



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV PROCESNÍHO INŽENÝRSTVÍ**

INSTITUTE OF PROCESS ENGINEERING

**OPTIMÁLNÍ PLÁNOVÁNÍ VÝROBY V  
PRŮMYSLOVÉM PODNIKU**

OPTIMAL PRODUCTION SCHEDULING IN AN INDUSTRIAL ENTERPRISE

**DIPLOMOVÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Bc. Radim Voňka

**VEDOUCÍ PRÁCE**

SUPERVISOR

Ing. Michal Touš, Ph.D.

BRNO 2022



# Zadání diplomové práce

Ústav: Ústav procesního inženýrství  
Student: **Bc. Radim Voňka**  
Studijní program: Procesní inženýrství  
Studijní obor: bez specializace Vedoucí  
práce: **Ing. Michal Touš, Ph.D.**  
Akademický rok: 2021/22

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijníma zkušebníma řádem VUT v Brně určuje následující téma diplomové práce:

## Plánování výroby pomocí pokročilých algoritmů

### **Stručná charakteristika problematiky úkolu:**

Plánování výroby je nezbytnou aktivitou výrobních podniků. Cílem je najít takový plán, který efektivně využívá výrobní kapacity a minimalizuje tak výrobní časy. V případě výrob, kdy se zpracovává více výrobků na více strojích s určitou posloupností prací, je hledání efektivního plánu bez optimalizačních algoritmů obtížné, ne-li nemožné. Cílem této práce je definice optimalizační úlohy plánování a její implementace a řešení v jazyce Python.

### **Cíle diplomové práce:**

Implementovat úlohu plánování výroby pomocí jazyka Python  
Vyřešit úlohu pro konkrétní zadání

### **Seznam doporučené literatury:**

MOHAN, Jatoth, Krishnanand LANKA a A. Neelakanteswara RAO. A Review of Dynamic Job Shop Scheduling Techniques. Procedia Manufacturing [online]. 2019, 30, 34-39 [cit. 2021-10-7]. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2019.02.006

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

---

prof. Ing. Petr Stehlík, CSc., dr. h. c.  
ředitel ústavu

---

doc. Ing. Jaroslav Katolický, Ph.D.  
děkan fakulty

## **Abstrakt**

Plánování výroby je nedílnou součástí všech továrních podniků. Dosažení efektivního plánování ve firmách zabývajících se kusovou produkcí je velmi komplikované. K řešení složitých problémů souvisejících s plánováním je využíváno optimalizačních algoritmů, které problémy dokáží účinně vyřešit v reálném čase, a tak dosáhnout požadovaných výsledků. Vytvoření a implementace kapacitního plánování s účelem minimalizování výrobních časů v podniku je tématem diplomové práce. K tvorbě nástroje pro optimální plánování výroby je využit programovací jazyk Python.

## **Klíčová slova**

Kapacitní plánování, model plánování, Python, programování s omezením

## **Abstract**

Production planning is an integral part of all factory companies. Achieving effective planning in piece production companies is very complicated. Optimization algorithms are used to solve complex planning-related problems, which can effectively solve problems in real time and thus achieve the desired results. The creation and implementation of capacity planning in order to minimize production times in the company is the topic of the thesis. The Python programming language is used to create a tool for optimal production planning.

## **Keywords**

Capacity planning, planning model, Python, constrained programming



## **Bibliografická citace**

VOŇKA, Radim. *Plánování výroby pomocí pokročilých algoritmů* [online]. Brno, 2022 [cit. 2022-05-05]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/140753>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav procesního inženýrství. Vedoucí práce Michal Touš.

## **Prohlášení**

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením pana Ing. Michala Touše, Ph.D., pracovníků firmy PHARMIX, s.r.o. a s použitím odborné literatury a pramenů, které jsou uvedeny v seznamu použité literatury.

V Brně dne 5. 5. 2022

.....

Bc. Radim Voňka

## **Poděkování**

Rád bych poděkoval svému vedoucímu Ing. Michalovi Toušovi Ph.D. za odborné vedení, ochotu, obětovaný čas a spoustu drahocenných rad během psaní diplomové práce.

Poděkování patří také technické ředitelce Ing. Gabriele Orlové, řediteli Františku Hrabalovi a jednatele Ing. Rostislavu Horčíčkovi za možnost zpracování diplomové práce právě ve společnosti PHARMIX, s.r.o.

Dále bych chtěl poděkovat celé mé rodině a své ženě za trpělivost a podporu, která mě provázela během celého studia.



## Obsah

<b>1 Úvod</b> .....	<b>13</b>
<b>2 Plánování v podniku</b> .....	<b>14</b>
2.1 Cíle plánování .....	14
2.2 Členění plánování .....	15
<b>3 Představení společnosti PHARMIX, s.r.o.</b> .....	<b>17</b>
3.1 Popis společnosti .....	17
3.1.1 organizační struktura .....	18
3.2 Informační systémy .....	19
3.2.1 Informační systém ERP .....	20
3.2.2 Informační docházkový systém .....	21
3.3 Vykazování splněné práce .....	21
3.4 průběh zakázky .....	22
3.5 Původní proces plánování výroby .....	25
3.6 Návrh na zkvalitnění plánování .....	26
<b>4 Metody řešení</b> .....	<b>27</b>
4.1 Matematický model .....	27
4.2 Volba algoritmu řešení .....	31
4.2.1 OR-Tools .....	32
4.3 Sběr dat.....	33
4.3.1 Původní systém sběru dat .....	33
4.3.2 Zmodernizovaný systém sběru dat.....	34
4.3.3 Chyby sběru dat .....	36
4.4 Zpracování dat .....	36
4.4.1 Rozdělení pracovišť.....	37
4.4.2 Zpracování dat pro kapacitní plánování.....	39
4.5 Implementace nástroje pro optimální plánování výroby .....	40
4.5.1 Příprava modulů a knihoven.....	40
4.5.2 Deklarace modelu.....	41
4.5.3 Vstupy dat .....	42
4.5.4 Definice dat.....	43
4.5.5 Definice proměnných.....	44
4.5.6 Definice omezení.....	46

4.5.7 Definice výstupů .....	47
4.5.8 Vypsání výsledků.....	48
4.5.9 Vykreslení řešení.....	49
<b>5 Řešení případové studie .....</b>	<b>51</b>
5.1 Vstupní data .....	51
5.2 Vypsání řešení .....	53
5.3 Vyhodnocení správnosti řešení .....	55
<b>6 Závěr .....</b>	<b>56</b>
<b>Seznam použité literatury .....</b>	<b>57</b>
<b>Seznam obrázků .....</b>	<b>59</b>
<b>Seznam tabulek .....</b>	<b>60</b>
<b>Seznam zkratk.....</b>	<b>61</b>
<b>Seznam příloh .....</b>	<b>62</b>

# 1 Úvod

V dnešní době je zákazníky na podniky vyvíjen čím dál větší tlak na snižování cen, vyšší kvalitu výrobků a co nejkratší možnou dobu výroby. Podniky se snaží požadavkům vyhovět, aby si zákazníky udržely a byly konkurenceschopné. K tomu potřebují přesné plánování, které jim zajistí plynulost výroby a dosažení minimálních časových prodlev.

Plánování v podnicích, zabývajících se kusovou výrobou, je obzvlášť komplikované. Přináší řadu problémů, které mohou při výrobě nastat. Jedná se o těžko předvídatelné dynamické události, jako např. příchody nových objednávek, náhlé změny situací při výrobě a poruchy strojů. Tyto problémy související s plánováním podniky řeší prostřednictvím speciálních nástrojů pro optimální plánování výroby vytvářených na míru dle specifické výroby podniku.

Společnost PHARMIX, s.r.o., zabývající se kusovou zakázkovou výrobou nerezových strojů a zařízení pro farmacii, chemii a potravinářství, je jedním z podniků potýkajících se s problémy souvisejícími s plánováním. Kvůli nesprávnému dodržování smluvených termínů výroby vznikl ve společnosti návrh na zavedení sofistikovaného systému plánování, který by vedl k šetření pracovních kapacit a snížil by délku výroby zakázek. Právě implementace úlohy plánování je tématem diplomové práce.

Cílem práce je optimalizovat plánování výroby za účelem minimalizace časových prodlev mezi operacemi a šetření pracovních kapacit. Řešení musí zohledňovat omezující podmínky, jako posloupnost výroby nebo nemožnost zpracování více úkolů na jednom stroji zároveň. Script pro optimalizaci je vytvářen v programovacím jazyce Python a využívá otevřený zdrojový kód softwarové sady OR-Tools s hybridním řešitelem CP-SAT (Constraint Programming-Boolean Satisfiability).

Práce je strukturována následovně. V úvodní kapitole č. 1 je obecně přiblíženo plánování v podniku s cíli a členěním plánování. Kapitola č. 2 je věnována podniku PHARMIX, s.r.o. Podrobně se zaměřuje na kapacitní plánování ve společnosti a vše s ním spojené. Metody řešení jsou popsány ve 3. kapitole. Zabývají se matematickým popisem plánování dle charakteru výroby v podniku, volbou vhodného algoritmu řešení a prací s daty. Součástí kapitoly je také objasnění tvorby scriptu pro optimalizaci kapacitního plánování v jazyce Python včetně jeho funkčnosti. Kapitola č. 4 se věnuje prezentování řešení případové studie na zkušebních zakázkách ve výrobě s využitím již vytvořeného scriptu z předchozí kapitoly. Závěrečná zhodnocující kapitola č. 5 shrnuje výsledky, kterých bylo dosaženo v diplomové práci.

## 2 Plánování v podniku

Plánování hraje v současné době intenzivních konkurenčních bojů zásadní roli, jež může ovlivnit, zda podnik bude na trhu úspěšný a vytvoří si co nejlepší podmínky, či nikoli. Je velmi těžké zvolit vhodnou metodu plánování, která bude poskytovat optimální řešení pro daný podnik.

Pojmem plánování se rozumí vymezení cílů a naznačení cest, jimiž by se mělo požadovaných cílů dosáhnout. K dosažení cílů slouží plán, který závisí na realizaci ostatních funkcí podniku, jako např. vedení, koordinace práce, organizace úkolů, kontrolování splněných úkolů a vhodná komunikace [1].

Proces plánování je závislý na mnoha faktorech, kterými mohou být: velikost, struktura, technická vyspělost a obor podnikání. Kvůli velké míře odlišnosti podniků se přistupuje ke specifickému plánování pro každý podnik. Jednotlivé plány podniků mají společnou tzv. Smart metodu, která by měla splňovat klíčové zásady [2]:

- Konkrétnost – plán musí obsahovat požadované výstupy (formu, kvalitu, čas ukončení, úplnost a podrobnosti)
- Měřitelnost – plán musí být vyjádřen vždy ve stejných jednotkách (čas nebo měna), aby bylo možné porovnat data z minulých let, a aby byla zároveň zachována vypovídající schopnost dat
- Dosažitelnost – plán musí být pro zaměstnance dostatečně motivující a dosažitelný, aby se s ním ztotožnili
- Reálnost – plán nesmí být nereálný, protože musí být v souladu s výrobními možnostmi podniku a konkurenceschopností
- Materiálnost – plán musí reflektovat hmatatelné akce a kroky v oblasti výroby, obchodu, nákupu a dalších činností

### 2.1 Cíle plánování

Bez předem jasně vymezených cílů by plánovací proces neměl smysl a jednalo by se o čistě náhodný proces. Jakmile jsou cíle předem přesně definované, tak je od zaměstnanců očekávána vyšší výkonnost, efektivita a motivace než u neplánovaného procesu [1].

Stanovení cílů plánování má úzkou spojitost s vizí podniku do budoucna. Je důležité, aby cíle podniku reflektovaly samotnou misi a směr, kterým se podnik chce ubírat a čeho chce dosáhnout. Takové cíle se nazývají globální neboli dlouhodobé a měly by být vydány vedením podniku formou písemného dokumentu. Globální cíle vymezují obecné výrobní cíle, mezi které patří např. zvýšení výnosů, zlepšení kvality výrobků a další. Pokud podnik chce dlouhodobých cílů dosáhnout, musí formulovat požadavky na jejich dosažení. Vzniká hierarchie stanovení cílů, jež bývá často vyobrazena jako tzv. pyramida cílů viz obrázek č. 2.1 [2].



*Obrázek 2.1 Pyramida cílů*

Pyramida může vypadat u různých podniků odlišně, protože podniky nemají vždy stejné priority cílů. Na obrázku č. 2.1 je vyobrazena pyramida cílů určena přímo pro společnost PHARMIX, s.r.o.

## 2.2 Členění plánování

Plánování lze rozdělit dle několika hledisek uvedených ve zdrojích [1], [2]:

### A) Časové hledisko

Plánování můžeme rozdělit pomocí časového horizontu na 3 základní úrovně plánování:

- Dlouhodobé plány – délka trvání plánu delší než pět let
- Střednědobé plány – délka trvání plánu v rozmezí jednoho až pěti let
- Krátkodobé plány – délka trvání plánu kratší než jeden rok

Podle potřeb podniku je možné plánování také rozdělit dle časového horizontu na plány roční, čtvrtletní, měsíční, týdenní, denní nebo i dokonce směnový plán na daný den či směnu.

### B) Úroveň rozhodovacího procesu

S ohledem na charakter rozhodovacího procesu rozlišujeme:

- Strategické plány – orientují se na globální cíle (velikost podniku, rozvoj, investice a další), jež souvisí s budoucím směrem vývoje a způsoby,

jakými požadovaných cílů dosáhnout. Převážně je vypracovává vedení podniku spolu se svým manažerským týmem. Jedná se zpravidla o dlouhodobé plány.

- Taktické plány – jsou využívány k uskutečnění strategických cílů a definují formu jejich dosažení. Konkrétní postupy a prostředky k dosažení cílů zajišťují organizační pracovníci jednotlivých úseků podniku.
- Operativní plány – vychází z taktických plánů a jsou nejnižší úrovní rozdělení. S jejich pomocí jsou řízeny operace postupů v krátkém časovém úseku (měsíce, týdny, dny nebo směny). Při tvorbě operativních plánů se vychází z konkrétních detailů taktických plánů. Určuje se odpovědná osoba, která má zodpovědnost na daný plán.

### C) Věcná naplň plánu

Jedná se o plány přímo související s předmětem činnosti podniku. Podniky mají jak společné okruhy plánů (finanční, obchodní, investiční, marketingové a další), tak i své vlastní specifické okruhy plánů korespondující přímo s náplní činnosti podniku.

### D) Účelové hledisko

Plány nemusí být jen řídicího charakteru, ale mohou sloužit i k jiným účelům. Jedná se např. o vyřizování úvěru či financování projektu (podnikatelské plány).

Podniky využívají v praxi jen některé plány dle svých potřeb. Nejčastěji je možné se setkat s plány strategickými, ročními a operativními [1].

### 3 Představení společnosti PHARMIX, s.r.o.

Diplomová práce byla vypracována ve spolupráci se společností PHARMIX, s.r.o., která je představitelem kusové výroby na zakázku. Zefektivnění plánování výroby je v současnosti jednou z hlavních priorit této společnosti.

#### 3.1 Popis společnosti

Společnost PHARMIX, s.r.o. byla založena v roce 2000. Sídlem společnosti je město Kroměříž nacházející se ve Zlínském kraji. V současnosti patří společnost mezi významné výrobce a dodavatele strojů a zařízení pro farmaceutický, potravinářský a chemický průmysl, a to nejen v České republice, ale i v několika evropských státech (Německo, Rakousko, Dánsko, Švédsko, Itálie a další). Své výrobky vyváží dále také do USA, Kanady, Číny, Mexika a rovněž do arabsko-perského teritoria. Výrobní program se skládá z vlastního portfolia strojů (např. vertikální homogénizátor, který vyráběla jako jedna z prvních firem na světě) a zařízení z nerezových materiálů, u kterých se jedná převážně o zakázkovou výrobu.



*Obrázek 3.1 Areál společnosti PHARMIX, s.r.o.*

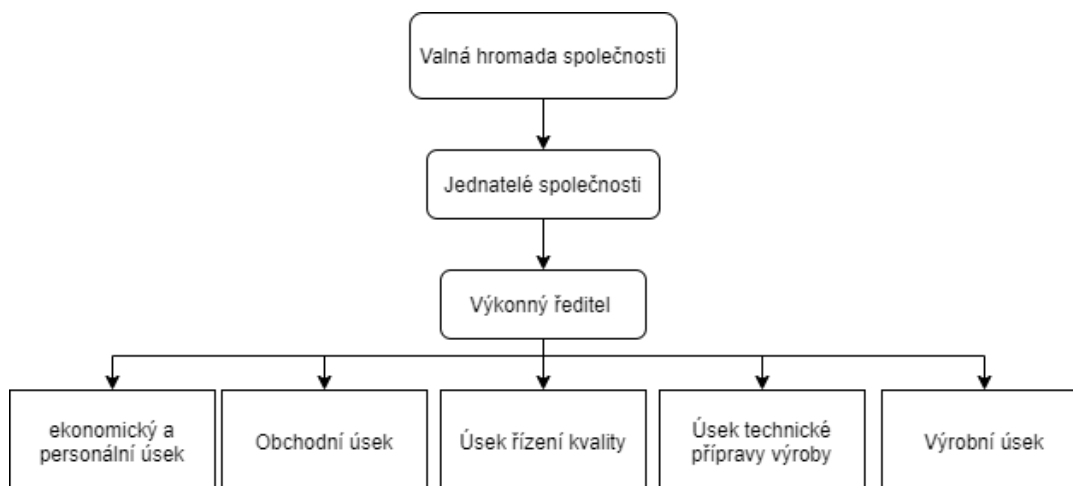
Hlavními oblastmi použití strojů a zařízení z výrobního portfolia jsou:

- Míchání a homogenizace sypkých směsí a kapalin
- Prosévání, granulace a separace
- Sušení
- Skladování a transport
- Zakázková výroba pro nspecifikované obory

Strategií do budoucího vývoje je zajišťování kvalitního výrobního programu s důrazem na perfektní kvalitu, dodání ve smluveném termínu a kvalitní servis. Aby společnost byla schopná dodržovat zmíněné cíle, je nutné, aby zabezpečovala kvalitní vzdělání pracovníků, rychle reagovala na aktuální situaci na trhu a neustále pracovala na inovacích produktů.

### 3.1.1 organizační struktura

Ke znázornění rozdělení vnitřního uspořádání společnosti slouží zjednodušené schéma struktury vedení podniku s jednotlivými podsektory uvedené na obrázku č. 3.2. Jedná se o aktuální schéma pro rok 2022.



*Obrázek 3.2 Organizační struktura společnosti*

#### 1. Ekonomický a personální úsek

Pracovníci zde zabezpečují komplexní činnosti v oblasti personalistiky a mezd pro interní potřeby příslušných úseků a pro externí potřeby vykazování za společnost.

#### 2. Obchodní úsek

Probíhá zde komunikace se zákazníkem od prvopočátečního požadavku na výrobu až po vyexportování zakázky klientovi. Obchodníci po schválení možné realizace zakázky zpracují nabídku, kterou předkládají zákazníkovi a po jejím odsouhlasení zakázku zaplánují do systému. Zaplánování zakázky je impulsem pro začátek procesu plánování, který prvotně obnáší stanovení předběžného termínu dokončení, odhad množství čerpání výrobních kapacit a způsobu výroby. Data stanovování jednotlivých okolností jsou čerpána z podobných zakázek zpracovaných v minulosti.

#### 3. Úsek řízení kvality

Úsek zajišťuje výrobu v souladu s požadovanými normami zakázky, dohled na bezpečnost práce a environmentální politiku společnosti. Pod vedením vedoucího pracovníka úseku, který připravuje požadovanou dokumentaci, certifikace a proškolení personálu, pracuje tým kontrolorů dohlížející na celý průběh výroby. Kontroly kvality probíhají paralelně s výrobou zakázek, proto se pracovní kapacita zaměstnanců úseku nepočítá mezi čerpané kapacity dělníků

potřebných na výrobu. Pracovní doba zaměstnanců úseku řízení kvality tak nebude v budoucím plánovacím systému výroby vystupovat.

#### 4. Úsek Technické přípravy výroby (TPV)

Probíhá zde vypracování konstrukčního zpracování spolu s technickým řešením procesu výroby. Úsek je v přímém kontaktu s obchodníky při tvorbě konstrukční (výkresová dokumentace, kusovník a grafické přílohy) i technologické dokumentace (technologický postup, seznam atestů a norem). Data ze zpracované technologické dokumentace slouží jako plnohodnotná vstupní data pro nástroj na kapacitní plánování výroby.

#### 5. Výrobní úsek

Zabezpečuje dohled nad výrobou zakázek. Podle aktuální situace ve výrobě a vytíženosti pracovišť mohou vedoucí pracovníci úseku rozhodnout o případném vyvezení výrobku do kooperace k vyhotovení požadovaných operací za účelem odlehčení zatížení konkrétních pracovišť.

Pořadí jednotlivých výrobních operací bude v budoucnu plně určovat nástroj na kapacitní plánování výroby, s jehož využitím dojde k minimalizaci časových prodlev mezi operacemi. Zakázka tak bude vyrobena v co nejrychlejším možném termínu vzhledem k aktuální situaci ve výrobě. Na dodržování sledu operací dohlížejí mistři a vedoucí výroby. Průchod dílců výrobním procesem zajišťuje plánovač, který bude rozdávat práci podle vzniklého harmonogramu operací ve výrobě.

### 3.2 Informační systémy

Informace jsou klíčem k úspěšnému fungování podniku se všemi jeho procesy včetně plánování. Ke sběru, uchovávání, přenosu, zpracování a poskytování informací slouží informační systémy, které jsou podniky využívány ke zkvalitnění služeb a zvýšení produktivity. Pro jednotlivé podniky se vytváří na míru nebo se již vytvořený systém upravuje pomocí přídatných modulů.

Na propojení informačních systémů s budoucím kapacitním plánováním je příkládán značný význam hlavně z toho důvodu, že ve výrobním podniku dochází k toku velkého objemu dat. Proto na sebe musí jednotlivé systémy plynule navazovat a nesmí se vzájemně blokovat. Díky provázanosti je práce se systémy uživatelsky dostupnější a zároveň šetří čas pracovníků.

Cílem je tedy získat automatické plánování, jež bude čerpat data z informačních systémů, které výrobní podnik již využívá. Právě automatické plánování je vhodné pro podniky, které mají větší objemy vyráběných dílců a dochází zde k minimálním korekcím v již vytvořeném plánu. Jedním z takových podniků je právě společnost PHARMIX s.r.o.

### 3.2.1 Informační systém ERP

Plánování podnikových zdrojů (Enterprise resource planning, ERP) je informační systém schopný správy a koordinace všech zdrojů, informací a funkcí podniku ze sdílených dat ukládaných do jedné centrální databáze [3]. Pomocí integrace sesbíraných dat jsou získávány a vyhodnocovány cenné informace, které napomáhají sjednotit veškeré procesy v podniku.

Společnost PHARMIX, s.r.o. využívá ERP systém Inuvio, který je produktem společnosti Asseco Solutions, a.s. Inuvio je často používán v malých nebo středně velkých podnicích jak v České republice, tak i jinde ve světě. Právě v České republice je společnost Asseco Solutions, a.s. největším producentem podnikových informačních systémů na domácím trhu. Mezi konkurenty patří světově nejpoužívanější ERP systémy, jako např. německý SAP nebo americký Oracle, jež jsou častěji využívány spíše u větších podniků.

Využívaný ERP systém Inuvio je založen na třech základních funkcích popsaných ve zdroji [4]:

1. Provozní funkce – zasahuje do zajišťování materiálu, výroby dílů, montáže až do dodání zakázky zákazníkovi
2. Funkce služeb – je spjata s přijetím objednávek, monitorováním pokroku, dodáním zakázky a se službami spojenými s údržbou a poradenskou činností ohledně správného provozu
3. Funkce podniku nebo obchodního procesu – postihuje tok zboží, služeb, informací, uskutečňování rozhodnutí a kontroly v celém podniku

Kromě základního rozhraní systému ERP společnost využívá ke své činnosti následující moduly:

- Technická příprava výroby – využívá úsek technické přípravy výroby (kusovník, technologický postup a výkresová dokumentace)
- Řízení výroby – využívá výrobní úsek (výrobní příkazy, kooperační objednávky a stav výroby)
- Oběh zboží – využívá úsek technické přípravy výroby (stav skladu, přejímky a výdejky)
- Účetnictví a fakturace – využívá ekonomický úsek (vedení účetnictví, zadávání dokladů, řešení dopravy)
- Mzdy a personalistika – využívá personální úsek (karty zaměstnanců, hodnocení zaměstnanců a stanovování mezd)
- Pokladna a číselníky – využívá obchodní úsek (organizace zakázek a vedení hotovosti v různých měnách)

Softwarem prochází veškerý tok dat, se kterými společnost pracuje, proto je důležité ho využívat v plném rozsahu. Jeho výhodou je velmi dobrá komunikace s programem Microsoft Excel, jenž je často využíván k nejrůznějšímu zpracování dat do podoby

výkazů. Na třídění, výpočet a analýzu dat je využíván nástroj programu Microsoft Excel kontingenční tabulka.

### 3.2.2 Informační docházkový systém

Společnost PHARMIX, s.r.o. využívá kromě hlavního ERP systému Inuvio také integrovaný modul informačního systému Kompas2. Systém je zaměřený na elektronickou evidenci docházky zaměstnanců a umožňuje kontrolovat jejich pracovní dobu. K fungování využívá bezkontaktní identifikační systém, který je nainstalovaný u vchodu do budovy podniku, aby pracovníci snadno zaevidovali příchod nebo odchod. Velkou výhodou modulu je kompletní provázanost s ERP systémem Inuvio, ve kterém jsou následně zpracovávány mzdy zaměstnanců.

Vize do budoucna je taková, že by byl docházkový systém propojen s nástrojem pro plánování. Jakmile by zaměstnanec zaevidoval odchod z podniku, tak by se automaticky pozastavily veškeré jeho rozpracované úkony. Provázanost systémů by tedy zajistila, že by při odchodu zaměstnance rozpracované operace dále nepokračovaly a nečerpaly tak výrobní kapacity určené pro daný vyráběný dílec.

### 3.3 Vykazování splněné práce

Dělníci k vykazování své splněné práce využívají tři rozmístěné ethernet (ETH) terminály s vestavěnou čtečkou. Jakmile dělník vykoná operaci, je povinen si ji zaevidovat pomocí čárového kódu, který je vždy přiřazen každé operaci na tištěném technologickém postupu. Ten prochází výrobou spolu s dílcem označeným stejným pořadovým číslem jako je na technologickém postupu pro snadnější identifikaci a zamezení možné záměny. Po zaznamenání operace na čtečce se informace o provedené operaci přenesou do ERP systému, kde je následně operace vyhodnocena jako splněná.

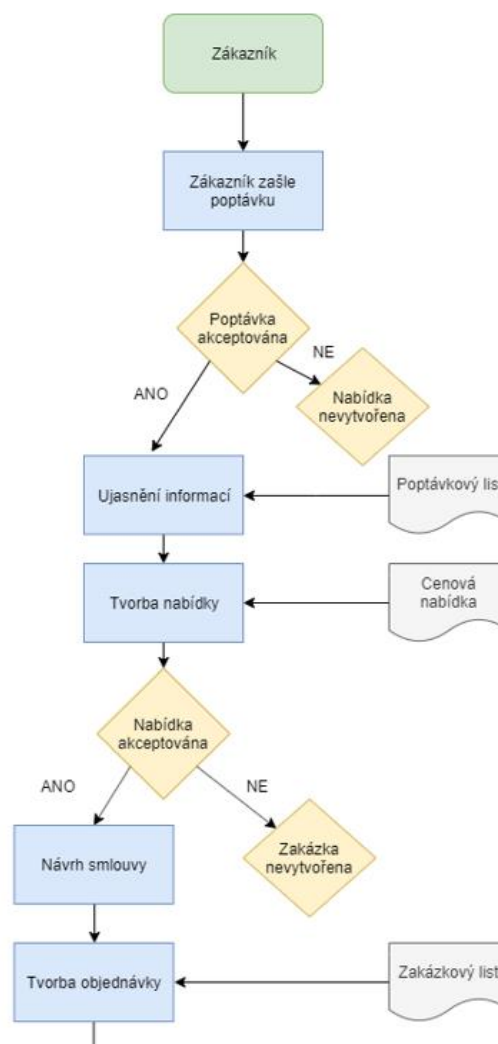
Stávající princip vykazování vyhotovené práce ve společnosti PHARMIX, s.r.o. je zastaralý a zaznamenané operace mají nedostatečné množství dat informující o stavu výroby. Ještě před zaváděním nového nástroje na plánování je zapotřebí pozměnit současný způsob vykazování splněné práce na zmodernizovanou formu, díky které by se do ERP systému dostalo více informací. Jednalo by se např. o informace o času začátku a konce operace, evidenci prostojů, evidenci poruch stroje a evidenci úklidu pracovišť včetně strojů. Všechny nově získaná data by sloužila ke zpřesnění návrhu potřebného času na vyhotovení operací v technologických postupech. Tyto časy by následně využíval nástroj pro plánování jako vstupní data.

### 3.4 průběh zakázky

V aktuální kapitole budou demonstrovány jednotlivé kroky postupu zakázky celým procesem od poptávky zákazníka až po převzetí dokončeného výrobku zákazníkem. Předem stanovený proces průběhu zakázky je důležitý pro zkvalitnění kapacitního plánování. Díky detailnímu rozboru kroků průběhu zakázky je možné zdokonalovat postupné dílčí kroky procesu.

Vývoj průběhu zakázky je rozdělen do čtyř hlavních částí složených z několika striktně dodržovaných kroků, které zajišťují vždy určité úseky společnosti.

- První částí je průběh zakázky úsekem obchodu:



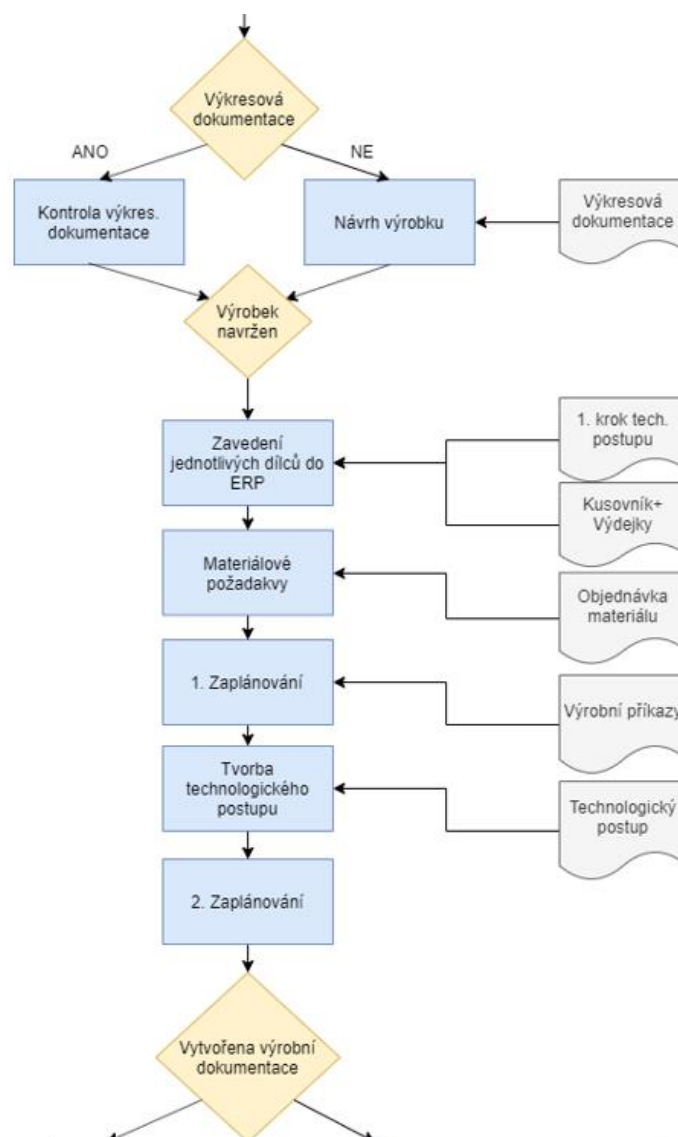
Obrázek 3.3 Průběh zakázky – úsek obchodu

Počáteční událostí ve výrobním řetězci je zaslání poptávky zákazníkem do společnosti s akceptovatelnými požadavky k výrobě. Jednotlivé požadavky musí projít schválením a dochází k ujasňování informací mezi společností a zákazníkem s následnou tvorbou nabídky. Pokud zákazník nabídku akce-

ptuje, je vystaven návrh smlouvy s následnou objednávkou. Všechny zmíněné kroky se uskutečňují v obchodním a ekonomickém úseku společnosti, který je v kontaktu s technologií a konstruktéry.

Již při návrhu smlouvy dochází ke stanovení předběžného termínu možného dokončení výroby zakázky a odhadu množství hodin, které bude zakázka z fondu dostupných pracovních hodin čerpat. V tomto kroku by měl být již využit nástroj pro plánování, díky kterému by předběžně zpracovaná zakázka byla vložena do aktuálního stavu výroby. Po vložení zakázky do aktuálního plánu výroby by bylo možné vyhodnotit změny ve vytiženosti jednotlivých pracovišť a podle toho následně upravit odhadnutý termín předběžného dokončení.

- Další částí je průběh zakázky úsekem TPV:

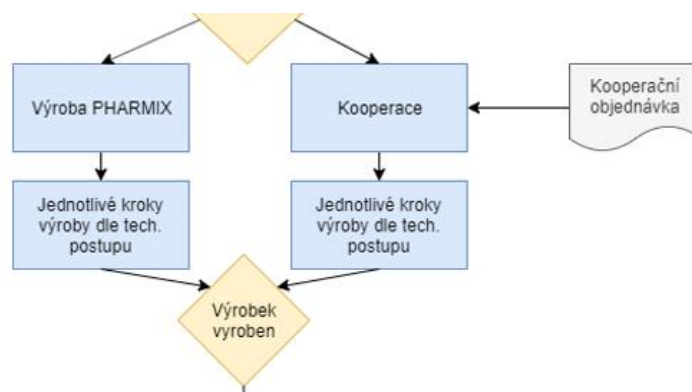


Obrázek 3.4 Průběh zakázky – úsek TPV

Zde dochází k návrhu výrobku s následnou tvorbou finální výkresové dokumentace. Pokud zákazník dodá vlastní výkresovou dokumentaci, dochází pouze ke kontrole, popřípadě přepracování dokumentace. Jakmile je výrobek navržen, dochází k důležité části zavedení jednotlivých dílců do ERP. Po zavedení se stanoví materiálové požadavky, vytvoří se technologické postupy (tvorba kusovníku, určení stromovité struktury dílců a sestav včetně stanovení mezd za jednotlivé operace) a dojde k finálnímu zaplánování zakázky do plánování.

Do tohoto okamžiku byla zakázka zaplánována jen předběžně bez bližších specifikací. Po vytvoření finální výkresové dokumentace jsou informace o zakázce přeneseny z ERP systému do plánovacího nástroje, kde budou data využita jako vstupní data. Zakázka je tedy kompletně zaplánována úsekem TPV do plánovacího systému výroby, podle kterého bude následně vyráběna ve výrobním úseku.

- Ve třetí části průběh zakázky prochází úsekem výroby:

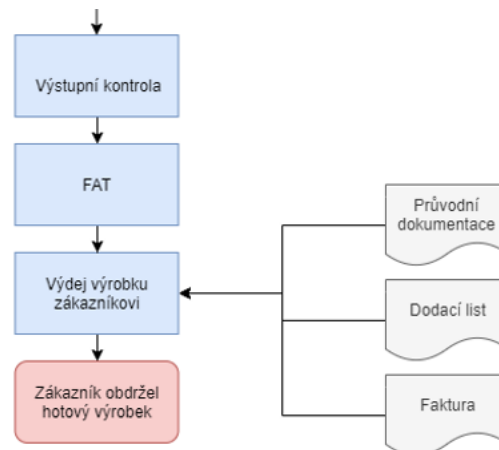


*Obrázek 3.5 Průběh zakázky – úsek výroby*

Po vytvoření výrobní dokumentace v předchozí fázi se zakázka přesouvá do výrobního úseku, kde začíná samotná výroba pod dohledem vedoucích pracovníků, kteří dohlížejí na dodržování termínů a správnost postupu výroby. Dochází k čerpání pracovních kapacit dělníků a strojů. Pokud je výroba natolik vytížená, že je předem v ohrožení smluvený termín dodání zakázky, může dojít k dokončení výroby v kooperaci.

Seřazení a pořadí operací ve výrobě bude probíhat pomocí nově vytvořeného nástroje na kapacitní plánování, který zajišťuje uspořádání úloh s minimálními časovými prodlevami. Úkolem výrobního úseku bude dodržování posloupnosti operací a případné řešení nečekaných problémů, jež mohou během výroby nastat a je potřeba na ně okamžitě reagovat.

- Posledním oddílem průběhu zakázky je úsek řízení kvality:



**Obrázek 3.6** Průběh zakázky – úsek řízení kvality

V konečné fázi průběhu zakázky je nutná výstupní kontrola, o kterou se stará úsek řízení kvality. Pokud výrobek splňuje všechny normy a výstupní kontrola proběhne v pořádku, je výrobek považován za dokončený. Jsou vystaveny potřebné dokumenty (průvodní dokumentace, dodací list a faktura) k následnému výdeji výrobku zákazníkovi. Po přijetí výrobku zákazníkem se zakázka uzavírá a je považována za vyfakturovanou.

Děni v posledním úseku průběhu zakázky nebude součástí kapacitního plánování. Odpracované hodiny pracovníků úseku řízení kvality nejsou přímo čerpané na zakázku, proto v plánování nefigurují.

### 3.5 Původní proces plánování výroby

V podniku neexistoval žádný sofistikovaný plánovací systém ani software a jednalo se pouze o plánování vycházející ze zkušeností vedoucích pracovníků výroby ve spolupráci s mistry. Důraz byl kladen na rozbor již dokončených zakázek formou jejich vzájemné podobnosti. Jelikož se jedná o zakázkovou výrobu, která se snaží vyhovět veškerým požadavkům zákazníka, jsou vyráběné stroje vždy minimálně malými úpravami odlišné. Právě díky jejich odlišnosti docházelo k odchylkám při určování potřebných kapacit na výrobu zakázky.

Nepřesnosti se v plánování objevují také ve správnosti dat. K jejich shromažďování je využíván ERP systém. Přímou v systému se s daty pracuje pomocí jeho nabízených funkcí a pokud jsou na některé činnosti nabízené funkce nedostatečné, jsou data vyexportována do sešitu Microsoft Excel, kde se následně zpracují pro vhodnou implementaci požadované funkce. Operace jsou navrhovány a evidovány v ERP systému v netradiční jednotce, kterou je česká koruna. Nejedná se tedy o standardní normu času v jednotkách sekund, minut nebo hodin. Kvůli datům uváděným v jednotkách měny dochází k odchylkám při posuzování podobnosti s předchozími dokončenými zakázkami. V každém roce se totiž platová situace na jednotlivých pracovištích společnosti mění se zvyšující se mírou růstu mezd. Poté jsou pak

výsledné kalkulace s klesajícím rokem výroby zakázky nižší než novější a musí se využít přepočtových koeficientů, při jejichž využívání dochází k odchýlkám.

Nejenom špatně nastavená jednotka vyjadřování odpracované práce v ERP systému se podílí na zkreslenosti dat. Celková hodnota čerpání kapacit na zakázce se počítá z dat, které stanovil technolog při tvorbě technologické dokumentace, nikoli z finálních dat po ukončení výroby doplněných o doplatky. Pokud stanovené peníze nejsou adekvátní náročnosti operace (může to být zapříčiněno výskytem špatně odhadovaných událostí, jako např. nedostatečná kvalita materiálu, nevhodně stanovený technologický postup, tvorba vadných kusů a poškození povrchů při montáži), tak je operace doplněna o doplatek k dorovnání adekvátní částky za operaci. K takovému kroku je nutné schválení nároku na doplatek mistrem. Doplatky mají za následek rozdíl mezi plánovanou cenou zakázky a reálnou cenou po dokončení výroby.

### 3.6 Návrh na zkvalitnění plánování

Ke zkvalitnění plánování by byl zapotřebí speciální software kapacitního plánování vytvořený na míru, který by sloužil k minimalizaci časových prodlev s cílem dokončit zakázky v nejkratším možném termínu s optimálním využitím kapacit pracovišť. Výsledkem by byl přesný harmonogram výroby po jednotlivých operacích, díky kterému by byla vhodně vytížena výroba na pracovištích. Společnost by získala přehled nad výrobou s reálnými předpoklady dokončení výroby zakázek při aktuálním stavu výroby. Software by měl také využít v obchodním úseku při stanovování předběžných termínů dokončení zakázek. K dosažení přesných výsledků pomocí plánovacího softwaru by byly nutné změny ve způsobu získávání a zpracování dat.

Jednou ze změn je vyjádření práce formou času (jednotka je minuta). Časové jednotky jsou oproti jednotkám mzdy konstantní a v průběhu časového horizontu neměnné. Pokud systém bude pracovat v jednotkách času, je potom možné změnit formu vykazování operací zaměstnanci, která neobsahuje dostatečné množství dat plynoucí z výroby. K přesnosti dat by vedlo zavedení „start-stop“ systému, kdy pracovník musí před začátkem práce na operaci svůj start zaevidovat pomocí ETH terminálu se čtečkou čárových kódů. Tímto zaznamenáním se zaeviduje start operace a po ukončení operace by stejným způsobem operaci ukončil a zaevidoval tak konec operace. V systému by byl vytvořen nový modul k porovnávání reálných časů z výroby s časy stanovenými technologem. Pomocí odchylek by byl postupně optimalizován potřebný čas na vykonání operace, a tím by byly minimalizovány potřebné doplatky. Při následující tvorbě technologického postupu zakázek by technolog data využíval a odchylka by se postupně minimalizovala. Plánovací software by využil již optimalizovaná data, která by se více blížila realitě. Celý proces by tak byl tímto způsobem zpřesněn.

K zavedení „start-stop“ systému by bylo nutné dokoupit větší množství ETH terminálů se čtečkami čárových kódů a strategicky je rozmístit po podniku tak, aby byly v blízkosti všech pracovišť pro zaznamenávání operací. Personál by byl proškolen na novou formu vykazování práce a celý proces by byl postupně zaveden po jednotlivých pracovištích v testovacím provozu, aby se odhalily případné problémy.

## 4 Metody řešení

Ve společnosti PHARMIX, s.r.o. probíhá zakázková výroba specifických kusů výrobků z nerez. Jedná se převážně o zakázky setrvávající ve výrobě několik měsíců a jejich výroba vyžaduje stovky až tisíce hodin práce. Takový charakter výroby je z hlediska plánování složitější než běžná sériová výroba.

Kvůli nepravidelnému přijímání jednotlivých zakázek v průběhu roku je výroba špatně předvídatelná, proto je nutné pracovat s dynamickým plánováním, jež se může maximálně přiblížit reálné situaci výroby v podniku.

Na výrobu může mít také vliv stochastické prostředí. Jedná se o neustále měnící se prostředí, které může být ovlivněno mnoha událostmi o různé míře pravděpodobnosti výskytu. Jedná se např. o již zmíněné náhodné přicházení zakázek do výroby, pozastavení výroby, snížení pracovních kapacit, ekonomické problémy ve světě, které mohou ovlivnit dodávky materiálu a odesílání již vyhotovených produktů a spousta dalších.

K navržení vhodné metody řešení problematiky, týkající se kapacitního plánování, je využít matematický model sloužící k popisu výrobního procesu ve společnosti PHARMIX, s.r.o. Při jeho použití je brán ohled na charakter výroby v podniku. Matematický model bude využit při volbě algoritmu řešení s vhodným řešitelem a bude také tvořit základ pro následující tvorbu scriptu sloužícího k optimalizaci kapacitního plánování. Dříve, než se přejde k samotné tvorbě scriptu, musí dojít ke sběru a zpracování dat, které budou sloužit jako vstupy pro budoucí nástroj pro plánování.

### 4.1 Matematický model

Klasický problém plánování ve výrobě je složen ze sady nezávislých úloh (výrob zařízení). Každá úloha má uspořádanou sadu operací, z nichž každá operace má předem definovaný stroj, na kterém bude vyhotovena a pevné pořadí v sadě operací. Cílem je zajistit posloupnost operací na strojích tak, aby byla minimalizována doba na dokončení všech úloh [5]. K formulaci problému plánování ve výrobě slouží optimalizační modely [6].

Matematický model plánování výroby ve společnosti PHARMIX, s.r.o. je lineární a vychází z obecného matematického modelu, který je tvořen z účelové funkce, rozhodovacích proměnných, parametrů a omezení. Obecný matematický model pro úlohu plánování je popsán v bakalářské práci [7]. Obecný matematický model je nutné upravit tak, aby jeho části co nejlépe popisovaly skutečnou výrobu ve společnosti.

Upravený matematický model pro společnost PHARMIX, s.r.o. je tvořen z následujících čtyř částí:

- Účelová funkce

Vyjadřuje se jako matematická funkce rozhodovacích proměnných [8]. Účelová funkce má za cíl minimalizaci časových prodlev mezi jednotlivými operacemi

za účelem snížit celkový potřebný čas na výrobu všech aktivních zakázek. Díky ušetřeným výrobním kapacitám budou nižší celkové náklady na výrobu zakázky.

Účelová funkce  $f$  je definovaná vztahem v lineárním tvaru:

$$\min z = f(t_{i,j}) \quad (1)$$

$z$  ... celková doba časových prodlev

$t_{i,j}$  ... rozhodovací proměnná

- Rozhodovací proměnné

Jedná se o  $n$  (množinu přirozených čísel) číselných proměnných, které ovlivňují výsledné řešení. Rozhodovací proměnná se zapisuje ve tvaru  $t_{i,j}$ , kde písmeno  $t$  vyjadřuje počáteční čas operace a indexy  $(i, j)$  vyjadřují pořadí dané operace v zakázce.

$t_{i,j}$  ... rozhodovací proměnná

$i$  ... index zakázky

$j$  ... index pořadí operace v posloupnosti operací zakázky

Vysvětlení zápisu je demonstrováno na zápisu příkladné operace označené indexy (1, 2). Index s hodnotou 1 značí číslo zakázky (zakázka číslo 1) a index s hodnotou 2 vyjadřuje pořadí operace zakázky (druhá operace první zakázky).

- Parametry

Parametry číselně popisují operace zakázek. U obecného modelu jsou parametry označovány dvojicí čísel. Zde muselo dojít ke změně na označení pomocí tří hodnot  $(m, p, s)$ , kde písmeno  $m$  definuje pracoviště či stroj, na kterém bude operace vyhotovena. V některých případech se nemusí jednat přímo o pracoviště nebo stroj, ale o profesi pracovníka, jenž bude operaci vykonávat. Tento případ nastává u pracovišť, kde pracovníci neobsluhují stroj (např. mechanici, svářeči nebo brusiči). Písmeno  $p$  vyjadřuje čas potřebný na vyhotovení operace. Třetí pozice označená písmenem  $s$  znázorňuje místo ve stromové struktuře zakázky, které definuje posloupnost operací (např. nelze provést určitou operaci, pokud nebyly dokončeny jiné). Pomocí číselného označení je možné definovat omezení posloupnosti operací na zakázce, které bude popsáno v následující odrážce s názvem omezení.

Příklad označení fiktivních dvou operací zakázky číslo 1:

$$\text{zakázka 1} = [( \text{pracoviště frézy}, 3, ( ) ), ( \text{mechanik}, 2, (0) )] \quad (2)$$

Zápis popisuje dvě po sobě jdoucí operace zakázky číslo 1. Číselné označení operací se začíná počítat od 0 (0, 1, 2, 3...). Nultá operace bude vyhotovena na pracovišti frézy za dobu 3 hodin. Prázdná kulatá závorka značí, že

před danou operací není žádná jiná operace, na kterou by navazovala. První operaci bude vykonávat 2 hodiny pracovník z oddělení mechaniků. 0 v kulaté závorce značí, že práce na dané operaci může začít teprve až je vyhotovena nultá operace.

- Omezení

Znázorňují množinu podmínek, které musí vyhovovat výslednému řešení. Omezující podmínky jsou vyjadřovány pomocí rovnic nebo také často pomocí nerovnic [12].

Proces plánování budou ovlivňovat následující omezující podmínky:

- Omezující podmínka přednosti

Vyjadřuje, že na jednom pracovišti je dvojice po sobě jdoucích operací vykonávána tak, že první operace musí být splněna, než začne práce na druhé operaci. Podmínka je platná jen v rámci výroby dílce nebo sestavy, jinak mohou být operace na různých dílcích či sestavách zakázky vyráběny paralelně. Matematicky je možné provést zápis pomocí nerovnice, která je demonstrována na následujícím příkladu:

$$t_{0,1} + 3 \leq t_{0,2} \quad (3)$$

Nerovnice je zapsána pomocí rozhodovacích proměnných a délky trvání první z dvojice po sobě jdoucích operací na stejné zakázce. Operace označená (0, 1) vyjadřuje první operaci zakázky číslo 0 o délce trvání 3 hodiny. Následující operace (0, 2) znázorňuje druhou operaci na zakázce číslo 0. Druhá operace tedy musí začít minimálně za 3 hodiny po začátku první operace.

- Omezující podmínka překrývání

Podmínka formuluje, že stroj nebo pracovník nemůže pracovat na více než jedné operaci v jeden okamžik. Matematicky se omezující podmínka může zapsat pomocí nerovnice, která je demonstrována na následujícím příkladu:

$$t_{0,1} + 3 \leq t_{2,2} \quad (4)$$

Zápis vyjadřuje dvě operace dvou zakázek, které jsou vyráběny na pracovišti frézky. Jedná se o první operaci zakázky číslo 0 označené indexy (0, 1) s časem výroby 3 hodiny a o druhou operaci zakázky číslo 2 označené indexy (2, 2) s časem výroby 4 hodiny. Z hlediska plánování je výhodnější vyhotovit jako první operaci (0, 1) a proto může na pracovišti frézky začít následující operace s indexy (2, 2) minimálně za 3 hodiny.

- Omezující podmínka přerušení

Jakmile je zakázka zaplánována do výrobního plánu, musí její operace probíhat bez přerušení. Operaci nelze přerušit během jejího vykonávání z důvodů upřednostnění jiné operace.

- Omezující podmínka posloupnosti operací

Podmínka definuje výrobní strukturu zakázky včetně jednotlivých návazností operací ve výrobě dílců a kontinuity jednotlivých dílců, které tvoří sestavy. Zachování stupňovité (stromovité) struktury je zcela nezbytné pro následující tvorbu kapacitního plánování a jedná se o stěžejní omezující podmínku, na kterou budou navazovat předchozí omezující podmínky přednosti a překrývání.

Pořadí a návaznost operace určuje třetí člen v označení parametru. Člen číselně udává, na jakou operaci daná operace navazuje a je zapisován do kulaté závorky. Pokud je místo členu prázdná závorka, znamená to, že daná operace nezávisí na žádné předchozí operaci. Pokud je v závorce více čísel, tak operace závisí na více předchozích operacích. Zpravidla se v tomto případě jedná o sestavu, jež je tvořena více dílci nebo podsestavami.

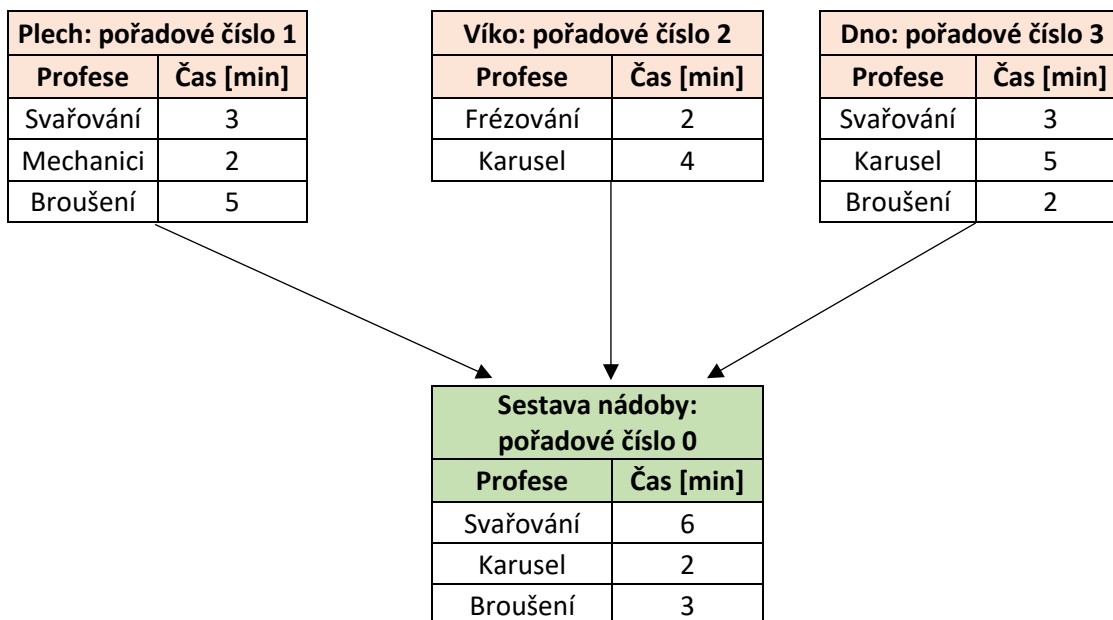
K vysvětlení implementace podmínek omezení bude využita zjednodušená vzorová zakázka nádoby tvořená ze třech dílců (plech, víko a dno) tvořící sestavu (nádobu – finální produkt). U výroby dílců spolu se sestavou bude platit omezující podmínka přednosti. Zároveň ale může výroba jednotlivých dílců probíhat paralelně. Teprve až budou vyhotoveny kompletně všechny dílce, může začít první operace sestavy. Strukturu návaznosti operací zajišťuje omezující podmínka posloupnosti operací. V celé zakázce platí omezující podmínka přerušení a překrývání.

Zakázka nádoby je tvořena 11 operacemi a zápis pomocí parametrů je následující:

$$\text{zakázka nádoby} = \left[ \begin{array}{l} (\text{svařování}, 3, ()), (\text{mechanici}, 2, (0)), (\text{broušení}, 5, (1)), \\ (\text{frézování}, 2, ()), (\text{karusel}, 4, (3)), \\ (\text{svařování}, 3, ()), (\text{karusel}, 5, (5)), (\text{broušení}, 2, (6)) \\ (\text{svařování}, 6, (2,4,7)), (\text{karusel}, 2, (8)), (\text{broušení}, 3, (9)) \end{array} \right] \quad (5)$$

Grafický zápis stromovité struktury zakázky s využitím programu Microsoft Excel je zobrazen v tabulce č. 4.1.

Tabulka 4.1 Zjednodušená zakázka výroby nádoby



## 4.2 Volba algoritmu řešení

Po podrobné analýze všech podnikových činností a definování matematického modelu společnosti se pokračuje k volbě algoritmu řešení, který vymezuje postup, jímž lze úlohu vyřešit.

Jelikož společnost pracuje s rozsáhlými zakázkami složenými z velkého množství operací (u velkých zakázek jsou to stovky operací), nesmí mít algoritmus problém pracovat s velkým množstvím dat. Proces výroby zakázky je ovlivněn také omezeními, které jsou popsány v matematickém modelu, jenž je dopodrobna rozebrán v předešlé kapitole č. 4.1 a je nutné je dodržet. Algoritmus musí být schopen naplánovat pořadí operací na pracovištích s požadovanými omezeními tak, aby délka plánu byla minimální a všechny úkoly byly dokončeny.

Požadavky na algoritmus:

- Práce s velkým množstvím vstupních dat
- Rychlý výpočetní čas
- Zavedení omezení

Vhodných voleb řešitele nebo jejich kombinací se jeví hned několik. Všechny mohou být využity pro dané požadavky na algoritmus. Úlohu plánování lze řešit jako úlohu

smíšeného celočíselného lineárního programování (Mixed-integer Linear Programming, MILP), kdy se hledá řešení pomocí metod jako metoda větví a mezí nebo speciálně navržených heuristických algoritmů. MILP pracuje s lineárním modelem obsahující alespoň jednu celočíselnou proměnnou, jež je zpravidla binární. Řešení MILP úlohy je často značně složité [9]. Další využitelnou možností je formulovat problém jako programování s omezeními (Constraint Programming, CP). CP k řešení využívá dynamických algoritmů [10]. K hledání řešení úlohy plánování lze také využít tzv. SAT řešitele (z anglického Boolean Satisfiability Solver), který využívá Booleovy algebry. Logické proměnné Booleovy algebry nabývají pouze dvou hodnot, zpravidla 1 a 0 (popřípadě ano nebo ne). SAT řešitel si díky tomu dokáže snadno poradit s velkým množstvím dat v relativně rychlém čase [11].

K řešení plánování ve společnosti PHARMIX, s.r.o. se využije kombinace CP a SAT řešitele. Vzniká tedy hybridní řešitel CP-SAT, který kombinuje silné stránky a vyhýbá se slabým stránkám obou přístupů. Jeho dominantou je rychlé a velmi efektivní řešení rozsáhlých úloh s využitím zavedených omezujících podmínek. CP-SAT využívá řešič tzv. generování líné klauzule (Lazy Clause Generation, LCG) nad řešičem SAT. Jádro řešení pomocí LCG je jednoduché. Využívá řešitele, jenž je založený na porozumění akcí, které se vztahují na základní sadu proměnných představující celočíselné proměnné CP modelu [12]. Řešitel se využívá k řešení problémů, jako jsou např. plánování výroby nebo plánování práce pro zaměstnance a mezi jeho hlavní dominanty patří vysoká úroveň modelování, účinné vyhýbání se opakování stejného dílčího vyhledávání, silné autonomní vyhledávání a programovatelnost vyhledávání [13]. Bližší informace o fungování jsou zpracované v práci *Search is Dead, Long Live Proof* [13].

#### 4.2.1 OR-Tools

Pro řešení kapacitního plánování s využitím řešitele CP-SAT byla po srovnání výhod zvolena knihovna OR-Tools od společnosti Google, jež vyhrála v roce 2021 čtyři zlaté medaile v mezinárodní soutěži v programování s omezeními (MiniZinc Challenge). Rozšířená uživatelská komunita, velké množství materiálů k implementacím řešení, potvrzená funkčnost a spokojenost v praxi byly hlavními aspekty při výběru této OR-Tools knihovny. Další adekvátní možností bylo řešení od společnosti IBM s názvem CP Optimizer, které stejně jako OR-Tools disponuje širokou škálou využití, nejmodernějšími specializovanými algoritmy a podporou nejčastějších programovacích jazyků. OR-Tools převyšuje CP Optimizer v množství a zpracování materiálů, což byl rozhodující faktor pro výběr právě OR-Tools knihovny.

Otevřený zdrojový kód softwarové sady OR-Tools slouží k optimalizaci nejnáročnějších problémů v jazycích C++, Python, C# nebo Java. Sada dokáže řešit problémy optimalizace omezení (programování s omezením), lineární programování, problémy se směřováním vozidel, plánování, balení a další. K řešení problémů OR-Tools využívá několik řešitelů jako např. Gurobi nebo CPLEX, otevřené zdrojové řešení jako SCIP, GLPK, Google GLOP a často využívaný CP-SAT. S jejich pomocí dokáže najít optimální řešení v případě obrovského množství řešení, které by počítač nebyl schopen prohledat [14].

## 4.3 Sběr dat

Další fází před tvorbou nástroje pro plánování je sběr dat. K implementaci zvoleného hybridního CP-SAT řešitele jsou zapotřebí kvalitně připravená data, jež budou zároveň vstupy do scriptu budoucího nástroje pro plánování. Ve snaze získat optimální řešení plánování kapacit ve výrobě musí být data popisující výrobu co nejpřesnější, aby bylo vůbec možné takového výsledku dosáhnout. Zpřesnění délek časů potřebných k vyhotovení operací lze docílit zmodernizováním původního systému sběru dat a následným zautomatizováním vyhodnocování dat z výroby. Nově získaná data tak budou mít dostatečnou vypovídající hodnotu pro následné zpracování.

### 4.3.1 Původní systém sběru dat

Ve společnosti existovaly dva způsoby získávání dat z výroby, a to evidování splněných operací pomocí čteček čárových kódů přiřazených každé operaci na technologickém postupu, nebo pomocí zapisování mzdových listů zaměstnanci. Mistři vyhotovili na konci měsíce mzdu pro všechny zaměstnance tak, že zpracovali vykázané operace zaměstnance z ERP systému. Pokud nějaké operace chyběly, bylo nutné jen poté doplnit mistrem na základě zápisu mzdového listu vedeného zaměstnancem. Ten sloužil jako kontrola odpracovaných hodin zaznamenaných v ERP systému.

Záznam vyhotovené operace v ERP systému nesl následující data:

- Datum a čas vykázaní operace
- Osobní číslo zaměstnance
- Pracoviště, na kterém byla operace provedena
- Počet vyhotovených bezchybných kusů
- Počet vyhotovených vadných kusů

Žádné další informace evidence operací neobsahovala, tudíž nebylo známo, jak dlouho se operace reálně vykonávala. Technolog nezískával potřebnou zpětnou vazbu o jeho vhodném navržení potřebného času na vyhotovení operace. Nedocházelo proto ke zdokonalování technologických postupů a optimalizaci potřebných časů na vyhotovení různých operací. Data byla v některých případech nepřesná a docházelo k odchýlkám od reálného stavu při výrobě.

Na obrázku č. 4.1 je vyobrazeno prostředí ERP systému se zaznamenanými operacemi. Zelená kolečka ve sloupci „Splněno1“ znázorňují splněné operace a červená kolečka ještě nesplněné operace.

Typ	Číslo zakázky	Pořad... /	Název 1	Operace	Název	Splněno1
▶ Jednicová oper...	19/215	1	Basket machined-sest...	10	Karusel	✓
Jednicová oper...	19/215	1	Basket machined-sest...	20	Mechanici	✓
Jednicová oper...	19/215	1	Basket machined-sest...	30	Svařovna	✓
Jednicová oper...	19/215	1	Basket machined-sest...	40	Karusel	✗
Jednicová oper...	19/215	1	Basket machined-sest...	50	FrH	✗
Jednicová oper...	19/215	1	Basket machined-sest...	60	FrH	✗

Obrázek 4.1 Původní zobrazení zaznamenaných operací v ERP systému

### 4.3.2 Zmodernizovaný systém sběru dat

Změna oproti předchozímu systému sběru dat spočívá v evidenci operací zaměstnanci, kteří již nezaznamenávají jen vyhotovenou operaci, ale začínají evidovat začátek práce na operaci, ukončení operace a popřípadě prostoje. Jedná se o již zmíněný „start-stop“ systém. Při delších operacích trvajících více směn je nutné operaci na konci směny pozastavit (zahájit prostoj) a následující den před pokračováním v práci znovu spustit (ukončit prostoj). Díky novému systému sběru dat je možné monitorovat přesný čas potřebný na vyhotovení operace, který je následně srovnán s navrženým časem od technologa. K vyhodnocování slouží tzv. semafor, který je navržen tak, aby pomocí barev (zelená – v termínu  $\pm 2\%$  odchylka od stanoveného termínu, oranžová – odchylka od termínu 2-5 %, červená – odchylka od termínu více než 5 %) vyhodnotil splnění navrženého času. Pokud semafor signalizuje červenou barvu, je nutné odůvodnit zaměstnancem výchylku od navrženého času mistrovi nebo vedoucímu pracovníkovi, který je povinen důvod zpoždění zaznamenat do předem vytvořené šablony v ERP systému pomocí číselníku odchylek, jenž udává důvody, proč došlo k odchylce oproti plánovanému času. Mezi důvody zpoždění operace obsažené v číselníku patří:

- Pracovník špatně zaevidoval operaci
- Neodůvodněné nedodržení časové normy
- Chyba ve výkresové dokumentaci
- Chyba v technologickém postupu
- Technická závada stroje či nástroje na vyhotovení operace
- Zvýšená náročnost na zpracování oproti odhadu technologa

Odhalené a odůvodněné odchylky by mohly vést k objevení slabých míst v plánování výroby, které je nutné eliminovat ještě před přechodem na nový způsob plánování výroby s využitím nástroje na plánování. Technolog bude moci vycházet při sestavování nových technologických postupů z předešlých dat porovnaných s reálnou výrobou. Díky tomu by mělo postupem času docházet k minimalizaci časových odchylek plánu od reality.

Na obrázku č. 4.2 je zobrazeno zmodernizované prostředí ERP systému s vyhodnocováním operací pomocí semaforu. Ve sloupci „TAC strojní“ je navržený čas na vykonání operace od technologa a skutečný odpracovaný čas je ve vedlejším sloupci s názvem „Skutečný odpracovaný čas“. Barva ve sloupci „Semafor“ je

určována podle již okomentovaného principu založeném na procentuální odchylce rozdílu skutečného času a času navrženého technologem.

	Datum pořízení	Číslo zaká...	Pořadové číslo	Operace	Název	Název 1	TAC strojní (min)	Skutečně odprac. č...	Semafor
▶	08.10.2021 20:34:53	21/049	29	70	Karusel	Flange	320	373,616667	●
□	08.10.2021 13:21:16	21/049	24	70	Karusel	Ring	395	438,133333	●
□	20.07.2021 10:05:08	21/049	101	30	Karusel	Plate	1313	665,9	●
□	20.07.2021 9:59:33	21/049	101	30	Karusel	Plate	1313	175,616667	●

Obrázek 4.2 Aktuální zobrazení zaznamenaných operací v ERP systému

Při vyhodnocování dat lze nejen porovnávat časy potřebné na výrobu operace, ale také je možné porovnávat např. výkonnost jednotlivých pracovišť či zaměstnanců, výnosnost zakázek a také porovnávat očekávané plány s reálnými daty přímo z výroby.

Před zavedením zmodernizovaného systému evidence byly ve společnosti jen 3 ETH terminály sloužící k evidenci operací. Při zavádění systému „start-stop“ bylo nutné rozšířit síť ETH terminálů, aby zaměstnanci měli vždy terminál u pracoviště a nemuseli ztrácet čas docházením na vzdálenější místa. Množství ETH terminálů bylo zvýšeno na 11 kusů a byly strategicky rozmístěny v halách společnosti. Ovládací menu ETH terminálu bylo vytvořeno dle požadavků dodavatelem. Vzhled nově nakoupených ETH terminálů pro projekt zavedení kapacitního plánování je na obrázku č. 4.3.



Obrázek 4.3 ETH terminál sloužící k evidenci operací

### 4.3.3 Chyby sběru dat

Pro následující zpracování dat je podstatné jejich správné zaznamenávání. Chyby by mohly vést ke značným odchylkám v plánování výroby, proto je důležitá neustálá kontrola dat jak nadřízenými pracovníky, tak pomocí vyhodnocování ERP systémem. Pokud dojde ke zjištění chyby, je nezbytné na chyby reagovat, zjistit příčinu vzniku chyby a odstranit ji, aby se již neopakovala. Nejpravděpodobnější chyby, které mohou během evidence ve výrobě nastat, jsou:

- Nezaevidování či chybné zaevidování operace do ERP systému
- Zaevidování nesprávné operace
- Chyba v technologickém postupu, kvůli které není možné operaci zaevidovat
- Chybný čárový kód

K odhalení zmíněných chyb byl vytvořen kontrolní mechanismus, který chybu označí v ERP systému tak, že ve sloupci „Semafor“ je chybná operace označena modrou barvou. Při každodenní kontrole operací pak mistr jednoduše odhalí chybnou operaci a podle příčiny vzniku chyby následně chybu opraví. Při opakovaných výskytech chyb je nutné vymyslet potřebná opatření, díky kterým bude zajištěno, aby se chyby nadále neopakovaly.

Na obrázku č. 4.4 jsou znázorněné chybné operace, které nastaly během výroby na pracovišti karuselu. Mistr chyby následně okomentoval ve sloupci s názvem „Důvod zpoždění operace“. Chyby je možné vyfiltrovat v libovolném časovém úseku a určit jejich četnost výskytu.

	Datum pořízení	Číslo zakázky	Pořadové číslo	Operace	Název	TAC strojní (min)	Skutečně odprac. čas obsluhy	Semafor	Důvod zpoždění operace
<input type="checkbox"/>	17.11.2021 13:21:25	20/204	190	160	Karusel	15999	0	●	chyba v tech. postupu
<input type="checkbox"/>	17.11.2021 13:21:17	20/204	190	50	Karusel	130	0	●	nezaevidováno zaměstnancem
<input type="checkbox"/>	17.11.2021 13:21:09	20/204	190	120	Karusel	67	0	●	chybný čárový kód
<input type="checkbox"/>	16.11.2021 14:43:07	21/125	103	30	Karusel	368	592,15	●	
<input type="checkbox"/>	12.11.2021 17:44:05	21/134	227	20	Karusel	148	264,75	●	
<input type="checkbox"/>	11.11.2021 20:17:04	20/205	226	20	Karusel	395	219,416667	●	

Obrázek 4.4 Zobrazení chybně zaznamenaných operací v ERP systému

## 4.4 Zpracování dat

Po sběru dat z výroby je zapotřebí data zpracovat do podoby vhodné pro následnou implementaci do scriptu nástroje pro plánování. Výstupem po zpracování dat bude soubor programu Microsoft Excel pro každou zakázku zvlášť. Tyto soubory budou uloženy ve společné složce na firemním síťovém úložišti. Jednotlivé soubory zakázek budou následně automaticky importovány pomocí příkazu přímo ve scriptu.

Data jsou ve společnosti zpracovávána dvěma informačními systémy. Jedním je ERP systém Inuvio a druhým je docházkový modul informačního systému Kompas2, který slouží primárně pro evidenci docházky a tvorbu mezd zaměstnanců. Jeho dalším využitím je vyhodnocování celoroční docházky a průměrné nemocnosti. Z hlediska

výroby systém neukládá žádná důležitá data, proto v další části této kapitoly nebude uváděn.

Data sloužící k vyhodnocování výroby jsou získávána z ERP systému. Ke kvalitním týdenním, měsíčním nebo ročním reportům je potřebné velké množství kvalitních informací, jež musí být vhodně zpracovány do požadované formy. Ke zpracování dat je ve společnosti nejčastěji využíván program Microsoft Excel. Komunikace mezi ERP systémem a programem Microsoft Excel funguje bez problémů a je velmi rychlá. V programu Microsoft Excel se nejčastěji zpracovávají:

- Týdenní (měsíční a roční) reporty vykonané práce
- Protokoly vyhodnocené práce jednotlivých zaměstnanců a pracovišť
- Soubory potřebné pro plánování (tabulky, dokumenty se vstupními daty pro plánování a dokumenty se stromovitou strukturou zakázek)
- Tabulky vyhodnocující jednotlivé vyhotovené zakázky

#### 4.4.1 Rozdělení pracovišť

Ve společnosti PHARMIX, s.r.o. pracuje v roce 2022, kdy je zpracovávána diplomová práce, celkem 43 pracovníků provozu na 9 pracovních pozicích. Pracoviště jsou pojmenována podle strojů na pracovišti nebo podle profese. Každé pracoviště je také označeno zkratkou, která se objevuje při zápisech v technologických postupech a v ERP systému. Počty pracovníků se s nástupem nebo odchodem pracovníka pravidelně aktualizují.

Rozdělení po pracovnících je optimální řešení problému přípravy pracovišť. Jednotliví pracovníci vykonávají vlastní specializovanou práci, proto lze daný krok rozdělení pracovišť učinit. V řešení kapacitního plánování je potom možné zanedbat rozdílné pracovní fondy pracovišť (pracoviště v podniku mají různý počet zaměstnanců, a proto mají různé pracovní fondy), které by zásadně komplikovaly řešení.

Každý pracovník je zaměřen na práci dle své specializace. Práce je přiřazována pracovníkovi mistrem nebo plánovačem. Změna jeho práce může nastat jen ve výjimečných situacích, např. při nedostatku pracovních sil či dlouhodobé nemoci některého ze zaměstnanců. Bližší specializace jednotlivých pracovníků nebude na žádost společnosti PHARMIX, s.r.o. z důvodu důvěrnosti dat v tabulce č. 4.2 uvedena. Pracovníci mají v tabulce přiřazené označení jak zkratkou, tak i číselné, které bude využito v budoucím nástroji pro plánování výroby. Číselný formát je jednodušší při importu dat do programovacího jazyka Python, ve kterém bude probíhat tvorba scriptu.

Tabulka 4.2 Rozdělení pracovišť

Profese/pracoviště	Počet pracovníků	Specializace	Označení zkratkou	Označení číselné
Klempíř	1	Klempíř	kl	1
Frézka a obrážka	2	Frézka	fr1	2
		Obrážka	fr2	3
Mechanik	6	Vrtačka1	me1	4
		Vrtačka2	me2	5
		Mechanik1	me3	6
		Mechanik2	me4	7
		Lakovna1	me5	8
		Lakovna2	me6	9
Soustruh a strojní broušení	3	Soustruh1	sou1	10
		Soustruh2	sou2	11
		Soustruh3	sou3	12
Horizontka	4	Horizontka1	frh1	13
		Horizontka2	frh2	14
		Horizontka3	frh3	15
		Horizontka4	frh4	16
Karusel	4	Karusel1	kar1	17
		Karusel2	kar2	18
		Karusel3	kar3	19
		Karusel4	kar4	20
Svářeč	10	Svářeč1	sv1	21
		Svářeč2	sv2	22
		Svářeč3	sv3	23
		Svářeč4	sv4	24
		Svářeč5	sv5	25
		Svářeč6	sv6	26
		Svářeč7	sv7	27
		Svářeč8	sv8	28
		Svářeč9	sv9	29
		Svářeč10	sv10	30
Ruční brusič	12	Ruční brusič1	rb1	31
		Ruční brusič2	rb2	32
		Ruční brusič3	rb3	33
		Ruční brusič4	rb4	34
		Ruční brusič5	rb5	35
		Ruční brusič6	rb6	36
		Ruční brusič7	rb7	37
		Ruční brusič8	rb8	38
		Ruční brusič9	rb9	39
		Ruční brusič10	rb10	40
		Ruční brusič11	rb11	41
		Ruční brusič12	rb12	42
Balotina	1	Balotina	bal	43

#### 4.4.2 Zpracování dat pro kapacitní plánování

Data potřebná pro nástroj kapacitního plánování vycházejí z technologické dokumentace výroby dané zakázky. Požadovaná podoba vstupních parametrů, které popisují operace zakázky, je popsána v matematickém modelu a je vyjádřena v již zmíněném tvaru ( $m$  – pracoviště, kde bude operace vykonávána;  $p$  – potřebný čas na vyhotovení operace;  $s$  – místo ve stromové struktuře zakázky). K vyjádření operací v potřebném tvaru je nejprve nutné zakázku zaevidovat a následně zpracovat konstruktéry spolu s technologií. Pro snadnější upravování a přehlednost jsou data následně z ERP systému vyexportována do předem připravené šablony v programu Microsoft Excel. Pomocí šablony je možné v programu evidovat splněné operace, které procentuálně vyhodnocují stav výroby zakázky. Procentuální plnění zakázek je využíváno při tvorbě pravidelných souhrnů informujících o stavu výroby a je také poskytováno konečnému zákazníkovi v reportech stavu výroby.

Vstupní data pro řešení pomocí OR-Tools v Pythonu je nutné definovat pouze v celých číslech kvůli zvýšení výpočetní rychlosti řešiče CP-SAT. Data proto budou uváděna v jednotkách minut, stejně jako v ERP systému. K exportování dat do programu Microsoft Excel bylo nutné vytvořit speciální formulář od společnosti Asseco Solutions, díky kterému budou data v následujícím formátu připravená k následnému zpracování. Tabulka hodnot v programu Microsoft Excel je uvedena v tabulce č. 4.3. Zobrazení tabulky je demonstrováno na fiktivní zakázce číslo 1, která je složená ze dvou dílců (pořadové číslo 1 a 2), jež tvoří finální sestavu (pořadové číslo 3).

Tabulka 4.3 Zpracování fiktivní zakázky v programu Microsoft Excel

Zakázka 1			
Pořadové číslo	Stroj	Čas [min]	Umístění
1	1	3	
1	2	4	0
2	1	3	1
2	3	1	
2	1	2	3
3	2	3	2, 4

Zakázka je rozdělená podle operací, kdy každá operace je součástí dílce či sestavy označených pořadovým číslem. Operace jsou popsány již zmíněnými třemi parametry ( $m$  – sloupec stroj,  $p$  – sloupec čas,  $s$  – sloupec umístění). Pořadové číslo je v Microsoft Excelu uvedeno jen pro lepší přehlednost. Do Pythonu sloupec „Pořadové číslo“ nebude importován.

## 4.5 Implementace nástroje pro optimální plánování výroby

V předchozích podkapitolách byl postupně popsán matematický model, výběr algoritmu řešení, sběr a zpracování dat, tedy vše potřebné pro přípravu na implementaci optimalizačního nástroje, který bude v aktuální kapitole popsán.

Výběr vhodného programovacího jazyka je jedním z prvních kroků při tvorbě nástroje na kapacitní plánování. Programovací jazyk je prostředkem pro zápis algoritmů, které mohou být realizovány na počítači. Takový zápis algoritmu pomocí programovacího jazyka se jmenuje program.

V dnešní době existuje velké množství programovacích jazyků. Pro některé záležitosti jsou programovací jazyky vhodnější než pro jiné. Mezi nejčastěji využívané patří např. Java, C, Python, C++, C#, Visual Basic .NET a JavaScript. K řešení problémů souvisejících s kapacitním plánováním jsou nejčastěji využívány Python, C++, Java a C#. Všechny čtyři zmíněné programovací jazyky mají podporu knihovny OR-Tools od společnosti Google.

Jednou z možností výběru programovacího jazyka na vytvoření kapacitního plánování je již jmenovaný jazyk Java, který je objektově orientovaný a patří mezi nejpoužívanější jazyky. Jeho využití a množství materiálů na řešení požadované problematiky však není tak rozsáhlé jako u jazyka Python. K vytvoření nástroje pro kapacitní plánování je proto využít programovací jazyk Python, jemuž patří podle indexu popularity programovacích jazyků (Popularity of Programming Language, PYPL) pro rok 2021 první místo před Javou. Python je silný procedurální objektově orientovaný funkční jazyk vytvořený koncem 80. let 20. století. Jazyk se v současné době používá v různých odvětvích, jako např. při vývoji softwaru, vývoji webových aplikací, vývoji grafického uživatelského rozhraní pro stolní počítače, vzdělávání a vývoji vědeckých aplikací [15]. Volba Pythonu pro řešení je primárně podnícena jeho jednoduchostí a velkým množstvím knihoven rozšiřující programovací možnosti.

V následující části podkapitoly budou popsány hlavní části scriptu, kterým bude řešen problém kapacitního plánování ve společnosti PHARMIX, s.r.o. Kód vychází z návrhu na řešení tzv. „job shop“ problému naplánování úloh na strojích tak, aby se dosáhlo minimální délky plánu. Obecná podoba scriptu je uvedena v internetovém zdroji [14]. Popis metody tvorby kódu a řešení CP-SAT modelů je uveden v internetových zdrojích [16] a [17], ze kterých budou informace sloužící k tvorbě kódu čerpány. Ke zpracování řešení v jazyce Python bude využito vývojového prostředí PyCharm, které bude současně sloužit k prezentaci scriptu.

### 4.5.1 Příprava modulů a knihoven

Python využívá mnoho modulů a knihoven třetích stran, které napomáhají k řešení požadovaných úkolů. Knihovna představuje obsáhlou řadu služeb a funkcí. Je rozdělena na moduly, díky kterým zajišťuje požadované programovací potřeby. Modul je soubor kódu obsahující definice (např. funkce, třídy a proměnné) a příkazy. Některé moduly jsou přímo předdefinované a lze je využívat bez nutnosti importování pomocí

příkazu. Seskupení souvisejícího kódu do modulu usnadňuje pochopení a použití kódu. Díky tomu je kód logicky uspořádán.

K základnímu řízení modulů se využívají dva příkazy:

- `import` (načítá celý modul)
- `from` (načítá daná data z modulu do lokálního prostoru)

K tvorbě scriptu budou využity následující knihovny a moduly. Jejich vyvolání představuje první část kódu nazvanou *Importy*. Mezi knihovny a moduly, které budou během tvorby kódu využity, patří:

- `import collections` – modul implementuje specializované datové typy kontejnerů, jež poskytují alternativy k obecným vestavěným kontejnerům v Pythonu (např. `list` a `tuple`)
- `import ortools.sat.python.cp_model` – metoda tvorby a řešení CP-SAT modelů
- `import pandas as pd` – softwarová knihovna pro manipulaci a analýzu dat
- `import plotly.figure_factory as ff` – modul pro vytváření jedinečných grafů, které jsou zahrnuty v knihovně *Plotly*
- `from plotly.basedatatypes import BaseTraceType` – základní třída napomáhající vykreslování grafů knihovny *Plotly* pro všechny typy obrazců
- `import math` – modul obsahující matematické metody a konstanty, které lze využít k řešení matematických úloh
- `import os` – modul poskytující přenosný způsob používání funkcí závislých na operačním systému, včetně interakce se systémem souborů

#### 4.5.2 Deklarace modelu

Funkce sloužící k řešení kapacitního plánování je definována v úvodu scriptu příkazem *def*, za kterým je zapsáno pojmenování funkce. Jelikož se Pythonu nedeklarují proměnné, jsou proměnné lokální. K řešení problému je využita deklarace pomocí příkazu *global*, díky kterému budou proměnné uvedené za příkazem na globální úrovni.

```
def Minimalizace_casovych_prodlev():  
    global job_id, task_id, task, pridelene_prace
```

Popis proměnných:

- `job_id` – proměnná udávající číslo zakázky u každé operace
- `task_id` – proměnná určující číselné označení pořadí operace na zakázce
- `task` – proměnná nesoucí veškeré informace operace vyjádřené parametry (ve tvaru:  $m$  – pracoviště,  $p$  – délka operace,  $s$  – pořadí operace)
- `pridelene_prace` – proměnná sloužící k uspořádání plánu operací (ve formátu: `start` – doba startu operace, `job` – číslo zakázky, `index` – pořadí operace na zakázce, `duration` – délka trvání operace)

Následně je ve funkci deklarován model řešitele CP-SAT.

```
model = cp_model.CpModel()
```

### 4.5.3 Vstupy dat

Než se data popisující výrobu zakázek stanou plnohodnotnými vstupními daty, musí projít přechodovým můstkem mezi ERP systémem a vývojovým prostředím PyCharm, kterým je Microsoft Excel. V programu Microsoft Excel dochází k uspořádání dat a seřazení do konečného formátu pomocí vytvořené šablony zpracovávající data z ERP systému. Finální vzhled uspořádaných dat se nachází v tabulce hodnot, která je demonstrována na fiktivní zakázce v kapitole č. 4.4.2 v tabulce č. 4.3.

Ve snaze zajistit co největší míru automatizace, jsou vstupní data importována pomocí knihovny *os* s použitím příkazu *os.listdir()*. Použitím příkazu je získán seznam všech souborů v zadaném umístění s příponou *.xlsx*.

```
main_dir = './'
all_filenames = [(main_dir + file) for file in
os.listdir(main_dir) if file.endswith(".xlsx")]
```

Formát vstupních dat je přesně definovaný a data jsou zapsána pomocí třech parametrů (*m* – pracoviště, *p* – délka operace, *s* – pořadí operace) tvořící operaci. Jelikož parametr *s* znázorňuje místo ve stromové struktuře zakázky, může být vyjádřen více čísly (k tomu dochází převážně u sestav, které jsou tvořeny několika díly). Zápis parametru *s* je ve tvaru *n*-tice (zápis *n*-tice se vytváří pomocí kulatých závorek), která je neměnná a uspořádaná. Každá operace je zapsána také ve formě *n*-tice s přesně daným pořadím jednotlivých operací. Zakázky jsou tvořené operacemi a jsou zapsané v seznamu (zápis seznamu se vytváří pomocí hranatých závorek). Ten se používá k uložení více položek do jedné proměnné. Všechny zakázky tvoří seznam s názvem výroba, do něhož jsou proměnné postupně ukládány.

Část kódu definující finální podobu dat je zapsán s využitím *for* cyklů s podmínkami *if*, *elif* a *else*. Data jsou do *for* cyklu načítána ze seznamu všech *.xlsx* souborů uložených v zadané složce (každá zakázka je uložena v samostatném *.xlsx* souboru). Načítání dat probíhá po sloupcích, které tvoří *n*-tice. U sloupců *continuity*, popisující pořadí operace, je různý počet sloupců podle toho, na kolik dílců operace navazuje. K definování počtu sloupců je použita metoda *math.isnan()*, která zkontroluje, zda je hodnota prázdná (*NaN*) či nikoli. Jakmile je vytvořena *n*-tice sloupců popisující pořadí a sloupců s pracovištěm a časem, jsou obě *n*-tice sloučeny a přidány do seznamu zakázky v seznamu výroby.

```

vyroba = []
for f in all_filenames:
    df = pd.read_excel(f, header=0)
    zakazka = []
    for index, row in df.iterrows():
        tup_1 = [row['machine'], row['time']]
        tup_2 = ()
        for i in range(df.columns.get_loc("continuity"),
len(df.columns)):
            if math.isnan(row.iloc[i]):
                if i == df.columns.get_loc("continuity"):
                    tup_2 = ()
                    break
                elif i == df.columns.get_loc("continuity")+1:
                    tup_2 = tup_2 + ()
                    break
            else:
                break
            tup_2 = tup_2 + (row.iloc[i],)
        tup_1.append(tup_2)
        tup_1 = tuple(tup_1)
        zakazka.append(tup_1)
    vyroba.append(zakazka)
print(vyroba)

```

#### 4.5.4 Definice dat

##### Počet strojů

Stanovení počtu pracovišť podílejících se na výrobě zakázek vychází ze vstupních dat. Pracoviště je uvedeno u operace na 0. místě (Python v tomto případě počítá s označením míst od nuly, ne od jedné) v kulaté závorce popisující operaci parametry. K určení celkového počtu pracovišť je nutné projít všechny pracoviště jednotlivých zakázek a operací pomocí funkce *for*.

```
pocet_stroju = 1 + max(task[0] for job in zakazky for task in
job)
```

Počet pracovišť je vyjádřen ve formě rozsahu. K takové formě zápisu je využita funkce *range*. Ta tvoří posloupnost čísel, která ve výchozím nastavení začíná od 0 a zvyšuje se vždy o 1. Posloupnost končí před uvedeným číslem v kulaté závorce za funkcí *range*. Počet pracovišť ve formátu rozsahu se využije například v iteračních smyčkách *for*, kdy postupně projdou iterací všechny pracoviště.

```
vsechny_stroje = range(pocet_stroju)
```

## Horizon

K definici proměnných je nutné stanovit dynamický výpočet možného času na výrobu zadaných zakázek, tzv. *horizon*. Jedná se o sumu všech délek trvání operací zakázek. *Horizon* představuje neoptimální řešení, při kterém se operace plánu nepřekrývají. Proto musí být optimální řešení plánu vždy menší než *horizon*.

Časy operací vychází ze vstupních dat, kde jsou uvedeny na 1. místě v kulaté závorce popsané operace pomocí vstupních parametrů. K určení *horizonu* je využita funkce *for*, která prochází postupně všechny časy operací jednotlivých zakázek a operací.

```
horizon = sum(task[1] for job in zakazky for task in job)
```

### 4.5.5 Definice proměnných

Proměnné slouží k odkazování se na určitá data, která jsou uložena v paměti počítače. Vytváří se určité datové typy, jež jsou následně přiřazeny do proměnných. Lze je využít například při volání metod ve funkcích.

V Pythonu jsou proměnné dynamicky typové, a proto nemusí deklarovat s jejich datovým typem. To znamená, že proměnným lze přiřadit různé datové typy (např. text, objekt uživatele nebo čísla) a jazyk se s tím vždy vypořádá, protože dokáže vnitřně automaticky měnit datový typ. Dochází proto k podstatnému zkrácení kódu ve srovnání s ostatními programovacími jazyky.

#### Datové kontejnery

Před definováním proměnných je nezbytné předpřipravít místa (tzv. datové kontejnery), kde se bude s proměnnými pracovat a zároveň zde budou ukládány. Vytvoření datových kontejnerů poskytuje modul *collection*. Datový kontejner je objekt, který se používá pro ukládání různých objektů. Poskytuje způsob přístupu k obsaženým objektům v datovém kontejneru a umožňuje přes ně iterovat.

K práci s proměnnými jsou vytvořeny tři datové kontejnery:

- Pojmenovaná n-tice pro ukládání informací o vytvořených proměnných. Vytváří n-tici podtříd s pojmenovanými poli.

```
typ_operace = collections.namedtuple('typ_operace', 'start  
end interval')
```

- Pojmenovaná n-tice pro manipulaci s informacemi o řešení. Vytváří n-tici podtříd s pojmenovanými poli.

```
prirazeni_typu_operace = collections.namedtuple  
( 'prirazeni_typu_operace', 'start job index duration')
```

- Podtřída slovníku, která volá funkci pro dodání chybějících hodnot. Vytváří intervaly zakázek a přidává je do příslušných seznamů strojů.

```
intervaly_zakazek = collections.defaultdict(list)
```

Všechny operace popsané proměnnými jsou přidány do prázdného slovníku, který je zapsán pomocí dvou složených závorek.

```
vsechny_operace = {}
```

Slovník je asociativní pole obsahující dva typy prvků. Jedním z nich je klíč a druhým je hodnota, která je vždy přiřazena pod jeden z klíčů.

### Intervaly zakázek

S využitím vestavěné funkce Pythonu *enumerate* lze iterovat přes cyklus *for*. Funkce vždy iteruje hodnotu, kterou následně vrátí ve formě výčtového objektu zpět do smyčky a postupně tvoří slovník. Během iterace jsou očíslovány jednotlivé operace v zakázkách, ke kterým jsou ještě přiřazeny další potřebné informace týkající se vytvoření proměnných.

Metoda řešitele *NewIntVar* je použita pro každou operaci k vytvoření celočíselné proměnné s doménou. Metodou je zapsán čas zahájení operace (*start*) a čas ukončení operace (*konec*) ve tvaru *NewIntVar* (instance třídy, dolní mez pro proměnnou, horní mez pro proměnnou, název proměnné). Horní mez pro *start* a *konec* tvoří *horizont* (součet časů na vyhotovení všech operací zakázek), který byl již definován v podkapitole definice *dat*.

Pro vytvoření intervalové proměnné je využita metoda *NewIntervalVar* ve tvaru (instance třídy, začátek intervalu – *start*, délka intervalu – *délka*, konec intervalu – *konec*, název proměnné intervalu). Řešení vždy musí splňovat základní podmínku, a to sice že délka intervalu nesmí být menší nebo rovna. Matematický zápis podmínky délky intervalu je následující:

$$\text{konec intervalu} - \text{začátek intervalu} \geq 0 \quad (6)$$

Intervalová proměnná je omezení, které se využije v definici omezení u omezující podmínky překrývání s názvem *NoOverlap*.

Kód k vytvoření potřebné formy zápisu intervalů zakázek s následným uložením do předem vytvořeného prázdného slovníku pojmenovaného *vsechny\_operace* je následující:

```
for job_id, job in enumerate(vyroba):
    for task_id, task in enumerate(job):
        stroj = task[0]
        duration = task[1]
        suffix = '_%i_%i' % (job_id, task_id)
        start = model.NewIntVar(0, horizont, 'start' + suffix)
        end = model.NewIntVar(0, horizont, 'end' + suffix)
        interval = model.NewIntervalVar(start, duration, end,
                                       'interval' + suffix)
        vsechny_operace[job_id, task_id] = typ_operace
                                   (start=start,
                                   end=end,
                                   interval=interval)
        intervaly_zakazek[stroj].append(interval)
```

### 4.5.6 Definice omezení

Omezení představují důležitou část kódu, díky které je nástroj pro optimální plánování výroby přizpůsoben výrobě ve specifickém podniku. Vychází z matematického modelu uvedeného v kapitole č. 4.1, kde jsou omezení popsána pomocí matematického zápisu ve formě nerovnic (omezení přednosti a překrývání) nebo specifikované podmínky (omezení přerušení a posloupnosti operací).

- Omezující podmínka překrývání

K implementaci omezení do kódu je využita metoda řešení pomocí funkce `model.AddNoOverlap`, která zajišťuje, že se časové intervaly operací vypracovávaných na jednom stejném pracovišti nebudou nikdy časově překrývat.

```
for machine in range(pocet_stroju):  
    model.AddNoOverlap(intervals[machine])
```

- Omezující podmínka posloupnosti operací a přednosti

Omezení slouží k zachování pořadí operací u dílců, sestav nebo na samotné zakázce. Dále také zabraňuje, aby se po sobě jdoucí operace u dílce či sestavy nepřekrývaly, ale výroba dílců či sestav probíhala paralelně. Návaznosti operací jsou popsány vstupními parametry zapsanými v kulaté závorce na posledním 2. místě kulaté závorky (parametru návaznosti operací je sériové číslo značené písmenem s).

K definici omezení je nejprve využita funkce `for`, která postupně prochází zakázky, operace a nakonec sériová čísla. Zápis cyklu je následující:

```
for job_id, job in enumerate(zakazky):  
    for task_id in range(len(job) - 1):  
        for SerialNumber in job[task_id][2]:
```

K definici omezující podmínky posloupnosti operací a přednosti lze využít třídu `LinearExpr`, protože se jedná o lineární omezení definované pomocí celočíselného lineárního výrazu. Lineární výrazy řešitel CP-SAT řeší buď pomocí definice omezení, nebo definicí účelové funkce. Zde se jedná o definici omezení vyjadřující to, že čas ukončené operace nastal před časem zahájení další operace dílce nebo sestavy.

```
model.Add(tasks[job_id, SerialNumber].end <= tasks[job_id,  
task_id].start)
```

- Omezující podmínka přerušení

Metoda řešení pomocí CP-SAT řešitele obsahuje takové základní omezení, že jakmile je operace spuštěna, musí být i dokončena. Řešitel neumožňuje řešení, které obsahuje nutnost pozastavování operací. Situace pozastavené zakázky ve prospěch jiné zakázky během standardní výroby nemůže nastat, proto

nemusí být možnost zaváděno do kódu. Všechny přijaté zakázky mají ve výrobě stejnou prioritu a nejsou vůči sobě nijak nadřazené.

#### 4.5.7 Definice výstupů

Žádaným výstupem nástroje pro plánování je minimalizovaný čas potřebný na vyrobení všech operací. K dosažení požadovaného výstupu bude využita účelová funkce minimalizace časových prodlev.

V účelové funkci vystupuje proměnná *max\_cas*, jejíž hodnota je maximální možný čas na vyhotovení všech operací zakázek. Proměnná je zapsána metodou řešitele *NewIntVar*.

```
max_cas = model.NewIntVar(0, horizont, 'makespan')
```

Pro zápis účelové funkce je vytvořen model *AddMaxEquality*, který je určen pro lineární omezení, a to ve tvaru *model.AddMaxEquality* (instance třídy, cílová proměnná, výraz ve tvaru proměnné). Cílová proměnná je maximální čas výroby (*max\_cas*) a proměnnou výrazu jsou všechny operace.

```
model.AddMaxEquality(max_cas, [
    vsechny_operace[job_id, len(job) - 1].end
    for job_id, job in enumerate(vyroba)
])
```

Účelová funkce je zapsána pomocí modelu *Minimize*, který nastaví cíl modelu na minimalizaci, ve tvaru *model.Minimize* (instance třídy, cílový objekt).

```
model.Minimize(max_cas)
```

#### Vyvolání řešitele

Hlavní třídou řešitelů je *CpSolver*. Účelem třídy je vytvořit řešitele *Solve*. Jakmile je metoda *Solve* vyvolána, umožňuje kontrolovat postup řešení metodami *Value*, *BooleanValue* a metodou *obecné statistiky*.

```
solver = cp_model.CpSolver()
status = solver.Solve(model)
```

K omezení délky hledání řešení je nastaven časový limit řešení na 30 sekund. Časový limit je možné měnit podle potřeby.

```
solver.parameters.max_time_in_seconds = 30.0
```

#### 4.5.8 Vypsání výsledků

Zobrazení výsledků může mít různé podoby. Od společnosti PHARMIX, s.r.o. byl vznesen požadavek, aby byl kladen vyšší důraz převážně na propracovanou grafickou interpretaci výsledků. Číselné zobrazení výsledků není tak důležité pro budoucí vizi plánování v podniku.

Finální zpracování výsledků je navrženo tak, aby byly jednotlivé operace rozřazeny na pracovištích dle optimálního řešení plánu. Na pracovišti vznikne harmonogram operací k vyhotovení s pevně danou posloupností. Každá operace nese informace o číselném označení operace a časovém intervalu, jenž určuje rozmezí od kdy do kdy se má na operaci pracovat.

Kód zobrazující výsledky je tvořen několika *for* cykly. Nejprve musí být ale splněna podmínka, že je řešení optimální. Pokud je podmínka splněna, přistoupí se k zápisu kódu pro zobrazení výsledků. Tvoří ho slovník přidělených operací na pracovištích. Do slovníku jsou přes dva *for* cykly postupně ukládány operace.

```
if status == cp_model.OPTIMAL:
    pridelene_prace = collections.defaultdict(list)
    for job_id, job in enumerate(vyroba):
        for task_id, task in enumerate(job):
            stroj = task[0]
            pridelene_prace[stroj].append(
                prirazeni_typu_operace(start=solver.Value(
                    vsechny_operace[job_id, task_id].start),
                    job=job_id,
                    index=task_id,
                    duration=task[1]))
```

Následně se vytvoří výstupní řádky. Na každém řádku bude jedno pracoviště, ke kterému budou postupně přidávány operace podle řešení. Přidělené operace jsou seřazeny podle počátečního času operace. Kvůli přehlednosti jsou přidána volná místa na výstupu po jednotlivých operacích. Vzniknou tak zarovnané sloupce. Výsledné zobrazení výsledků lze následně vyvolat pomocí příkazu *print*.

Část zobrazení vypsání výsledků se může v průběhu používání programu v reálné výrobě upravovat podle potřeb uživatelů. Výsledky se také mohou upravit do podoby, jež bude možno uložit do souboru *.xlsx*. Díky tomu by se s výsledky mohlo dále pracovat a využívat je pro další účely. Systém kapacitního plánování by se případně v budoucnu dal spárovat se systémem zaznamenávání operací, kde by se naplánované operace rovnou po vyhotovení evidovaly. Jednalo by se již o pokročilejší systém, který by k zavedení potřeboval odborné pracovníky se specializací v oboru informatiky.

### 4.5.9 Vykreslení řešení

Na grafickou část řešení je kladen zvýšený důraz, protože bude využíván k řízení plánování ve výrobě. Jako nejvhodnější grafické zpracování řešení kapacitního plánování je zvolen Ganttův diagram. Jedná se o druh pruhového diagramu často využívaného v průmyslu při plánování a řízení výroby. S jeho pomocí lze přehledně znázornit posloupnost naplánovaných operací v čase. OR-Tools nenabízí grafické zpracování řešení, které by šlo otevřeně navázat na kód. Proto byly využity jiné knihovny, které grafické zpracování nabízí. Během návrhu nejvhodnějšího designu pro podnik byly vyzkoušeny různé implementace Ganttova diagramu na řešení. Ke znázornění Ganttova diagramu jsou nejčastěji využívány knihovny *Matplotlib* a *Plotly*. Oba způsoby znázornění řešení byly vyzkoušeny a jako vhodnější řešení bylo zvoleno řešení s použitím knihovny *Plotly* s modulem „*figure\_factory*“. Informace o způsobu fungování knihovny vychází ze zdroje [18].

Modul *plotly.figure\_factory* využívá *plotly.express.density\_heatmap()*. Výsledné řešení je upraveno do požadovaného formátu pomocí předem vytvořené funkce. Funkce pro vizualizaci je zapsána v úvodu scriptu, aby nebyla součástí funkce minimalizace časových prodlev. Funkce je ve tvaru *vizualizace (pridelene\_prace – nese informace o jednotlivých operacích zakázek, vsechny\_stroje – funkce rozsahu celkového počtu všech pracovišť, pocatecni\_cas – počáteční datum začátku první operace)*. Data z funkce jsou ukládána do prázdného datového seznamu *finalni*. Funkce *vizualizace()* je zapsána:

```
def vizualizace(pridelene_prace, vsechny_stroje,
pocatecni_cas):
    finalni = []
    for pracoviste in vsechny_stroje:
        pridelene_prace[pracoviste].sort()
        for assigned_task in pridelene_prace[pracoviste]:
            name = 'Zakázka %i' % assigned_task.job
            temp = dict(Task=pracoviste, Start=pocatecni_cas +
pd.DateOffset(minutes=assigned_task.start),
                    Finish=pocatecni_cas +
pd.DateOffset(minutes=(assigned_task.start +
assigned_task.duration)),
                    Resource=name)
            finalni.append(temp)
    finalni.sort(key=lambda x: x['Task'])
    return finalni
```

V poslední části kódu ve funkci *Minimalizace\_casovych\_prodlev()* je využita vytvořená funkce *vizualizace()* k tvorbě Ganttova diagramu. Při vyvolání modulu *plotly.figure\_factory* lze zadat doplňující informace týkající se vzhledu diagramu, jako např. barvy polí, výška a šířka grafu, název grafu a zobrazení mřížky. Ohraničení polí pro lepší přehlednost při více operacích jedné zakázky (označené jednou barvou) je zajištěno modulem *plotly.basedatatypes*.

```

res = vizualizace(pridelene_prace, vsechny_stroje,
pocatecni_cas)
fig = ff.create_gantt(res, index_col='Resource',
title='Kapacitní plánování výroby',
show_colorbar=True, showgrid_x=True,
showgrid_y=True, group_tasks=True,
show_hover_fill=True, )
fig.update_traces(mode='lines', line_color='white',
selector=dict(fill='toself'))
trace: BaseTraceType
fig.show()

```

Během používání nástroje pro plánování v podniku lze zobrazení výsledků upravovat dle potřeby, aniž by byla narušena funkčnost scriptu. V budoucnu se nabízí možnost ke zlepšení v provázanosti nástroje pro plánování s evidencí operací. Pokud by operace byly spojené s plánováním, byl by Ganttův diagram dynamický a měnil by se v závislosti s vyhotovenými operacemi. V tomto případě by bylo vhodnější využít knihovnu *Mathplotlib* s modulem „*pyplot*“ [19]. S využitím modulu je možné procentuálně sledovat výrobu zakázek v reálném čase.

## 5 Řešení případové studie

Případová studie je demonstrována na zjednodušeném stavu výroby ve společnosti PHARMIX, s.r.o. Ve výrobě jsou aktivní dvě zakázky složené z dílů tvořící finální sestavu. K naplánování výroby jednotlivých operací je využit nástroj pro optimální plánování výroby, který má za cíl sestavit výrobní plán v co nejkratším možném čase. Popis fungování nástroje na plánování je popsáno v předešlé kapitole č. 4.5.

### 5.1 Vstupní data

Znázornění podoby zakázek je vytvořené v programu Microsoft Excel. První zakázka je pojmenovaná jako *zakázka 0*. Je tvořena třemi dílci (dno, plášť a víko), které dohromady tvoří sestavu nádoby. Až jsou všechny operace dílců hotové, lze teprve začít s první operací sestavy nádoby. Všechny časy jsou uvedené v jednotkách minut, protože při zpracování dat algoritmus nedokáže pracovat s desetinnými čísly. Jednotky hodin by byly nepřesné kvůli velkému vlivu zaokrouhlování na celá čísla, proto je nejvhodnějším řešením vyjádření času v jednotkách minut. Schéma zakázky 0 je zobrazeno v tabulce č. 5.1.

Tabulka 5.1 Schéma zakázky 0

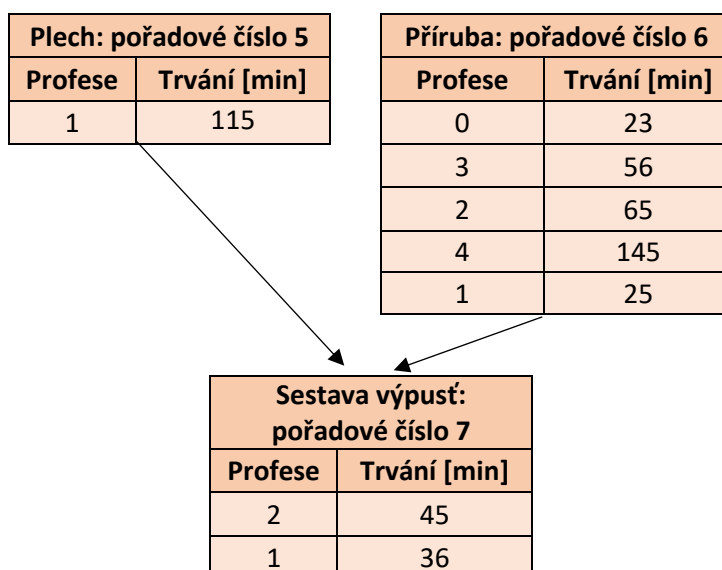
Dno: pořadové číslo 1		Plášť: pořadové číslo 2		Víko: pořadové číslo 3	
Profese	Trvání [min]	Profese	Trvání [min]	Profese	Trvání [min]
2	50	0	20	0	45
0	23	2	85	1	58
1	76	1	32	2	63

Sestava nádoba: pořadové číslo 4	
Profese	Trvání [min]
2	55
1	84

*Zakázka 1* je tvořena jen ze dvou dílců (plech a příruba), které tvoří finální sestavu výpusti. Práce na první operaci sestavy výpusti musí také začít jako u předchozí zakázky až po kompletním vyhotovení obou dílců. Schéma zakázky 1 je uvedeno v tabulce č. 5.2.

*Tabulka 5.2 Schéma zakázky 1*



Profese vyskytující se u obou zakázek jsou číselně označené od 0 do 4. Číselné označení bylo zvoleno proto, že při importování dat do prostředí Pythonu se s ním pracuje lépe než se slovními názvy jednotlivých profesí. K číselnému označení profesí existuje legenda, která vyjadřuje, co které číslo znázorňuje. Legenda je uvedena v tabulce č. 5.3.

*Tabulka 5.3 Legenda popisující označení profesí*

Legenda	
Označení	Profese/pracoviště
0	Mechanik
1	Ruční broušení
2	Svařování
3	Horizontální frézka
4	Karusel

Jakmile dojde k zaplánování zakázek, tak zpracování technologické dokumentace zakázek přebírá technolog, který vyhotoví technologické postupy výroby. V ERP systému se technologické postupy výroby exportují do sešitu Microsoft Excel, který slouží jako přechodový můstek pro importování dat do Pythonu. V Microsoft Excel jsou data automaticky upravena do požadovaného formátu pomocí předpřipravené šablony. Data jsou následně nahrána pomocí příkazu do prostředí jazyka Python.

Při nahrávání do Pythonu jsou vstupní data upravena do požadované podoby vstupních dat pro výpočet pomocí CP-SAT řešitele. Potřebný formát vstupních dat v Pythonu musí být následující:

```
vyroba = [
    [(2, 50, ()), (0, 23, (0,)), (1, 76, (1,)), (0, 20, ()),
    (2, 85, (3,)), (1, 32, (4,)), (0, 45, ()), (1, 58, (6,)),
    (2, 63, (7,)), (2, 55, (2, 5, 8)), (1, 84, (9,)), (2, 0,
    (10,))], # zakázka 0
    [(1, 115, ()), (0, 23, ()), (3, 56, (1,)), (2, 65, (2,)),
    (4, 145, (3,)), (1, 25, (4,)), (2, 45, (0, 5)), (1, 36, (6,)),
    (2, 0, (7,))] # zakázka 1
]
```

Jakmile jsou vstupní data připravená, je zadán počáteční čas, který definuje čas startu první operace výroby. Pro zpracování případové studie je zvolen začáteční čas dne 12. 4. 2022 v 7:00 hodin.

```
pocatecni_cas = pd.to_datetime('12/05/2022 07:00:00')
```

Po splnění těchto kroků přípravy nástroje pro plánování je vše přichystáno ke spuštění hledání optimálního řešení případové studie.

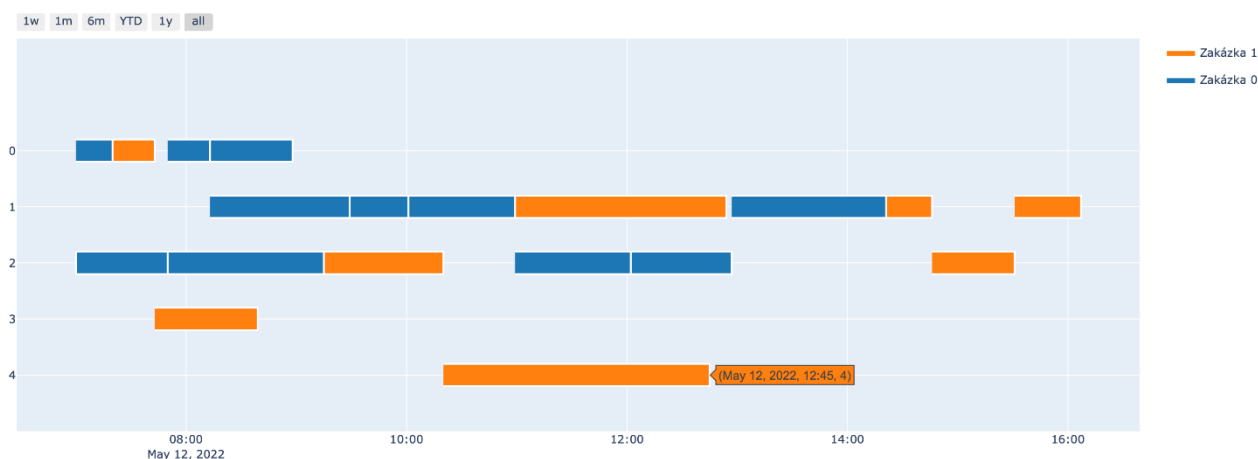
## 5.2 Vypsání řešení

Řešení optimálního uspořádání dvou zakázek ve výrobě je v podobě číselného řešení a grafického řešení vyjádřeného pomocí Ganttova diagramu. Výsledné uspořádání koresponduje se zadanými omezeními, které jsou uvedené ve scriptu v části omezení. Jedná se o omezující podmínku přerušení, přednosti, překrývání a posloupnosti operací. U řešení nejsou patrné žádné zjevné prodlevy mezi operacemi, plynule na sebe navazují.

Výsledné řešení zpracování zakázek je 547 minut (délka trvání od 7:00 dne 12. 4. 2022 do 16:07 dne 12. 4. 2022). O jakou se jedná zakázku lze jednoduše poznat pomocí barvy. Jaká barva odpovídá dané zakázce je uvedeno v legendě umístěné v pravém horním rohu Ganttova diagramu na obrázku č. 5.1. Pokud uživatel nasměruje kurzor myši na jakoukoli operaci zobrazenou v Ganttově diagramu, zobrazí se tabulka s popisem vybrané operace. V popisu je uveden počáteční a konečný čas definující dobu k vyhotovení operace. Další zajímavou funkcí interaktivního Ganttova diagramu je možnost zobrazení plánu v rozmezí jednoho týdne, měsíce, půl roku, roku a od počátku plánování do aktuálního data (zkratka YTD). Tato funkce je při zobrazování řešení případové studie zbytečná, protože doba vyhotovení obou zakázek je v rámci jednoho dne, ale při zobrazení plánu celé výroby ve společnosti je díky funkci možné se přehledně pohybovat při práci s plánem.

Zobrazení Ganttova diagramu je přeneseno automaticky po nalezení optimálního řešení z vývojového prostředí PyCharm do přednastaveného webového prohlížeče, kde je možné s plánem interaktivně pracovat. Výsledné řešení lze uložit do běžně používaných formátů obrázků jako je *.jpg* a *.png*.

### Kapacitní plánování výroby



**Obrázek 5.1** Ganttův diagram řešení případové studie

Číselné řešení je zobrazeno na obrázku č. 5.2 v konzoli vývojového prostředí PyCharm. Výsledné uspořádání sledu operací je rozděleno po pracovištích. Každá operace je popsána názvem a časovým intervalem. Název je uveden ve formátu *job\_x\_y*, kde písmeno *x* značí číslo zakázky. Písmeno *y* vyjadřuje číslo operace, jež je shodné s pořadovým číslem, pod kterým byla operace importována ve vstupních parametrech zakázky. Časový interval definující čas vyhrazený k vyhotovení operace je vymezen intervalem v hranaté závorce nacházející se pod názvem operace. V řešení se objevují 2 nulové operace naplánované na pracovišti číslo 2. Tyto operace se objevují v řešení jako pojistka proti zacyklení operací. Pokud by vždy poslední fiktivní operace zakázky nebyla nulová, mohlo by dojít k chybě při hledání optimálního řešení. Operace jsou nulové a nijak neovlivňují finální výsledek.

Podobu výsledků lze z konzole vývojového prostředí PyCharm exportovat do textového formátu, ve kterém se s výsledky dá následně pracovat. Jelikož společnost export do textového formátu nevyžaduje, tak tento krok není součástí scriptu.

```
pracoviště 0: job_0_3      job_1_1      job_0_1      job_0_6
              [0,20]      [20,43]      [50,73]      [73,118]
pracoviště 1: job_0_2      job_0_5      job_0_7      job_1_0      job_0_10     job_1_5      job_1_7
              [73,149]     [149,181]    [181,239]    [239,354]    [357,441]    [441,466]    [511,547]
pracoviště 2: job_0_0      job_0_11     job_1_8      job_0_4      job_1_3      job_0_8      job_0_9      job_1_6
              [0,50]      [0,0]        [0,0]        [50,135]     [135,200]    [239,302]    [302,357]    [466,511]
pracoviště 3: job_1_2
              [43,99]
pracoviště 4: job_1_4
              [200,345]
```

**Obrázek 5.2** Číselný zápis řešení případové studie

### 5.3 Vyhodnocení správnosti řešení

Výsledné uspořádání operací bylo srovnáno s plánem, který by vznikl bez pomoci nástroje na plánování. Šlo by tedy jen o intuitivní seřazení jednotlivých operací tak, aby bylo dosaženo kratšího času potřebného na výrobu zakázek uvedených v případové studii. Vhodnějšího uspořádání operací se nepodařilo dosáhnout. Řešení pomocí plánovacího nástroje lze proto považovat za správné a nástroj pro plánování za vhodný způsob hledání optimálního řešení pro plánování výrobních kapacit ve společnosti PHARMIX, s.r.o.

Plánování s využitím nástroje pro plánování bylo také odzkoušeno při zkušebním provozu na výrobě dvou duplicitních výrobků víka, které je zobrazeno na obrázku č. 5.3. Výroba probíhala za standardních podmínek a srovnatelného stavu vytíženosti pracovních kapacit.



Obrázek 5.3 Víko

Výroba prvního kusu s využitím nástroje pro plánování přinesla přibližně 15% úsporu potřebného času k vyhotovení víka. Jedná se pouze o odhad úspory času, který nebylo možné číselně podložit. Odhad vychází z dostupných dat z výroby, které byly posouzeny vedoucími pracovníky. U rozsáhlejších zakázek s větším počtem dílců je předpokládána ještě vyšší úspora času potřebného na výrobu.

## 6 Závěr

K dosažení efektivního plánu využívající účinně výrobní kapacity a zároveň minimalizující výrobní časy byl vytvořen optimalizační nástroj, který využívá hybridní řešitel CP-SAT. Ten si dokáže velmi rychle a výkonně poradit i s rozsáhlými úlohami. Hledání optimálního řešení je omezeno podmínkami, jimiž je charakterizována výroba. Pro tvorbu nástroje byl využit programovací jazyk Python v integrovaném vývojovém prostředí PyCharm.

Výstupem z nástroje pro plánování výrobních kapacit je plán posloupnosti výroby operací na jednotlivých pracovištích společně s grafickým zpracováním v podobě Ganttova diagramu, který bude společnost využívat převážně k řízení výroby. Funkčnost nástroje pro plánování byla již otestována ve zkušebním provozu. Data získána ze zkušebního provozu byla porovnána s výrobními daty z provozu využívající dosud starý způsob plánování bez využití nástroje pro optimální plánování výroby. Po vyhodnocení bylo dosaženo velice uspokojivého výsledku v podobě ušetření pracovních kapacit ve výrobě o přibližně 15 %.

Nový nástroj pro plánování výrobních kapacit je velkým pokrokem oproti předešlému principu plánování výroby, který spočíval převážně v odhadech a podobnostech s již vyhotovenými zakázkami. Stále ale zůstává prostor pro zdokonalování a úpravy. Systém plánování by mohl být v budoucnu přímo propojen s ERP systémem a vyhotovené operace z výroby by se rovnou evidovaly nejen v ERP systému, ale také v nástroji pro plánování, kde by se rovnou zpracovávaly pro potřeby plánování. Dále by bylo vhodné získat v průběhu výroby více dat, díky kterým by bylo dosaženo ještě přesnějšího odhadu času pro vyhotovení operací uvedených v technologické dokumentaci. Při vyšší úrovni automatizace těchto úkonů by se tak nástroj dal považovat za prvek průmyslu 4.0. Celý proces dosažení optimálního plánovacího systému v podniku tak bude vyžadovat ještě mnoho úsilí, ale základní princip je již vyřešen.

## Seznam použité literatury

- [1] VEBER, Jaromír. *Management – Základy, moderní manažerské přístupy, výkonnost a prosperita*. 2. aktualizované vydání. Praha: Management Press, 2014. ISBN 978-80-7261-274-1.
- [2] ŽŮRKOVÁ, Hana. *Plánování a kontrola – klíč k úspěchu*. Grada Publishing a.s., 2007. ISBN 978-80-247-1844-6.
- [3] APPELBAUM, Deniz, Alexander KOGAN, Miklos VASARHELYI a Zhaokai YAN. ERP definice. *International Journal of Accounting Information Systems* [online]. 2017, 25, 29–44. [cit. 2021-09-03]. ISSN 14670895. Dostupné z: doi:10.1016/j.accinf.2017.03.003
- [4] SHTUB, Avraham a Reuven KARNI. *ERP – Helios Orange funkce* [online]. Boston, MA: Springer US, 2010 [cit. 2021-09-21]. ISBN 978-0-387-74523-7. Dostupné z: doi:10.1007/978-0-387-74526-8
- [5] ÖZGÜVEN, Cemal, Lale ÖZBAKIR a Yasemin YAVUZ. Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Applied Mathematical Modelling* [online]. 2010, 34(6), 1539–1548. [cit. 2021-10-01]. ISSN 0307-904X. Dostupné z: doi:10.1016/j.apm.2009.09.002
- [6] JAIN, A.S. a S. MEERAN. Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research* [online]. 1999, 113(2), 390–434. [cit. 2021-10-03]. ISSN 0377-2217. Dostupné z: doi:10.1016/S0377-2217(98)00113-1
- [7] VOŇKA, Radim. *Optimální plánování výroby v průmyslovém podniku*. 2020. Fakulta strojního inženýrství VUT, ústav procesního inženýrství. Vedoucí bakalářské práce Michal Touš.
- [8] JANÁČEK, Jaroslav a Lubomír BUZNA. *Optimization in networks*. Žilina: University of Žilina, 2009. 1st ed. ISBN 978-80-8070-985-3.
- [9] ROSELLI, Sabino, Kristofer BENGTSSON a Knut ÅKESSON. SMT Solvers for Job-Shop Scheduling Problems: *Models Comparison and Performance Evaluation*. [online]. 2018, s. 547–552. [cit. 2021-10-24]. Dostupné z: doi:10.1109/COASE.2018.8560344
- [10] NAGATA, Yuichi a Isao ONO. A guided local search with iterative ejections of bottleneck operations for the job shop scheduling problem. *Computers & Operations Research* [online]. 2018, 90, 60–71. [cit. 2021-12-11]. ISSN 0305-0548. Dostupné z: doi:10.1016/j.cor.2017.09.017
- [11] FRANCO, John a John MARTIN. Chapter 1. A History of Satisfiability. In: Armin BIERE, Marijn HEULE, Hans VAN MAAREN a Toby WALSH, ed. *Frontiers in Artificial Intelligence and Applications* [online]. IOS Press, 2021. [cit. 2022-03-16]. ISBN 978-1-64368-160-3. Dostupné z: doi:10.3233/FAIA200984

- [12] STUCKEY, Peter J. Lazy Clause Generation: Combining the Power of SAT and CP (and MIP?) Solving. In: Andrea LODI, Michela MILANO a Paolo TOTH, ed. *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* [online]. Berlin, Heidelberg: Springer, 2010, s. 5–9. Lecture Notes in Computer Science. [cit. 2022-03-28]. ISBN 978-3-642-13520-0. Dostupné z: doi:10.1007/978-3-642-13520-0\_3
- [13] STUCKEY, Peter J. *Search is Dead Long Live Proof* [online prezentace]. 2012 [cit. 2022-04-15]. Dostupné z: <https://people.eng.unimelb.edu.au/pstuckey/PPDP2013.pdf>
- [14] Google OR-Tools. *The Job Shop Problem* [online]. [cit. 2022-04-20]. Dostupné z: <https://developers.google.com/optimization?hl=cs>
- [15] BHASIN, Harsh. *Python Basics: A Self-Teaching Introduction* [online]. Mercury Learning and Information, 2019 [cit. 2022-04-22]. ISBN 978-1-68392-353-4. Dostupné z: <https://1lib.cz/book/3641986/812370>
- [16] API Documentation. *ortools.sat.python.cp\_model*. Developers Google [online]. [cit. 2022-04-22]. Dostupné z: [https://google.github.io/ortools/python/ortools/sat/python/cp\\_model.html#CpModel](https://google.github.io/ortools/python/ortools/sat/python/cp_model.html#CpModel)
- [17] Python. *Python.org* [online]. [cit. 2022-04-23]. Dostupné z: <https://www.python.org/>
- [18] Plotly. *Dash Enterprise* [online]. [cit. 2022-04-25]. Dostupné z: <https://plotly.com/>
- [19] CARVALHO, Thiago. Gantt charts with Python's Matplotlib. *Towards data science* [online]. 2021. [cit. 2022-04-26]. Dostupné z: <https://towardsdatascience.com/gantt-charts-with-pythons-matplotlib-395b7af72d72>

## Seznam obrázků

<b>Obrázek 2.1</b> Pyramida cílů.....	15
<b>Obrázek 3.1</b> Areál společnosti PHARMIX, s.r.o. ....	17
<b>Obrázek 3.2</b> Organizační struktura společnosti .....	18
<b>Obrázek 3.3</b> Průběh zakázky – úsek obchodu.....	22
<b>Obrázek 3.4</b> Průběh zakázky – úsek TPV.....	23
<b>Obrázek 3.5</b> Průběh zakázky – úsek výroby .....	24
<b>Obrázek 3.6</b> Průběh zakázky – úsek řízení kvality.....	25
<b>Obrázek 4.1</b> Původní zobrazení zaznamenaných operací v ERP systému .....	34
<b>Obrázek 4.2</b> Aktuální zobrazení zaznamenaných operací v ERP systému.....	35
<b>Obrázek 4.3</b> ETH terminál sloužící k evidenci operací.....	35
<b>Obrázek 4.4</b> Zobrazení chybně zaznamenaných operací v ERP systému.....	36
<b>Obrázek 5.1</b> Ganttův diagram řešení případové studie.....	54
<b>Obrázek 5.2</b> Číselný zápis řešení případové studie.....	54
<b>Obrázek 5.3</b> Víko .....	55

## Seznam tabulek

<b>Tabulka 4.1</b> Zjednodušená zakázka výroby nádoby .....	31
<b>Tabulka 4.2</b> Rozdělení pracovišť .....	38
<b>Tabulka 4.3</b> Zpracování fiktivní zakázky v programu Microsoft Excel .....	39
<b>Tabulka 5.1</b> Schéma zakázky 0 .....	51
<b>Tabulka 5.2</b> Schéma zakázky 1 .....	52
<b>Tabulka 5.3</b> Legenda popisující označení profesí.....	52

## Seznam zkratk

CP-SAT	-	Constraint Programming-Boolean Satisfiability
TPV	-	Technická příprava výroby
ERP	-	Plánování podnikových zdrojů (Enterprise resource planning)
ETH	-	Ethernet
MILP	-	Smíšené celočíselné lineární programování (Mixed-integer linear Programming)
CP	-	Programování s omezeními (Constraint Programming)
LS	-	Lokální vyhledávání (Local search)
SAT	-	Boolean Satisfiability Solver
LCG	-	Generování líné klauzule (Lazy clause Generation)
PYPL	-	Popularity of Programming Language

## **Seznam příloh**

Příloha 1 – Nástroj pro optimální plánování výroby