove_casy[
nahodny_vyrobok]
                    potrebna_kapacita = vypocitaj_potrebnu_kapacitu(stroj+1, obdobie, [vrstva_1, vrstva_2,
vrstva_3])
            return([vrstva_1, vrstva_2, vrstva_3])

# Funkcia odburavania sklzu
def odburaj_sklz (riesenie):
    vrstva_1 = np.copy(riesenie[0])
    vrstva_2 = np.copy(riesenie[1])
    vrstva_3 = np.copy(riesenie[2])
    for obdobie in range(dopyt.shape[1]):
        sklz = vypocitaj_sklz(vrstva_3[:,obdobie+1])
        nenulove = np.nonzero(sklz)[0]
        while len(nenulove) > 0:
            nahodny_vyrobok = nenulove[randint(0, len(nenulove)-1)]#nahodny vyrobok
            if vrstva_1[nahodny_vyrobok,obdobie] != 0:
                mozne_produkovat = (kapacita[vrstva_1[nahodny_vyrobok,obdobie]-1,obdobie]-
vypocitaj_potrebnu_kapacitu(vrstva_1[nahodny_vyrobok,obdobie], obdobie, [vrstva_1, vrstva_2, vrstva_3]))/
cyklove_casy[nahodny_vyrobok]
                if mozne_produkovat >= 1:
                    vrstva_3[nahodny_vyrobok,obdobie] += min(sklz[nahodny_vyrobok],mozne_produkovat)
                    sklz[nahodny_vyrobok] = 0
                    nenulove = np.nonzero(sklz)[0]
                else:
                    sklz[nahodny_vyrobok] = 0
                    nenulove = np.nonzero(sklz)[0]
            else:
                nahodny_stroj = zastupitelnost(nahodny_vyrobok)[randint(0, len(zastupitelnost(nahodny_vyrobok))
-1)]
                pretypy = pretyp(nahodny_stroj, [vrstva_1, vrstva_2, vrstva_3])
                z = posledne_nastavenie_strojov[nahodny_stroj-1] if len(np.where(pretypy[:,2] <= obdobie)[0])
== 0 else pretypy[np.where(pretypy[:,2] <= obdobie)[0][-1]][1]
                mozne_produkovat = (kapacita[nahodny_stroj-1,obdobie]-vypocitaj_potrebnu_kapacitu(nahodny_stroj
, obdobie, [vrstva_1, vrstva_2, vrstva_3])-casy_pretypovania_strojov[z, nahodny_vyrobok])/cyklove_casy[
nahodny_vyrobok]
                if mozne_produkovat >= 1:
                    vrstva_3[nahodny_vyrobok,obdobie] += min(sklz[nahodny_vyrobok],mozne_produkovat)
                    vrstva_2[nahodny_vyrobok,obdobie] = len(np.where(pretypy[:,2] == obdobie)[0])+1
                    vrstva_1[nahodny_vyrobok,obdobie] = nahodny_stroj
                    sklz[nahodny_vyrobok] = 0
                    nenulove = np.nonzero(sklz)[0]
                else:
                    sklz[nahodny_vyrobok] = 0
                    nenulove = np.nonzero(sklz)[0]
            return([vrstva_1, vrstva_2, vrstva_3])

# Tato funkcia generuje vychodiskovu populaciu
def inicializacia (velkost_populacie):
    vychodiskove_riesenia = []
    for x in range (velkost_populacie):
        vrstva_1 = np.zeros_like(dopyt)
        vrstva_2 = np.zeros_like(dopyt)
        vrstva_3 = np.zeros_like(dopyt)
        vychodiskove_riesenia.append(odburaj_sklz([vrstva_1, vrstva_2, vrstva_3]))
    return (vychodiskove_riesenia)

```

```

# Turnajovy vyber
def selekcia (popul):
    orig_fit = [HUF(x) for x in popul]
    nova_popul = [popul[np.argmax(orig_fit)]]
    for x in range (len(popul)-1):
        prvvy_index = randint(0, len(popul)-1)
        druhy_index = randint(0, len(popul)-1)
        while druhy_index == prvvy_index:
            druhy_index = randint(0, len(popul)-1)
        if orig_fit[prvvy_index] <= orig_fit[druhy_index]:
            nova_popul.append(popul[prvvy_index])
        else:
            nova_popul.append(popul[druhy_index])
    return(nova_popul)

# Funkcia ktora vykonava rekombinaciu
def crossover (popul):
    nova_populacia = []
    while len(nova_populacia)<len(popul):
        prvvy_individual = randint(0, len(popul)-1)
        druhy_individual = randint(0, len(popul)-1)
        while druhy_individual == prvvy_individual:
            druhy_individual = randint(0, len(popul)-1)
        t_obdobia = len(popul[0][0][0])
        pozicia_zlomu = randint(0,t_obdobia-1)
        vrstva1 = np.concatenate((popul[prvvy_individual][0][:,0:pozicia_zlomu], popul[druhy_individual][0][:,
        pozicia_zlomu:t_obdobia]),axis = 1)
        vrstva2 = np.concatenate((popul[prvvy_individual][1][:,0:pozicia_zlomu], popul[druhy_individual][1][:,
        pozicia_zlomu:t_obdobia]),axis = 1)
        vrstva3 = np.concatenate((popul[prvvy_individual][2][:,0:pozicia_zlomu], popul[druhy_individual][2][:,
        pozicia_zlomu:t_obdobia]),axis = 1)
        potomok = [vrstva1,vrstva2,vrstva3]
        nova_populacia.append(oprav_riesenie(potomok))
        vrstva1 = np.concatenate((popul[druhy_individual][0][:,0:pozicia_zlomu], popul[prvvy_individual][0][:,
        pozicia_zlomu:t_obdobia]),axis = 1)
        vrstva2 = np.concatenate((popul[druhy_individual][1][:,0:pozicia_zlomu], popul[prvvy_individual][1][:,
        pozicia_zlomu:t_obdobia]),axis = 1)
        vrstva3 = np.concatenate((popul[druhy_individual][2][:,0:pozicia_zlomu], popul[prvvy_individual][2][:,
        pozicia_zlomu:t_obdobia]),axis = 1)
        potomok = [vrstva1,vrstva2,vrstva3]
        nova_populacia.append(oprav_riesenie(potomok))
    return(nova_populacia)

# Funkcia ktora vykonava mutaciu
def mutacia (popul,pravdepodobnos_mutacie):
    nova_populacia = np.copy(popul)
    for index, riesenie in enumerate(popul):
        testovacia_pravdepodobnost = uniform(0,1)
        if testovacia_pravdepodobnost <= pravdepodobnos_mutacie:
            vrstva_1 = np.copy(riesenie[0])
            vrstva_2 = np.copy(riesenie[1])
            vrstva_3 = np.copy(riesenie[2])
            typ_mutacie = randint(0, 2)
            if typ_mutacie == 0:
                nahodny_vyrobok = randint(0, vrstva_1.shape[0]-1)
                nahodne_obdobie = randint(0, vrstva_1.shape[1]-1)
                pripustne_stroje = zastupitelnost(nahodny_vyrobok)
                pripustne_stroje.append(0)
                pripustne_stroje.remove(vrstva_1[nahodny_vyrobok, nahodne_obdobie]) # Vymaz predosly stroj z
                pripustnej_mnoziny (nechceme zvolit ten isty)
                nahodny_stroj = pripustne_stroje[randint(0, len(pripustne_stroje)-1)] # Iny nahodny stroj
                zmen = np.where((vrstva_1[:, nahodne_obdobie] == vrstva_1[nahodny_vyrobok, nahodne_obdobie]) & (
                vrstva_2[:, nahodne_obdobie] > vrstva_2[nahodny_vyrobok, nahodne_obdobie]))
                vrstva_2[zmen[0], nahodne_obdobie] -= 1
                vrstva_1[nahodny_vyrobok, nahodne_obdobie] = 0
                vrstva_2[nahodny_vyrobok, nahodne_obdobie] = 0
                vrstva_3[nahodny_vyrobok, nahodne_obdobie] = 0
                if nahodny_stroj != 0:
                    vrstva_1[nahodny_vyrobok, nahodne_obdobie] = nahodny_stroj
                    zmen = np.where(vrstva_1[:, nahodne_obdobie] == nahodny_stroj)
                    vrstva_2[zmen[0], nahodne_obdobie] += 1
                    potrebna_kapacita = vypocitaj_potrebnu_kapacitu(nahodny_stroj, nahodne_obdobie, [vrstva_1,
                    vrstva_2, vrstva_3])
                    dispo_kapa = kapacita[nahodny_stroj-1, nahodne_obdobie]
                    mozne_produkovat = (dispo_kapa - potrebna_kapacita)/cyklove_casy[nahodny_vyrobok]
                    sklz = sum(dopyt[nahodny_vyrobok]-vrstva_3[nahodny_vyrobok])
                    if sklz > 0 and mozne_produkovat > 0 :
                        vrstva_3[nahodny_vyrobok, nahodne_obdobie] = min(sklz, mozne_produkovat)
                    else:
                        zmen = np.where(vrstva_1[:, nahodne_obdobie] == nahodny_stroj)
                        vrstva_2[zmen[0], nahodne_obdobie] -= 1
                        vrstva_1[nahodny_vyrobok, nahodne_obdobie] = 0
                        nova_populacia[index] = [vrstva_1, vrstva_2, vrstva_3]
            elif typ_mutacie == 1:
                nahodne_obdobie = randint(0, vrstva_1.shape[1]-1)
                nahodny_vyrobok_1 = randint(0, vrstva_1.shape[0]-1)
                nahodny_vyrobok_2 = randint(0, vrstva_1.shape[0]-1)
                while nahodny_vyrobok_1 == nahodny_vyrobok_2:
                    nahodny_vyrobok_2 = randint(0, vrstva_1.shape[0]-1)
                if vrstva_1[nahodny_vyrobok_1, nahodne_obdobie] == vrstva_1[nahodny_vyrobok_2, nahodne_obdobie]:
                    zapamataj = vrstva_2[nahodny_vyrobok_1, nahodne_obdobie]
                    vrstva_2[nahodny_vyrobok_1, nahodne_obdobie] = vrstva_2[nahodny_vyrobok_2, nahodne_obdobie]
                    vrstva_2[nahodny_vyrobok_2, nahodne_obdobie] = zapamataj
                nova_populacia[index] = oprav_riesenie([vrstva_1, vrstva_2, vrstva_3])

```

```

elif typ_mutacie == 2:
    nahodny_vyrobok = randint(0, vrstva_1.shape[0]-1)
    nahodne_obdobie_1 = randint(0, vrstva_1.shape[1]-1)
    nahodne_obdobie_2 = randint(0, vrstva_1.shape[1]-1)
    while nahodne_obdobie_1 == nahodne_obdobie_2:
        nahodne_obdobie_2 = randint(0, vrstva_1.shape[1]-1)
    if vrstva_1[nahodny_vyrobok, nahodne_obdobie_1] != 0 and vrstva_1[nahodny_vyrobok,
nahodne_obdobie_2] != 0:
        vyuzita_kapa_1 = vypocitaj_potrebnu_kapacitu(vrstva_1[nahodny_vyrobok, nahodne_obdobie_1],
nahodne_obdobie_1, [vrstva_1, vrstva_2, vrstva_3])
        dispo_kapa_1 = kapacita[vrstva_1[nahodny_vyrobok, nahodne_obdobie_1]-1, nahodne_obdobie_1]
        volna_kapa_1 = dispo_kapa_1 - vyuzita_kapa_1
        vyuzita_kapa_2 = vypocitaj_potrebnu_kapacitu(vrstva_1[nahodny_vyrobok, nahodne_obdobie_2],
nahodne_obdobie_2, [vrstva_1, vrstva_2, vrstva_3])
        dispo_kapa_2 = kapacita[vrstva_1[nahodny_vyrobok, nahodne_obdobie_2]-1, nahodne_obdobie_2]
        volna_kapa_2 = dispo_kapa_2 - vyuzita_kapa_2
        if volna_kapa_1 > cyklove_casy[nahodny_vyrobok]:
            y = min(vrstva_3[nahodny_vyrobok, nahodne_obdobie_2], volna_kapa_1/cyklove_casy[
nahodny_vyrobok])
            vaha = uniform(0, 1)
            objem_produkcie = round(vaha*y, 0)
            vrstva_3[nahodny_vyrobok, nahodne_obdobie_1] += objem_produkcie
            vrstva_3[nahodny_vyrobok, nahodne_obdobie_2] -= objem_produkcie
            if vrstva_3[nahodny_vyrobok, nahodne_obdobie_2] == 0:
                zmen = np.where((vrstva_1[:, nahodne_obdobie_2] == vrstva_1[nahodny_vyrobok,
nahodne_obdobie_2]) & (vrstva_2[:, nahodne_obdobie_2] > vrstva_2[nahodny_vyrobok, nahodne_obdobie_2]))
                vrstva_2[zmen[0], nahodne_obdobie_2] -= 1
                vrstva_1[nahodny_vyrobok, nahodne_obdobie_2] = 0
                vrstva_2[nahodny_vyrobok, nahodne_obdobie_2] = 0
            elif volna_kapa_2 > cyklove_casy[nahodny_vyrobok]:
                y = min(vrstva_3[nahodny_vyrobok, nahodne_obdobie_1], volna_kapa_2/cyklove_casy[
nahodny_vyrobok])
                vaha = uniform(0, 1)
                objem_produkcie = round(vaha*y, 0)
                vrstva_3[nahodny_vyrobok, nahodne_obdobie_2] += objem_produkcie
                vrstva_3[nahodny_vyrobok, nahodne_obdobie_1] -= objem_produkcie
                if vrstva_3[nahodny_vyrobok, nahodne_obdobie_1] == 0:
                    zmen = np.where((vrstva_1[:, nahodne_obdobie_1] == vrstva_1[nahodny_vyrobok,
nahodne_obdobie_1]) & (vrstva_2[:, nahodne_obdobie_1] > vrstva_2[nahodny_vyrobok, nahodne_obdobie_1]))
                    vrstva_2[zmen[0], nahodne_obdobie_1] -= 1
                    vrstva_1[nahodny_vyrobok, nahodne_obdobie_1] = 0
                    vrstva_2[nahodny_vyrobok, nahodne_obdobie_1] = 0
                if vrstva_1[nahodny_vyrobok, nahodne_obdobie_1] == 0 and vrstva_1[nahodny_vyrobok,
nahodne_obdobie_2] != 0:
                    volna_kapacita = []
                    pripustne_stroje = []
                    for x in zastupitelnost(nahodny_vyrobok):
                        volna_kapacita.append(kapacita[x-1, nahodne_obdobie_1] - vypocitaj_potrebnu_kapacitu(x,
nahodne_obdobie_1, [vrstva_1, vrstva_2, vrstva_3]))
                        pripustne_stroje.append(x)
                    lepsi_stroj = pripustne_stroje[np.argmax(volna_kapacita)]-1
                    volna_kapa_1 = volna_kapacita[np.argmax(volna_kapacita)]
                    pretyp = pretyp(lepsi_stroj+1, [vrstva_1, vrstva_2, vrstva_3])
                    z = posledne_nastavenie_strojov[lepsi_stroj] if len(np.where(pretyp[:, 2] <
nahodne_obdobie_1)[0]) == 0 else pretyp[np.where(pretyp[:, 2] < nahodne_obdobie_1)[0][:-1]][1]
                    if volna_kapa_1 > casy_pretypovania_strojov[z, nahodny_vyrobok] + cyklove_casy[
nahodny_vyrobok]:
                        y = min(vrstva_3[nahodny_vyrobok, nahodne_obdobie_2], (volna_kapa_1 -
casy_pretypovania_strojov[z, nahodny_vyrobok])/cyklove_casy[nahodny_vyrobok])
                        vaha = uniform(0, 1)
                        objem_produkcie = round(vaha*y, 0)
                        vrstva_3[nahodny_vyrobok, nahodne_obdobie_1] += objem_produkcie
                        vrstva_3[nahodny_vyrobok, nahodne_obdobie_2] -= objem_produkcie
                        vrstva_1[nahodny_vyrobok, nahodne_obdobie_1] = lepsi_stroj+1
                        zmen = np.where(vrstva_1[:, nahodne_obdobie_1] == vrstva_1[nahodny_vyrobok,
nahodne_obdobie_1])
                        vrstva_2[zmen[0], nahodne_obdobie_1] += 1
                        if vrstva_3[nahodny_vyrobok, nahodne_obdobie_2] == 0:
                            zmen = np.where((vrstva_1[:, nahodne_obdobie_2] == vrstva_1[nahodny_vyrobok,
nahodne_obdobie_2]) & (vrstva_2[:, nahodne_obdobie_2] > vrstva_2[nahodny_vyrobok, nahodne_obdobie_2]))
                            vrstva_2[zmen[0], nahodne_obdobie_2] -= 1
                            vrstva_1[nahodny_vyrobok, nahodne_obdobie_2] = 0
                            vrstva_2[nahodny_vyrobok, nahodne_obdobie_2] = 0
                        if vrstva_1[nahodny_vyrobok, nahodne_obdobie_2] == 0 and vrstva_1[nahodny_vyrobok,
nahodne_obdobie_1] != 0:
                            volna_kapacita = []
                            pripustne_stroje = []
                            for x in zastupitelnost(nahodny_vyrobok):
                                volna_kapacita.append(kapacita[x-1, nahodne_obdobie_2] - vypocitaj_potrebnu_kapacitu(x,
nahodne_obdobie_2, [vrstva_1, vrstva_2, vrstva_3]))
                                pripustne_stroje.append(x)
                            lepsi_stroj = pripustne_stroje[np.argmax(volna_kapacita)]-1
                            volna_kapa_1 = volna_kapacita[np.argmax(volna_kapacita)]
                            pretyp = pretyp(lepsi_stroj+1, [vrstva_1, vrstva_2, vrstva_3])
                            z = posledne_nastavenie_strojov[lepsi_stroj] if len(np.where(pretyp[:, 2] <
nahodne_obdobie_2)[0]) == 0 else pretyp[np.where(pretyp[:, 2] < nahodne_obdobie_2)[0][:-1]][1]
                            if volna_kapa_1 > casy_pretypovania_strojov[z, nahodny_vyrobok] + cyklove_casy[
nahodny_vyrobok]:
                                y = min(vrstva_3[nahodny_vyrobok, nahodne_obdobie_1], (volna_kapa_1 -
casy_pretypovania_strojov[z, nahodny_vyrobok])/cyklove_casy[nahodny_vyrobok])
                                vaha = uniform(0, 1)
                                objem_produkcie = round(vaha*y, 0)
                                vrstva_3[nahodny_vyrobok, nahodne_obdobie_2] += objem_produkcie
                                vrstva_3[nahodny_vyrobok, nahodne_obdobie_1] -= objem_produkcie
                                vrstva_1[nahodny_vyrobok, nahodne_obdobie_2] = lepsi_stroj+1
                                zmen = np.where(vrstva_1[:, nahodne_obdobie_2] == vrstva_1[nahodny_vyrobok,
nahodne_obdobie_2])
                                vrstva_2[zmen[0], nahodne_obdobie_2] += 1

```

```

        if vrstva_3[nahodny_vyrobok,nahodne_obdobie_1] == 0:
            zmen = np.where((vrstva_1[:,nahodne_obdobie_1] == vrstva_1[nahodny_vyrobok,
nahodne_obdobie_1]) & (vrstva_2[:,nahodne_obdobie_1] > vrstva_2[nahodny_vyrobok,nahodne_obdobie_1]))
            vrstva_2[zmen[0],nahodne_obdobie_1] -= 1
            vrstva_1[nahodny_vyrobok,nahodne_obdobie_1] = 0
            vrstva_2[nahodny_vyrobok,nahodne_obdobie_1] = 0
            nova_populacia[index] = oprav_riesenie([vrstva_1, vrstva_2, vrstva_3])
    return (nova_populacia)
#####
#_Vyuzi algoritmus_#
#####
# Najde najlepsie riesenie v celej populacii a vypise riesenie a jeho fitness hodnotu
def naj_riesenie (popul):
    fitness_hodnoty = []
    for riesenie in popul:
        fitness_hodnoty.append(HUF(riesenie))
    return(min(fitness_hodnoty), popul[np.argmin(fitness_hodnoty)])

# Formatovanie vystupu
def df_kapa_riesenie (riesenie, stroj):
    index = np.where(riesenie[0] == stroj)
    df_poradie = np.zeros_like(riesenie[1])
    df_poradie[index] = riesenie[1][index]
    iba_vyroba = np.zeros_like(riesenie[2], dtype="float64")
    iba_vyroba[index] = riesenie[2][index]*cyklove_casy[index[0]]
    davky = np.zeros_like(riesenie[2], dtype="float64")
    davky[index] = riesenie[2][index]
    poradie = pretyp(stroj, riesenie)
    iba_pretyp = np.zeros_like(riesenie[2], dtype="float64")
    for x in poradie:
        iba_pretyp[x[1],x[2]] = casy_pretypovania_strojov[x[0],x[1]]
    spolu_kapa = iba_vyroba + iba_pretyp
    vyuzita_kapacita = np.sum( spolu_kapa, axis = 0)
    pretypovanie_spolu = np.sum( iba_pretyp, axis = 0)
    vyroba_spolu = np.sum( iba_vyroba, axis = 0)
    plan_pretypovania = pd.DataFrame(df_poradie, index = vyrobky, columns = data.columns[3:15])
    spolu_kapacita = pd.DataFrame( spolu_kapa, index = vyrobky, columns = data.columns[3:15])
    spolu_kapacita.loc["Disponibilna kapacita"] = kapacita[stroj-1]
    spolu_kapacita.loc["Vyuzita kapacita"] = vyuzita_kapacita
    spolu_kapacita.loc["Zatazenost stroja (%)"] = round( spolu_kapacita.loc["Vyuzita kapacita"]/spolu_kapacita.
loc["Disponibilna kapacita"]*100,2).fillna(0)
    spolu_kapacita.loc["Z toho pretyp"] = pretypovanie_spolu
    spolu_kapacita.loc["Z toho vyroba"] = vyroba_spolu
    spolu_kapacita.loc["Pomer pretypovania (%)"] = round( spolu_kapacita.loc["Z toho pretyp"]/spolu_kapacita.loc[
"Vyuzita kapacita"]*100,2).fillna(0)

    return(plan_pretypovania,
pd.DataFrame(davky, index = vyrobky, columns = data.columns[3:15]),
spolu_kapacita,
pd.DataFrame(iba_pretyp, index = vyrobky, columns = data.columns[3:15]),
pd.DataFrame(iba_vyroba, index = vyrobky, columns = data.columns[3:15]))

# Spusti optimalizacny algoritmus
def optimalizuj (velkost_populacie,pocet_generacii,pravdepodobnos_mutacie, hranica_nezlepsenia):
    naklady = {"Naklady sklzu":[], "Naklady zasob":[], "Naklady pretypovania":[], "Celkove naklady":[]}
    najlepsie_riesenie = [0,0,0]
    populacia = inicializacia(velkost_populacie)
    naj_fit_riesenie = naj_riesenie(populacia)
    naklady["Celkove naklady"].append(naj_fit_riesenie[0])
    najlepsie_riesenie[0] = naj_fit_riesenie[0]
    naj_predosle = naj_fit_riesenie[0]
    najlepsie_riesenie[1] = naj_fit_riesenie[1]
    sklz = sum(naklady_sklzu @ vypocitaj_sklz(naj_fit_riesenie[1][2])[:,1:])
    naklady["Naklady sklzu"].append(sklz)
    zasoby = sum(naklady_skladovania @ vypocitaj_zasoby(naj_fit_riesenie[1][2])[:,1:])
    naklady["Naklady zasob"].append(zasoby)
    naklady["Naklady pretypovania"].append(naj_fit_riesenie[0] - sklz - zasoby)
    print("Inicializacia -",0,naj_fit_riesenie[0], "€")
    counter = 0
    for generacia in range(pocet_generacii-1):
        populacia = selekcia(populacia)
        populacia = crossover(populacia)
        populacia = mutacia(populacia, pravdepodobnos_mutacie)
        naj_fit_riesenie = naj_riesenie(populacia)
        naklady["Celkove naklady"].append(naj_fit_riesenie[0])
        sklz = sum(naklady_sklzu @ vypocitaj_sklz(naj_fit_riesenie[1][2])[:,1:])
        naklady["Naklady sklzu"].append(sklz)
        zasoby = sum(naklady_skladovania @ vypocitaj_zasoby(naj_fit_riesenie[1][2])[:,1:])
        naklady["Naklady zasob"].append(zasoby)
        naklady["Naklady pretypovania"].append(naj_fit_riesenie[0] - sklz - zasoby)
        # Uchovanie najlepsiho riesenia
        if naj_fit_riesenie[0] < najlepsie_riesenie[0]:
            najlepsie_riesenie[0] = naj_fit_riesenie[0]
            najlepsie_riesenie[1] = naj_fit_riesenie[1]
            najlepsie_riesenie[2] = generacia+1
        # Uchovanie riesenia od posledneho nezlepsenia
        if naj_fit_riesenie[0] < naj_predosle:
            naj_predosle = naj_fit_riesenie[0]
            counter = 0
        else:
            naj_predosle = naj_fit_riesenie[0]
            counter+=1
        print("Generacia -",generacia+1,"HUF -",naj_fit_riesenie[0], "€", "Riesenie sa nezlepsilo -",counter,"x"
)

```

```
if counter == hranica_nezlepsenia:
    counter = 0
    print("Inicializujem 80% novych jedincov v populacii.")
    vymaz = math.ceil(len(populacia)*0.8)
    populacia = populacia[vymaz:]
    generuj_novu = np.array(inicializacia(vymaz))
    populacia = np.concatenate((populacia, generuj_novu))
df_nakaldy = pd.DataFrame(naklady)
print("Najlepsie riesenie algoritmus dosiahol v ", najlepsie_riesenie[2], ". generacii.")
print("Hodnota ucelovej funkcie :", najlepsie_riesenie[0])
return najlepsie_riesenie, df_nakaldy, fig.show()
```

Zdroj: vlastné spracovanie