



Diplomová práce

Dynamické rozvrhování výroby v TOS VARNSDORF a.s.

Studijní program:

N0788A270004 Inovační a průmyslové inženýrství

Autor práce:

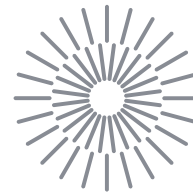
Bc. Michal Podroužek

Vedoucí práce:

Ing. František Koblasa, Ph.D.

Katedra výrobních systémů a automatizace

Liberec 2024



Zadání diplomové práce

Dynamické rozvrhování výroby v TOS VARNSDORF a.s.

<i>Jméno a příjmení:</i>	Bc. Michal Podroužek
<i>Osobní číslo:</i>	S22000264
<i>Studijní program:</i>	N0788A270004 Inovační a průmyslové inženýrství
<i>Zadávací katedra:</i>	Katedra výrobních systémů a automatizace
<i>Akademický rok:</i>	2023/2024

Zásady pro vypracování:

Cílem diplomové práce je vytvořit rozvrhovací algoritmus pro modul systému TOSControl, který bude sloužit tvorbu rozvrhu výroby na strojích v interní výrobě TOS VARNSDORF a.s. Rozvrh výroby bude respektovat různá kritéria dynamického rozvrhování (např. technický stav stroje, odvádějí hotové výroby, délka operací a seřízení) s cílem minimalizovat seřizování (počet a celková doba) a zároveň minimalizovat zpoždění od stanoveného termínu odvedení zakázky zákazníkovi.

1/ Úvod do problematiky rozvrhování výroby včetně jejich dynamických aspektů. Úvod do problematiky vícekritériálního hodnocení. Úvod do systémového inženýrství – popis algoritmizace (např. UML či BPMN).

2/ Analýza současného stavu rozvrhování – datové základny nutné pro tvorbu rozvrhu s ohledem na evidenci a řízení zakázek, technické přípravy výroby (technologický postup včetně časových norem, systému stanovování délky trvání činností), technického stavu strojního zařízení, aj.

3/ Tvorba testovacího datového modelu.

4/ Návrh rozvrhovacího algoritmu a způsobu vícekritériálního hodnocení rozvrhů. Rozbor možností optimalizace dílenského rozvrhu.

5/ Verifikace rozvrhovacího algoritmu na testovacích instancích. Testování a vyhodnocení algoritmu na modelu s reálnými daty.

6/ Vyhodnocené přínosů a podmínek implementace.

Rozsah grafických prací:
Rozsah pracovní zprávy: 50-65
Forma zpracování práce: tištěná/elektronická
Jazyk práce: čeština

Seznam odborné literatury:

- [1] PINEDO, Michael L. Planning and scheduling in manufacturing and services. ISBN 978-1-4419-0909-1.
- [2] OUELHADJ, Djamilia; PETROVIC, Sanja. A survey of dynamic scheduling in manufacturing systems. Journal of scheduling, 2009, 12: 417-431.
- [3] ANDERER, Simon, et al. Meta Heuristics for Dynamic Machine Scheduling: A Review of Research Efforts and Industrial Requirements. IJCCI, 2018, 192-203.
- [4] ANGEL-BELLO, Francisco; VALLIKAVUNGAL, Jobish; ALVAREZ, Ada. Two approaches to handle the dynamism in a scheduling problem with sequence-dependent setup times. Expert Systems with Applications, 2021, 167: 114137.
- [5] COPIL, Karina, et al. Simultaneous lotsizing and scheduling problems: a classification and review of models. OR spectrum, 2017, 39: 1-64.

Vedoucí práce: Ing. František Koblasa, Ph.D.
Katedra výrobních systémů a automatizace

Datum zadání práce: 6. listopadu 2023
Předpokládaný termín odevzdání: 6. května 2025

doc. Ing. Jaromír Moravec, Ph.D.
děkan

L.S.

doc. Ing. Petr Lepšík, Ph.D.
garant studijního programu

V Liberci dne 6. listopadu 2023

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědom toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Dynamické rozvrhování výroby v TOS VARN-SDORF a.s.

Abstrakt

Diplomová práce se zaměřuje na analýzu a implementaci metod dynamického rozvrhování s cílem optimalizovat výrobní procesy ve firmě TOS VARNSDORF a.s. Algoritmus pro rozvrhování výroby byl naprogramován v programovacím jazyce MATLAB. Program využívá strategii reaktivního rozvrhování, kdy je nejprve vytvořen statický rozvrh, který je po narušení systému přerozvrhnut. Pro tvorbu statického rozvrhu byl testován genetický algoritmus a prioritní pravidla SPT a LPT. Pro dynamické změny rozvrhu byly testovány prioritní pravidla LST a CR, které byly testovány vůči simulaci mistra prioritním pravidlem RS. Výstupem této práce je algoritmus pro vytvoření statického rozvrhu a dynamické přerozvrhování, který bude sloužit při vývoji modulu pokročilé správy stroje TOScontrol.

Klíčová slova

Průmysl 4.0, Dynamické rozvrhování, Genetický algoritmus, Prioritní pravidla, MATLAB, Algoritmizace

Dynamic scheduling in TOS VARNSDORF a.s.

Abstract

This master's thesis focuses on implementing and analyzing dynamic scheduling methods to optimize the production of TOS VARNSDORF a.s. The scheduling algorithm was programmed using the MATLAB programming language. The program uses a reactive scheduling strategy. Firstly, the program generates a static schedule using a genetic algorithm. Other methods used in this thesis for comparison were dispatching rules SPT and LPT. For dynamic changes in the schedule, the program uses a modified Giffler-Thompson algorithm for non-delay schedules with the addition of dispatching rules LST and CR. The dynamic algorithm was compared against non-delay schedules using the RS priority rule. The outcome of this thesis is algorithms for creating static schedules and algorithms for dynamic rescheduling, which will be used to develop an advanced machine control system, TOScontrol.

Keywords

Industry 4.0, Dynamic scheduling, Genetic algorithm, Dispatching rules, MATLAB, Algorithms

Poděkování

Rád bych touto cestou poděkoval Ing. Františkovi Koblasovi, Ph.D. za cenné rady a odborné vedení této diplomové práce. Zároveň bych chtěl poděkovat firmě TOS VARNSDORF a.s. za poskytnuté materiály pro zpracování diplomové práce a spolupráci na vývoji algoritmů. Dále bych chtěl poděkovat své rodině a přátelům za podporu při psaní této práce a její korekturu.

Obsah

1	ÚVOD	13
2	POPIS ROZVRHOVACÍCH MODELŮ	14
2.1	DRUHY ROZVRHOVACÍCH MODELŮ	14
2.1.1	<i>Otevřený rozvrhovací problém (Open Shop Scheduling Problem).....</i>	<i>14</i>
2.1.2	<i>Proudový rozvrhovací problém (Flow Shop Scheduling Problem)</i>	<i>15</i>
2.1.3	<i>Sekvenční rozvrhovací problém (Job Shop Scheduling Problem).....</i>	<i>16</i>
2.2	ÚČELOVÁ FUNKCE.....	17
2.3	TŘÍDY ROZVRHOVACÍCH MODELŮ	19
2.3.1	<i>Rozvrh bez zpoždění.....</i>	<i>20</i>
2.3.2	<i>Aktivní rozvrh</i>	<i>20</i>
2.3.3	<i>Semi-aktivní rozvrh.....</i>	<i>20</i>
3	DYNAMICKÉ ROZVRHOVÁNÍ.....	21
3.1	DYNAMICKÉ UDÁLOSTI.....	21
3.2	STRATEGIE DYNAMICKÉHO ROZVRHOVÁNÍ.....	22
3.2.1	<i>Úplně reaktivní rozvrhování.....</i>	<i>22</i>
3.2.2	<i>Prediktivně-reaktivní rozvrhování</i>	<i>22</i>
3.2.3	<i>Proaktivní rozvrhování</i>	<i>22</i>
3.3	VYHODNOCOvacÍ UKAZATELE DYNAMICKÉHO JSSP	23
3.4	ROZVRHOVACÍ METODY DYNAMICKÉHO ROZVRHOVÁNÍ.....	23
3.4.1	<i>Přesné metody</i>	<i>23</i>
3.4.2	<i>Přibližné metody řešení.....</i>	<i>24</i>
3.5	KDY MĚNIT ROZVRH	36
4	VÍCEKRITERIÁLNÍ HODNOCENÍ ROZVRHU	39
4.1	MODEL VÁŽENÉHO SOUČTU (WSM)	39
4.2	MODEL VÁŽENÉHO SOUČINU (WPM)	40
5	SOUČASNÝ STAV VE FIRMĚ TOS VARNSDORF A.S.	42
5.1	SOUČASNÝ STAV ROZVRHOVÁNÍ VE FIRMĚ TOS VARNSDORF	43
5.2	MAPOVÁNÍ INFORMAČNÍCH TOKŮ.....	44
5.2.1	<i>Oprava obráběcích hlav</i>	<i>44</i>
5.2.2	<i>Výroba obráběcích hlav.....</i>	<i>45</i>
5.2.3	<i>Výroba na těžké mechanice</i>	<i>46</i>
5.3	MODUL TOSCONTROL	47
5.4	POŽADAVKY NA SYSTÉM ROZVRHOVÁNÍ (HODNOCENÍ KVALITY ROZVRHU).....	52

5.5	ANALÝZA DATOVÉ ZÁKLADNY.....	52
5.5.1	<i>Tvorba testovacího datového modelu.....</i>	53
5.5.2	<i>Informace o zakázkách.....</i>	53
6	NÁVRH ALGORITMŮ.....	57
6.1	STATICÝ ROZVRH.....	57
6.1.1	<i>Prioritní pravidla</i>	61
6.1.2	<i>Genetický algoritmus</i>	61
6.2	DYNAMICKÝ ROZVRH	67
6.2.1	<i>Příchod nové zakázky</i>	69
6.2.2	<i>Porucha výrobního stroje</i>	70
6.2.3	<i>Dynamická změna výrobní přesnosti strojů</i>	71
6.2.4	<i>Dynamické přerozdělení pomocí prioritních pravidel.....</i>	72
7	TESTOVÁNÍ ALGORITMŮ	74
7.1	TESTOVÁNÍ ALGORITMŮ NA TEORETICKÝCH MODELECH.....	74
7.2	TESTOVÁNÍ NA REÁLNÉM MODELU	77
7.3	STANOVENÍ VAH VÍCEKRITÉRIÁLNÍHO HODNOCENÍ.....	79
7.4	TESTOVÁNÍ DYNAMICKÉHO ROZVRHOVÁNÍ	79
7.5	VYHODNOCENÍ VÝSLEDKŮ.....	83
7.5.1	<i>Statický rozvrh.....</i>	83
7.5.2	<i>Dynamický rozvrh.....</i>	84
7.6	IMPLEMENTACE ALGORITMU.....	85
8	ZÁVĚR.....	87
9	POUŽITÁ LITERATURA	89
10	SEZNAM OBRÁZKŮ	94
11	SEZNAM TABULEK	96
12	SEZNAM PŘÍLOH.....	97

Seznam zkratek

3D	Trojrozměrný
A	Aktivní rozvrh
A_i	Alternativa i
a_{ij}	Hodnota kritéria j zakázky i
APS	Pokročilý plánovací systém
BPML	Modelovací jazyk firemních procesů
BPMN	Modelování a notace firemních procesů
C	Průběžný čas výroby
CD	Vzdálenost řešení
C_j	Čas dokončení zakázky
C_{max}	Maximální dokončení všech zakázek
CNC	Počítačové číslicové řízení
CR	Kritický poměr
d_j	Termín odvedení zakázky
DR	míra diskontinuity
EDD	Nejdřívější datum odvedení
e_{ij}	Termín zahájení operace
ERP	Celopodnikové plánování zdrojů
F	Celkový čas výroby zakázek
FCFS	První příchozí je první obslužený
FIFO	První dovnitř, první ven
f_{ij}	Termín ukončení operace
FJSSP	Flexibilní sekvenční rozvrhovací problém
FSSP	Proudový rozvrhovací problém
GA	Genetický algoritmus
H-MOIA	Hybridní víceúčelový imunitní algoritmus
J	Zakázka
JSR	Nejmenší poměr délky rezervy
JSSP	Sekvenční rozvrhovací problém
L_j	Zpoždění zakázky
L_{max}	Maximální zpoždění všech zakázek
LPT	Nejdelší procesní čas

LSh	Posun zakázky vlevo
LST	Časová rezerva do data odvedení
LWKR	nejméně práce zbývá
M	Stroj
MES	Výrobní informační systém
MIDOS	Minimální dopad narušené operace na rozvrh
MIROS	Minimální dopad přerozvrhnuté operace na rozvrh
MS	Microsoft
MWKR	Nejvíce práce zbývá
ND	Rozvrh bez zpoždění
n_{ij}	Normalizovaná hodnota kritéria j zakázky i
NSGA-II	Genetický algoritmus s nedominovaným tříděním
O	Operace
OEE	Celková efektivnost zařízení
OSSP	Otevřený rozvrhovací problém
P_i	Preference zakázky
p_{ij}	Procesní čas operace
P_j	Jednotková penalizace zakázky
PP	Prioritní pravidlo
PSO	Optimalizace rojením částic
QA	Kvantové žíhání
r_j	Termín uvolnění zakázky
RS	Náhodný výběr
RSh	Posun zakázky vpravo
S-1/OP	Rezerva zbývajících času na počet operací
SA	Semi aktivní rozvrh
SCADA	Dispečerské řízení a sběr dat
SGR	Redukce posouvající se mezery
SPT	Nejkratší procesní čas
T	Průměrná odchylka od termínu zakázek
T_j	Odchylka od termínu zakázky
TOS	Továrny obráběcích strojů
U	Průměrné využití strojů
ÚF	Účelová funkce

U_k	Využití stroje
UML	Unifikovaný modelovací jazyk
W_j	Váha kritéria j
WPM	Model váženého součinu
WSM	Model váženého součtu

1 Úvod

Tato diplomová práce se zaměřuje na analýzu a implementaci metod dynamického rozvrhování s cílem optimalizovat výrobní procesy ve firmě TOS VARNSDORF a.s. Primární zaměření společnosti je na vývoj a výrobu horizontálních frézovacích a vyvrtávacích strojů, obráběcích center a portálů.

V současném průmyslovém prostředí, kde konkurenceschopnost firem značně ovlivněna jejich schopností reagovat rychle a efektivně na změny, se dynamické rozvrhování stává nezbytným nástrojem. Dynamické rozvrhování je jedním z klíčových aspektů moderního řízení výroby, který umožňuje flexibilní a efektivní přizpůsobení výrobních plánů měnícím se podmínkám a požadavkům. Tento přístup k rozvrhování se odlišuje od statických metod svou schopností reagovat na nečekané události, jako jsou poruchy strojů, změny priorit zakázek či variace ve výrobní kapacitě.

Cílem diplomové práce je navrhnout rozvrhovací algoritmus pro modul systému TOS-control, který bude sloužit k tvorbě rozvrhu výroby na strojích v interní výrobě TOS VARNSDORF a.s. Rozvrh výroby bude respektovat různá kritéria dynamického rozvrhování (např. technický stav stroje, odvádění hotové výroby, délku operací a seřízení) s cílem minimalizovat zpoždění od stanoveného termínu odvedení zakázky zákazníkovi a zároveň minimalizovat seřizování.

Dílní cíle práce obsahují:

- uvedení do problematiky rozvrhování výroby včetně jejich dynamických aspektů,
- uvedení do problematiky vícekriteriálního hodnocení rozvrhů,
- uvedení do systémového inženýrství (popis algoritmizace),
- analýza současného stavu techniky,
- analýza současného stavu rozvrhování a datové základny,
- tvorba testovacího datového modelu,
- návrh rozvrhovacího algoritmu,
- návrh způsobu vícekriteriálního hodnocení rozvrhů,
- verifikace rozvrhovacího algoritmu na testovacích instancích,
- testování a vyhodnocení algoritmu na modelu s reálnými daty,
- vyhodnocení přínosů a podmínek implementace.

Teoretická část

2 Popis rozvrhovacích modelů

V literatuře zabývající se teorií rozvrhování výroby se často mluví o rozvržení n zakázek J_i , kde $i = 1, 2, \dots, n$ na m strojů M_j , kde, $j = 1, 2, \dots, m$. Rozvržení zakázky znamená přiřazení zakázky na jeden nebo více strojů v jednom nebo více časových intervalech. Pokud zakázka J vyžaduje rozvržení více o operací O_k , kde $k = 1, 2, \dots, o$, pak je odkazováno na rozvrhovací operaci zakázky j na stroji i ve formě $O_k(ij)$. Časové informace operace nesou proměnné procesní čas p_{ij} , termín zahájení operace e_{ij} a doba ukončení operace f_{ij} . Zavádí se také notace modelů ve formě $\alpha|\beta|\gamma$ [1]. Problémem notace rozvrhovacích modelů je popsán například v práci Pineda M.[2]:

- α udává druh rozvrhovacího modelu,
- β udává omezení rozvrhovacího modelu,
- γ udává optimalizovanou účelovou funkci.

Notace $FJ|brkdn|\Sigma T_j$ pak bude popisovat flexibilní sekvenční problém s možností výskytu poruchy stroje a optimalizací celkové odchylky od termínů zakázek. V polích pro β a γ může být více notací zároveň.

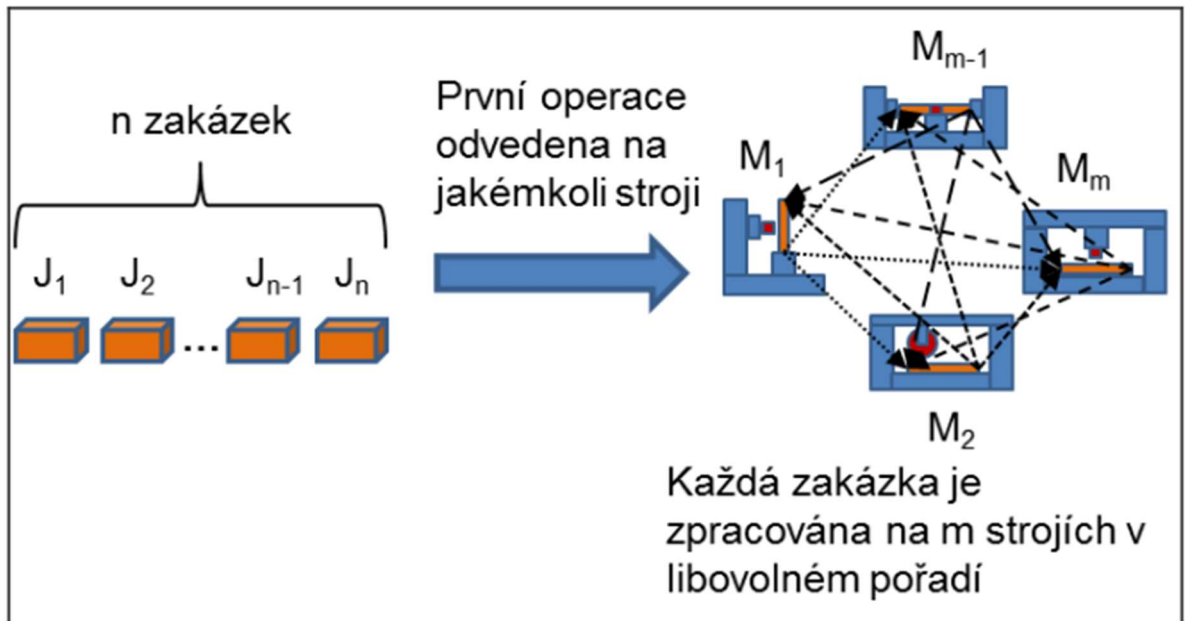
2.1 Druhy rozvrhovacích modelů

Druh rozvrhovacího problému je dán notací α , ve kterém je často specifikován počet výrobních strojů (J3 – sekvenční problém se třemi stroji).

2.1.1 Otevřený rozvrhovací problém (Open Shop Scheduling Problem)

V otevřeném rozvrhovacím problému (OSSP Obrázek 1, notace: $\alpha=0$) figuruje v modelu n nezávislých zakázek a m paralelních strojů. Každá zakázka $i = 1, 2, \dots, n$, má předem dané operace O_{ij} , $j = 1, 2, \dots, m$, kde každá operace O_{ij} zakázky i musí být zpracována na stroji j za čas p_{ij} . Každý stroj může najednou zpracovávat pouze jednu operaci a zároveň může být prováděna pouze jedna operace ze zakázky. Nemůže se tedy stát, že se najednou na stroji vykonávají operace na dvou zakázkách a nemůže zároveň nastat v rozvrhu případ, kdy je v jeden moment předepsána zakázka na více strojů zároveň. Operace každé zakázky mohou být zpracovávány v libovolném pořadí. Tento problém je často řešen

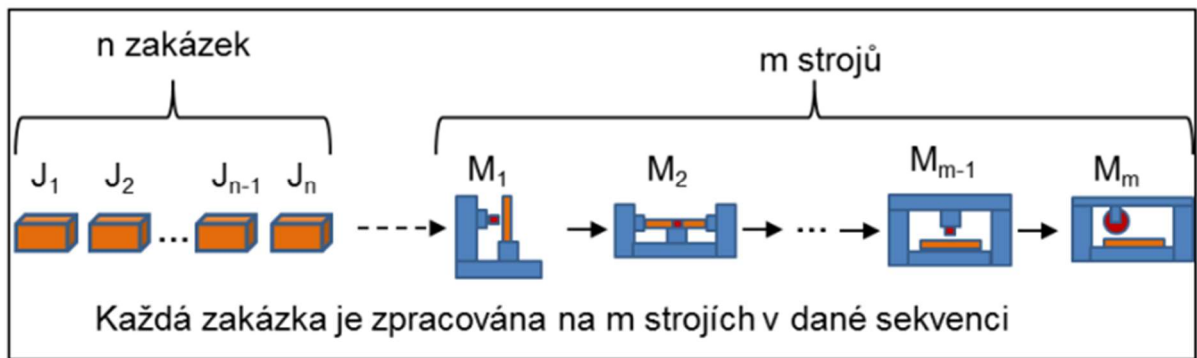
v případech jako jsou například rozvrhy hodin ve školách nebo údržbářské práce. Ačkoliv tento model přináší větší svobodu při tvoření rozvrhů, tak zároveň velice zvyšuje množství možných řešení [1, 3].



Obrázek 1 Otevřený rozvrhovací problém [1]

2.1.2 Proudový rozvrhovací problém (Flow Shop Scheduling Problem)

V proudovém rozvrhovacím problému (FSSP Obrázek 2, notace: $\alpha=F$) figuruje v modelu m strojů v sérii. Každá ze zakázek musí být zpracována na každém ze strojů a všechny zakázky musejí následovat stejnou cestu výrobním systémem. Kupříkladu musejí být všechny zakázky být zpracovány na stroji A poté na stroji B a tak dále. Ve chvíli, kdy je zakázka dokončena na jednom stroji, tak se zařadí do fronty práce následujícího stroje. V této frontě se pak chovají zakázky zpravidla podle pravidla FIFO (First In, First Out ~ první příchozí je prvním obsluženým)[1, 2]. Jedním z příkladů proudového problému jsou výrobní linky.



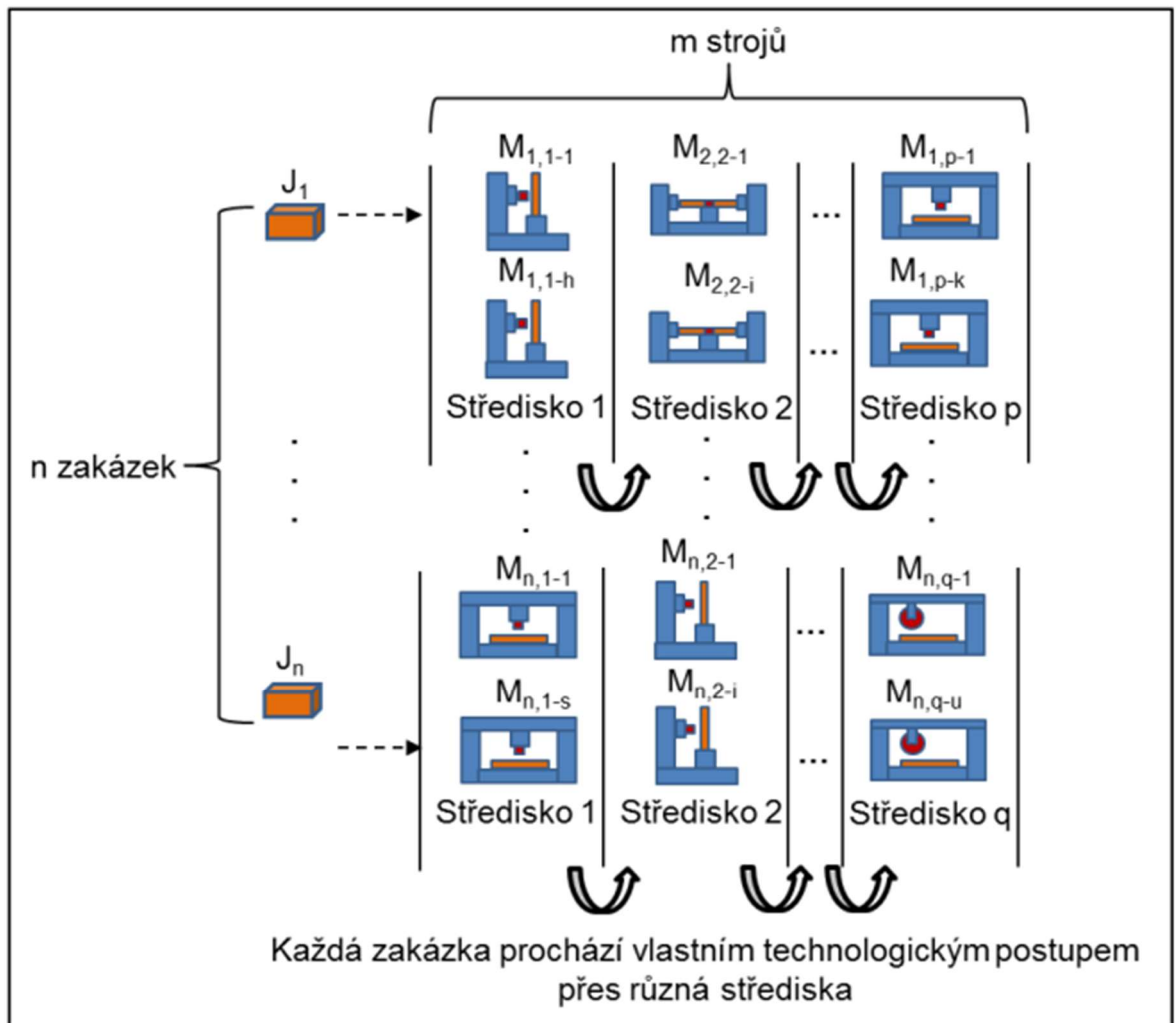
Obrázek 2 Proudový rozvrhovací problém [1]

2.1.3 Sekvenční rozvrhovací problém (Job Shop Scheduling Problem)

V modelu sekvenčního problému (JSSP **Chyba! Nenalezen zdroj odkazů.**, notace: $\alpha=J$) se uvažuje m strojů zpracovávajících n zakázek. V sekvenčním problému má každá zakázka jiný výrobní postup čili jinou sekvenci strojů, na kterých má být zpracována.

Sekvenční problém se dělí na problém, kdy se žádná zakázka nevrací na již využitý stroj a na problémy, kdy k tomuto jevu dochází[2]. Jedná se o nejobecnější model rozvrhování výroby, u kterého pak označení strojů M_{nm} zachycuje n jako číslo zakázky a m pořadí operace v sekvenci zakázky.

V praxi, zejména ve strojírenské výrobě, je časté rozšíření problému přidáním alternativních strojů do technologického postupu. Operace pak mohou být zpracovávány nejen na primárně určeném stroji, ale i na jiných. Často nastává problém, kdy je na alternativních strojích jinak dlouhý čas seřízení, nebo procesní čas. Tento problém je pak označován jako flexibilní sekvenční problém (FJSSP Obrázek 3, notace: $\alpha=FJ$). Oproti klasickému sekvenčnímu problému zde roste složitost problému, jelikož je nutné nejen rozvrhovat pořadí operací na jednotlivých výrobních střediscích, ale také je nutné přiřadit operaci na daný stroj.



Obrázek 3 Flexibilní sekvenční rozvrhovací problém[1]

2.2 Účelová funkce

Notace γ ukazuje na účelovou funkci, která je minimalizována. Čas dokončení operace zakázky j na stroji i je označována jako C_{ij} . Čas dokončení zakázky je označován jako C_j , čas začátku zpracování zakázky je r_j a termín odvedení zakázky jako d_j . Literatura [2, 4, 5] identifikuje několik základních cílů, které je třeba minimalizovat:

- L_j (*lateness*) = zpoždění zakázky j (viz (1)),
- T_j (*tardiness*) = odchylka od termínu zakázky j (viz (2)),
- P_j (*unit penalty*) = jednotková penalizace zakázky j (viz (3)).

Následující rovnice ((1 až (8) ukazují příklady výpočtů účelových funkcí zohledněných v práci.

zpoždění zakázky:

$$L_j = C_j - d_j \quad (1)$$

odchylka od termínu zakázky:

$$T_j = \max(C_j - d_j, 0) = |L_j| \quad (2)$$

jednotková penalizace zakázky:

$$P_j = \begin{cases} 1 & \text{pokud } C_j > d_j \\ 0 & \text{v opačném případě} \end{cases} \quad (3)$$

průměrná odchylka od termínů zakázek:

$$T = \frac{1}{n} \sum_{j=1}^n T_j \quad (4)$$

využití zdrojů:

$$U_k = \frac{\text{Celkový čas užívání zdroje}}{\max(C_i)} \quad (5)$$

průměrné využití zdrojů:

$$U = \frac{1}{m} \sum_{k=1}^m U_k \quad (6)$$

průběžný čas výroby:

$$C = \frac{1}{n} \sum_{j=1}^n \sum_{i=1}^o C_{ij} \quad (7)$$

celkový čas výroby zakázek:

$$F = \sum_{j=1}^n (C_j - r_j) \quad (8)$$

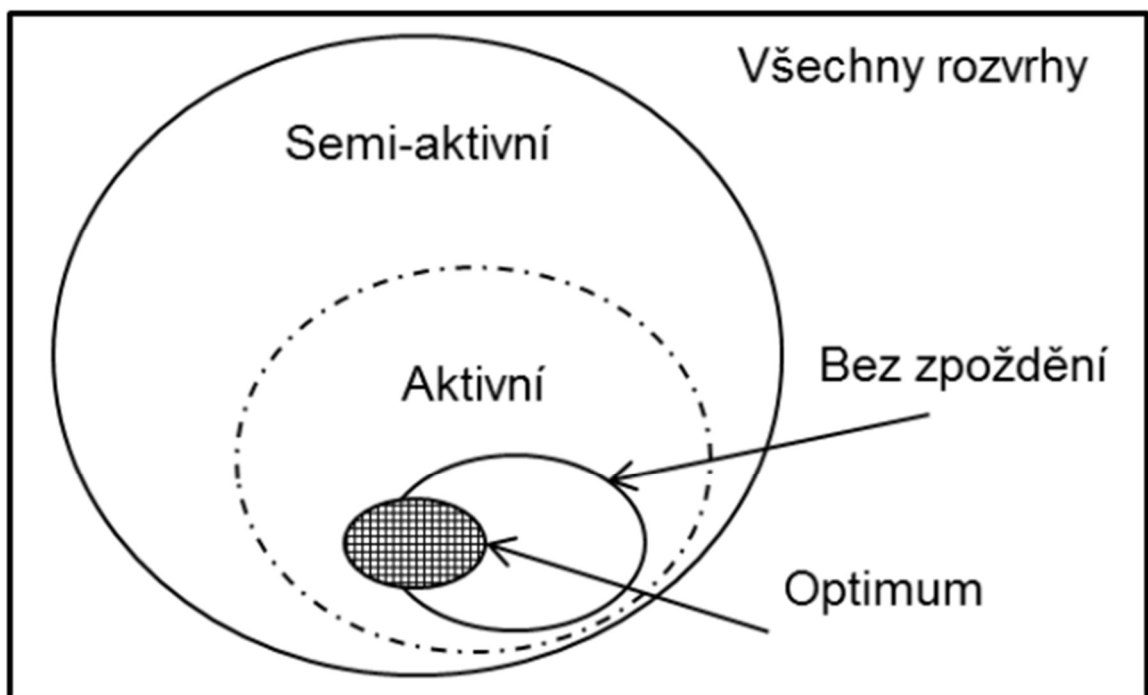
Notace γ může obsahovat dále například[1, 2]:

- $\gamma = C_{max} \sim (\text{Makespan})$: Maximální dokončení všech zakázek. Definována jako $\max(C_1, \dots, C_n)$. Je ekvivalentní s časem dokončení poslední zakázky.

- $\gamma=L_{max}$: Maximální zpoždění všech zakázek. Definována jako $\max(L_1, \dots, L_n)$. Ukazuje nejhorší zpoždění zakázek.
- $\gamma=\Sigma w_j C_j$: Celkový vážený čas dokončení všech zakázek. Často brán pouze jako ΣC_j a označován jako celkový průběžný čas všech zakázek.
- $\gamma=\Sigma w_j T_j$: Celková vážená odchylka od termínu. Často brána pouze jako ΣT_j a označována jako celková odchylka od termínu.
- $\gamma=\Sigma w_j P_j$: Celková vážená penalizace za nedodržení termínu. Často brány pouze jako ΣP_j a označovány jako celková penalizace za nedodržení termínu.

2.3 Třídy rozvrhovacích modelů

V rozvrhovací terminologii je kladen důraz na rozdíly mezi pojmy sekvence, rozvrh a rozvrhovací strategie. Jako sekvence je označováno pořadí zakázek přiřazených na daný stroj, rozvrh je komplexnější náhled na problém. V rozvrhu je určeno, v jaký čas a na jakém stroji budou zakázky zpracovány. Rozvrhovací strategie je používána převážně ve stochastickém prostředí a rozhoduje, k jaké reakci dojde ve všech stavech systému[1, 2]. Obrázek 4 ukazuje Vennův diagram tříd rozvrhovacích modelů.



Obrázek 4 Třídy rozvrhovacích modelů [1]

2.3.1 Rozvrh bez zpoždění

Rozvrh bez zpoždění (anglicky Non-Delay Shedule – ND) je takový proveditelný rozvrh, ve kterém nezůstává žádný zdroj nevytížen, když je dostupná operace pro zpracování. Ekvivalentně lze říct, že jsou zakázány nevynucené nečinnosti. Pro mnoho modelů existují optimální rozvrhy, které jsou právě rozvrhy bez zpoždění. Je ale nutné říct, že existují rozvrhy, ve kterých jsou nevynucené nečinnosti žádané[1, 2].

2.3.2 Aktivní rozvrh

Aktivní rozvrh (anglicky Active Schedule – A) je takový rozvrh, ke kterému nelze vytvořit alternativní rozvrh skrze změnu sekvence operací na stroji, aby ve výsledném alternativním rozvrhu skončila alespoň jedna operace dříve než v původním rozvrhu a žádná z operací neskončila později než v původním rozvrhu. Jinými slovy nelze přesunout operaci na stroji do okna jeho nečinnosti tak, aby byla dodržena jeho splnitelnost, tedy aby byly dodrženy postupy zakázek[1, 2]. Ačkoliv i aktivní rozvrh může být optimální, tak budou v práci uvažovány převážně rozvrhy bez zpoždění, jelikož aktivní rozvrhy pracují s tím, že všechny zakázky budou dostupné, tak rozvrh bez zpoždění pracuje s těmi, které jsou dostupné právě v daném časovém okamžiku.

2.3.3 Semi-aktivní rozvrh

Semi-aktivní rozvrh (anglicky Semi-Active Schedule – SA) je takový rozvrh, ve kterém nelze operaci dokončit dříve tak, aby nedošlo ke změně sekvence operací na stroji. Každý aktivní rozvrh je rozvrhem semi-aktivním ale nikoliv naopak[2].

3 Dynamické rozvrhování

V reálném výrobním systému je prostředí z větší části dynamické a je ovlivňováno například neplánovanými poruchami a nepravidelnou pracovní docházkou (absencí). Tyto vlivy jsou nevyhnutelné a nastolují potřebu řešení pomocí dynamického rozvrhování. V současnosti je většina prací založena na statické poptávce a deterministických procesních časech, kdežto většina JSSP se zabývá dynamickou poptávkou a stochastickými procesními časy [6].

- Deterministické = dovolují výpočet budoucí události bez přítomnosti náhody. Modely vykazují vždy stejné chování a jejich výstup je vždy předem určený.
- Stochastické = výpočet budoucí události je ovlivněn náhodou, výsledky modelu jsou vždy jiné i když jsou počáteční podmínky vždy stejné[7].

Statický a dynamický model rozvrhování byl poprvé definován Jacksonem v roce 1957 [8]. Statické rozvrhování zpracovává rozvrh, kde jednou zpracovaný rozvrh již nemění svoji podobu. Ve skutečném prostředí, z důvodu neustálých změn v provozním prostředí a z důvodu výskytu rušivých faktorů jako jsou zpracování při přesčase, naléhavá objednávka nebo nepřesné odhady času zpracování, se stávají plány často rychle zastaralé. V souladu s těmito změnami ve skutečných podmínkách musíme neustále upravovat plán výroby. Toto je oblast zkoumání dynamického rozvrhování zdrojů[6].

3.1 Dynamické události

Stochastické události, které využívá dynamické rozvrhování se nazývají dynamické události a jsou rozděleny do čtyřech tříd[9].

- Události spjaté s vyráběným dílem:
 - neurčitý procesní čas dílu,
 - datum přejímky se nenadále mění,
 - priorita vyráběných dílů a objednávky se nenadále mění.
- Události spjaté se stroji:
 - poškození stroje omezující jeho dostupnost,
 - konflikty mezi kapacitou produkce a skutečným využitím stroje.
- Události spjaté s výrobním procesem:
 - výrobní proces je zpožděn,
 - kontrola kvality odmítla kus,

- Další:
 - absence pracovníka,
 - zpoždění dodávky materiálu,
 - příchod spěšné zakázky,
 - stornování zakázky.

3.2 Strategie dynamického rozvrhování

Strategie dynamického rozvrhování lze roztrždit na základě toho, jak reagují na změny ve výrobním prostředí na:

- úplně reaktivní rozvrhování,
- prediktivně-reaktivní rozvrhování,
- proaktivní rozvrhování.

3.2.1 Úplně reaktivní rozvrhování

Nevyužívá před-plánovací schéma. Rozvrhovací strategií je generace nového rozvrhu na základě výskytu dynamických událostí ve výrobě. Jelikož úplně reaktivní plánování vytváří výrobní rozvrh bez zvažování možných budoucích dynamických událostí, nemůže poskytovat plánovací měřítko pro příslušné aktivity dílny. Toto má za důsledek časté úpravy rozvrhovacího plánu a tím zvýšené náklady na rozvrhování při výskytu dynamické události [6].

3.2.2 Prediktivně-reaktivní rozvrhování

Prediktivně-reaktivní rozvrhování je běžnou strategií, která zahrnuje dva kroky postupu:

- generace před-plánovacího schématu bez ohledu na dynamické události, které se vyskytnou ve výrobní dílně v budoucnosti,
- aktualizace před-plánovacího plánu, což je spouštěno dynamickými událostmi a dynamickým plánováním, které může zlepšit rozvrhovací výkon [6].

3.2.3 Proaktivní rozvrhování

Proaktivní rozvrhování je strategický přístup, který vytváří robustní základní rozvrhy, které eliminují jejich potenciální narušení. Tyto rozvrhy předvídají na statistickém

základě změny v rozvrhu. Proaktivní rozvrhování je tvořené například algoritmy používající neuronové sítě[10].

3.3 Vyhodnocovací ukazatele dynamického JSSP

Ukazatele dynamického JSSP jsou rozděleny do několika skupin [6].

- Časové ukazatele:
 - doba dodání zakázky,
 - doba dokončení zakázky,
 - průměrná doba výroby a další.
- Ekonomické ukazatele:
 - náklady na výrobu,
 - penále za nedodání výrobku včas,
 - skladovací náklady pro dřívější dokončení výrobku a další.
- Systémové ukazatele:
 - dostupnost zařízení,
 - produktivitu a další.

Obvyklé plánování nemůže brát v potaz všechny indikátory, ale dělá rozhodnutí na základě specifických požadavků, ze kterých vybírá [6].

3.4 Rozvrhovací metody dynamického rozvrhování

Tato část diplomové práce se věnuje rešerši algoritmů použitých při řešení problému dynamického rozvrhování zdrojů v sekvenčních problémech rozvrhování.

3.4.1 Přesné metody

Metody přesného řešení vytvářejí jeden nebo více optimalizačních modelů cílové funkce, aby získaly optimální řešení. Převádí rozvrhovací problémy na matematické problémy omezení a získávají vždy optimální řešení. Většina plánovacích problémů je již prokázána jako NP problémy (nejsou řešitelné v polynomiálním čase). Jak se problém plánování rozrůstá s rostoucím počtem strojů, jeho možné řešení se zvyšuje exponenciálně a přesné metody nemohou snadno poskytnout optimální řešení v přijatelném čase. Proto přesné metody nelze použít k řešení praktických problémů [6].

3.4.2 Přibližné metody řešení

Přibližné metody řešení hledají na základě jejich algoritmu takové řešení, které se blíží optimálnímu rozvrhu, je splnitelné a může být následně přijatelným rozvrhem. Jsou používány zejména z důvodu nemožnosti nalezení přesného řešení u velké většiny praktických problémů.

3.4.2.1 Prioritní pravidla

Nejtradičnější metodou plánování výrobních a zpracovatelských úloh je prioritní pravidlo. Nejstarší prioritní pravidlo zavedl Jackson[11], mohlo by být použito v dynamickém systému plánování v reálném čase kvůli své jednoduchosti, srozumitelnosti, nízké výpočetní náročnosti atd. V posledních letech vzniká mnoho nových pravidel [6]. Jeho nevýhodou je pak nemožnost zaručení nalezení optimálního rozvrhu.

Jacksonovo prioritní rozvrhovací pravidlo je heuristické pravidlo, které je široce používáno v problémech s jedním strojem a u JSSP problémů. Jackson (1955) představil pravidlo EDD (Earliest Due Date), které říká, že při rozvrhování nezávislých úkolů na jeden stroj lze dosáhnout optimálního řešení tím, že se nejprve rozvrhují zakázky s bližším datem odvedení[11]. Rozšířené Jacksonovo pravidlo funguje na principu minimalizace makespanu, při zjednodušování JSSP problému na FSSP problém. V notaci zapsané jako $J2|n_i \leq 2|C_{\max}$ na $F2||C_{\max}$. Tento problém jde již za pomoci Johnsonova pravidla zpracovat v polynomiálním čase. Johnsonovo pravidlo je pravidlo pro vytvoření sekvence zakázek ve FSSP problému na základě plánování nejkratší operace prvního stanoviště první a nejkratší operace druhého stanoviště poslední[12].

Tyto algoritmy rozvrhují postupně operace, na základě výběru právě dostupných operací. Dostupná operace je taková, která je stále nerozvrhnutá a všechny operace, které jí mají předcházet jsou již v rozvrhu, nebo dokončeny. Tyto algoritmy generují rozvrhy například na základě algoritmu Gifflera a Thompsna a jeho modifikace pro rozvrhy bez zpoždění. V praxi je potřeba dbát na další omezení, jako jsou směnnost nebo čas uvolnění materiálu do výroby (a jiné). Z těchto dostupných operací se vybere stroj, na který je předepsána operace s předpokládaným nejdřívějším časem dokončení. Dále se poté podle prioritního pravidla rozhodne, kterou další operaci na tento stroj zanést. Toto rozhodnutí je provedeno na základě specifikovaného prioritního pravidla, které je určeno z požadavků na výrobu, a zároveň z dat z informačního systému, které umožňují ruční přiřazení

priorit důležitým zakázkám na základě ekonomických faktorů jako jsou například penále za nedodržení termínu nebo požadovaný termín odvedení zakázky. V případě nutnosti výroby dvou zakázek, kterým byla předepsána stejná priorita (mají například stejné datum dodání zákazníkovi), využívají se prioritní pravidla. Ty vycházejí z vlastností daného rozvrhu, respektive technologického postupu[1].

Prioritní pravidla se standardně využívají pro tvorbu statických rozvrhů. Jejich jednoduchost lze využít například pro tvorbu prvotního plánu reaktivního plánování. Výzkum v této oblasti probíhá již dlouhou dobu například Iskanderem a Panwalkerem [13] a jejich známo velké množství. Patří mezi ně například:

- SPT (Shorter Processing Time) operace s nejkratším procesním časem. Prioritu na stroji dostává operace s nejkratší dobou zpracování.
- LPT (Longest Processing Time) operace s nejdelším procesním časem. Prioritu na stroji dostává operace s nejdelší dobou zpracování.
- MWKR (Most Work Remaining) zakázka, na které zbývá nejvíce práce. Prioritu na stroji dostává zakázka s nejdelším zbývajícím časem.
- LWKR (Least Work Remaining) zakázka, na které zbývá nejméně práce. Prioritu na stroji dostává zakázka s nejkratším zbývajícím časem.
- FCFS (First Come First Served = FIFO) první zakázka, která přijde, je první zpracovávanou zakázkou.
- RS (Random Select) náhodný výběr zakázky

V dynamickém rozvrhování dochází při nenadálých změnách ve výrobním systému ke komplikacím, které je nutné řešit mírně modifikovanými pravidly. Nastává v nich také potřeba dynamicky měnit priority jednotlivých zakázek a operací. Konstruktivní algoritmus využívající tato pravidla musí být tedy schopen dynamicky měnit priority jednotlivých zakázek. Často se hovoří o pojmu „Slack time“ což je časová rezerva, kterou má zakázka mezi plánovaným datem odvedení zákazníkovi a datem dokončení její poslední operace. Pro dynamické rozvrhování lze použít například tyto prioritní pravidla:

- EDF (Earliest Deadline First) zakázky jsou priorityzovány na základě datumů jejich odvedení. Nejvyšší prioritu zde dostávají zakázky s nejbližším datem odvedení zákazníkovi[14].

- LST (také S-1, Least Slack Time) zakázky jsou priorityzovány na základě jejich zbývajících časových rezerv do data odvedení zákazníkovi pro zbývajících operace[13].
- S-1/OP (Slack Time/Operations) zakázky jsou priorityzovány na základě podílu rezervy zbývajících času a počtu operací. Nejvyšší priorita je přidělena zakázce s nejnižší hodnotou podílu[13].
- JSR (Least Job Slack Ratio) zakázky jsou priorityzovány na základě podílu rezervy zbývajících času a celkového času do data odvedení zákazníkovi[13].
- CR (Critical ratio) – zakázky jsou priorityzovány na základě poměru zbývajících času do odvedení zákazníkovi a zbývajících času výroby[5].

V následující tabulce (Tabulka 1) je přehled některých rozvrhovacích pravidel (rovnice (9) až (19)) s matematickým vyjádřením indexu priorit π . Index priorit je poté přiřazen každé dostupné operaci a dle pravidla je poté z množiny indexů π vybrána operace pro zařazení do rozvrhu. Notace v tabulce je následující:

- Index i = na číslo zakázky
- Index k = stroj přiřazený operaci
- $p_{i,k}$ = procesní čas operace
- R_i = suma zbývajících procesních časů zakázky i
- no_i = počet zbývajících operací zakázky i
- d_i = datum odvedení zakázky i
- $O_{i,dost.}$ = množina dostupných operací
- t = aktuální čas

Tabulka 1 Přehled prioritních pravidel

Prioritní pravidlo	Matematické vyjádření
Statické rozvrhování	
SPT	$\pi_{i,k} = p_{i,k}$ (9)
LPT	$\pi_{i,k} = p_{i,k}$ (10)
MWKR	$\pi_{i,k} = R_i$ (11)
LWKR	$\pi_{i,k} = R_i$ (12)
FCFS	$first\{O_{i,dost.}\}$ (13)
RS	$rnd\{O_{i,dost.}\}$ (14)
Dynamické rozvrhování	
EDF	$\pi_{i,k} = d_i$ (15)
LST	$\pi_{i,k} = d_i - t - R_i$ (16)
S-1/OP	$\pi_{i,k} = \frac{d_i - t - R_i}{no_i}$ (17)
JSR	$\pi_{i,k} = \frac{d_i - t - R_i}{d_i - t}$ (18)
CR	$\pi_{i,k} = \frac{d_i - t}{R_i}$ (19)

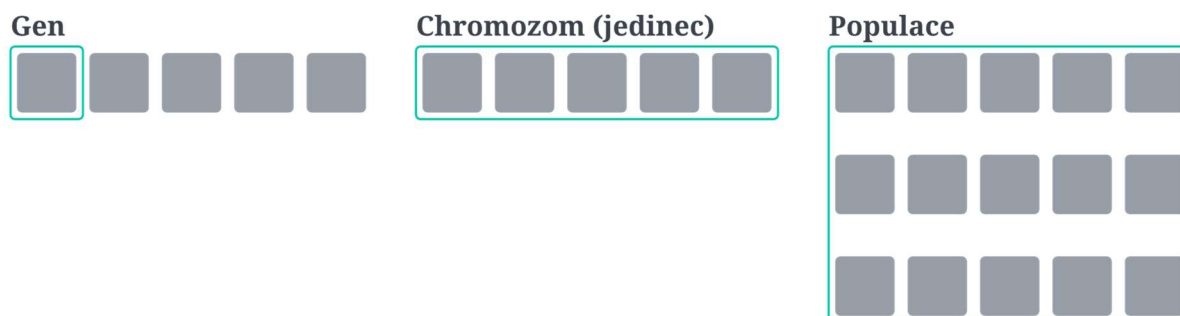
3.4.2.2 Heuristika hrdla lahve

Bottleneck (hrdlo lahve) je termín označující nejužší místo v systému, respektive stroj, který svou produkcí definuje maximální čas dokončení všech zakázek. Hlavní princip této metody je přerozvrhování výrobních zakázek na úzkém místě, které se po přerozvrhování přesouvá na jiný zdroj. Podobně jako konstruktivní algoritmus Gifflera a Thompsona je tento algoritmus iterativní. Nerozvrhuje však zakázky jednu po druhé, ale tvoří fixovaný rozvrh pro samostatné stroje[1].

3.4.2.3 Genetické algoritmy (GA)

Výhodami GA jsou vyspělá technologie kódování a jednoduché operace. Při řešení velkých kombinatorických optimalizačních problémů ale program existuje ve velkém prohledávacím prostoru. Tím se prodlužuje prohledávací čas a snadno nastane předčasná konvergence. Způsob využití GA pro řešení rozvrhovacích problémů efektivně je považován za náročný problém[6].

GA jsou heuristický proces inspirovaný aplikací biologických evolučních principů. Dokáží řešit komplexní problémy (jako je například rozvrhování) napodobováním procesu evoluce k iterativnímu zlepšování každé populace. Tyto algoritmy pracují se sadami potenciálních řešení, zakódovanými jako binární kód nebo jako jiná obdobná datová struktura. Tyto potenciální řešení se nazývají jedinci a celá množina řešení je označována jako populace. Jedinci jsou tvořeny jednotlivými geny[15]. Obrázek 5 ilustruje názvosloví používané v genetických algoritmech.



Obrázek 5 Názvosloví používané v genetických algoritmech[16]

Genetické algoritmy, které jsou pro názornost popsány v následujícím příkladu plánování trasy mezi několika body ve městě, se skládají z několika kroků[15]:

- Zakódování řešení – Potenciální řešení bude zakódováno jako permutace genů čili míst ve městě, které budou reprezentovány písmeny.
 - (příklad A – zubař, B – školka, ..., F – pekař, ...)
- Inicializace – Vytvoří se základní populace jedinců, která bude mít danou velikost a jedinci v ní budou náhodné permutace genů (A, B, C, ...)
- Ohodnocení populace – podle účelové funkce (ÚF), v našem případě se může jednat například o délku trasy, čas na cestě nebo jiných parametrů.
 - V tomto kroku se také zaznamenává průměrný výsledek populace a nejlepší výsledek nejlepšího jedince populace
- Selektce – z populace se vyberou cesty (jedinci) s dobrou hodnotou ÚF, které budou součástí další populace. V našem příkladě se například pro dobu cesty vyberou cesty s kratším časem. Běžné metody výběru rodičů další populace jsou ruletová, turnajová, nebo pořadová selektce.
- Modifikace jedinců – v tomto kroku dochází k modifikaci chromozomů jedinců, které zajišťují vývoj populace. Může se jednat například o křížení, nebo mutaci:

- Křížení – Vytváření nových cest kombinací genů jednotlivých rodičů. Obrázek 6 ukazuje příklad prohození segmentů cest pro vytvoření dvou nových potomků.

Rodič 1:	A	B	C	D	E	F
Rodič 2:	F	C	D	B	E	A

Potomek 1:	A	C	D	B	E	F
Potomek 2:	F	B	C	D	E	A

Obrázek 6 Příklad křížení chromozomů

- Mutace – zavedení náhodných změn do chromozomu na základě předem určené pravděpodobnosti. Mutace v daném příkladě může zahrnovat náhodnou výměnu dvou bodů na cestě, přidáním nového bodu do trasy nebo změnu pořadí v částech jednotlivých tratí. Obrázek 7

Před mutací:	A	B	C	D	E	F	-->	Po mutaci:	A	G	C	D	H	F
Před mutací:	F	C	D	B	E	A	-->	Po mutaci:	F	B	D	C	E	A

Obrázek 7 Příklad mutace chromozomů

- Nová generace – rodiče z původní populace a noví potomci se znovu ohodnotí a proběhne výběr jedinců pro vstup do nové populace.
- Terminace – GA iterativně pokračuje, dokud není dosaženo podmínky terminace. Podmínka terminace může být definována počtem generací, dosažení řešení s určitou hodnotou ÚF (byla nalezena cesta, jejíž délka je vyhovující) nebo v populaci došlo ke konvergenci nejlepších řešení s průměrnou hodnotou řešení.
- Výběr finálního řešení – po dosažení podmínky terminace je z populace vybrán vhodný jedinec, typicky jedinec s nejlepší hodnotou ÚF v poslední generaci. Je také nutno podotknout, že pro správný chod GA je nutné správně nastavit parametry jako je počet generací či velikost populace.

3.4.2.4 Genetický algoritmus s nedominovaným tříděním (NSGA-II)

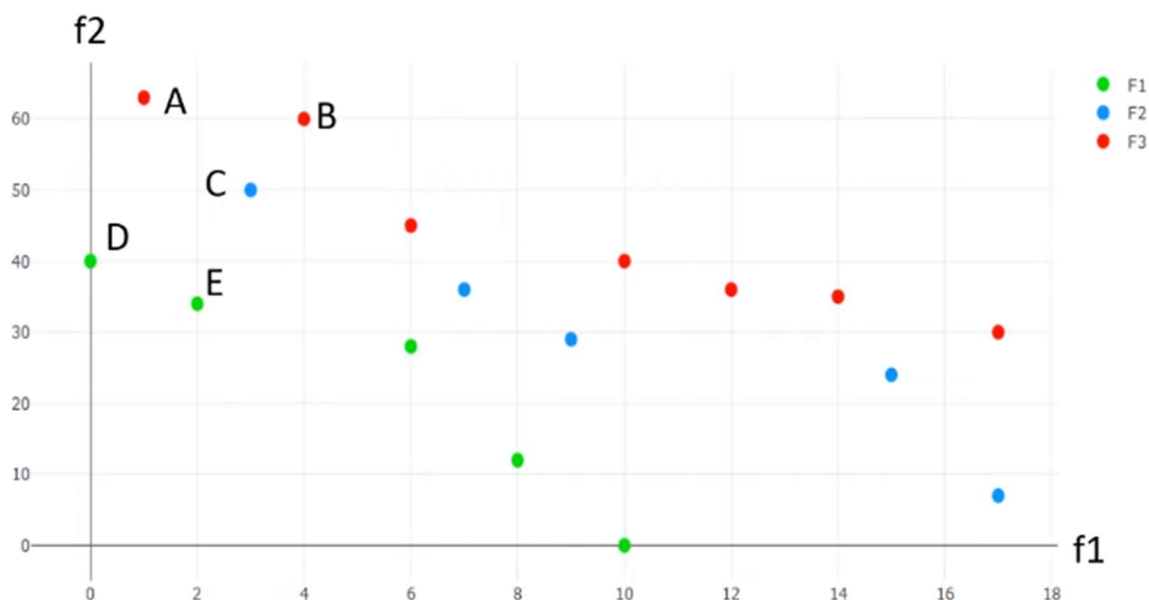
NSGA-II (z anglického Non-dominated Sorting Genetic Algorithm) je evoluční algoritmus postupující podle standardní osnovy genetických algoritmů s upraveným křížením a

výběrem jedinců pro následující generace. V NSGA-II jsou potomci, jako u běžných genetických algoritmů, přidáni k rodičům do množiny řešení, která je následně ohodnocena podle daných kritérií a roztríděna do jednotlivých „front“. Počet front poté zhruba odráží kvalitu jedinců do nich přiřazených. Čím menší počet front, tím vyšší kvalita[17].

Rozřazování do front je iterativní proces, ve kterém se hledají jedinci se stejnou hodnotou dominance. Dominantní jedinec je pak takový, který splňuje daná kritéria ((20)). Každému jedinci je pak přiřazeno pole, ve kterém je zobrazeno, kterým jiným jedincům v populaci dominuje. Nejprve se hledají jedinci první fronty (Obrázek 8 zeleně). Jedincům se přiřadí parametr, který ukazuje kolik jiných jedinců mu dominuje. V momentě, kdy jsou nalezeni jedinci s nulovou hodnotou tohoto parametru, označí se jako fronta 1. U jedinců z fronty 1 se poté projde pole jedinců, kterým dominuje a pro každého jedince, na kterého je odkázáno v tomto poli je odečteno 1. Poté co se projdou všichni jedinci z fronty 1, hledají se jedinci, kteří mají parametr počtu dominantních řešení 0. Tito jedinci pak tvoří frontu 2. U dalších front se pak postupuje analogicky[18].

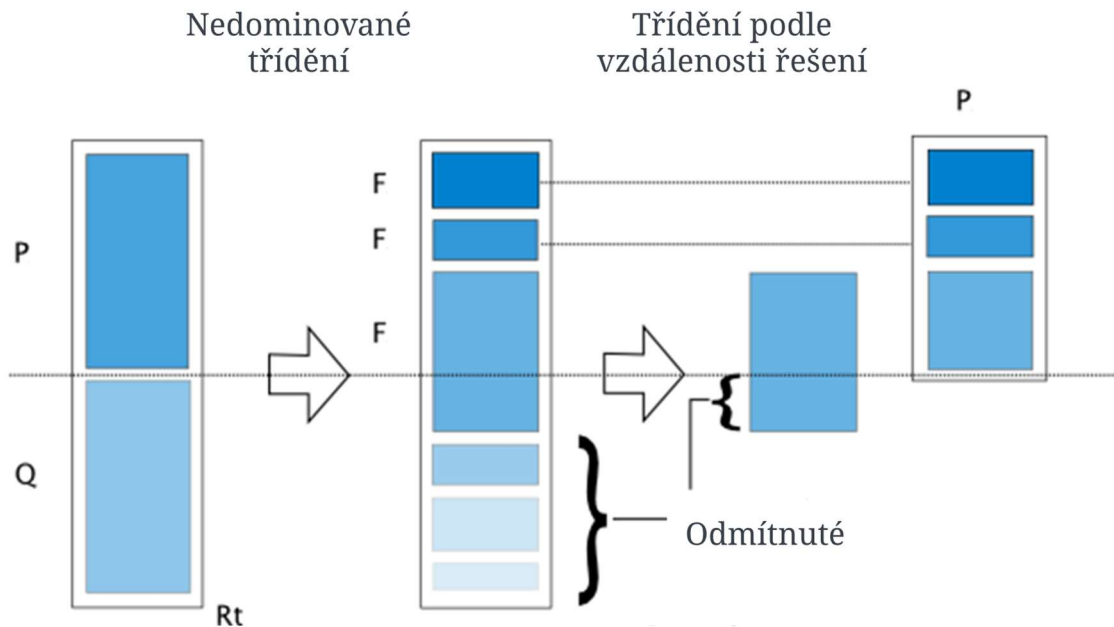
kritéria dominance rozvrhu 1 oproti rozvrhu 2[18]

$$(x_1 \leq x_2 \ \&\& \ y_1 \leq y_2) \ \&\& \ (x_1 < x_2 \ || \ y_1 < y_2) \quad (20)$$



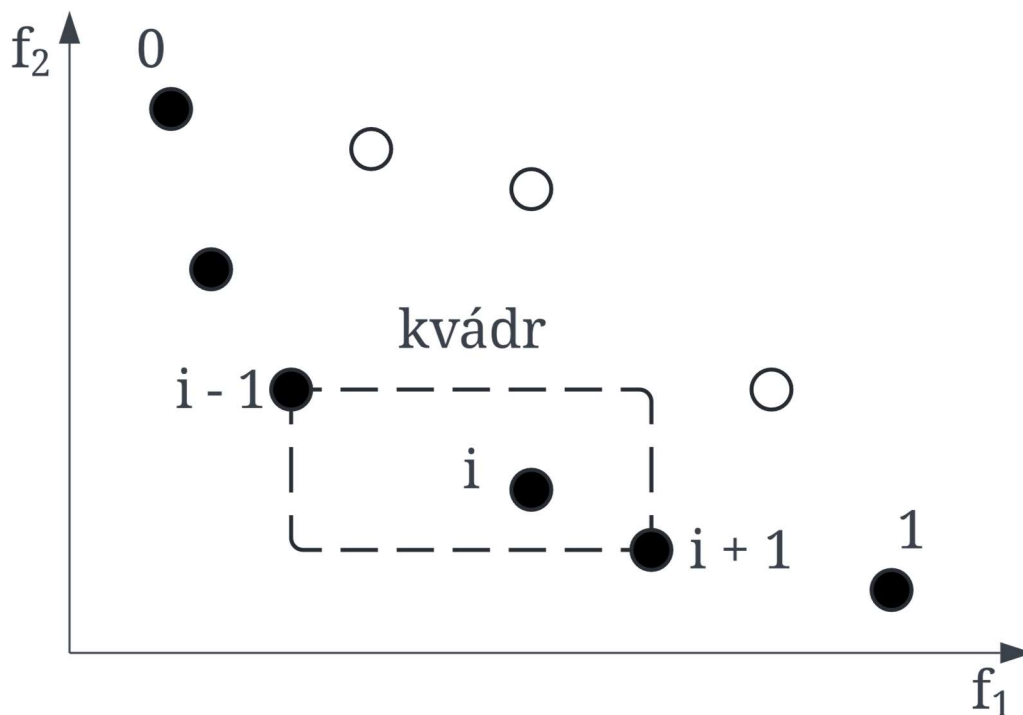
Obrázek 8 Barevné rozlišení jedinců do jednotlivých front[18]

V dalším kroku se populace seřadí podle front (Obrázek 9) a stejně jako v jiných genetických algoritmech se vybere polovina populace s vyšší kvalitou řešení. Jak je vidět na obrázku (Obrázek 9), do výsledné populace se nevejde celá fronta 3[17].



Obrázek 9 Výběr další generace jedinců[17]

Pro zajištění dobrého rozložení jedinců a pro vyhnutí se lokálním extrémům, využívá NSGA-II metodu zvanou „Crowding distance“. Vzdálenost řešení (CD ~ Crowding Distance), je Manhattanská vzdálenost a je spočtena pomocí rovnice (21) a poté se vyberou nejprve jedinci s vyšší hodnotou vzdálenosti řešení. Toto se provádí u fronty populace, která není celá přijata do další generace, pro zajištění diversity řešení. Pro zachování diversity, jsou vždy vybráni jedinci, kteří jsou nejlepší ze všech podle jednoho kritéria (CD je jim nastavena na nekonečno).



Obrázek 10 Vzdálenost řešení [19]

Výpočet vzdálenosti řešení (viz Obrázek 10):

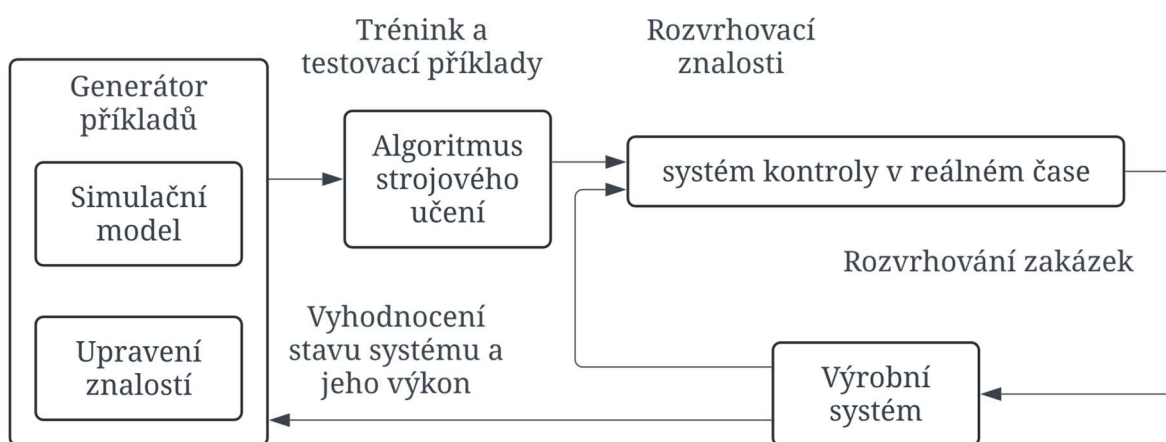
$$CD(i) = CD(i) + \frac{o(i+1) - o(i-1)}{o(max) - o(min)} \quad (21)$$

Produkování nových jedinců je dále zprostředkováno turnajovou metodou. Rodiče jsou porovnáváni na základě hodnoty ÚF. Podle toho, který je optimálnější, tak tomu je umožněno vytvořit potomka. Pokud mají stejnou hodnotu ÚF, jsou porovnáváni na základě CD. Vítězný jedinec poslední populace je vybrán z nejlepší pareto fronty vhodným výběrem[17].

3.4.2.5 Další bio-inspirované algoritmy

Prvním příkladem je použití neuronové sítě. Umělá neuronová síť je programovací systém, který je generovaný na základě konceptu organizované struktury lidského mozku a jeho operačního mechanismu. Mezi její výhody patří: velké ukládací místo, silná ukládací schopnost, schopnost učit se, dobrá tolerance chyb, jednoduchá klasifikace. Oproti tomu lze za její nevýhody považovat: nízkou efektivitu učení, malou rychlost, obtížné vyjádření širšího spektra znalostí. Neuronová síť je aplikovaná pro rozvrhování výroby již 15 let. Je schopna generovat proaktivní rozvrhy [6, 20].

Priore, Gómez, Pino a Rosillo [21] se ve svém článku zabývali tím, jak může strojové učení pomoci vybrat nejlepší pravidla pro plánování úloh v továrnách na základě minulých výkonů. Pojednává o potřebě systému (Obrázek 11), který dokáže přizpůsobit pravidla plánování různým situacím ve výrobním prostředí. Existuje několik strategií, kdy je třeba změnit rozvrh: ve stanovených časech (periodické), při výskytu událostí (řízené událostmi). Studie ukazuje, že rozvrhovací pravidla ve výrobě se liší v úspěšnosti v závislosti na mnoha faktorech, jako je nastavení systému a pracovní zatížení. Předpokládá se, že změna rozvrhovacích pravidel podle situace by mohla fungovat lépe než použití fixního pravidla.



Obrázek 11 Algoritmus strojového učení pro výběr prioritních pravidel [21]

Systémy rozvrhování v reálném čase musí rychle vybírat pravidla na základě aktuálních informací, aniž by docházelo ke zpoždění. Tyto systémy využívají "informace o rozvrhování" k rychlému rozhodování v měnícím se prostředí, ale získat tyto informace je obtížné. Strojové učení, jako jsou neuronové sítě, pomáhá najít informace potřebné pro plánování, ale vyžaduje správná data a algoritmy. Systém se učí ze simulací pro výběr pravidel použitelných v různých situacích a aktualizuje své znalosti na základě své výkonnosti[21].

Dalším používaným algoritmem v literatuře je algoritmus simulovaného žihání. Algoritmus simulovaného žihání je typem sekvenčního optimalizačního algoritmu s nízkou výpočetní efektivitou, inspirovaný žiháním kovových slitin s cílem získání jejich optimálních vlastností. Největší výhodou tohoto algoritmu je možnost jeho kombinace s jiným algoritmem[6].

Optimalizace pomocí metody mravenčí kolonie je další metodou, která najde nejvýhodnější řešení daného problému pomocí skupinových informací. Jejími výhodami jsou jednoduché výpočty, rychlá konvergence, ale je náchylná k inklinaci k lokálnímu minimu[6], [22].

V literatuře je také prozkoumána metoda rojení částic. Optimalizace metodou rojení částic je rozšířená pro řešení celé řady optimalizačních problémů z důvodů její vysoké účinnosti, menšímu počtu parametrů potřebujících úpravy (kontrolu), velké robustnosti a dalším charakteristikám. Její největší nevýhodou je možnost dosažení předčasné konvergence a chabá lokální optimalizace [3].

Dalším algoritmem, který je inspirovaný přírodou je imunitní algoritmus. Umělý imunitní algoritmus je globální prohledávací algoritmus, který simuluje funkce biologického imunitního systému. Imunitní algoritmy, díky své rychlé konvergenci a silné schopnosti prohledávat celý systém globálně, umožnili nové směry zkoumání rozvrhovacích problémů[6].

3.4.2.6 Další metody přibližného řešení rozvrhovacích problémů

Rozvrhovací expertní systém dokáže generovat komplexní heuristická rozvrhovací pravidla, které jsou „inteligentní“, ale jeho doba vývoje je dlouhá, náklady jsou vysoké a je složité získat potřebné zkušenosti a znalosti[6], [23].

Používaná je také simulační metoda. Z důvodů komplikovanosti výrobních systémů, je těžké je popsat a zkoumat pomocí exaktních analytických modelů. Nicméně simulace může vyprodukovat ideální model a mohla by ho kvalitativně posoudit a tím přijmout vhodnou rozvrhovací metodu pro rozvrhovací systém. Díky tomu je simulace stále jednou z nejpopulárnějších rozvrhovacích metod[6, 24].

Důkladně prozkoumaným algoritmem je metoda zakázaného prohledávání. Tabu Search (TS) je metaheuristika, která se používá k řešení optimalizačních problémů. Výkon TS byl oproti technice lokálního prohledávání zlepšen tím, že zavrhuje již navštívená řešení problému a jejich sousedy. Tím se zastaví hledání lokálních optimálních řešení. TS vytváří kandidáty, což je v podstatě množina různých celočíselných proměnných. Tito kandidáti se liší o jeden nebo více bitů od současného optimálního řešení a jsou vyřazeni ze seznamu vytvořeného TS. Dílčí problémy původního problému se pak řeší pro každého

kandidáta technikou založenou na gradientu. Pro vytvoření zárodku pro další generaci je vybrán jeden současný kandidát s nejlepší objektivní hodnotou [25].

Jednou z moderních metod je multiagentní metoda. Metoda multiagentního rozvrhování vzniká díky rozšíření široce distribuovaného rozvrhování, které je ideálním modelem rozvrhování. Dokáže se přizpůsobit dynamickým změnám v dílenské výrobě a dosáhnout sebezdokonalování[6].

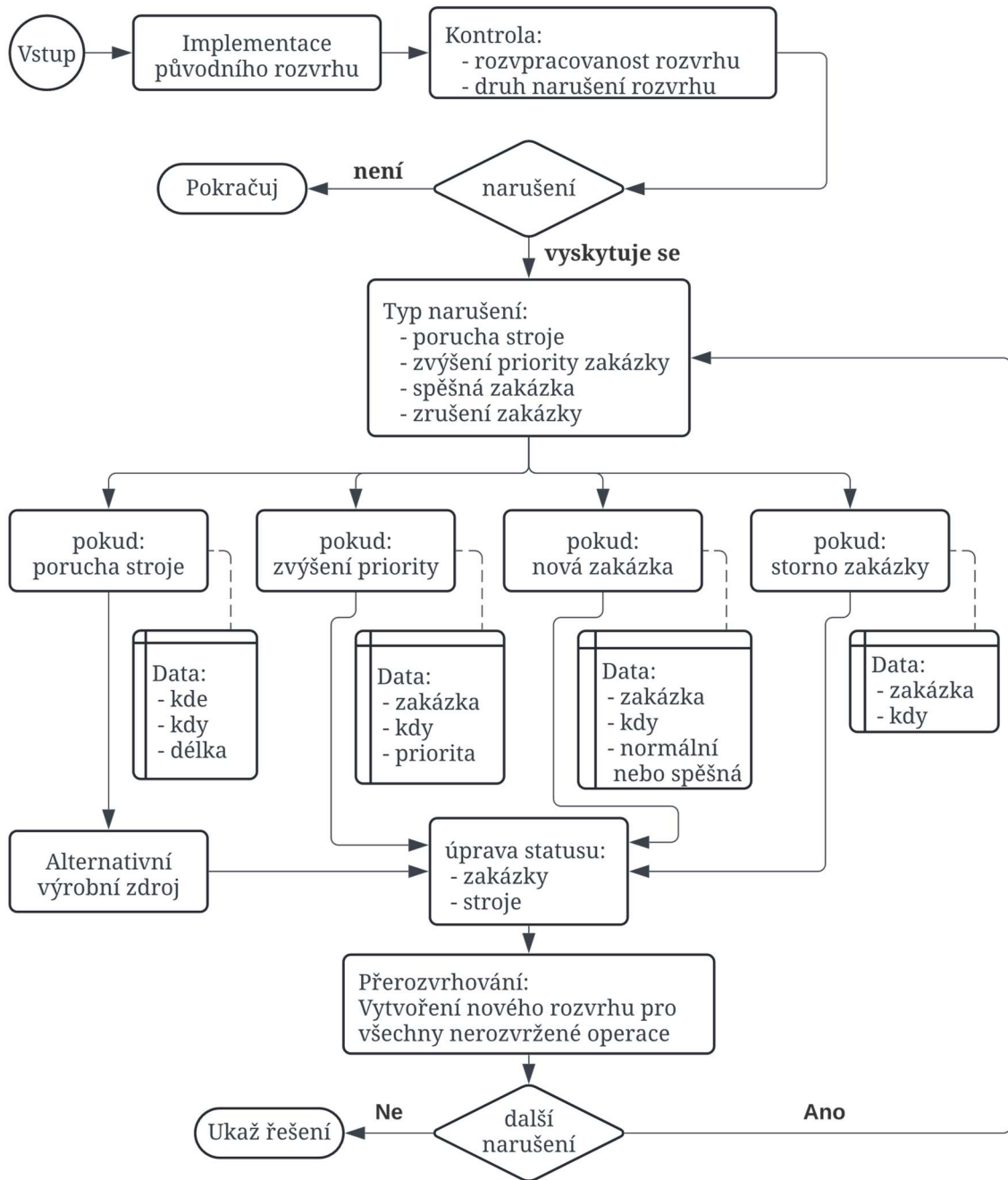
Další metodou je fuzzy logika. Základní myšlenka fuzzy logiky je využití matematických nástrojů k imitaci myšlení lidského mozku a vypořádání se s komplexními věcmi s použitím fuzzy procesingu. Podle náhodnosti a neurčitostí (fuzziness ~ neostrost, nejasnost, mlhavost, neurčitost) rozvrhovacích problémů jsou aplikována fuzzy pravidla na adaptivní rozvrhování v dynamickém výrobním prostředí. Podle stupně příslušnosti jsou ve výrobním systému aplikované prvky fuzzy teorie, pro určení pravděpodobnosti distribuce výrobních časů na pracovištích[6, 26].

Novým přístupem k problému se stává v éře rozvoje kvantových počítačů kvantové žíhání. Algoritmus kvantového žíhání je založen na algoritmu simulovaného žíhání. QA využívá skutečných fenoménů kvantové fyziky jako jsou kvantové provázání, princip superpozice a kvantové tunelování. Oproti simulovanému žíhání využívá kvantové žíhání kvantového tunelování, které algoritmu umožňuje náhlé přechody pomyslných energetických bariér a tím překonávat lokální extrémy. Schworm a kol.[27] uvedli použití tohoto algoritmu v dynamickém prostředí (DFJSSP). Systém funguje tak že, na počátku vygeneruje pomocí QA plán, který je regenerován pokaždé, když dojde ke změně v souboru se vstupy (ten kontroluje průběžně). Pracuje s pravidlem, že již probíhající aktivita nesmí být ukončena a nahrazena jinou.

3.5 Kdy měnit rozvrh

V literatuře se objevuje mnoho prací „Jak změnit rozvrh?“ , které se zabývají otázkami: „Kdy přerozvrhovat?“ a

Jedna z prací která se věnovala strategiím přerozvrhování byla uvedena v práci Elmaraghyho a Jaina[4]. Jejich algoritmus je zachycen na následujícím obrázku (Obrázek 12).



Obrázek 12 Architektura algoritmu pro dynamické rozvrhování v práci Elmaraghyho a Jaina[4]

Byly zde využity genetické algoritmy pro vytvoření prvotního rozvrhu a přerozvrhování probíhalo následně pomocí heuristických algoritmů. Nejprve algoritmus při výskytu změny ve výrobním procesu přerozvrhuje operace na volné alternativní zdroje, pokud toto není možné, dochází ke specifickým řešením v případě:

- rozbití stroje:
 - převedení na jiný stroj, pokud koliduje s jinou zakázkou tak se porovnají priority (převážně podle pravidla MWRK). Porovnání předpovědi času opravy a set-up timu zakázky na jiném stroji.
 - K lokálnímu přerozvrhování na alternativní stroje dojde pouze v případě, že jde o rozbití stroje a zahrnuje pouze zakázky ovlivněné tímto strojem.
- Příchodu spěšné zakázky:
 - je ihned zaplánováno do rozvrhu na patřičné stroje podle její priority vůči ostatním. Pokud není urgentní tak FIFO pro zařazení do fronty operací, nebo porovnání na základě data odvedení. Pokud má vyšší prioritu než některé, tak ty jsou posunuty doprava.
 - Převádění „předběhnutých“ zakázek na jiné stroje se v této práci neuvazuje.
- Zrušení objednávky:
 - Všechny rozvrhnuté operace jsou smazány a jsou vytvořeny volná místa pro další náhlé změny.

Kupříkladu Paprocka a Skolud[28] navrhli hybridní víceúčelový imunitní algoritmus (H-MOIA) s podporou heuristiky pro prediktivní a reaktivní plánování, jehož cílem je minimalizovat dopad přerušených operací na plán. Studie se zaměřuje na zavedení údržbových prací do rozvrhu s cílem omezit nepředvídaná přerušení způsobená poruchami strojů a zlepšit produktivitu, včasnou výrobu, stabilitu a robustnost rozvrhu. Studie sleduje parametry MIDOS (Minimální dopad narušené operace na rozvrh) a MIROS (Minimální dopad přerozvrhnuté operace na rozvrh). Ve studii byla použita prediktivní strategie generování rozvrhu a poté do něj byly přidány časové rezervy před operace s možným výskytem poruch.

V průběhu výroby poté algoritmus nejprve hledal ovlivněným operacím alternativní výrobní zdroje, poté odstranil ovlivněné zakázky posunem vpravo (Right-shifting – RSh), následně proběhl posun zakázek neovlivněných narušením systémem vlevo (Left-Shifting – LSh). V posledním kroku probíhalo pomocí heuristiky SGR (Shifted Gap Reduction) vkládání operací do předpřipravených časových rezerv.

Další práce Wanga, Zhanga a Yanga[29] se zabývá problematikou příchodu nových zakázek. V práci byl použit algoritmus rojení částic (PSO). K přeplánování zde došlo po každé s příchodem nové zakázky. Práce uvádí parametr míry diskontinuity (DR), který je podílem celkové doby zpoždění ku celkové době zpracování nových úloh. Algoritmus vloží novou zakázku do výrobního rozvrhu a převážně poté přesouvá operace na jednotlivých přímkách zdrojů podle omezení a parametru DR.

Lze tedy říct, že nejčastěji řešené úlohy bývají příchod nové zakázky, změna priorit zakázky, zrušení zakázky, náhlá nedostupnost výrobního zdroje, zmetkovitost a absence pracovníka.

4 Vícekriteriální hodnocení rozvrhu

Tato kapitola se zabývá vícekriteriálním hodnocením rozvrhů. Jedná se o výběr nejvhodnějšího rozvrhu z vytvořených rozvrhů na základě definovaných kritérií.

4.1 Model váženého součtu (WSM)

Jednou z nejjednodušších metod vícekriteriálního hodnocení je metoda váženého součtu. Funguje na základě přiřazování důležitosti (hodnoty) jednotlivým kritériím a poté dochází k výpočtu sumy vážených hodnot pro výpočet preference pro rozhodnutí. WSM probíhá v následujících krocích[30]:

- Identifikace faktorů – zde se identifikují alternativy výsledného řešení a jednotlivá kritéria. Faktory mohou být jak kvantitativní, tak kvalitativní povahy.
- Přiřazení vah – každému kritériu se přiřadí váha, která zohledňuje důležitost kritéria při rozhodování. Součet vah musí být 1.
- Sběr dat – každé alternativě je přiřazena hodnota ÚF jednotlivých kritérií.
- Výpočet váženého součtu – pro každou alternativu je spočtena preference P_i alternativ A_i , kde $i=1, 2, 3, \dots, M$, s váhami W_j , kde $j=1, 2, 3, \dots, N$, na základě rovnice (22)

Výpočet metody váženého součtu[30, 31]

$$P_i = \sum_{j=1}^N A_i * W_j \quad (22)$$

U této metody je nutné vypíchnout problematiku, kdy jsou pro jednotlivé alternativy hodnoty kritérií velmi rozdílné. Jelikož kritérium, které je několikařádově vyšší než jiné kritérium, bude mít na výsledném rozhodování větší vliv než kritérium s hodnotami výrazně nižšími, ale vyšší váhou. Zároveň musí být všechna kritéria ve stejných jednotkách, pokud tomu tak není, je problém označován jako multidimenzionální. Pro řešení takovýchto problémů je nutno data metody linearizovat (23– (27) podle vhodnosti použití[32] nebo použít jinou metodu.

Lineární normalizace sčítací (N3) [32]:

$$n_{ij} = \frac{a_{ij}}{\sum_{i=1}^M a_{ij}} \text{ nebo: } n_{ij} = \frac{1/a_{ij}}{\sum_{i=1}^M 1/a_{ij}} \quad (23)$$

Vektorová normalizace (N4) [32]:

$$n_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^M a_{ij}^2}} \text{ nebo: } n_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^M a_{ij}^2}} \quad (24)$$

Lineární normalizace maxima (N1) [32]:

$$n_{ij} = \frac{a_{ij}}{a_{max}} \text{ nebo: } n_{ij} = 1 - \frac{a_{ij}}{a_{max}} \quad (25)$$

Logaritmická normalizace (N5) [32]:

$$n_{ij} = \frac{\ln(a_{ij})}{\ln(\prod_{i=1}^M a_{ij})} \quad (26)$$

Lineární normalizace maxima-minima (N2) [32]:

$$n_{ij} = \frac{a_{ij} - a_{min}}{a_{max} - a_{min}} \text{ nebo: } n_{ij} = \frac{a_{max} - a_{ij}}{a_{max} - a_{min}} \quad (27)$$

Kde je:

- a_{ij} hodnota kritéria j alternativy i ,
- n_{ij} normalizovaná hodnota kritéria j alternativy i ,
- a_{max} maximální hodnota kritéria alternativy,
- a_{min} minimální hodnota kritéria alternativy.

4.2 Model váženého součinu (WPM)

Metoda váženého součinu je velmi podobná metodě WSM. Hlavním rozdílem je že místo součtu je v této metodě použit součin. Postup metody je také obdobný, co se týče identifikace faktorů, přiřazení vah a sběru dat. Rozdíl nastává až při výpočtu. V této metodě nejsou počítány preference jednotlivých alternativ, ale jsou porovnávány mezi sebou jednotlivé alternativy na základě součinu podílů kritérií jednotlivých alternativ. Každý podíl je zde umocněn relativní vahou jednotlivých kritérií[31].

Pro porovnání alternativ A_p a A_q je použita rovnice (28), kde je:

- $M \geq p, q \geq 1$,
- $a_{p,j}$ hodnota kritéria j alternativy p ,
- $a_{q,j}$ hodnota kritéria j alternativy q ,
- N počet kritérií,
- w_j váha kritéria j .

porovnání metodou váženého součinu[31]:

$$R\left(\frac{A_p}{A_q}\right) = \prod_{j=1}^N \left(\frac{a_{p,j}}{a_{q,j}}\right)^{w_j} \quad (28)$$

Výsledný poměr $R(A_p/A_q)$ ukazuje, která z alternativ je vhodnějším řešením. Pokud $R(A_p/A_q) \geq 1$ (pro maximalizační případ), pak je alternativa p nadřazena alternativě q.

Výhodou této metody může být například schopnost porovnat dvě řešení i když jsou jednotlivá kritéria několikařádově rozdílná. Dále může být nevýhodou pro iterativní programy fakt, že metoda nepřihradí každé alternativě jedno číslo, na jehož základě by bylo schopné alternativy seřadit, ale poskytne porovnání jednotlivých alternativ mezi sebou. Musí se tedy provést výpočet pro všechny páry alternativ, což může být například pro 50 alternativ 1250 porovnání. Pokud však existuje alternativa, která má vůči všem ostatním $R(A_p/A_q) \geq 1$ (pro maximalizační případ), lze ji označit za nejlepší.

Experimentální část

5 Současný stav ve firmě TOS Varnsdorf a.s.

Společnost TOS Varnsdorf a.s. je středně velkou společností s ročním obratem 2,5 miliardy Kč, zaměstnávající přibližně 450 zaměstnanců (konec roku 2023). Její primární zaměření je na vývoj a výrobu horizontálních frézovacích a vyvrtávacích strojů, obráběcích center a portálů. Firma dále nabízí příslušenství ke svým strojům (jako jsou například obráběcí hlavy), servis strojů, zakázkové kovoobrábění, vysoce přesné měření a chemicko-tepelné zpracování kovů.



Obrázek 13 Mapa dceřiných společností korporátu TOS Varnsdorf a.s. [33]

Firma TOS Varnsdorf je následovníkem firmy „Arno Plauert, továrna na stroje“ která byla založena v roce 1903 a po turbulentním období první poloviny 20. století byla v 50. letech znárodněna a převedena pod hlavičku „Továren obráběcích strojů“. Po pádu komunismu byl varnsdorfský podnik zprivatizován pod nynějším názvem TOS Varnsdorf a.s. Firma TOS Varnsdorf má několik dceřiných společností, a to jak v regionu (Metalurgie Rumburk s.r.o., MTEq s.r.o., TOS Olomouc s.r.o.), tak ve světě (TOS Kunming Machine Tool Co., Ltd, TOS Trade North America, LLC, TOS Shanghai Machine tools Ltd.). Zobrazeny jsou na mapě (Obrázek 13). Tyto dceřiné společnosti nabízejí velkou

škálu služeb, které se společnost od společnosti liší. Služby jsou v rozsahu od obchodního zastoupení, přes slévárenské služby, finální montáž, servis až po závody samostatně vyrábějící stroje. Společnost TOS Varnsdorf provozuje také soukromou střední školu s učilištěm SPŠ TOS VARNSDORF.

Ve Varnsdorfském závodě společnosti TOS Varnsdorf a.s. byly identifikovány tyto strukturální střediska:

- Ekonomické oddělení
- Konstrukce
- Technologie
- Řízení výroby
- Výrobní střediska:
 - Sklad
 - Lakovna
 - Kalírna
 - Těžká mechanika (část výroby sdružující velké obráběcí stroje, montáž obráběcích strojů a žíhací pec)
 - Lehká mechanika (část výroby obsahující menší stroje, vyrábějící vřetena, vřeteníky a obráběcí hlavy)
 - OTS (Servis)
 - Montáž
 - Kontrola
- Expedice

5.1 Současný stav rozvrhování ve firmě TOS Varnsdorf

V létě roku 2023, bylo absolvováno „kolečko“ ve firmě TOS Varnsdorf a.s. Jeho cílem bylo seznámení se s výrobou. Při průzkumu výroby byla analyzována vhodnost jejich jednotlivých částí pro zavedení nového rozvrhovacího systému. Dále následovalo seznámení se s řízením výroby a ERP systémem od firmy QAD Inc. na středisku řízení výroby. Nynější stav zahrnuje plánování do neomezených kapacit, kdy jsou týmem, složeným ze zaměstnanců řízení výroby a zástupců z vedení a z technických úseků, určena data splnění jednotlivých fází zakázek. Tato data jsou poté vstupními informacemi ERP systému, který na jejich základě nastaví termíny splnění jednotlivých operací pro dokončení

zakázky. V ERP systému jsou momentálně plánované operace pro výrobu jak již objednaných strojů, tak strojů „virtuálních“ které slouží pro zásobu materiálem jednotlivých středisek a výrobu polotovarů potřebných při následné výrobě prodaných strojů. Toto a plánování do neomezených kapacit má za následek takzvané přetěžování kapacitních jednotek, kdy je jednotlivým výrobním zdrojům předepisována práce, která jim naplňuje kapacitu na více než 100 %.

5.2 Mapování informačních toků

Pro mapování informačních toků byly vytvořeny BPML grafy. Business Process Modeling Language (BPML) je způsob zpracování podnikových procesů do snadno srozumitelné grafické podoby. Grafy byly zpracovány v programu yEd Graph Editor od firmy yWorks GmbH a lze je nalézt v přílohách. Nejprve byl zhotoven graf základní struktury toku informace při příchodu nové zakázky do firmy (Příloha 1/ 2). Tento graf je rozdělen vodorovně na výrobní a nevýrobní střediska.

5.2.1 Oprava obráběcích hlav

Prvním výrobním procesem, který byl graficky znázorněn, byla oprava obráběcích hlav na středisku lehké mechaniky (Příloha 1/ 3). Proces byl mapován od bodu příchodu zakázky od zákazníka až po předání zakázky zákazníkovi. V grafu jsou červenými čarami znázorněny informace, jež tečou mimo informační systém firmy (ERP QAD) a zeleně ty toky informací, které se předávají skrze systém. Z grafu informačních toků lze pozorovat, že většina informací (dokumentace a příkazy k práci) se předávají mimo informační systém a jsou ze systému vytahovány do osobních excelovských souborů a řízení výroby o nich nemá přehled. ERP systém se zde používá jen pro „vytažení“ pracovního postupu pro mistra lehké mechaniky a pro odpis hodin pro splnění fondu hodin na pracovišti. Toto je jeden z problémů, které se v informačních tocích ve firmě vyskytují. Každé výrobní středisko používá ke svému řízení svůj vlastní systém, který je běžně řízen tabulkovým kalkulátorem Excel s přidanými makry a nastává zde problém absence zpětných informací o stavu výroby pro středisko řízení výroby. Z grafu lze pozorovat, že středisko servisu informuje e-mailem lehkou mechaniku o přicházející zakázce dříve než středisko řízení výroby. Práce na lehké mechanice tudíž začínají na opravovaném dílu dříve, než je zaplánován střediskem řízení výroby. Středisko řízení výroby poté posléze zaplánuje výrobní kapacity zpětně pro odpis hodin z pracovního fondu.

Z důvodu malé integrace do informačního systému byla tato operace vyhodnocena jako nevhodná pro cíle mého zkoumání v rámci DP. Pro aplikaci dynamického rozvrhování na této operaci je zapotřebí větší integrace do informačního systému.

5.2.2 Výroba obráběcích hlav

Druhý výrobní proces, kterému byl zhotoven graf informačního toku, byla výroba hlav na lehké mechanice (Příloha 1/ 4). Znovu byly použity barvy pro rozlišení informací, které tečou skrze/mimo informační systém. Tato operace je již z pohledu toku informací mezi výrobou a řízením výroby na lepší úrovni. To je z důvodu, že se jedná o standardnější proces oproti opravě obráběcích hlav. Můžeme zde vidět, že příkaz k výrobě přichází z řízení výroby skrze informační systém, ačkoliv zde stále „pro jistotu“ řízení výroby posílá upomínkový e-mail. Dalším problémem je doprava výrobních postupů na výrobu. Ta je prováděna manuálně pracovníkem řízení výroby, který pracovní postupy pro výrobu hlavy tiskne a ručně je poté nese na středisko lehké mechaniky. Data, která mistr dostane, poté vloží do osobního MS Excel souboru, který slouží jako hrubý dílenský plán. Mistr zde sleduje počet položek na skladě, které si manuálně importuje z ERP systému a popřípadě se „předzásobuje“. Tato předzásoba probíhá jako výroba menších dílů, které budou potřeba pro budoucí, již prodané, obráběcí hlavy, které mají datum přejímky ve vzdálenějším časovém horizontu. Tyto menší díly mají svůj výrobní postup a průběh předání výrobního postupu probíhá stejně jako u hlav samotných. Vytvoří se příkaz v informačním systému a výrobní postup se pak vytištěn donesen mistrovi lehké mechaniky. Ten má pro výrobní postupy těchto „menších dílů“ připravené fyzické přihrádky, kde každá přihrádka slouží jako zásobník práce pro jeden stroj.

Montáž obráběcích hlav jde pak v systému sledovat tak, že každá hlava má přiřazen určitý počet hodin práce a zaměstnanci, kteří jí kompletují postupně (na konci směny, občas v průběhu, při přerušení práce na zakázce) odepisují hodiny, které odpracovali. Lze tedy teoreticky tvrdit, že je možné sledovat procentuální hotovost jednotlivých hlav, ale toto sledování bude zkresleno nedostatečně častým odepisováním práce, jelikož pracovníci nejsou nuceni výrobní dokumentací odepisovat průběžně s odvedenou prací. Další zkreslení také přichází z toho, že jsou odepisovány odpracované hodiny, nikoliv zkompletované části. Můžeme se tedy v informačním systému hlavně dozvědět, zda se na hlavě pracuje, či nikoliv a jaký pracovník na ní pracuje. Dalším problémem je terminál pro

odepisování práce, jenž je na lehké mechanice pouze jeden. Skládá se z počítače, čtečky čárových kódů a CRT monitoru.

Pro cíle zkoumání DP byla tato operace také vyhodnocena jako nevyhovující, z důvodu nedostatečné integrace do informačního systému.

5.2.3 Výroba na těžké mechanice

Další operací, které byl vytvořen graf toku informací, byla výroba na hale „těžké mechaniky“ (Příloha 1/ 5). Znovu byly použity barvy pro rozlišení informací, které tečou skrze/mimo informační systém. Zde je vidět, že v podstatě všechny informace mezi řízením výroby a výrobou na těžké mechanice tečou skrze ERP systém. Mimo chodí pouze upomínková pošta a řízení výroby opět nosí výrobní postupy v tištěné formě vedoucímu a mistrovi těžké mechaniky.

Vedoucí těžké mechaniky si každý den v ERP systému zobrazuje pracovní požadavky na jeho oddělení, které poté formuje do několikadenního dílenského plánu v MS Excel jakožto tabulku, kde je v každém sloupci zásoba práce na jednotlivé stroje. Vedoucí těžké mechaniky sleduje využití strojů v časech, kdy mají být zpracovávány zakázky pomocí MES systému PROXIA a vytváří zpětně ručně OEE jednotlivých pracovišť a vyhodnocuje jednotlivé pracovníky. Podle toho poté přiřazuje odměny z platového fondu jeho oddělení. Je nutné říct, že toto vyhodnocení je opět omezeno pouze na tabulku MS Excel, do které nemá nikdo jiný přístup.

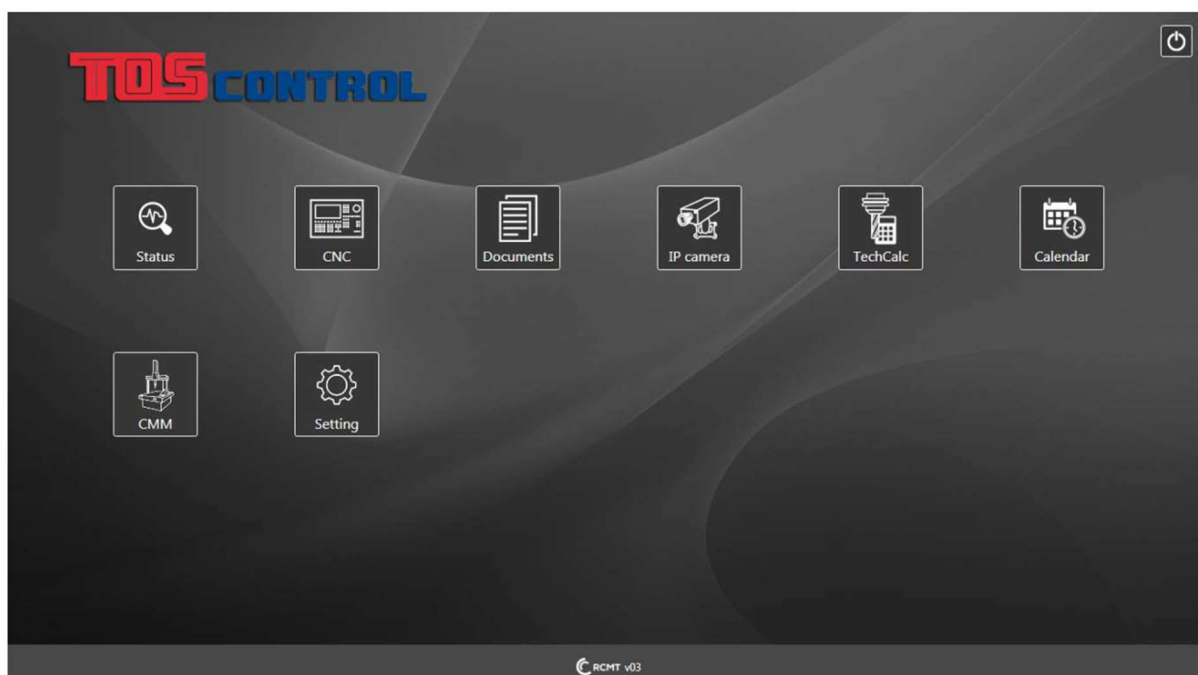
Systém PROXIA ve firmě TOS Varnsdorf pomalu končí a měl by být nahrazen vlastním řešením TOScontrol. Potenciál systému PROXIA jakožto MES systému byl značně nezaplněn. Pouze na jednom stroji bylo možné zobrazovat, na jaké zakázce se momentálně pracuje, ačkoliv se data o průběhu zpět do systému nedostávala. Systém PROXIA zde sloužil jako „informace navíc“ pro obsluhu. Odpis výroby zde probíhal obdobně jako na lehké mechanice, ačkoliv lze říct, že výroba na těžké mechanice je více připravena na potenciální zavedení dynamického systému řízení výroby. To jak z personálního, tak i z technického ohledu.

Na těžké mechanice také probíhá integrace systému TOScontrol, který slouží jakožto SCADA systém a měl by přejít v plnohodnotný MES systém. TOScontrol je momentálně implementován na dvou strojích WHT a je připraven pro vývoj aplikace pro rozvrhování

práce. Toto jsou důvody, na jejichž základě byla vybrána výroba na těžké mechanice jakožto vhodná část výroby pro DP.

5.3 Modul TOScontrol

TOScontrol je nadstavbou řídicího systému stojů firmy TOS Varnsdorf a.s., a to jak pro použití v této firmě, tak jako produkt k prodávaným strojům. Jedná se o soubor aplikací a funkcí pro pokročilou správu stroje. Jedná se o odpověď společnosti TOS Varnsdorf a.s. na trend Industry 4.0. Cílem je zvýšení konkurenceschopnosti v softwarové oblasti vlastních strojů. Motivací je pokud co možno nejvyšší komfort obsluhy s nízkými nároky na schopnosti operátora, ale také možnost implementace pokročilých řešení z hlediska diagnostiky, správy stroje a řízení výroby. Práce odvedená v této diplomové práci bude poté implementována v aplikaci Job management. Obrázek 14 zobrazuje jeho základní rozhraní.



Obrázek 14 Základní rozhraní TOScontrol [34]

Modul TOScontrol je standardně dodáván s následujícími aplikacemi:

- Status – přináší operátorovi pohled na základní informace o stroji jako jsou:
 - aktuální program, souřadnice nástroje, upozornění, id přihlášeného pracovníka a další (Obrázek 15).
- Řídicí systém CNC – zobrazí operátorovi standardní obrazovku řídicího systému pro provedení obráběcí operace.

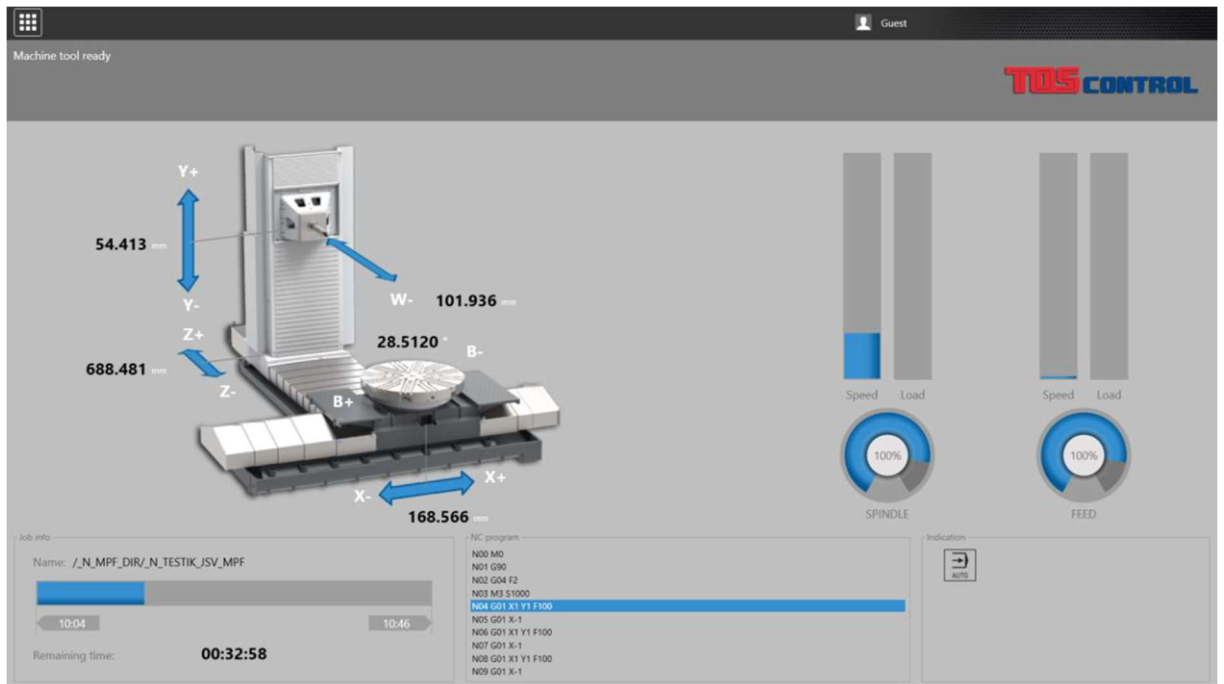
- Dokumentace – umožňuje zobrazení přidaných .PDF souborů jako jsou návod k obsluze, opravárenská příručka a jiné (Obrázek 17).
- Kalendář – zobrazí standardní kalendář pracovníka (Obrázek 16).
- Technologická kalkulačka – slouží operátorovi pro rychlé výpočty přímo na pracovišti.

Dále firma TOS Varnsdorf a.s. nabízí příplatkové aplikace:

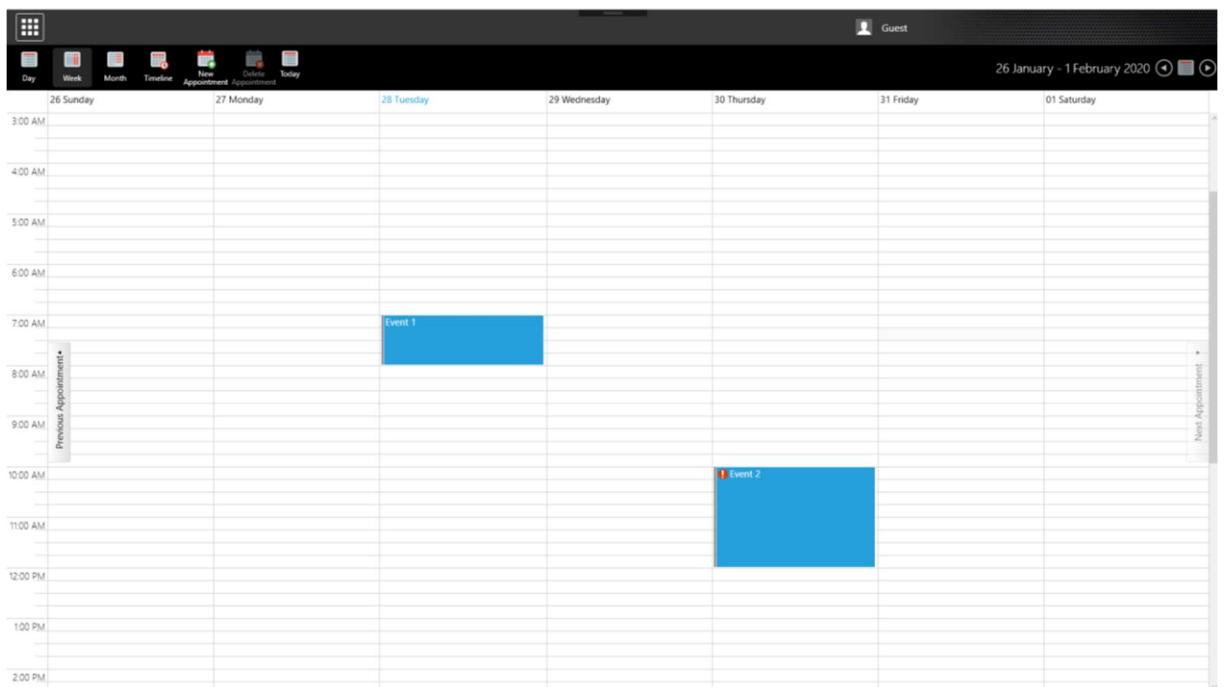
- IP kamera – umožňuje operátorovi zobrazit pohled z jedné nebo více motorizovaných kamer v pracovním prostoru (Obrázek 18).
- Kontrola a kompenzace – pro stroje vybavené dotykovou sondou umožňuje tato aplikace měření přímo na stroji. Toto umožňuje následnou automatickou kompenzaci chyb.

Ve vývoji je nespočet dalších aplikací, jako jsou například:

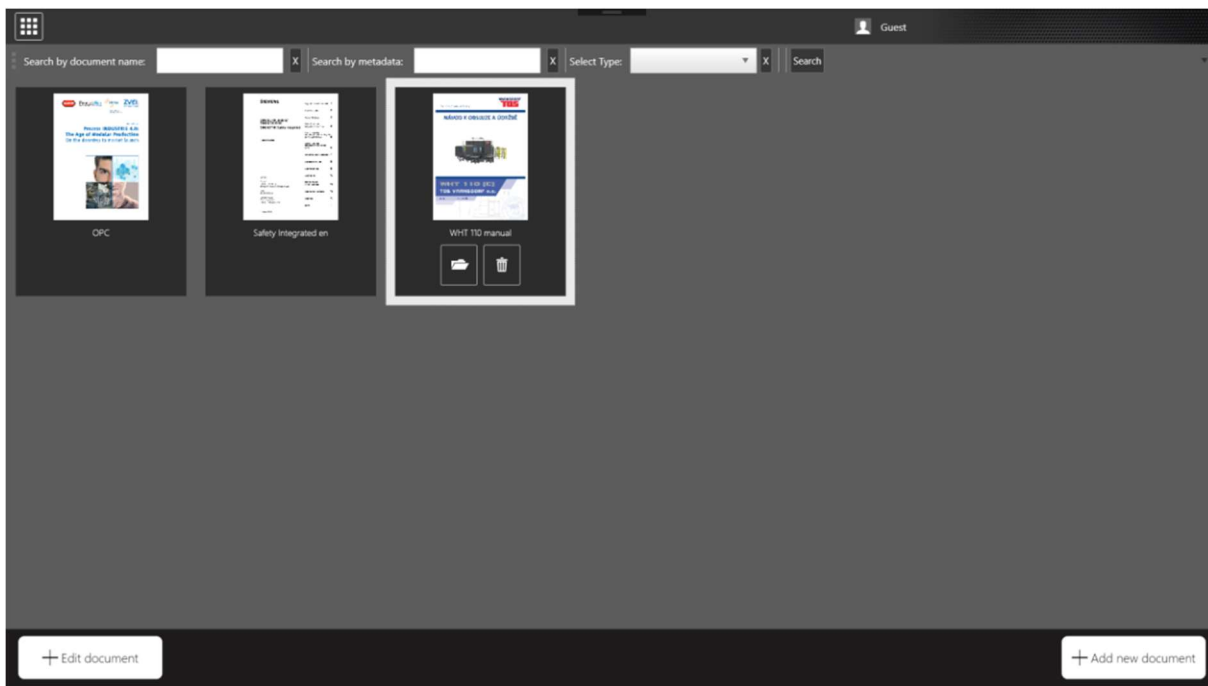
- Manažer zakázek (Job management) – zobrazí pohled na zakázku a přehled jejích operací na časové ose přímo na stroji. Bude umožňovat stahování dat o operaci z ERP přímo na stroji včetně dokumentace a NC programu (Obrázek 19, Obrázek 20).
- Monitoring stroje – Integrovaný systém pro monitorování historie používání stroje na časové ose. Zobrazuje základní stavy pro například následný výpočet OEE.
- Dalšími aplikacemi v ranném vývoji jsou: Teplotní kompenzace nástrojů, Charakteristiky strojů, Údržba strojů, Management nástrojů



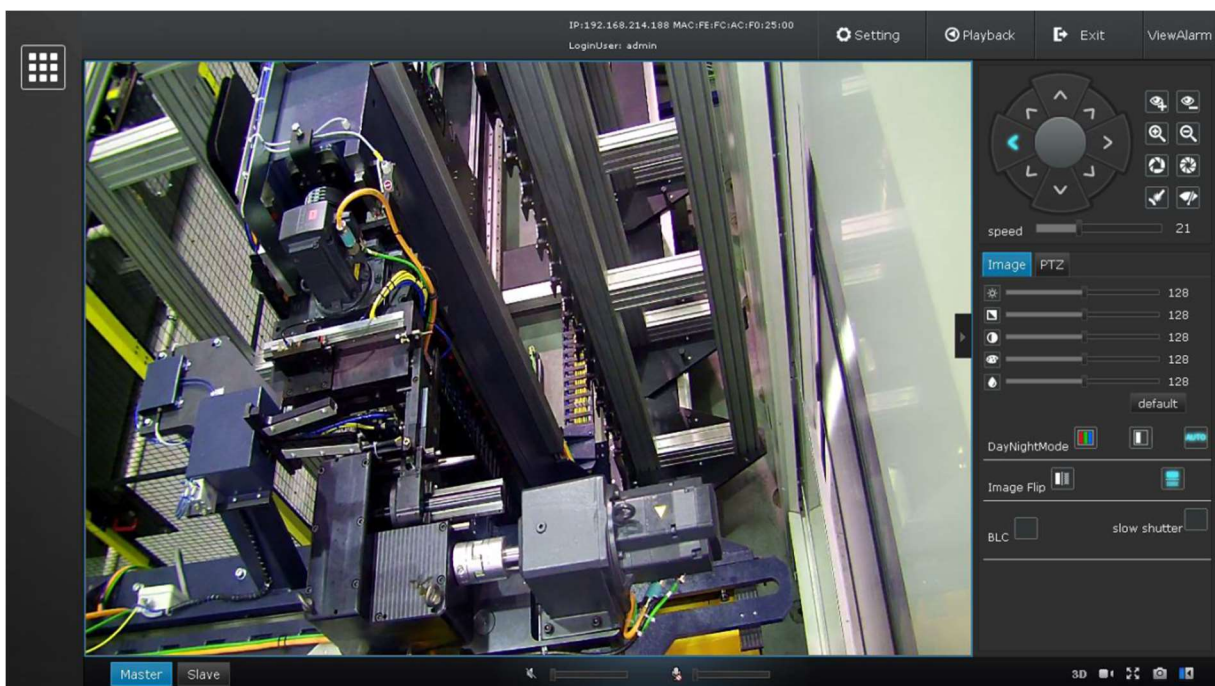
Obrázek 15 Aplikace Status[34]



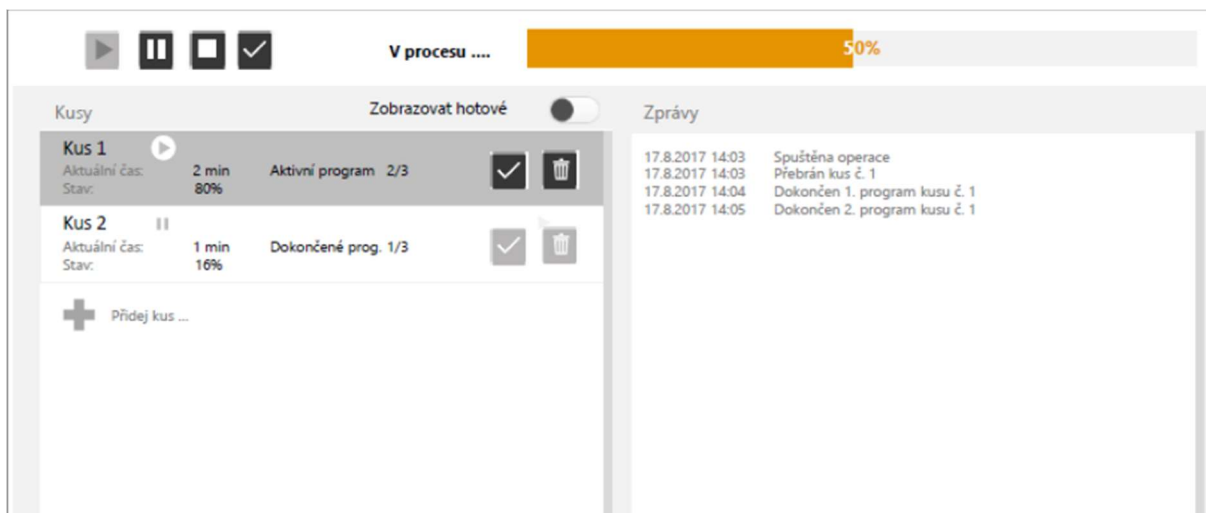
Obrázek 16 Aplikace Kalendář[34]



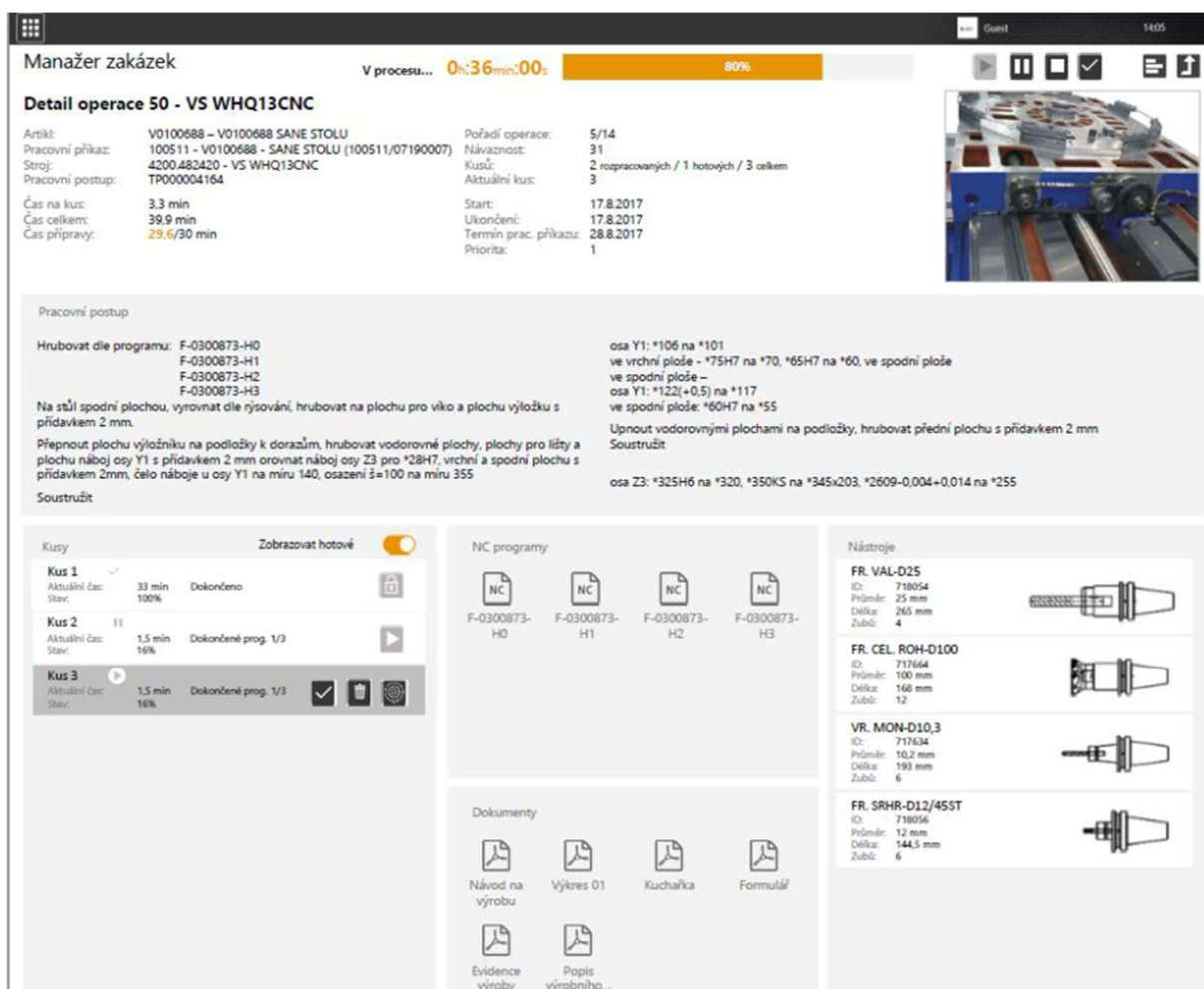
Obrázek 17 Aplikace Dokumentace[34]



Obrázek 18 Aplikace IP kamera[34]



Obrázek 19 Aplikace Manažer zakázek[34]



Obrázek 20 Stávající forma náhledu operace v Manažeru zakázek[34]

5.4 Požadavky na systém rozvrhování (hodnocení kvality rozvrhu)

Konzultací ve firmě TOS Varnsdorf a.s. byly určeny základní požadavky na rozvrhovací systém. Absolutní prioritou pro firmu TOS Varnsdorf a.s. je splnění termínů zakázek. Jak již bylo nastíněno v sekci 2.2, ukazatelem plnění termínů zakázek je například odchylka od termínu zakázky (T_j – tardiness). Tento ukazatel bude stěžejním pro algoritmus.

Dalším požadavkem je preference strojů, které v danou chvíli splňují požadavky zakázek na kvalitu výroby na stroji. Tento požadavek je stanoven technologickým postupem. Aktuální kvalita výroby na stroji, ve firmě často zmiňovanou jako technický stav stroje, je počítána průběžně kontrolou vyrobených dílů ať už přímo na stroji (u strojů vybavených kontaktním měřícím systémem) nebo na středisku kontroly kvality. Aktuální technický stav stroje je automaticky vypočten z metrologických reportů. 10% odchylka tolerance od nominálního rozměru povolené tolerance je brána jako optimální stav (kvalita = 1) následně hodnota kvality lineárně klesá až na hodnotu 0,5. Nesplnění tolerancí znamená posun pod 0,5. Pro příklad, kdy stroj vyrobil díl, který měl jednu sledovanou toleranci (22J8), byl tento rozměr předepsán na 22^{+2}_{-10} mm a následný změřený rozměr na výrobku byl 22,008 mm. Výsledný rozměr se vešel do dané tolerance a algoritmus mu vypočetl hodnotu kvality výroby 0,6 (60 %). Pro získání hodnoty technického stavu stroje je vypočten průměr všech hodnot kvality u technologickým postupem předepsaných tolerancí. Zároveň je tento parametr ovlivněn předešlými měřeními pro daný stroj. Hodnota technického stavu stroje je dána výpočtem klouzavého průměru z posledních 10 vyrobených dílců. Při hodnotě technického stavu stroje menší než 0,5 je stroj vyhodnocen jako nevhodující pro danou operaci a v intervalu (0,5-1) dochází k preferování stroje s vyšší hodnotou kvality, pokud je to možné.

Posledním parametrem účelové funkce je výpočet celkového času výroby. Celkový čas výroby je vypočten dle sekce 2.

5.5 Analýza datové základny

V této kapitole je představen náhled do datové základny dostupných dat z ERP a dat dostupných v teoretických publikacích[35].

5.5.1 Tvorba testovacího datového modelu

Testovací datové modely byly modely MK01 až MK10, pro které již bylo nalezeno brute-forcingem optimální řešení. Stejně testovací modely byly použity také v disertační práci Ing. Františka Koblasy, PhD.[1]. Modely jsou popsány (Tabulka 2). Zkratky v tabulce:

- *pocj* = počet zakázek
- *pocm* = počet strojů
- *poco* = minimální a maximální počet operací zakázky
- *meq* = maximální množství ekvivalentních strojů
- *proc* = maximální a minimální procesní čas operací
- *opt* = zjištěné teoretické optimální řešení makespanu

Tabulka 2 Přehled modelů MK01 – MK10

	<i>pocj</i>	<i>pocm</i>	<i>poco</i>	<i>meq</i>	<i>proc</i>	<i>opt</i>
MK01	10	6	5–7	3	1–7	36
MK02	10	6	5–7	6	1–7	24
MK03	15	8	10–10	5	1–20	204
MK04	15	8	3–10	3	1–10	48
MK05	15	4	5–10	2	5–10	168
MK06	10	15	15–15	5	1–10	33
MK07	20	5	5–5	5	1–20	133
MK08	20	10	5-10	2	5–20	523
MK09	20	10	10–15	5	5–20	299
MK10	20	15	10–15	5	5–20	165

5.5.2 Informace o zakázkách

Jako testovací model ze společnosti TOS VARNSDORF a.s. byla vybrána dvoutýdenní zásoba práce těžké mechaniky. Pro modelování řešení v této práci byly vyexportovány z ERP QAD ve formátu .XLS. ERP vypisuje zásobu práce v řádcích jakožto pracovní příkazy s těmito informacemi:

- 1. sloupec: Číslo pracovního příkazu
- 2. sloupec: Status operace (uvolněná/neuvolněná do výroby ~ R/P)

- 3. sloupec: Číslo artiklu = ID z interního kusovníku (číslo artiklu se může vyskytovat ve více pracovních příkazech, pokud se podle TP jedná o stejný díl)
- 4. sloupec: Název vyráběné položky
- 5. sloupec: Čas seřízení pro výrobní dávku (T_{bc})
- 6. sloupec: Čas výroby jednoho kusu ve výrobní dávce (t_{ac})
- 7. sloupec: Procesní čas výrobní dávky (T_a)
- 8. sloupec: Čas výroby celé výrobní dávky včetně seřízení (T_{sum})
- 9. sloupec: Kód výrobního střediska (pto TM = „4200“)
- 10. sloupec: ID pracovního příkazu
- 11. sloupec: Číslo operace z technologického postupu
- 12. sloupec: Název stroje
- 13. sloupec: ID stroje
- 14. sloupec: Objednané množství
- 15. sloupec: Dokončené množství kusů z výrobní dávky
- 16. sloupec: Počet zmetků
- 17. sloupec: Zbývající množství ve výrobní dávce
- 18. sloupec: Předepsaný start operace z MRP I
- 19. sloupec: Termín splnění pracovního příkazu

Dále je ERP systém QAD schopen vygenerovat směnnosti jednotlivých strojů na středisku TM také v .XLS formátu. V systému QAD je směnný kalendář, který ukazuje kolik daný stroj pracuje daný den hodin. Stroje podle tohoto směnného kalendáře běží po osmihodinových směnách, liší se však počet těchto směn pro jednotlivé stroje za den. Tyto rozdíly jsou od jedné do tří směn za den. Jednosměnný stroj pracuje 7,5 hodiny s 0,5hodinovou přestávkou, dvousměnný pracuje 15 hodin se dvěma 0,5hodinovými přestávkami a třisměnný stroj pracuje 22,5 hodiny se třemi 0,5hodinovými přestávkami. Směnný kalendář je zobrazen v následující tabulce (Tabulka 3). Názvy strojů v tabulce jsou anonymizovány, ID strojů v tabulce jsou ve formě, kterou generuje ERP. Jak je z této tabulky zřejmé, o víkendech se na středisku těžké mechaniky nepracuje.

Tabulka 3 Směnnost strojů na středisku těžké mechaniky

Stroj	Pondělí	Úterý	Středa	Čtvrtek	Pátek	Sobota	Neděle
348430	7.50	7.50	7.50	7.50	7.50	0.00	0.00
348431	7.50	7.50	7.50	7.50	7.50	0.00	0.00
448272	7.50	7.50	7.50	7.50	7.50	0.00	0.00
448280	15.00	15.00	15.00	15.00	15.00	0.00	0.00
448281	7.50	7.50	7.50	7.50	7.50	0.00	0.00
448781	15.00	15.00	15.00	15.00	15.00	0.00	0.00
448782	15.00	15.00	15.00	15.00	15.00	0.00	0.00
448783	15.00	15.00	15.00	15.00	15.00	0.00	0.00
452440	22.50	22.50	22.50	22.50	22.50	0.00	0.00
452450	22.50	22.50	22.50	22.50	22.50	0.00	0.00
482400	22.50	22.50	22.50	22.50	22.50	0.00	0.00
482410	7.50	7.50	7.50	7.50	7.50	0.00	0.00
482451	22.50	22.50	22.50	22.50	22.50	0.00	0.00
515400	7.50	7.50	7.50	7.50	7.50	0.00	0.00
567500	15.00	15.00	15.00	15.00	15.00	0.00	0.00
567610	15.00	15.00	15.00	15.00	15.00	0.00	0.00
941200	7.50	7.50	7.50	7.50	7.50	0.00	0.00
942100	7.50	7.50	7.50	7.50	7.50	0.00	0.00
952450	22.50	22.50	22.50	22.50	22.50	0.00	0.00
955300	7.50	7.50	7.50	7.50	7.50	0.00	0.00
986300	7.50	7.50	7.50	7.50	7.50	0.00	0.00

Pro plugin TOScontrolu budou po zavolání vstupů k dispozici tato výstupní data:

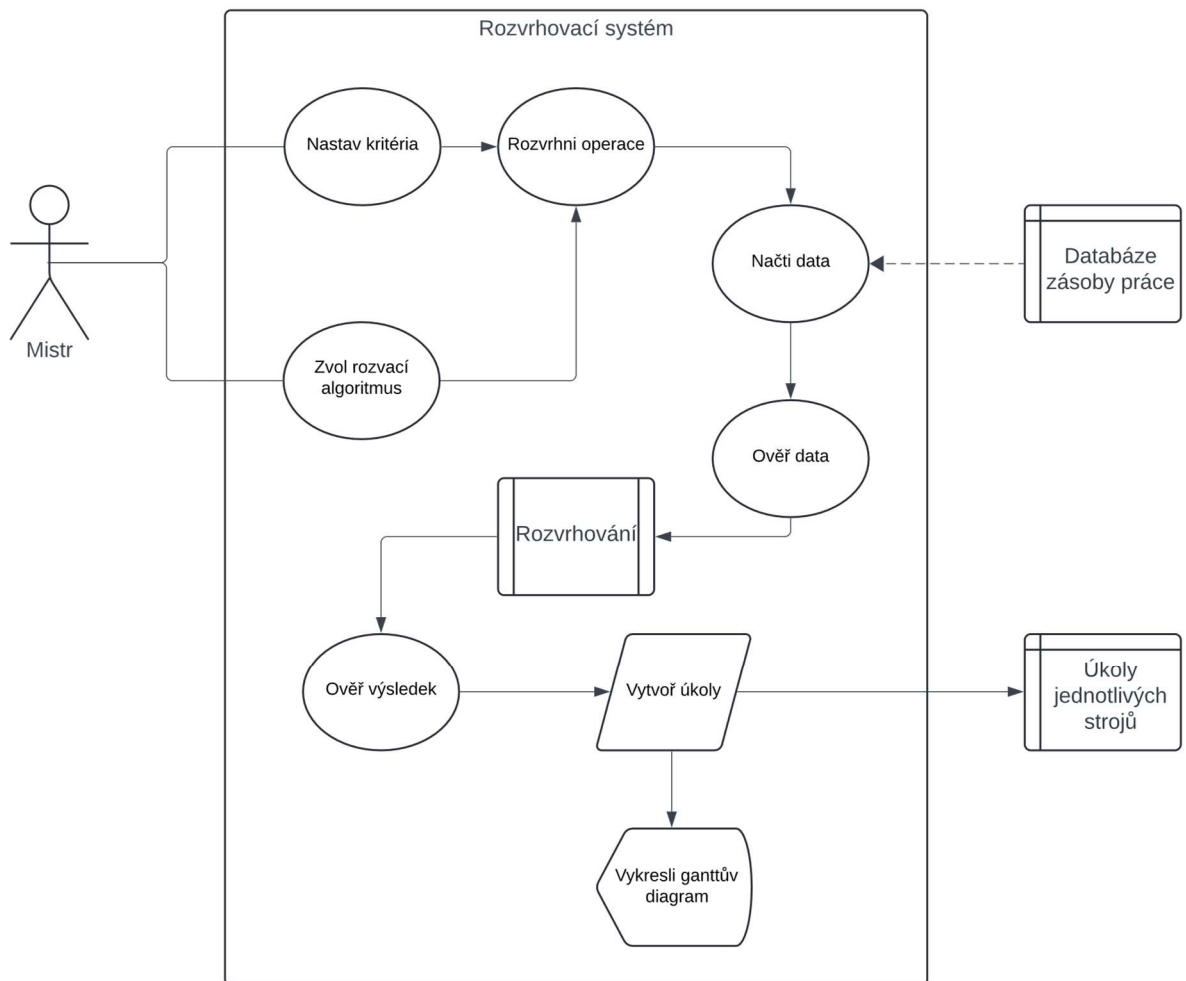
- Vstup:
 - Domain – Doména (konstanta TOSVDF)
 - Site – Místo (konstanta 10)
 - WrLot – ID Pracovního příkazu (nepovinné, vrací všechny pracovní příkazy)
 - WrOp - Číslo operace (nepovinné, vrací všechny op)
- Výstup:
 - Result – True / False (ukazuje, zdali operace skutečně existuje)
 - Site – Výrobní středisko
 - WrLot – ID Pracovního příkazu

- WrNbr – Číslo pracovního příkazu
- WoPart – ID vyráběného dílu z interního kusovníku
- PtDesc – Název vyráběného artiklu
- WrOp – Číslo operace v technologickém postupu
- WrDesc – Název operace
- WrSetup – Čas seřízení na celou dávku
- WrRun – Čas výroby na jeden kus
- WrQtyOrd – Požadované množství
- WrQtyComp – Dokončené množství (včetně kooperací)
- WrQtyRjct – Množství zmetků
- WrWkctr – Středisko operace
- WrMch – Stroj
- WcDesc – Název střediska
- TvwrcNcCode – podsegment JSON – Kódy NC Programů
- WrStart – Datum zahájení operace (Od data)
- WrDue – Termín operace
- WrStatus – Status operace (R/P)

6 Návrh algoritmů

Tato kapitola se bude zabývat popisem postupu tvorby algoritmů pro dynamické rozvrhování. Slovní popis postupu a funkcí jednotlivých algoritmů je doplněn o schématické znázornění daných algoritmů.

6.1 Statický rozvrh

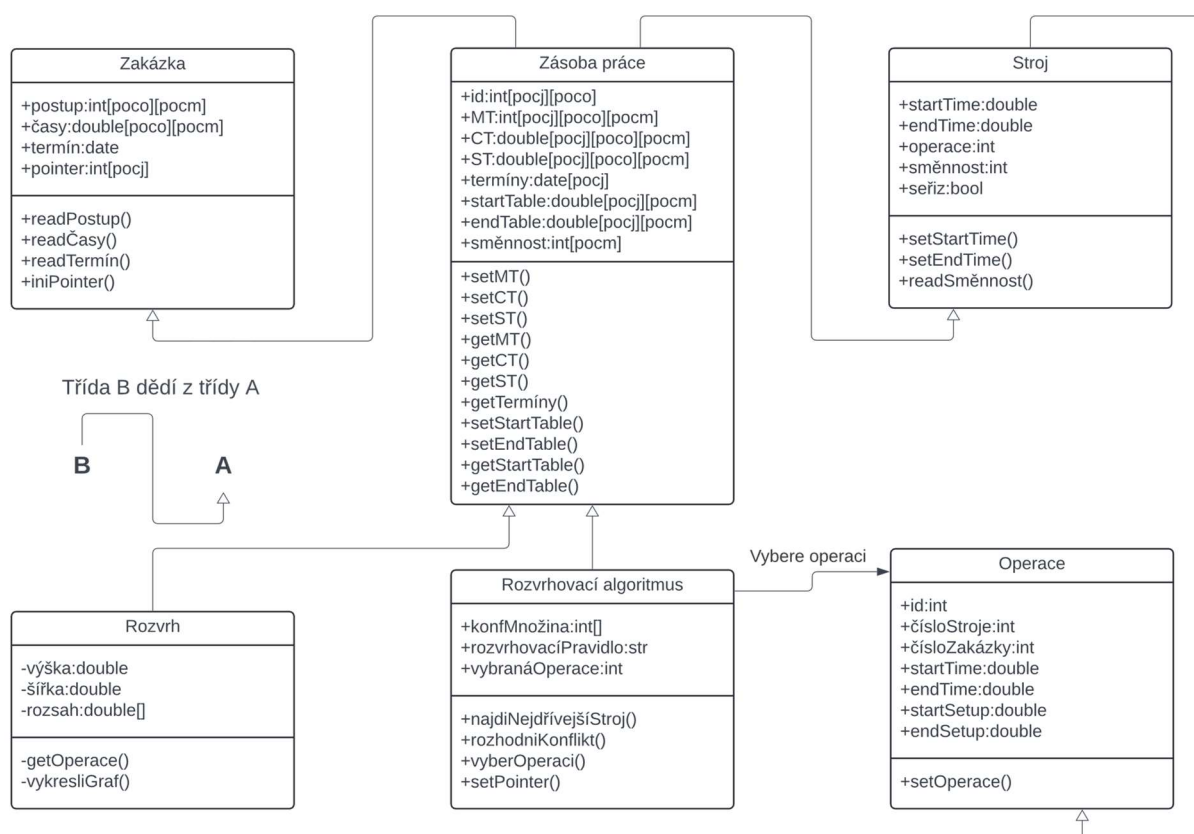


Obrázek 21 Struktura statického rozvrhování

Rozvrhy byly tvořeny metodou konstruktivního algoritmu. Konstruktivní algoritmus je takový algoritmus, jenž do zpočátku prázdného rozvrhu vkládá postupně operace podle stanovených pravidel, například na základě prioritních pravidel. Dokáže zpravidla pracovat s omezeními, jako jsou technologický postup, směnnost, požadavky na seřízení. Konstruktivní algoritmy mají zpravidla konečný počet kroků[1]. Obrázek 21 ukazuje základní strukturu fungování aplikace pro rozvrhování operací na stroje. Mistr nejprve nastaví kritéria a typ rozvrhovacího algoritmu. Aplikace posléze importuje data z databáze a ověří

se jejich kompletnost. Data z databáze zásoby práce jsou posléze rozvržena podle zvoleného algoritmu a výsledek rozvrhování je zobrazen na Ganttově diagramu a úkoly jsou odeslány na stroje vybavené modulem TOScontrol a vyexportovány ve formátu tabulky. Pro stroje nevybavené modulem TOScontrol jsou úkoly dávány standardně jako doposud.

Na následujícím obrázku (Obrázek 22) je vyobrazena struktura předávání informací mezi jednotlivými částmi algoritmu. Příklad je pro rozvrhování pomocí rozvrhovacího pravidla a vytvoření rozvrhu bez zpoždění. Je důležité si povšimnout, kterým směrem probíhá dědění informací mezi jednotlivými třídami algoritmu, jak je zobrazeno na obrázku.



Obrázek 22 Struktura rozvrhovacího algoritmu

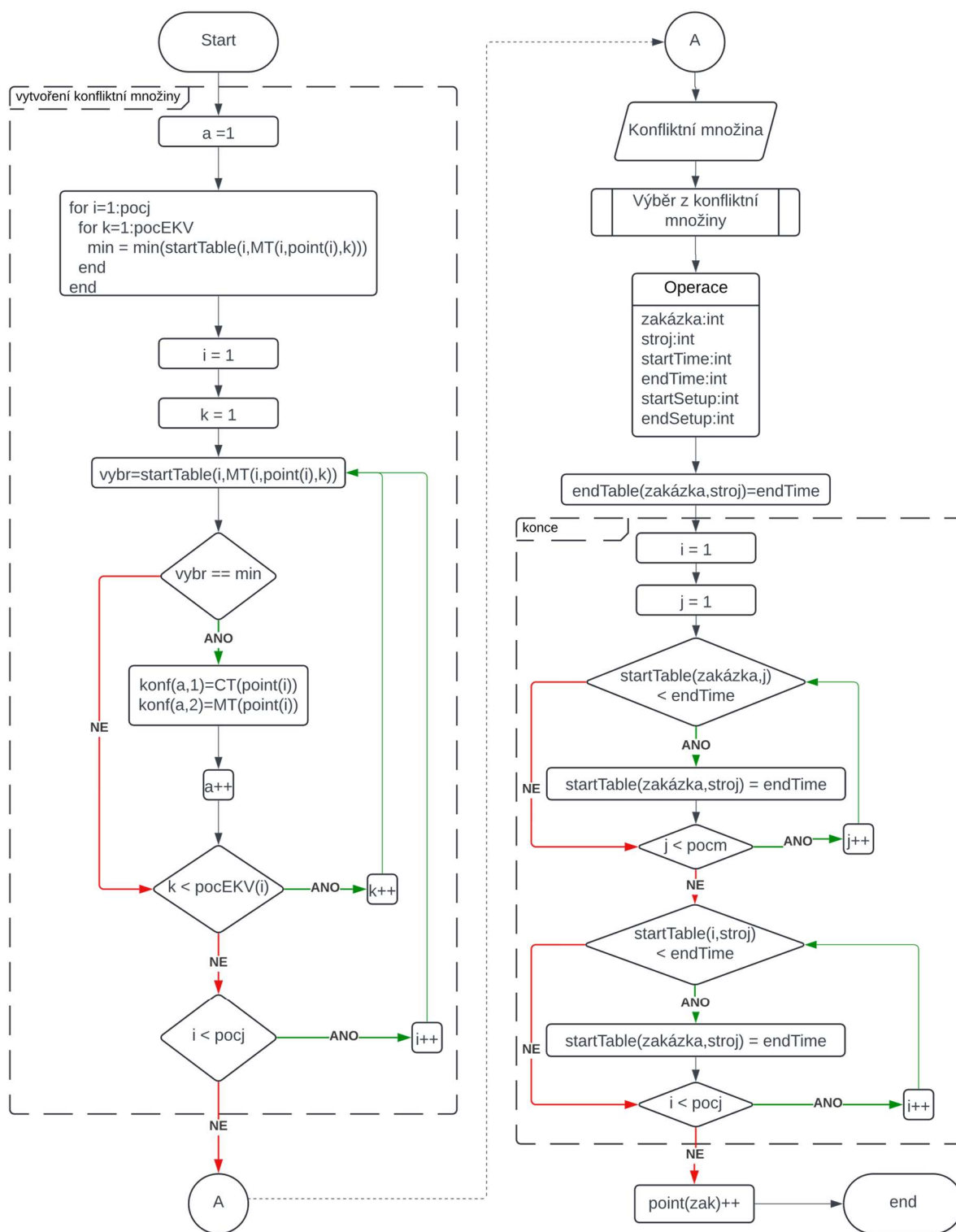
Tento algoritmus (Obrázek 22) funguje tak, že vytvoří z poskytnutých dat o strojích a zakázkách takzvanou zásobu práce, kterou uchovává ve formě matic CT, MT, ST, startTable, endTable a termíny. Jednotlivé časy vyjadřují:

- CT – časy zpracování dané operace na stroji, jedná se o 3D matici, kde:
 - x: řádek zobrazuje jednu zakázku,
 - y: operace seřazenými podle výrobního postupu ve sloupcích,
 - z: časy zpracování na alternativních strojích.

- ST – časy seřizování dané operace na stroji, jedná se o 3D matici, kde:
 - x: řádek zobrazuje jednu zakázku,
 - y: operace seřazenými podle výrobního postupu ve sloupcích,
 - z: časy zpracování na alternativních strojích.
- MT – Jedná se o 3D matici, kde každá buňka nese ID stroje, na kterém je zpracována daná operace s časem korespondujícím s CT maticí.
 - Pokud má například 3. operace 1. zakázky 4 alternativní stroje, je čas zpracování pro tento 4. stroj uložen na pozici CT(1,3,4) a čas zpracování je uložen na pozici MT(1,3,4).
- startTable – jedná se o tabulku zachycující nejdřívejší možné časy zaplánování operace zakázky X na stroji Y ($X=1\dots$ počet zakázek a $Y=1\dots$ počet strojů).
- endTable – jedná se o tabulku zachycující konce zaplánovaných operací zakázky X na stroji Y ($X=1\dots$ počet zakázek a $Y=1\dots$ počet strojů.).
- termíny – jedná se o matici zachycující předepsané data odvedení zakázek

Tyto data jsou poté předána modifikovanému rozvrhovacímu algoritmu Gifflera a Thompsona, který pro rozvrh bez zpoždění (Obrázek 23) hledá stroj, na které je možno rozvrhnout operaci co nejdříve. Rozvrhovací algoritmus poté vytvoří takovou konfliktní množinu, ve které jsou tyto nejdřívejší možné operace se všemi svými potřebnými informacemi (pro SPT a LPT například konfliktní množina nese data o umístění zakázky a o její délce). Algoritmus nadále například podle prioritního pravidla vybere z konfliktní množiny takovou operaci, která má nejvyšší index priority. Pokud je více zakázek s nejvyšší prioritou, vybírá z nich poté algoritmus pseudonáhodně jednu. Algoritmus v této práci používá rovnoměrně rozdělené pseudonáhodné veličiny.

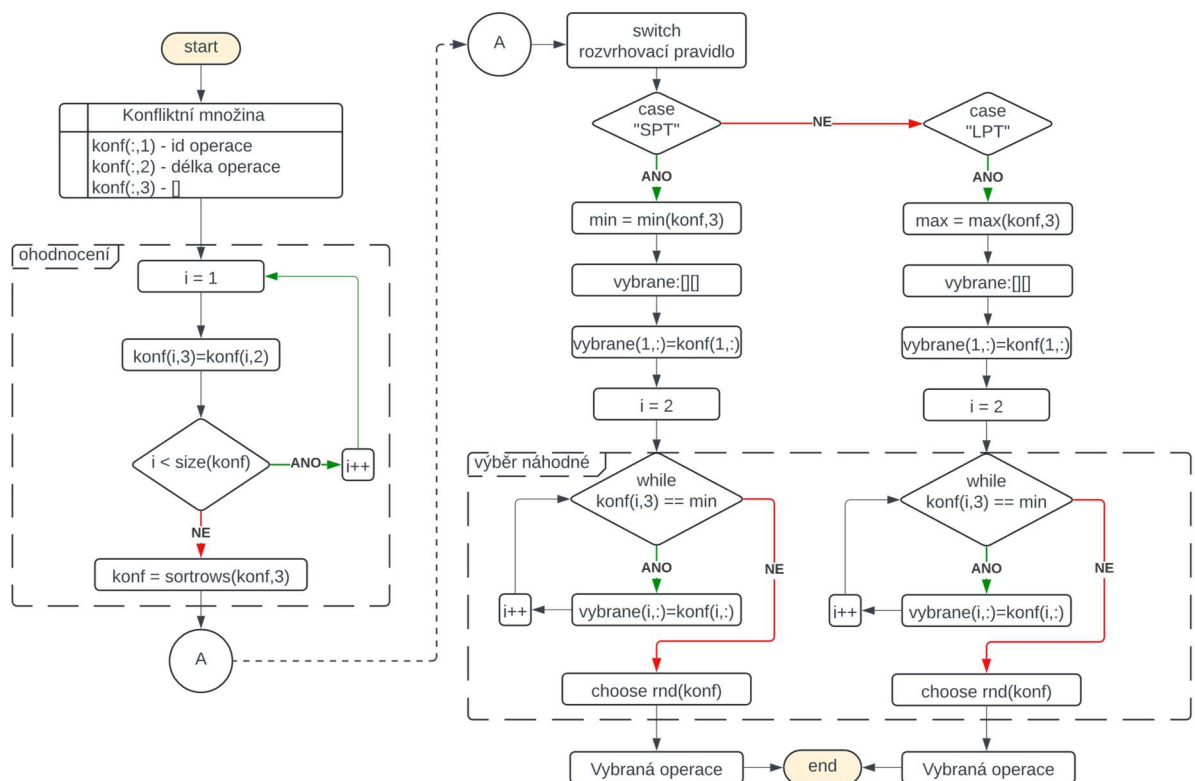
Vybraná operace je následně zařazena do seznamu vybraných úkolů (*vybr*) a je jí přiřazeno číslo výrobního stroje, na kterém bude zpracována, čas jejího počátku a čas jejího konce. Dále je jí přiřazen začátek a konec seřizování. Poté je zařazena do výrobního rozvrhu a je přepočítána matice endTable a podle ní poté i matice startTable. Ve chvíli, kdy jsou rozvrženy všechny operace, je vykreslen Ganttův diagram.



Obrázek 23 Modifikovaný algoritmus Gifflera a Thompsona pro vytvoření rozvrhu bez zpoždění

6.1.1 Prioritní pravidla

Obrázek 24 ukazuje algoritmus pro výběr operace z konfliktní množiny. Jedná se o proces nastíněný na předešlém obrázku (Obrázek 23). V první (pravé) části algoritmu je zobrazeno ohodnocení konfliktní množiny podle prioritních pravidel pro SPT a LPT zmíněných dříve (Tabulka 1, kapitola 3.4.2.1). Pro tyto dvě pravidla je výpočet ohodnocení stejný, liší se až výběrem z ohodnocené konfliktní množiny. Na obrázku jsou dále nastíněny dvě větve algoritmu, které se mění podle zvoleného prioritního pravidla. Podle použitého prioritního pravidla je poté hledána nejnižší hodnota indexu priority pro SPT v posledním sloupci konfliktní množiny a nejvyšší hodnota indexu priority pro LPT. Dále následuje cyklus výběru vybraných operací s minimální či maximální hodnotou prioritního indexu. Pokud je vybraných operací více, pak následuje pseudonáhodný výběr operace podle rovnoměrného rozložení.



Obrázek 24 Výběr z konfliktní množiny pomocí prioritních pravidel SPT a LPT

6.1.2 Genetický algoritmus

Obrázek 25 zachycuje strukturu genetického algoritmu použitého v této práci. Jedná se o jedno-chromozomový GA ve kterém křížení probíhá na základě výměny genů obsahujících id zakázek mezi jedinci. V prvním kroku jsou načteny data ze zásoby práce jako

v případě algoritmu pro prioritní pravidla. Načítána jsou opět CT, MT, ST, termíny, pointer, směnnost strojů a počet kusů pro jednotlivé zakázky. Dále jsou načítány proměnné:

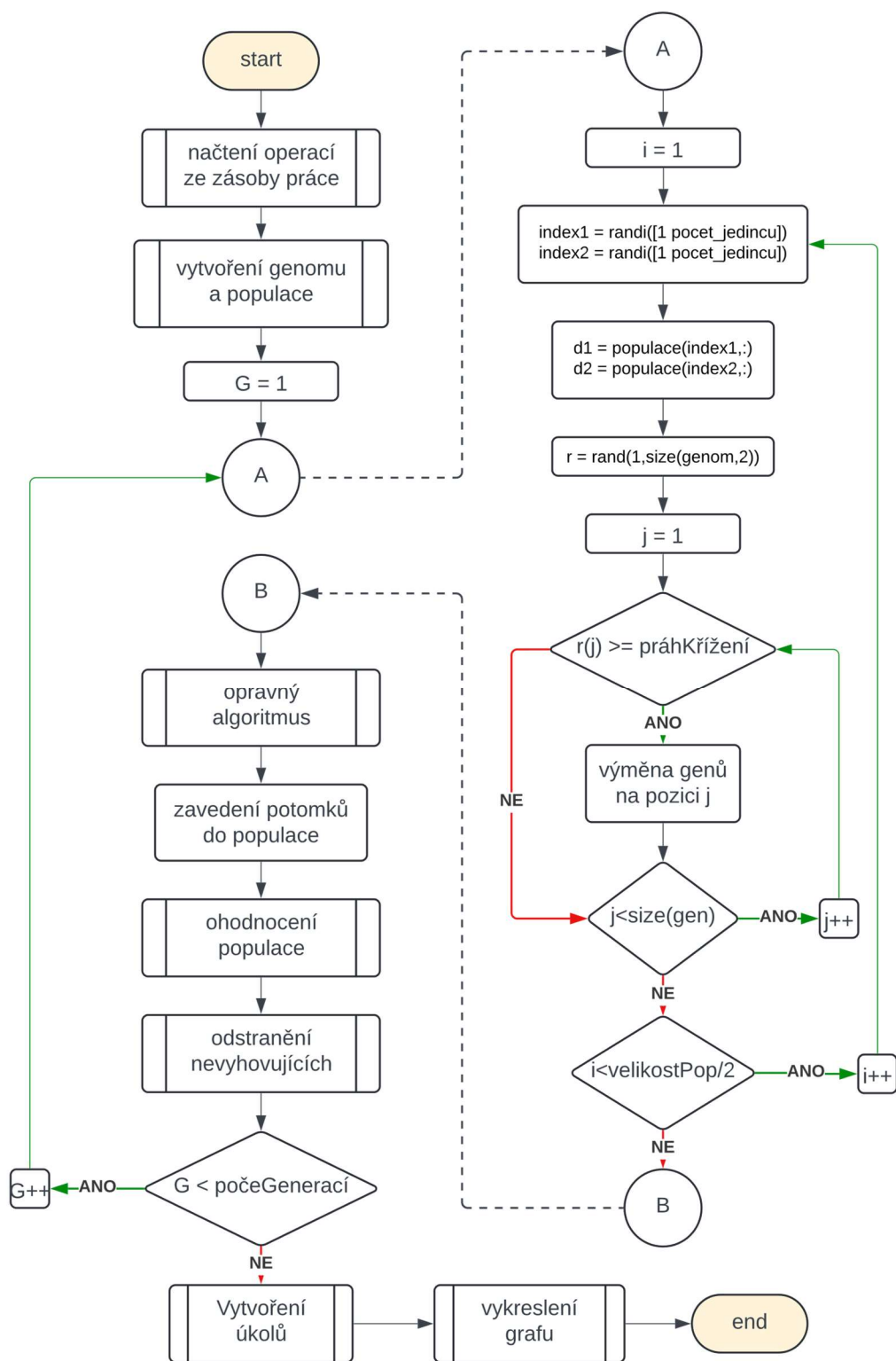
- pocj – počet zakázek,
- pocm – počet strojů,
- poco – maximální počet operací v jedné zakázce,
- poc_celk – celkový počet načtených operací,
- klic – obsahuje data pro jednotlivé zakázky jako jejich ID v ERP, id v algoritmu a termín odvedení dané zakázky.

Vytvoření genomu probíhá cyklicky, kdy se vytvoří pomocný vektor, který obsahuje počet operací pro jednotlivé zakázky a poté se z něj vytvoří vektor, který obsahuje daný počet ID zakázky jako je v pomocném vektoru. Pro zásobu práce, která bude složena ze tří zakázek, kde budou mít:

- zakázka 1 jednu operaci
- zakázka 2 tři operace
- zakázka 3 dvě operace

bude vytvořen genom: [1 2 2 2 3 3].

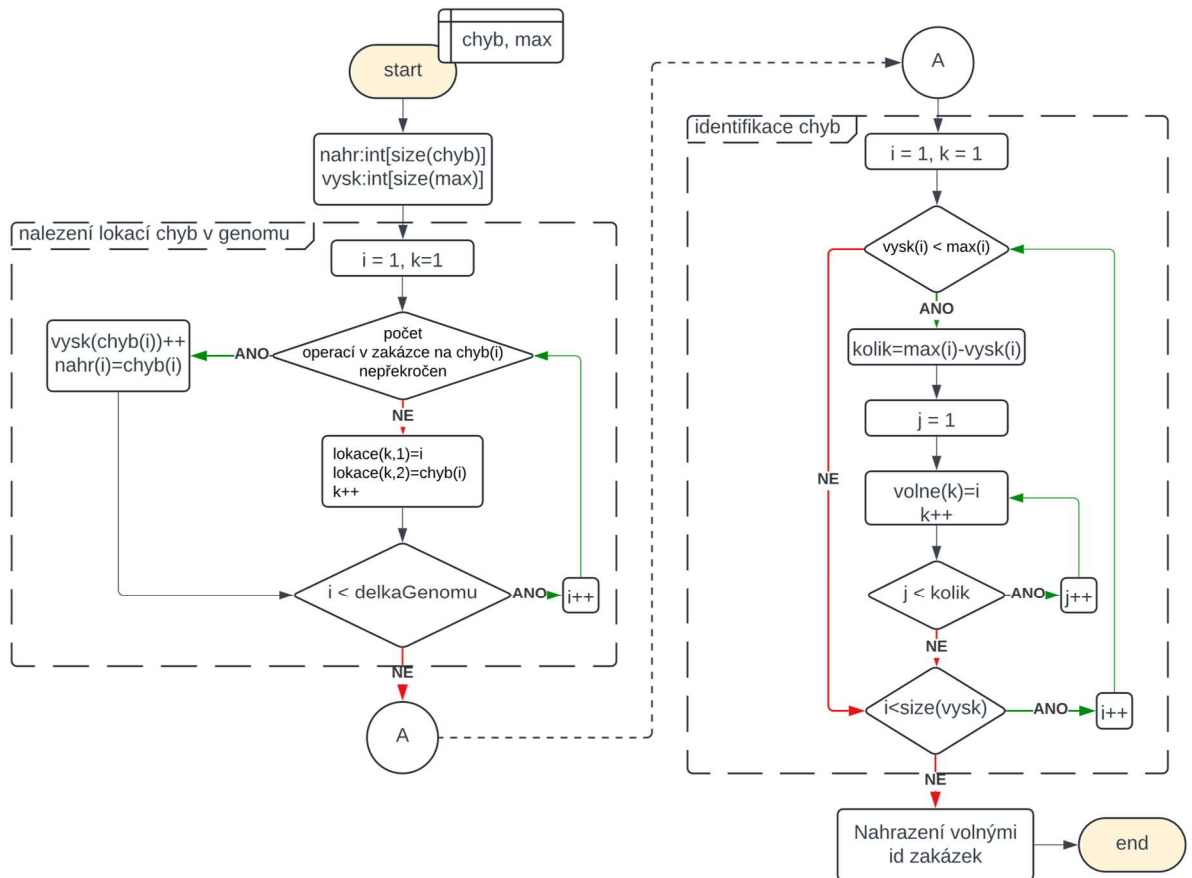
Dále následuje vytvoření populace jedinců, kdy se vytvoří daný počet jedinců. Každý jedinec bude permutací genomu, který je vytvořen, jak je uvedeno výše. GA (Obrázek 25 vpravo) probíhá pro předem daný počet generací G . V každé generaci probíhá takový počet křížení, jako je polovina velikosti populace, aby byla na konci populace dvojnásobná. GA poté vybere z populace pseudonáhodně podle rovnoměrného rozložení dva jedince a vyměňuje při překročení předem daného prahu křížení geny mezi rodiči a ukládá nové geny do dvou nových chromosomů potomků. Poté, kdy cyklus projde celou délku obou chromosomů a jsou vytvořeni dva unikátní potomci, začne probíhat opravný algoritmus chromosomů.



Obrázek 25 Struktura genetického algoritmu

Jelikož se jedná o sekvenční problém, ve kterém je přesně dáno kolik, kterých ID zakázek může v jednom chromosomu být, musel být navrhnout opravný algoritmus chromosomů, aby bylo toto pravidlo dodrženo. Obrázek 26 zachycuje tento opravný algoritmus, do kterého vstupují chybný chromosom (chyb) a vektor počtu operací v jednotlivých zakázkách

(max). V levé části algoritmu se nachází cyklus, který prochází chromozom a zjišťuje, zdali se není některé ID zakázek v chromozomu nevyskytuje příliš mnohokrát. Na svém konci vrátí lokace chyb a ID zakázky, která se takto provinila. Tyto informace jsou poté předány cyklu vpravo, který vrátí ID volných zakázek, jimž nebyl vyčerpán počet operací. Následně je na lokacích z prvního cyklu změněn gen chromosomu z chybného na gen obsahující ID volné nevyčerpané zakázky. Vektor volných je před vrácením do chromosomu zamíchán, aby se rozšiřovala diverzita.

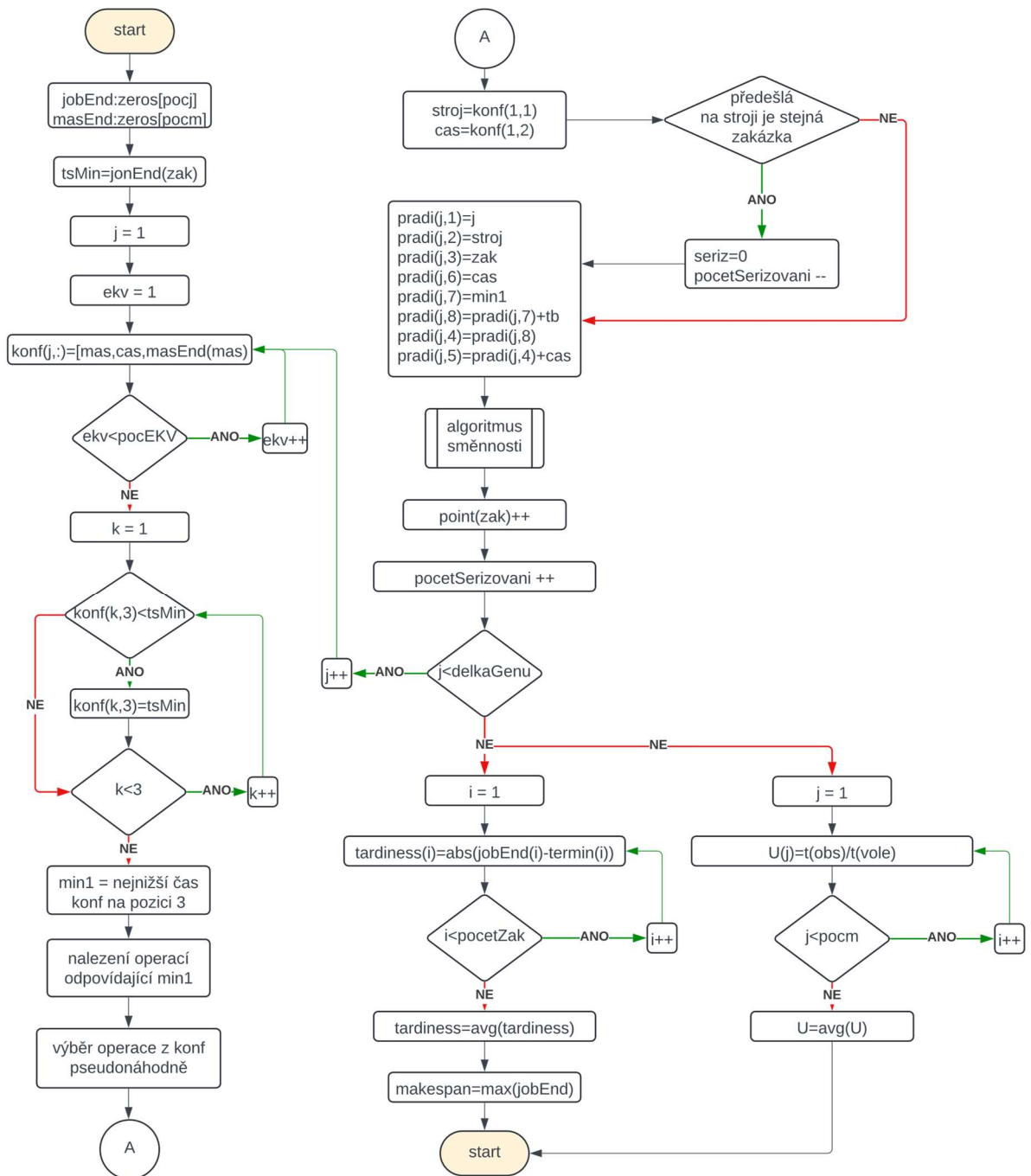


Obrázek 26 Opravný algoritmus genů

Ohodnocení populace (Obrázek 27) probíhá tak, že je vytvořen rozvrhovací algoritmus, který podle pořadí v genů v chromosomu jedince zaplánuje na nejdřívější volný termín operaci na nejdřívější volný stroj s ohledem na konec předešlé operace v zakázce. Pokud se v konfliktní množině vyskytne více strojů, které mohou začít ve stejný čas, dojde znovu k pseudonáhodnému výběru podle rovnoměrného rozložení. Jedná se o jednochromozový GA.

Operace se poté přidá do zásoby úkolů (pravá část algoritmu) kde na první pozici je pořadové číslo úkolu, na druhé je stroj, který bude zpracovávat operaci, ve třetím sloupci je

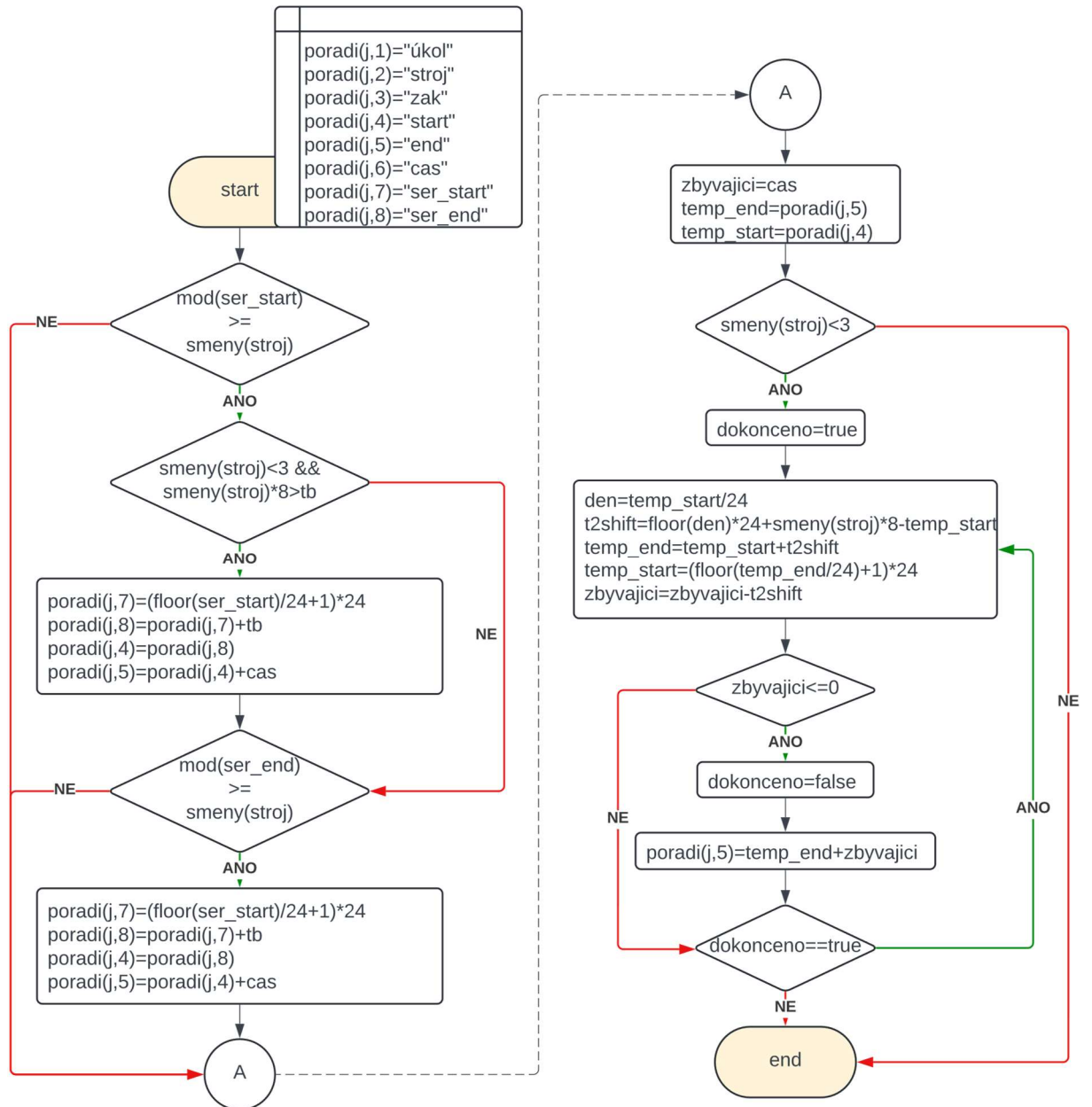
zakázka náležití operaci a ve čtvrtém je start operace. Poté přichází na řadu algoritmus směnnosti, který se stará o to, aby operace nebyla zaplánována v přestávce. Následně jsou vypočítány hodnoty ÚF makespan a tardiness.



Obrázek 27 Algoritmus vytvoření a ohodnocení jedince populace

Algoritmus na níže uvedeném obrázku ukazuje, jak je v programu řešena směnnost (Obrázek 28). Algoritmus řešení směnnosti hlídá, aby žádná z operací nezačala v přestávce mezi směnami a aby žádná ze zakázek neprobíhala ve zmíněné přestávce. Pokud přichází na stroj operace, která nepůjde stihnout v dané směně, nastává rozložení zakázky na části,

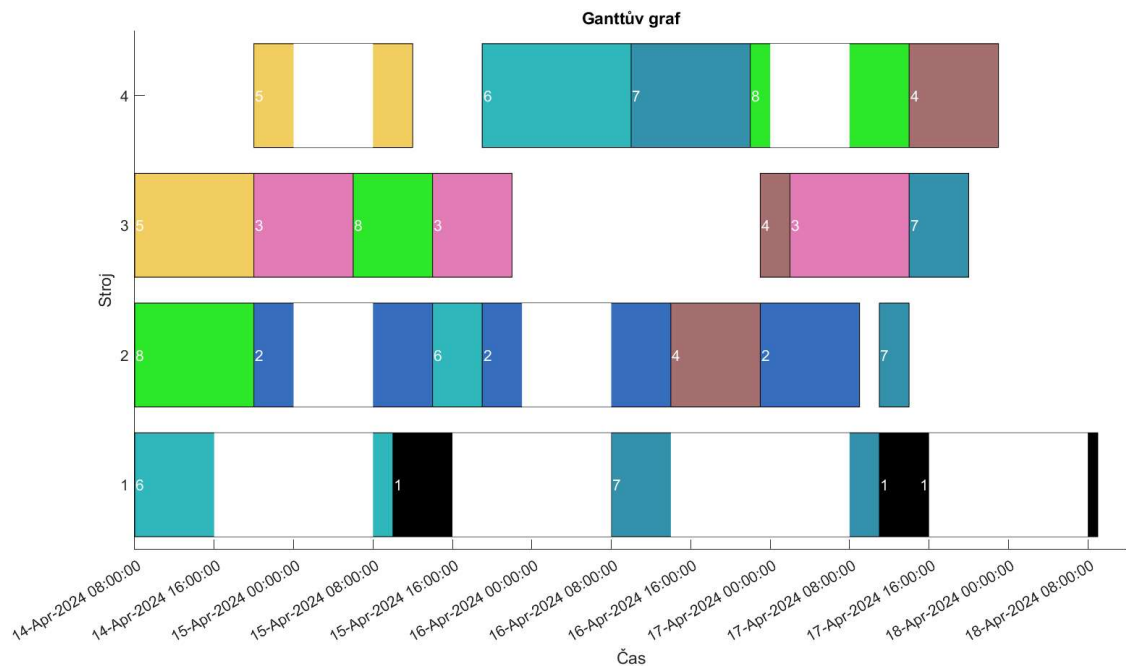
kteře budou zpracovávány ve více po sobě jdoucích směnách. Algoritmus nejprve zkontroluje, zda nezačíná, nebo nekončí seřizování operace v přestávce mezi směnami (levá část). V pravé části algoritmu, je algoritmus rozdělení operace do více po sobě jdoucích dní.



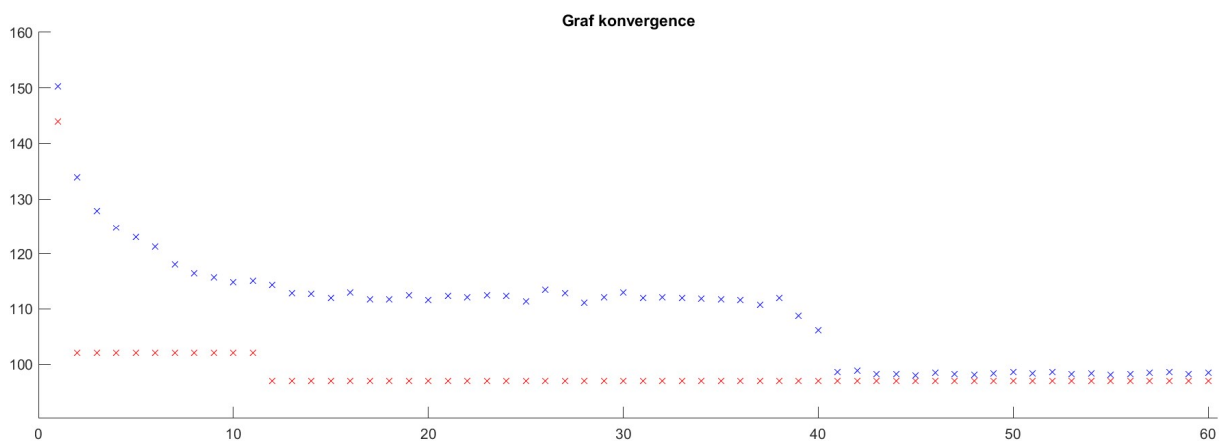
Obrázek 28 Algoritmus řešení směnnosti

Algoritmus „dělení dávek po kusech“, je v přílohách (Příloha 1/ 1). Na konci směny, ve které se celá výrobní dávka nepodaří zpracovat je zpracována pouze část, tedy počet kusů, které se stihnou vyrobit v dané směně. Algoritmus funguje tak že upravuje nejdřívější možné časy začátků dalších operací na daném stroji anebo v rámci stejné zakázky.

V GA následuje ohodnocení populace podle zvoleného vícekriteriálního hodnocení. Populace je poté seřazena podle ohodnocení a do další generace pokračuje pouze původní počet jedinců specifikovaný na začátku programu. Algoritmus, po projití daného počtu generací, následně vybere nejlepšího jedince z populace a stejným algoritmem jako pro ohodnocení jedince vytvoří seznam úkolů, který je poté předán algoritmu pro vytvoření Ganttova diagramu (Obrázek 29). V rámci algoritmu je také vykreslován graf (Obrázek 30) zachycující nejlepšího jedince a průměr populace.



Obrázek 29 Příklad Ganttova diagramu vytvořený algoritmem

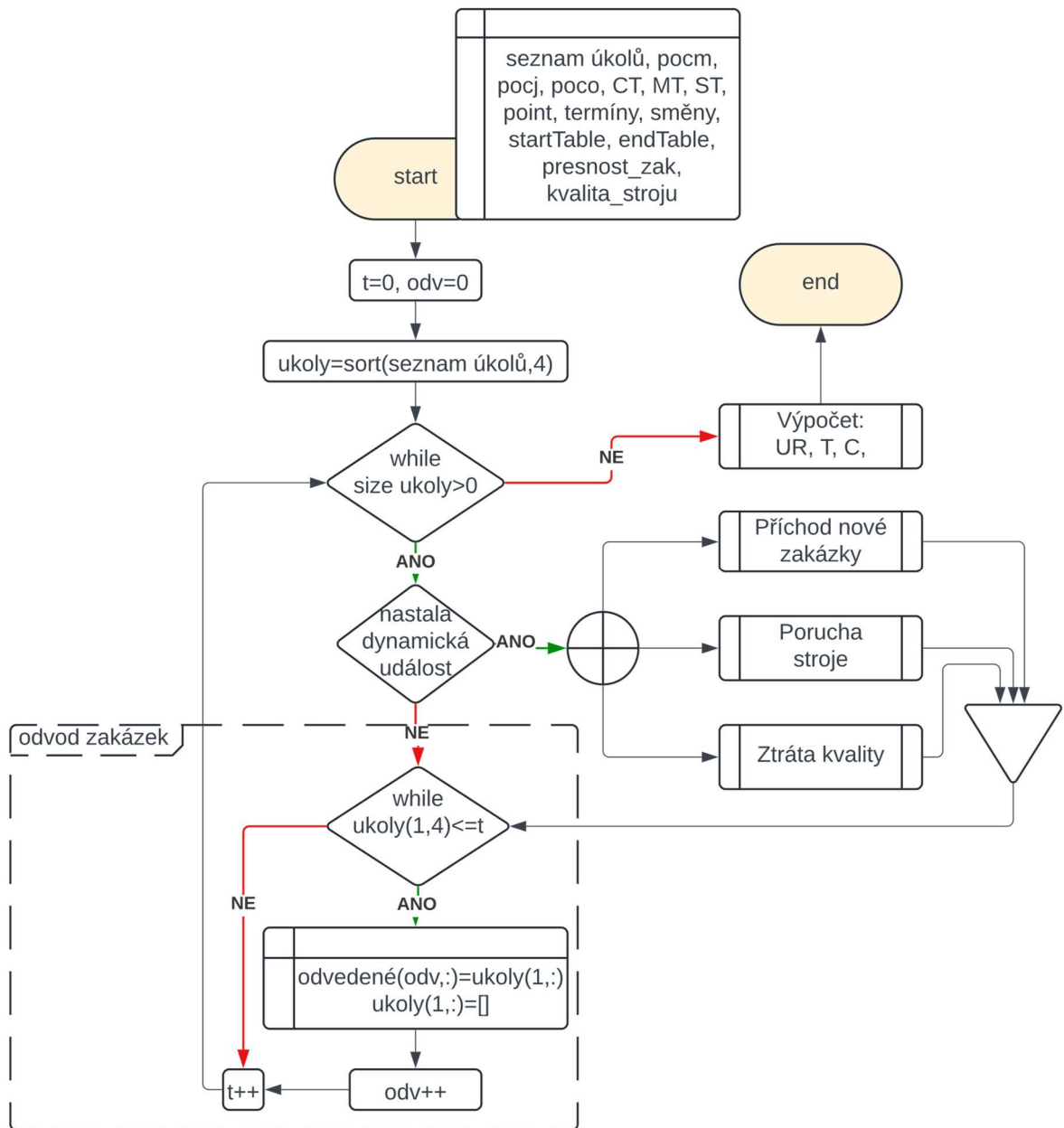


Obrázek 30 Graf zachycující nejlepšího jedince (červeně) a průměr populace (modře)

6.2 Dynamický rozvrh

Tvorba dynamického rozvrhu probíhala pomocí algoritmu, znázorněném na následujícím diagramu (Obrázek 31). Vstupními daty pro tento algoritmus jsou stejná data jako pro

tvorbu statického algoritmu s tím rozdílem, že zde seznam úkolů vyprodukovaný statickým rozvrhem nahrazuje zásobu práce statického algoritmu.



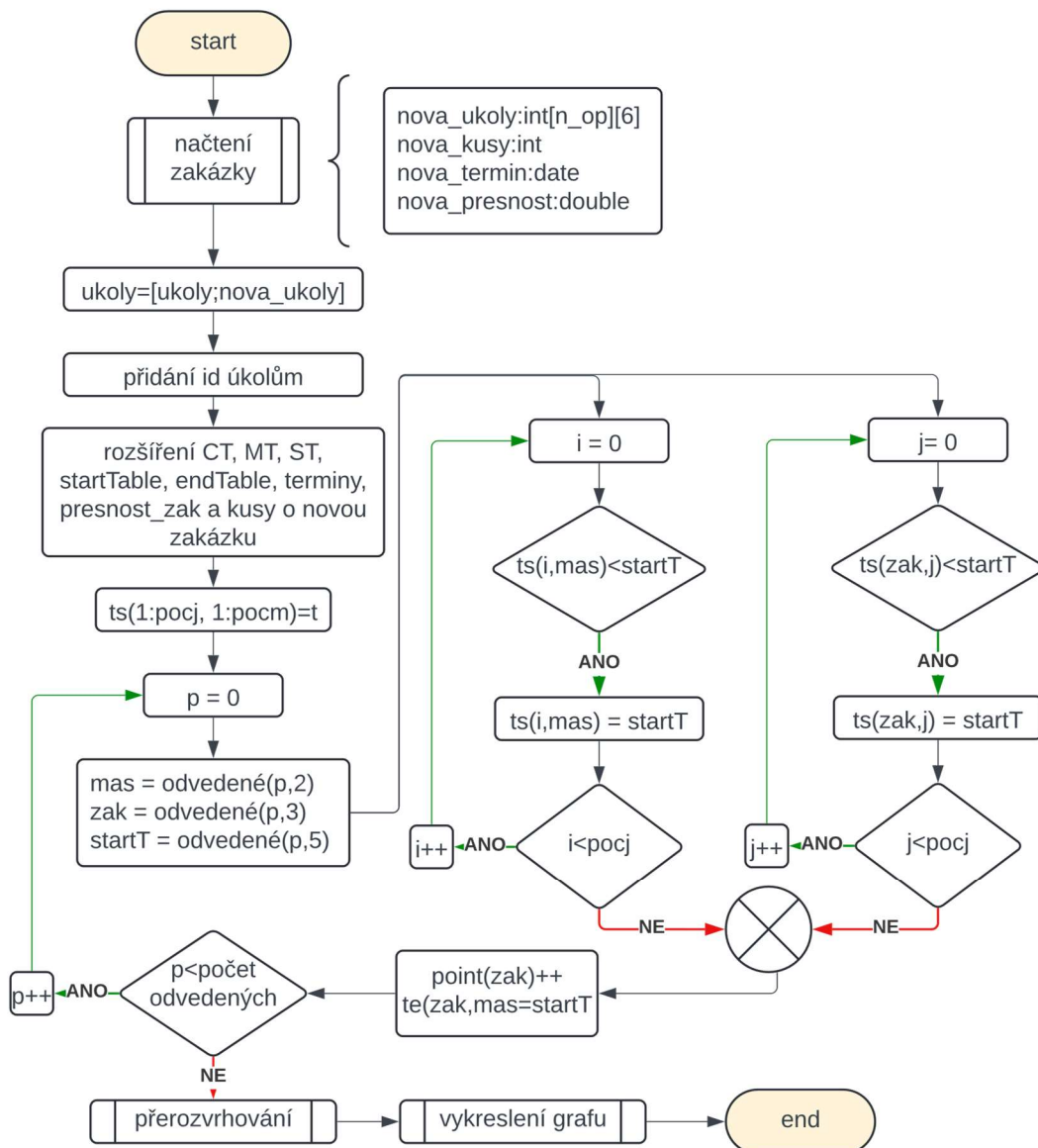
Obrázek 31 Diagram dynamického algoritmu

Po vytvoření seznamu úkolů statického rozvrhu a začátku výroby je nastaven počáteční čas algoritmu $t=0$, zároveň je vynulován počet odvedených zakázek výrobou odv a seznam úkolů je seřazen podle časů začátku jednotlivých úkolů (operací). Dynamické rozvrhování pak běží ve smyčce, dokud je v seznamu úkolů alespoň jeden úkol. Algoritmus, pokud v čase t nastává dynamická událost, přerovňuje výrobu podle zvoleného dynamického prioritního pravidla. Na konci každé t iterace jsou poté odvedeny operace, které

začaly být zpracovávány výrobou. Tyto operace jsou zavedeny do seznamu odvedených operací a vymazány ze seznamu úkolů.

6.2.1 Příklad nové zakázky

Algoritmus řešení nové zakázky (Obrázek 32) ukazuje, jak algoritmus dynamického rozvrhování řeší příchod nové zakázky s předem daným technologickým postupem, počtem kusů a termínem odvedení.

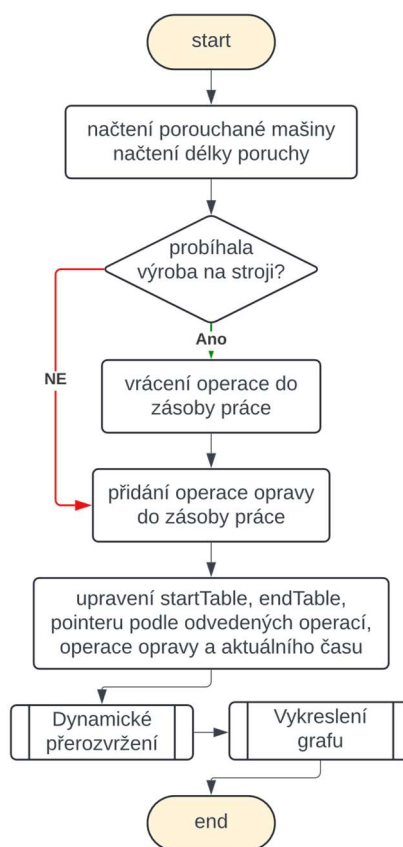


Obrázek 32 Algoritmus pro přerovrhování výrobního rozvrhu při příchodu nové zakázky V prvním kroku algoritmus načte data týkající se zakázky a vytvoří jim úkoly ve formě korespondujícím se seznamem úkolů, které poté přidá na začátek seznamu úkolů. Dále dochází k rozšíření CT, ST, MT, startTable, endTable, seznamu termínů a seznamu počtu

kusů o data nové zakázky. V dalším kroku je nastavena hodnota startTable v každé buňce na hodnotu času, kdy přišla tato nová zakázka. Poté se algoritmus dostává do cyklu, ve kterém jsou přenastaveny hodnoty v maticích startTable a endTable a vektor pointeru podle již odvedených zakázek. Následuje přerozvrhování rozvrhu, vykreslení grafu a dynamický algoritmus pokračuje.

6.2.2 Porucha výrobního stroje

Další dynamickou událostí, která může nastat pro tento algoritmus je porucha stroje. Algoritmus znázorněný na přiloženém schématu (Obrázek 33). K události poruchy stroje je zde přístupováno tak, že je v systému dostatek informací o poruchách strojů a jde tedy předvídat délka opravy tohoto výrobního stroje.



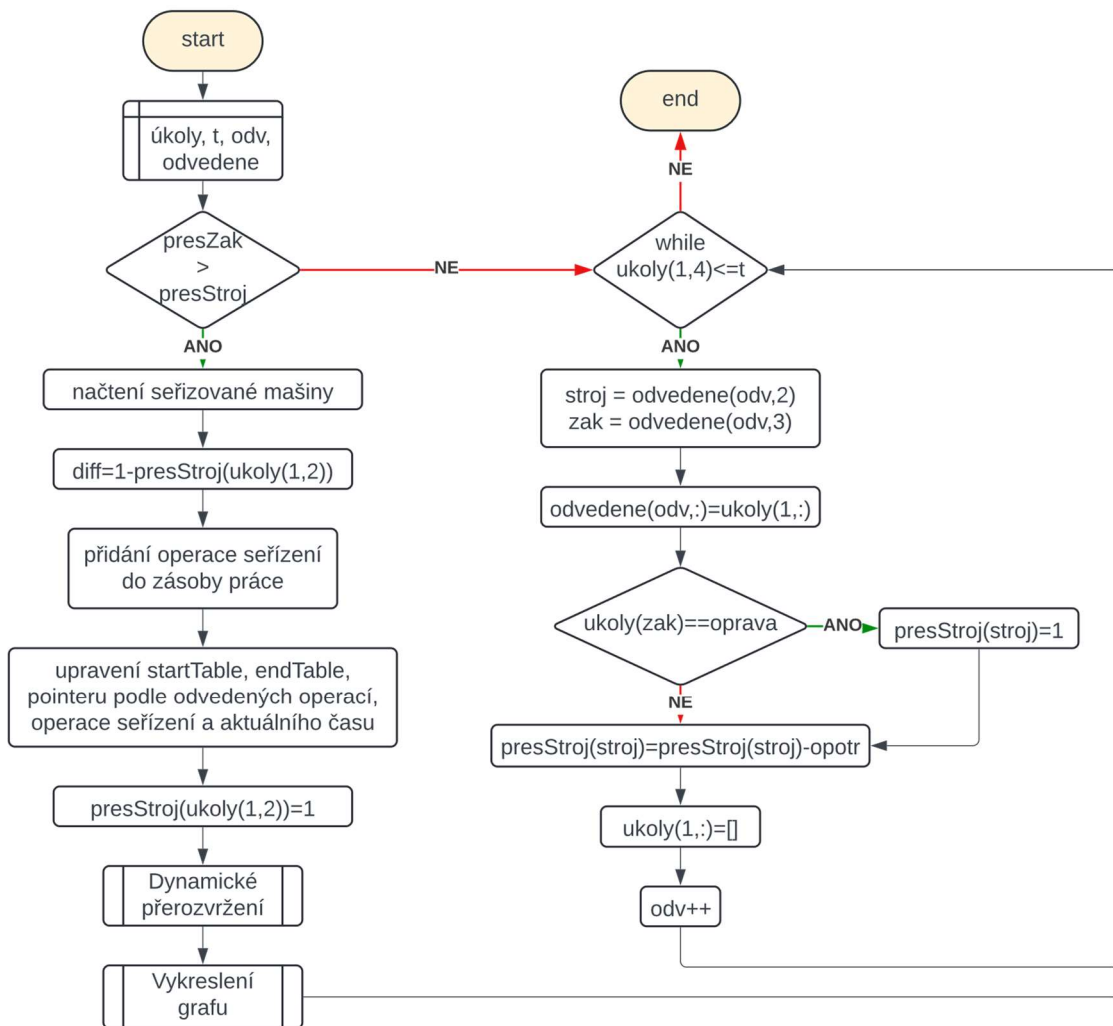
Obrázek 33 Algoritmus pro přerozvrhování výrobního rozvrhu při poruše stroje

Nejprve se v tomto algoritmu načtou data o předpokládané poruše (její délka) a poté algoritmus zjišťuje, zdali probíhá v čase poruchy na stroji výroba. Pokud výroba na tomto stroji probíhala, pak je operace zpracovávaná v čase poruchy vrácena z množiny odvedených operací do seznamu úkolů. Dále nastává obdobně jako u případu příchodu nové zakázky k úpravě maticí startTable a endTable a pointeru podle již odvedených zakázek,

aktuálního času a délky opravy stroje. V posledním kroku proběhne opět dynamické přerozvrhování a vykreslení grafu.

6.2.3 Dynamická změna výrobní přesnosti strojů

Tento algoritmus (Obrázek 34) vyžaduje změnu nejen v části dynamického algoritmu zabývající se dynamickými událostmi (vlevo), ale také změnu v části řešící odvádění operací (vpravo). Algoritmus je navržen na základě faktu, že výrobní stroje se opotřebovávají a ztrácí při výrobě svou přesnost.



Obrázek 34 Algoritmus pro přerozvrhování výrobního rozvrhu při změně přesnosti výrobních strojů (dynamická událost – vlevo, změna v odvádění zakázek – vpravo)

Tento algoritmus nejprve zkontroluje, zdali stroj, na kterém má být odvedena následující operace (presStroj) splňuje kritéria na přesnost výroby dané technickým postupem zakázky (presZak). V případě, že tato podmínka není splněna, dostává se algoritmus do své levé větve, ve které načte, který stroj má být seřizen. Dále je vypočten rozdíl mezi

ideálním stavem stroje a aktuálním stavem stroje (diff). Pro přidání operace seřizování do zásoby práce je použit obdobný algoritmus jako u případu s poruchou stroje. Neplatí zde však, že algoritmus kontroluje, zdali je na stroji zpracovávána operace, jelikož není připuštěno, že přesnost stroje klesne pod požadovanou hodnotu v průběhu výroby. Délka seřizování je přímo závislá na velikosti rozdílu mezi ideálním stavem stroje a aktuálním stavem stroje. Přerozvrhovacímu algoritmu jsou poté upraveny hodnoty matic startTable a endTable a pointeru podle již odvedených zakázek, aktuálního času a délky seřizování stroje. Následně je upravena hodnota přesnosti stroje na 100 %.

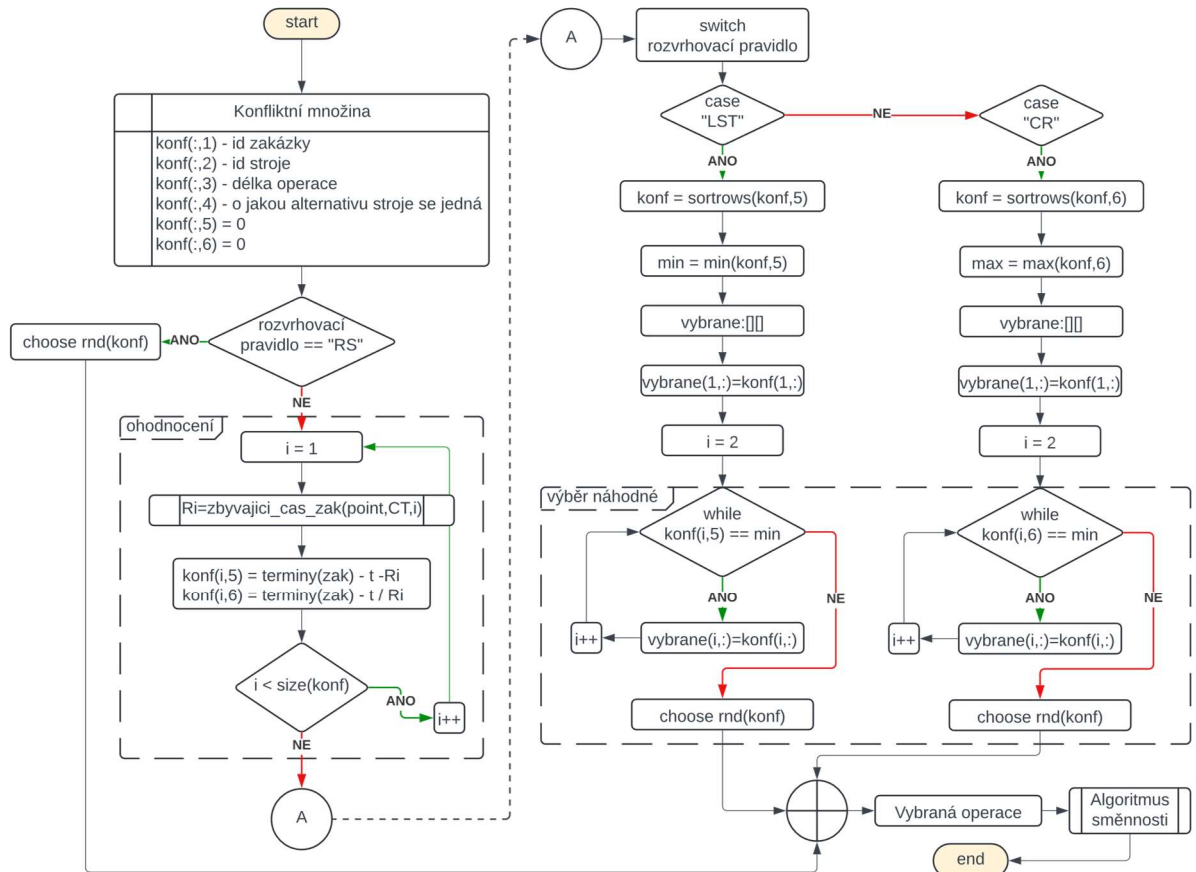
Pravá větev algoritmu na diagramu, která se zabývá změnou v odvádění zakázek, je upravena tak, že pokud je odveden úkol opravy stroje, tak se předpokládá, že po opravě bude stroj plně seřízen a bude tedy jeho výrobní přesnost 100%. Dále se zde při odvedení zakázky na daném stroji sníží přesnost stroje. V tomto algoritmu je přesnost snižována o pseudonáhodnou hodnotu (opotr), vybranou z rovnoměrného rovnoběžného rozložení, přímo úměrnou délce odvedené operace. Tento přístup byl zvolen s ohledem na potřeby firmy TOS VARNSDORF a.s., kde je snaha o výpočet hodnoty přesnosti stroje po každé dokončené operaci.

V praxi by poté přesnost stroje neměla měnit svou velikost o pseudonáhodnou hodnotu, ale měla být nahrazována z kontroly výsledných rozměrů odvedeného dílu. Ve firmě je také snaha o předvídaní hodnoty přesnosti výroby pro následující operace, toto by mohlo znamenat přechod finálního rozvrhovacího programu z reaktivní formy na prediktivně reaktivní. Změna algoritmu by nastala v tomto případě ve výše zmíněné pravé větvi algoritmu. Další otázkou je dále linearita času seřizování, která je přímo závislá na rozdílu od ideálního stavu a zdali v reálném prostředí bude stroj ztrácet také svou přesnost v závislosti na tom, jakou kvalitu momentálně má. Jinými slovy, zdali stroj ztratí svoji přesnost stejně rychle při změně z 80 % na 70 %, nebo z 90 % na 80 %. Pokud by stroj ztrácel svoji přesnost stále rychleji (změna z 80 % na 70 %, by proběhla rychleji než z 90 % na 80 %) bylo by dobré uvážit, zdali je vhodné seřizovat až do 100 % nebo bude vhodnější seřizovat stroj na nižší hodnotu přesnosti.

6.2.4 Dynamické přerozvržení pomocí prioritních pravidel

Algoritmus (Obrázek 35) zachycuje výběr operace k rozvržení z konfliktní množiny. Konfliktní množina vzniká stejně jako u statického rozvrhu algoritmem pro rozvrh bez

zpoždění (Obrázek 23). Algoritmus je obdobný jako pro statická prioritní pravidla (Obrázek 24). Prvním rozdílem je vstup aktuálního času do algoritmu a rozšíření dat v konfliktní množině, jelikož prioritní pravidla LST a CR, potřebují ke svému výpočtu více hodnot. Data pro rozhodnutí, která má být vybrána jsou zobrazena na schématu algoritmu.



Obrázek 35 Prioritní pravidla použitá pro dynamické rozvrhování

Algoritmus je oproti statickým pravidlům obohacen o funkci, která počítá zbývající výrobní čas zakázky dané operace. Výsledek (R_i) této funkce poté vstupuje do rovnic ohodnocení operací v konfliktních množinách. Schéma zachycuje tři různá prioritní pravidla LST, CR a Random select (RS). Pravidlo RS je zmíněno z důvodu, že s ním bude simulována práce mistra, kdy RS bude vytvářet rozvrhy bez zpoždění, které budou porovnávány s rozvrhy vytvořenými jinými prioritními pravidly. Použité pravidlo závisí na uvážení uživatele systému.

Levá část algoritmu ukazuje, jak je konfliktní množina ohodnocena. Pravá část algoritmu ukazuje výběr operace s nejlepším prioritním indexem pro jednotlivá prioritní pravidla. Znovu zde platí, že pokud má více operací nejlepší prioritní index je z nich vybrána pseudonáhodně podle rovnoběžného rovnoměrného rozložení jedna operace pro rozvrhnutí.

7 Testování algoritmů

7.1 Testování algoritmů na teoretických modelech

Testování algoritmů pro rozvrhování pomocí prioritních pravidel a GA probíhalo nejprve na teoretických modelech, kterým bylo již zjištěno nejnižší hodnoty celkové průběžné doby výroby (optima). Testována byla prioritní pravidla SPT a LPT a dále GA s ohodnocením váženého součtu. Váhy kritérií byly:

- celková průběžná doba výroby: 1
- průměrná odchylka od termínu zakázky: 0
- celkový počet seřízení: 0
- průměrné využití strojů: 0

Každý model byl otestován prioritním pravidlem 20 000krát, kdy délka každého experimentu probíhala přibližně 30 minut. Oproti tomu byl GA testován na každém modelu 10krát, kdy délka trvání algoritmu byla mezi 6 a 46 minut.

Výsledky jednotlivých algoritmů na teoretických modelech jsou znázorněny v následujících tabulkách (Tabulka 4 až Tabulka 6). V tabulkách jsou zeleně zvýrazněné buňky ukazující nejlepší řešení daného modelu v porovnání s ostatními rozvrhovacími metodami. Žlutě zvýrazněné buňky u GA ukazují na modely, jejichž průměrné řešení pomocí GA bylo lepší než minimální řešení zbylých dvou rozvrhovacích algoritmů. Červeně zvýrazněné buňky ukazují na modely, u kterých bylo nejhorší řešení GA lepší než minimální hodnoty řešení pomocí prioritních pravidel. Oranžově zvýrazněné buňky ukazují na modely, jimž byl daný algoritmus schopen nalézt optimální řešení.

Tabulka 4 Výsledky rozvrhů bez zpoždění vytvořené genetickým algoritmem

ND – GA(WSM)

Model	Opt.	Min.	Odchylka od optima	Průměr	směr. odch.	Odchylka od optima	Max.	Odchylka od optima
MK01	36	41	13.89 %	41,80	0.40	16.11 %	42	16.67 %
MK02	24	30	25.00 %	30,50	0.50	27.08 %	31	29.17 %
MK03	204	204	0.00 %	204,00	0.30	0.05 %	204	0.00 %
MK04	48	67	39.58 %	67,38	0.67	40.63 %	68	41.67 %
MK05	168	178	5.95 %	179,30	1.00	6.73 %	180	7.14 %
MK06	33	75	127.27 %	78,70	3.38	138.48 %	84	154.55 %
MK07	133	163	22.56 %	167,10	4.01	25.64 %	171	28.57 %
MK08	523	523	0.00 %	528,10	5.56	0.98 %	537	2.68 %
MK09	299	326	9.03 %	337,70	10.05	12.94 %	351	17.39 %
MK10	165	241	46,06 %	253,75	10.66	54.42 %	267	61.82 %
Průměrná odchylka			28,93 %			32,31 %		35,96 %

Z výsledků v tabulce Tabulka 4 si lze povšimnout, že GA našel optimální řešení pro dva z deseti teoretických modelů. Pro osm z deseti modelů vygeneroval GA lepší, nebo ekvivalentní řešení k řešením prioritních pravidel. Pro šest z deseti modelů bylo nejhorší řešení GA stále lepší než nejlepší řešení pomocí prioritních pravidel. U sedmi z deseti modelů vygeneroval GA lepší průměrné řešení než nejlepší řešení pomocí prioritních pravidel. Průměrné řešení GA pro každý z modelů, bylo také lepší oproti prioritním pravidlům.

Tabulka 5 Výsledky rozvrhů bez zpoždění vytvořené prioritním pravidlem SPT

ND – SPT

Model	Opt.	Min.	Odchylka od optima	Průměr	směr. odch.	Odchylka od optima	Max.	Odchylka od optima
MK01	36	44	22.22 %	55.17	5.60	53.25 %	70	94.44 %
MK02	24	35	45.83 %	45.95	5.11	91.45 %	57	137.50 %
MK03	204	248	21.57 %	284.98	18.69	39.70 %	326	59.80 %
MK04	48	65	43.75 %	79.06	5.20	64.71 %	92	91.67 %
MK05	168	196	16.67 %	219.79	11.49	30.83 %	247	47.02 %
MK06	33	72	118.18 %	80.19	4.00	142.99 %	90	172.73 %
MK07	133	187	40.60 %	213.81	13.66	60.76 %	247	85.71 %
MK08	523	523	0.00 %	614.53	32.32	17.50 %	680	30.02 %
MK09	299	381	27.42 %	437.46	24.24	46.31 %	491	64.21 %
MK10	165	295	78.79 %	344.27	22.38	108.65 %	397	140.61 %
Průměrná odchylka			41.87 %			65.62 %		92.37 %

Prioritní pravidlo SPT (Tabulka 5) bylo schopné nalézt při 20 000 iteracích optimální řešení pro jeden z teoretických modelů a dvě nejlepší řešení vůči zbylým metodám. Prioritní pravidlo LPT (Tabulka 6) našlo také jedno optimální řešení.

Tabulka 6 Výsledky rozvrhů bez zpoždění vytvořené prioritním pravidlem LPT

ND – LPT

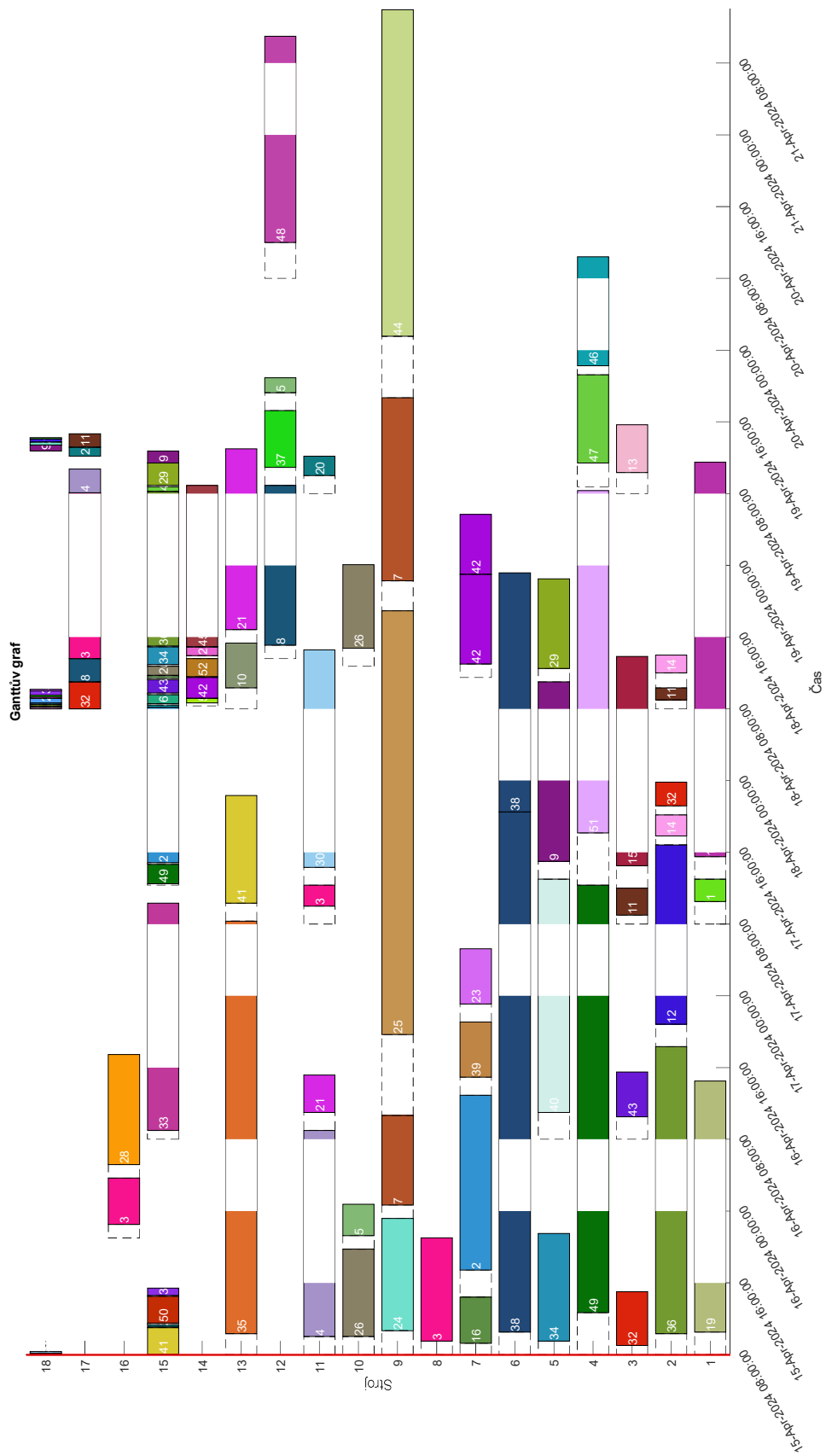
Model	Opt.	Min.	Odchylka od optima	Průměr	směr. odch.	Odchylka od optima	Max.	Odchylka od optima
MK01	36	56	55.56 %	67.71	5.46	88.08 %	87	141.67 %
MK02	24	45	87.50 %	63.04	5.61	162.68 %	79	229.17 %
MK03	204	225	10.29 %	292.45	27.81	43.36 %	368	80.39 %
MK04	48	76	58.33 %	101.63	8.51	111.73 %	132	175.00 %
MK05	168	190	13.10 %	216.14	8.98	28.66 %	252	50.00 %
MK06	33	104	215.15 %	126.39	8.03	283.01 %	153	363.64 %
MK07	133	237	78.20 %	282.86	15.50	112.68 %	326	145.11 %
MK08	523	523	0.00 %	592.57	29.90	13.30 %	676	29.25 %
MK09	299	372	24.41 %	469.91	31.64	57.16 %	549	83.61 %
MK10	165	334	102.42 %	434.80	35.79	163.51 %	528	220.00 %
Průměrná odchylka			63.39 %			106.42 %		151.78 %

Při porovnání průměrných odchylek od optima pro jednotlivé rozvrhovací algoritmy je zřejmé, že ačkoliv GA je schopen (za podobný časový úsek jako prioritní pravidla) vygenerovat pouze jedno finální řešení, tak je v průměru lepší než řešení prioritních pravidel. Průměrná odchylka od optima rozvrhu vytvořeným GA byla 32,13 %, kdežto u SPT byla průměrná odchylka 41,87 % a u LPT se jednalo o odchylku 63,39 % vůči optimálním řešením.

7.2 Testování na reálném modelu

Jako testovací modely pro GA a jako výchozí modely pro dynamické rozvrhování, byly použity dva výběry z databáze zásoby práce těžké mechaniky. První model (Obrázek 29), byl zkrácený model s 25 operacemi a druhý (Obrázek 36) byl týdenní model výroby na těžké mechanice, který obsahoval 97 operací. Jeho výsledné hodnoty kritérií ÚF byly:

- $C_{\max} = 126,62$ hodin
- $T = 14,36$ hodin
- Počet seřízení = 95
- Průměrné využití strojů = 43,95 %



Obrázek 36 Ganttův diagram vygenerovaný genetickým algoritmem

7.3 Stanovení vah vícekriteriálního hodnocení

Pro hodnocení rozvrhů generovaných GA bylo použito model váženého součtu, jelikož je to jednoduché řešení, které přiřadí každému rozvrhu hodnotu jakožto jedno číslo. Oproti tomu model váženého součtu není pro porovnávání tolika alternativ vzhledem k dostupnému výpočetnímu času tolik vhodný. Pro správné použití metody je nutné stanovit váhy jednotlivých kritérií. Stanovení vah jednotlivých kritérií pro GA bylo provedeno Saatyho maticí (Tabulka 7), kde byla určena významnost jednotlivých kritérií mezi sebou po poradě s vývojovým týmem:

Tabulka 7 výpočet vah porovnáním jednotlivých kritérií

Kritérium	C_{max}	T	Seřizování	U	G. průměr	Váha
C_{max}	1	0,33	3,00	1,00	1,00	0,205
T	3,00	1	4,00	3,00	2,45	0,503
Seřizování	0,33	0,25	1	0,33	0,41	0,084
U	1,00	0,33	3,00	1	1,00	0,206
Suma	5,33	1,92	11,00	5,33	4,86	

Jakožto nejvýznamnější kritérium hodnocení rozvrhu byla vyhodnocena průměrná odchylka od termínu zakázky (T), dále průběžná doba výroby (C_{max}) a využití strojů (U), jako nejméně podstatné bylo vyhodnoceno kritérium počet seřizování (Seřizování). Jak již bylo nastíněno v kapitole 4, pro model váženého součtu, je nutné hodnoty jednotlivých kritérií normalizovat. K tomuto účelu byla použita logaritmická normalizace (**Chyba! Nenalezen zdroj odkazů.**).

7.4 Testování dynamického rozvrhování

Pro testování dynamického rozvrhování byl vytvořen algoritmus, který vnášel do simulovaného výrobního prostředí dynamické změny. Tyto změny byly:

- příchod nové zakázky,
- oprava stroje,
- dynamická změna výrobní kvality strojů.

Jako algoritmy vhodné pro rychlou změnu rozvrhu byly testovány algoritmy obsahující dynamická prioritní pravidla LST (Least Slack Time) a CR (Critical ratio). Rozvrhy

vytvořené těmito prioritními pravidly byly rozvrhy bez zpoždění, jejichž algoritmus byl nastíněn v kapitole 6.2.4. Tvorba rozvrhu mistrem byla simulována jako tvorba rozvrhu bez zpoždění pomocí pravidla RS (Random Select), které z konfliktní množiny nejdřívějších možných startů vybírá operace náhodně.

Pro příchod nové zakázky byla do modelu zavedena nová zakázka, která pro testování všech prioritních pravidel přicházela v náhodný čas. Zakázka měla předem danou délku 4 operací na strojích 1 až 3 s předem určenou délkou.

V případě poruchy výrobního stroje bylo vycházeno z předpokladu, že jsou délky poruch výrobních strojů z historie známy nebo že lze odhadnout délku poruchy výrobního stroje. Tento stroj byl poté ponechán nedostupný po tuto dobu. Čas poruchy stroje byl volen opět pseudonáhodně v průběhu simulované výroby.

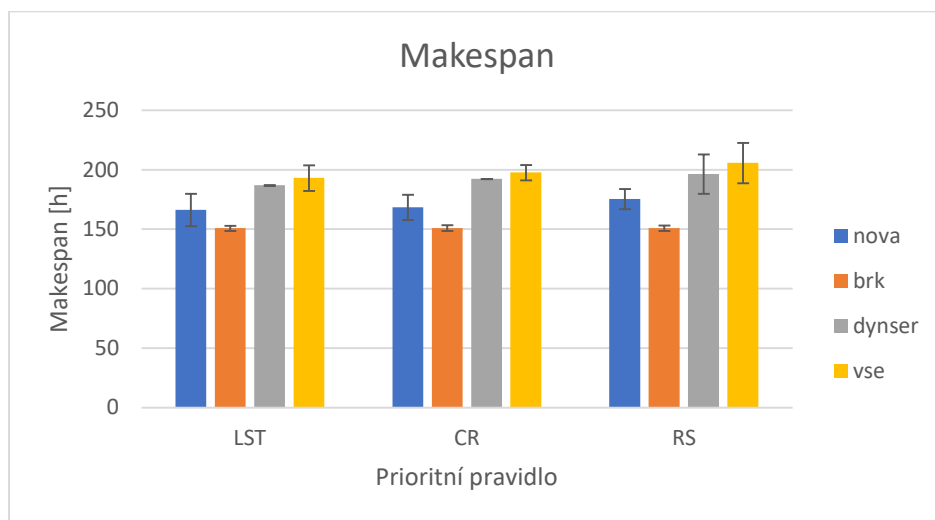
Při testování modelu, ve kterém probíhala dynamická změna kvality výrobních strojů, byla každé zakázce vygenerována minimální požadovaná kvalita výrobního stroje. Toto číslo bylo generováno v rozmezí 0,50 až 1,00. Jak již bylo zmíněno v sekci 5.4, budoucí systém vyhodnocuje stroj s hodnocením kvality 0,50 jako nevyhovující. Každému stroji byla taktéž vygenerována náhodná startovací kvalita v rozmezí 0,50 až 1,00. Pro všechny algoritmy byla vygenerována tato náhodná kvalita před jejich spuštěním. Po každé zpracované operaci byla stroji snížena výrobní kvalita o pseudonáhodnou hodnotu, která byla přímo závislá na délce trvání operace. Změna výrobní kvality stroje v průběhu operace se neuvažuje, jelikož v budoucím systému bude výrobní kvalita stroje vždy přepočítávána až po odvedení zakázky.

Posledním testovaným algoritmem byl algoritmus, který kombinoval všechny tři předchozí dynamické události (příchod nové zakázky, poruchu výrobního stroje a dynamickou změnu kvality výrobních strojů). Zde je dobré upozornit na předpoklad, se kterým se v modelech počítá, a to že když je stroj v poruše a je opravován, tak po konci opravy se zvedne jeho výrobní kvalita opět na 1,00 (100 %). Výsledky testování dynamického rozvrhování lze vidět na následujících obrázcích (Obrázek 37 až Obrázek 40).

Testování prioritních pravidel probíhalo po dobu 5 až 10 minut oproti tomu simulace náhodného výběru mistrem probíhala 20 až 30 minut. Kratší délka testování pro prioritní pravidla je zapříčiněna menší variabilitou výběru z konfliktní množiny oproti simulaci mistra, která je náročnější na výpočetní výkon. Dále stojí za uvážení, zda dokáže výběr z konfliktní množiny pomocí pravidla RS dosahovat stejných výsledků v případě, kdy

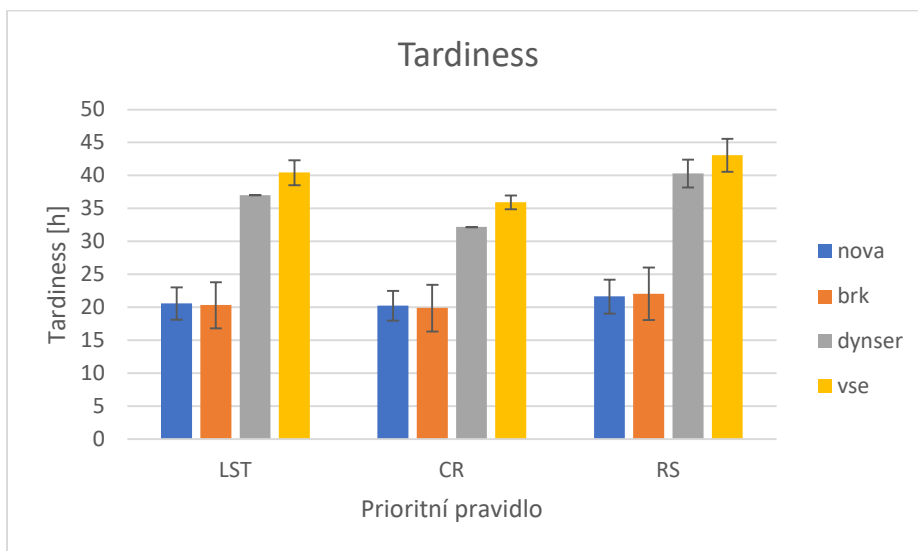
bude vyžadováno velmi rychlé řešení, tzv. „řešení na tlačítko“. Prioritní pravidla LST a CR dokáží vytvářet uspokojivé rozvrhy téměř okamžitě.

Při pohledu na graf (Obrázek 37) porovnání výsledků kritéria C_{max} lze pozorovat, že si prioritní pravidla LST a CR vedla lépe ve všech případech, kromě náhlé poruchy stroje oproti prioritnímu pravidlu RS. V případě náhlé poruchy stroje si všechna prioritní pravidla vedla téměř identicky. LST bylo mírně lepší oproti pravidlu CR.

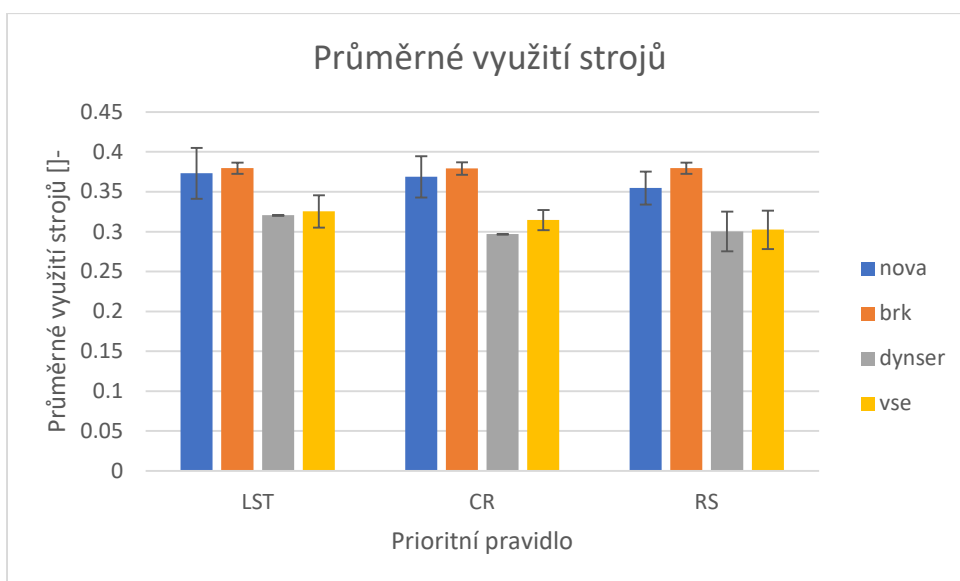


Obrázek 37 Výsledky dynamických testů pro kritérium průběžné doby výroby

Při pohledu na graf (Obrázek 38) porovnání výsledků kritéria T lze zřetelně pozorovat narůstající náročnost jednotlivých dynamických situací. Nejlepších výsledků dosahovalo prioritní pravidlo CR, které vykazovalo lepší hodnoty o 7 až 20 % oproti simulaci mistra pomocí náhodného přiřazování operací bez zpoždění. Pravidlo LST produkovalo pro jednotlivé případy o 5 až 10 % lepší rozvrhy v kritériu průměrné nedochvilnosti zakázek. Pravidla LST a CR podávala lepší výsledky než RS ve všech případech. Tento výsledek je zapříčiněný tím, že pravidla LST a CR jsou navržena právě z důvodu minimalizace průměrné odchylky od termínu a prioritizují operace, které náleží zakázkám, které mají nejmenší rezervu do data odvedení.

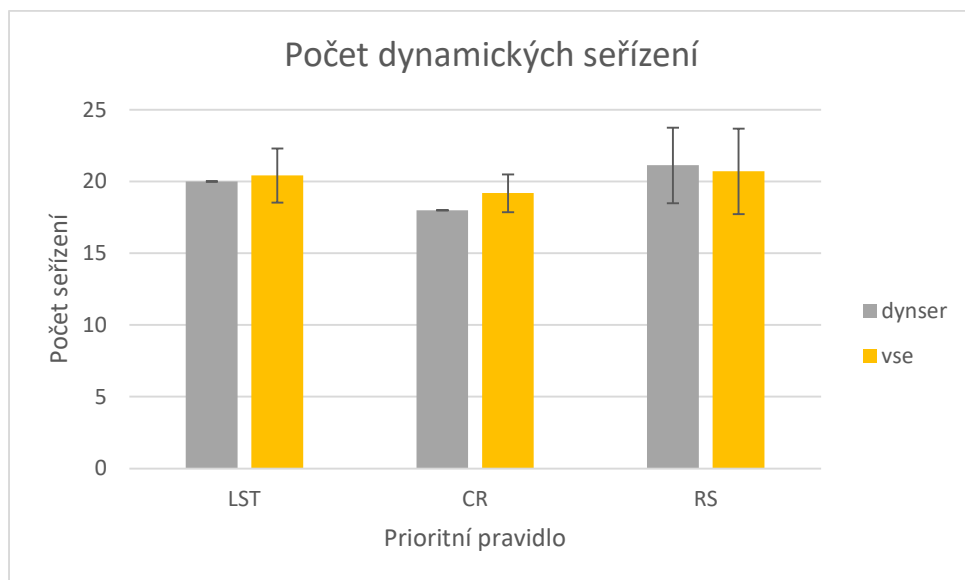


Obrázek 38 Výsledky dynamických testů pro kritérium průměrnou odchylku od termínu. Při pohledu na graf (Obrázek 39) porovnání výsledků kritéria U (průměrné využití strojů) lze znovu pozorovat vzrůstající náročnost jednotlivých dynamických situací. U tohoto grafu je však důležité dát pozor, jelikož se toto kritérium maximalizuje oproti zbylým funkcím, u kterých je snaha o jejich minimalizaci. Co se týče porovnání jednotlivých pravidel, tak pravidlo LST dosahovalo nejlepších výsledků pro případ nově přichozí zakázky, seřizování z důvodu dynamické změny výrobní kvality strojů a v případě experimentu se všemi dynamickými událostmi. V případě experimentu s poruchou výrobního stroje byl výsledek všech prioritních pravidel téměř identický. Pravidla LST a CR dosahovala pro všechny experimenty lepších nebo ekvivalentních výsledků oproti simulaci mistra pomocí pravidla RS.



Obrázek 39 Výsledky dynamických testů pro kritérium průměrné využití strojů

Posledním kritériem, které bylo hodnoceno, je počet dynamických seřízení ve výrobě (Obrázek 40), jelikož u případů nově příchozí zakázky a případu neočekávané poruchy stroje se dynamická změna kvality výrobních strojů neuvažuje. Lze si povšimnout, že prioritní pravidla LST a CR opět generovala kvalitnější rozvrhy oproti pravidlu RS. Nejméně dodatečných seřizování, se vykytovalo při použití pravidla CR.



Obrázek 40 Výsledky dynamických testů pro kritérium počet dynamických seřízení

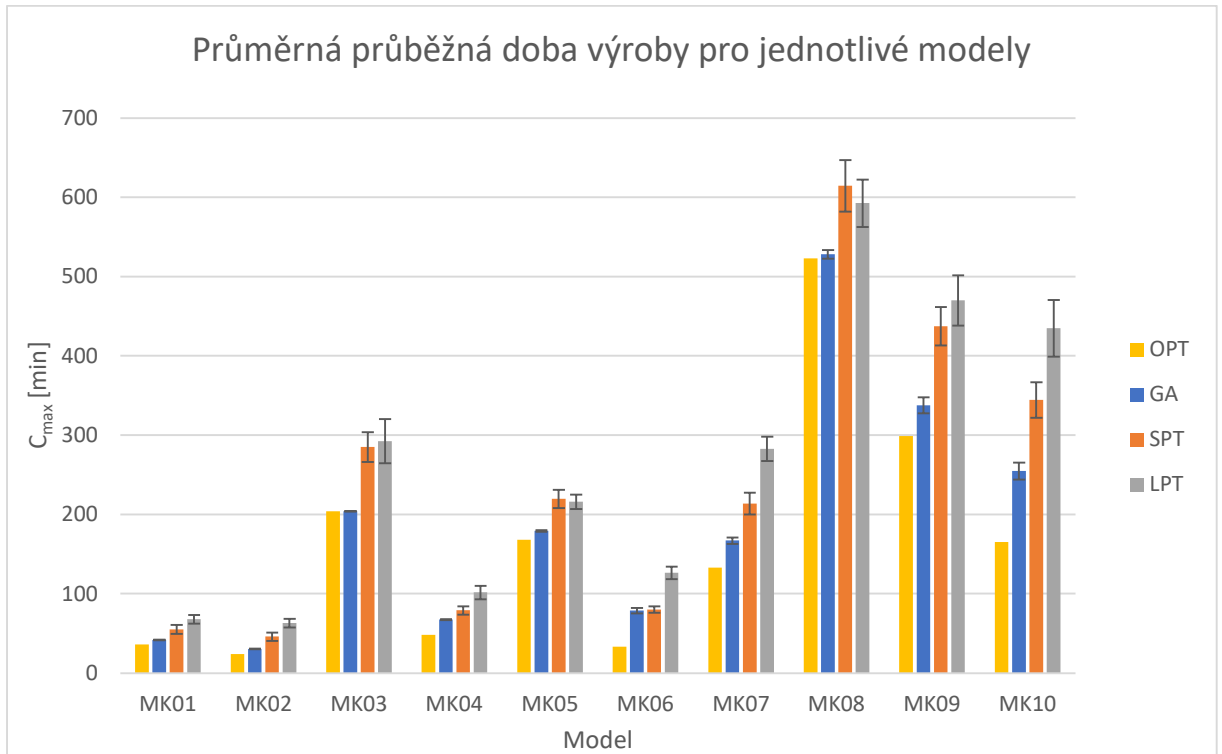
7.5 Vyhodnocení výsledků

7.5.1 Statický rozvrh

Při výběru algoritmu pro vytváření statického rozvrhu je bráno v potaz, že generování tohoto rozvrhu bude probíhat v době, kdy neprobíhá zrovna výroba, nebo probíhá jen na třísměnných strojích. Generování statického rozvrhu může probíhat tedy například o víkend, kdy standartně neprobíhá výroba, nebo přes noc, kdy výroba na jednosměnných a dvousměnných standartně neprobíhá. Generování statického rozvrhu může probíhat i delší časový úsek, například několik hodin.

Z výsledků v kapitole 7.1 je patrné, že nejlepší výsledky celkově generoval GA následovaný prioritním pravidlem SPT. Nejhorších výsledků dosahovalo prioritní pravidlo LPT. Lze také očekávat, že GA algoritmus vygeneruje za stejně velký delší časový úsek lepší řešení než řešení pomocí prioritních pravidel, co se týče celkové průběžné doby výroby. Co se týče kvality rozvrhu z hlediska kritéria průměrné odchylky od termínu zakázky, lze očekávat, že vytvoří GA také kvalitnější rozvrh, jelikož je u něj možné preferovat rozvrhy

s nižší odchylkou. Graf na obrázku (Obrázek 41) ukazuje srovnání průměrného výsledku jednotlivých rozvrhovacích algoritmů pro jednotlivé modely. Vzhledem k možnosti jeho přizpůsobení a vzhledem k tomu, že při delším časovém úseku generuje GA kvalitnější rozvrhy, byl GA vybrán jako metoda pro vytváření statických rozvrhů.



Obrázek 41 Graf srovnání průměrných rozvrhů vyprodukovaných jednotlivými algoritmy

7.5.2 Dynamický rozvrh

Z výsledků porovnání jednotlivých kritérií mezi sebou v kapitole 7.4 lze usoudit, že prioritní pravidlo CR (Critical Ratio) je nejvhodnější pro rozvrhy, ve kterých je důraz na plnění termínů, ačkoliv je to vykompenzováno horšími výsledky v kritériích průběžné doby výroby a s tím spjatým vytížením strojů. Oproti tomu prioritní pravidlo LST dosahovalo relativně dobrých výsledků pro jednodušší problémy, zejména pro kritérium průměrné využití výrobních strojů. Tyto prioritní pravidla byla testována proti náhodnému vybírání z konfliktní množiny, které díky své variabilitě dosahovalo relativně dobrých hodnot kritérií pro složitější problémy. Je však nutné podotknout, že pro rozsáhlejší modely je výpočetní čas tohoto pravidla RS (Random Select) výrazně delší oproti pravidlům LST a CR, které dokáží vytvořit uspokojivý rozvrh už po několika vteřinách.

Pro jednotlivé případy byly jednotlivým prioritním pravidlům vypočteny výsledné hodnoty účelové funkce. Kritéria byla logaritmicky normalizována a váhy jednotlivých

kritérií byly dosaženy využitím Saatyho matice. Hodnoty účelové funkce pak byly dosaženy metodou váženého součtu a jsou uvedeny v tabulce (Tabulka 8). Z výsledků v tabulce, kde jsou zeleně označeny nejlepší hodnoty pro jednotlivé experimenty a červeně nejhorší výsledky, je patrné, že nejlépe si vedlo prioritní pravidlo CR.

Tabulka 8 hodnoty ÚF pomocí váženého součinu normalizovaných kritérií

	LST	CR	RS
Nová zakázka	0,1968	0,1965	0,1989
Porucha stroje	0,1965	0,1960	0,1983
Dynamické seřizování	0,2075	0,2048	0,2092
Všechny události	0,2093	0,2070	0,2105

7.6 Implementace algoritmu

Výsledky dosažené v této práci naznačují vhodnost využití prioritního pravidla CR pro řešení problému dynamického rozvrhování. Algoritmus pro vytváření rozvrhů byl vytvořen ve dvou variantách, které řešily problematiku, kdy byla délka výroby dávky delší než délka směny výrobního stroje dvěma způsoby. První přístup byl rozdělení výrobních dávek na kusy, kdy se algoritmus snažil rozvrhnout výrobu co největšího počtu kusů výrobní dávky v počáteční směně a poté rozvrhoval zbylé kusy do následujících směn na daném stroji. V případě že byl kusový čas delší než délka směny stroje, rozvrhoval algoritmus operaci v celku, tudíž i do času, který nebyl vyhrazený danému stroji směnným kalendářem. V rozvrhu tudíž docházelo k vytváření přesčasů, při kratší průběžné době výroby a vyššímu vytížení strojů. Druhým přístupem bylo přerušování výroby kusu na konci směny a následné pokračování v další směně vyhrazené pro daný stroj, což pro výrobu složenou z obráběcích operací může být validní řešení.

Algoritmus také navazoval na vývoj systému pro predikci seřizování strojů vyvíjeným ve firmě TOS VARNSDORF a.s. ve spojení s RCMT ČVUT. V budoucnu je možná modifikace tohoto algoritmu z reaktivního rozvrhování na předvídavě reaktivní plánování, kdy na základě dat z měření jednotlivých dílů bude možné vytvořit prognózy, kdy stroj ztratí svou výrobní kvalitu. Dalším tématem, které rozvrhovací algoritmus řeší, je spojování výrobních dávek stejných dílů pro více zakázek a snaha o zpracování operací stejných dílů na stroji za sebou pro minimalizaci dávkového času. Co se týče spojování výrobních

dávek stejných dílů pro různé zakázky, tak tímto se algoritmus nezabýval, a to z důvodu nedostatku dat v systému. Dalšími daty, které je potřeba pro plný chod algoritmu získat, jsou data o alternativních strojích pro zpracování operace. Algoritmus je dostatečně robustní na to, aby se potýkal s problematikou flexibility systému, ale data v ERP pro tuto funkci momentálně chybí. Další data, která bude potřeba zavést do systému jsou data o poruchách strojů, kdy algoritmus počítá s určitou délkou odstávky stroje, která však nemusí být pro daný stroj a poruchu známá. Tato data by se mohla do systému dostat ve chvíli, kdy bude implementován MES systém, který bude také součástí TOScontrolu.

Při implementaci algoritmu je také nutné brát ohled na to, kdy vytvářet statický rozvrh. Z rešerše a výsledků experimentů lze doporučit vytvářet statický rozvrh vždy přes noc nebo přes víkend s daty, která mu budou předána dynamickým algoritmem. Další problematika, které je nutné se věnovat, je zásah mistra do rozvrhu, kdy mistr bude moci vyměnit pořadí operací. Toto je velmi komplexní problém z důvodu časové náročnosti pro vytvoření statického rozvrhu. Pro tuto problematiku lze doporučit dynamické přerozvrhování algoritmem v časovém úseku od konce přesunuté operace. Algoritmus by mohl být v budoucnu modifikován i pro případy zpomalení nebo zrychlení výroby, kdyby mohl prediktivně měnit délku operace v závislosti na historických datech a vytvářet tak robustnější rozvrhy.

8 Závěr

V této práci bylo představeno téma dynamického rozvrhování výroby a jeho přínos pro výrobní systémy. Teoretická část práce seznamuje čtenáře s rozvrhovacími modely, jejich notací, typy a třídami. Dále se tato část zabývá popisem dynamických událostí, které mohou nastat v reálném výrobním prostředí a strategiemi pro dynamické přeplánování. Následoval průzkum současného stavu techniky v oblasti dynamického rozvrhování, který obsáhl širokou škálu přístupů k dané problematice, a to od přesných metod řešení, přes přibližná heuristická řešení a bio-inspirované algoritmy až po řešení problematiky dynamického rozvrhování pomocí kvantových výpočtů. Součástí rešerše současného stavu techniky bylo také hledání odpovědi na otázku: „Kdy dynamicky rozvrh přeplánovat?“ a „Jak hodnotit výsledný rozvrh na základě více kritérií?“

V další kapitole byla představena firma TOS VARNSDORF a.s. a proběhla analýza současného způsobu rozvrhování ve firmě. Jednotlivým vybraným výrobním procesům byly vytvořeny BPMN modely. Pro vytvoření testovacího modelu byla následně vybrána týdenní zásoba práce na středisku „těžké mechaniky“. Testovací model obsahoval 56 zakázek s 1 až 4 operacemi na 18 strojích. Představen byl také modul TOScontrol, jehož součástí bude v budoucnu algoritmus vytvořený v této práci. Pro dynamické rozvrhování byla vybraná reaktivní metoda, která mění statický rozvrh výroby ve chvíli, kdy dojde k dynamické události. Pro vytvoření statického rozvrhu byl zvolen genetický algoritmus, který byl verifikován na teoretických flexibilních instancích MK01 až MK10 a jeho výsledky byly porovnány s optimálním řešením těchto instancí pro rozvrhy bez zpoždění. Rozvrhy generované GA byly pro pokračování do další generace ohodnoceny metodou váženého součtu. Pro dvě z deseti řešených instancí našel genetický algoritmus optimální řešení a celkově generoval rozvrhy 28,93 % od optimálního řešení. Pro porovnání byl GA opakovaně testován na těchto instancích vůči prioritním pravidlům SPT a LPT, kde dokázal vytvořit lepší řešení v osmi z deseti případů. Struktura algoritmů je podrobně popsána v práci v grafické a textové podobě. GA byl poté testován na datech výroby těžké mechaniky, kdy používal dostupné informace z ERP a dokázal vytvořit rozvrh s průběžnou dobou výroby jednoho pracovního týdne a minimální odchylkou od termínu.

Rozvrhy vygenerované GA poté sloužily jako vstupní data pro algoritmus dynamického přerovrhování, který simuloval výrobu na těžké mechanice s náhodným výskytem třech různých dynamických událostí. Dynamické události v experimentech zahrnovaly příchod nové zakázky, nenadálou poruchu výrobního stroje a dynamickou změnu výrobní kvality

stroje. Pro vytvoření dynamických rozvrhů bez zpoždění byla použita prioritní pravidla LST a CR, která byla porovnána se simulací práce mistra, která vytvářela rozvrhy bez zpoždění pomocí prioritního pravidla RS. Výsledky testování dynamického rozvrhování ukázaly, že prioritní pravidla LST a CR dokáží vytvářet kvalitnější rozvrhy než simulace mistra, a to zejména pro kritérium rozvrhu průměrného odchýlení od termínu zakázky, které bylo vyhodnoceno jako nejdůležitější kritérium rozvrhu, rozvrhy v tomto kritériu byly až o 20 % lepší než u CR. Struktura algoritmů je podrobně popsána v práci v grafické a textové podobě. Výsledný algoritmus také bral ohled na návaznost zakázek na výrobních strojích, kdy měnil délku seřizování vybraného stroje v závislosti na předešlé zakázce, která byla na stroji zpracovávána.

Práce ukázala možnost řešení dynamického rozvrhování na středisku těžké mechaniky ve firmě TOS VARNSDORF a.s. Jejím výsledkem je navržený algoritmus pro řešení dynamického rozvrhování pro tři dynamické události, které mohou nastat na tomto středisku. Dalším pokračováním práce v budoucnu by mohlo být řešení dalších dynamických událostí a modifikace reaktivního rozvrhování na předvídavě-reaktivní rozvrhování ve spojení se sběrem dat dynamické změny kvality výrobních strojů, které může tvořit robustnější rozvrhy, které nebude třeba měnit tak často. Implementace tohoto algoritmu závisí také na implementaci MES systému pro všechny výrobní stroje.

9 Použitá literatura

- [1] KOBLASA, František. *Uplatnění heuristických optimalizačních metod v oblasti rozvrhování strojírenské výroby středních a malých podniků* [online]. v Liberci, 2013. Technická Univerzita v Liberci. Dostupné z: <https://dspace.tul.cz/handle/15240/39274>
- [2] PINEDO, Michael L. *Scheduling: Theory, Algorithms, and Systems* [online]. Cham: Springer International Publishing, 2022 [vid. 2024-02-06]. ISBN 978-3-031-05920-9. Dostupné z: doi:10.1007/978-3-031-05921-6
- [3] LIAW, Ching-Fang. An efficient tabu search approach for the two-machine preemptive open shop scheduling problem. *Computers & Operations Research* [online]. 2003, **30**(14), 2081–2095. ISSN 03050548. Dostupné z: doi:10.1016/S0305-0548(02)00124-7
- [4] JAIN, A. K. a H.A. ELMARAGHY. Production scheduling/rescheduling in flexible manufacturing. *International Journal of Production Research* [online]. 1997, **35**(1), 281–309. ISSN 0020-7543, 1366-588X. Dostupné z: doi:10.1080/002075497196082
- [5] MICIETA, Branislav, Jolanta STASZEWSKA, Matej KOVALSKY, Martin KRAJCOVIC, Vladimira BINASOVA, Ladislav PAPANEK a Ivan ANTONIUK. Innovative System for Scheduling Production Using a Combination of Parametric Simulation Models. *Sustainability* [online]. 2021, **13**(17), 9518. ISSN 2071-1050. Dostupné z: doi:10.3390/su13179518
- [6] MOHAN, Jatoth, Krishnanand LANKA a A. Neelakanteswara RAO. A Review of Dynamic Job Shop Scheduling Techniques. *Procedia Manufacturing* [online]. 2019, **30**, 34–39. ISSN 23519789. Dostupné z: doi:10.1016/j.promfg.2019.02.006
- [7] *Stochastic vs Deterministic Models: Understand the Pros and Cons* [online]. [vid. 2024-05-21]. Dostupné z: <https://blog.ev.uk/stochastic-vs-deterministic-models-understand-the-pros-and-cons>
- [8] JACKSON, James R. Simulation research on job shop production. *Naval Research Logistics Quarterly* [online]. 1957, **4**(4), 287–295. ISSN 00281441, 19319193. Dostupné z: doi:10.1002/nav.3800040404

- [9] SURESH, V. a Dipak CHAUDHURI. Dynamic scheduling—a survey of research. *International Journal of Production Economics* [online]. 1993, **32**(1), 53–63. ISSN 09255273. Dostupné z: doi:10.1016/0925-5273(93)90007-8
- [10] SHEN, Xiaoning, Leandro L. MINKU, Rami BAHSOON a Xin YAO. Dynamic Software Project Scheduling through a Proactive-Rescheduling Method. *IEEE Transactions on Software Engineering* [online]. 2016, **42**(7), 658–686. ISSN 0098-5589, 1939-3520, 2326-3881. Dostupné z: doi:10.1109/TSE.2015.2512266
- [11] JACKSON, J. R. *Scheduling a production line to minimize maximum tardiness* [online]. B.m.: Office of Technical Services, 1955. Dostupné z: <https://books.google.cz/books?id=4jnPJgAACAAJ>
- [12] MINIMIZING THE MAKESPAN IN TWO-MACHINE JOB SHOP SCHEDULING PROBLEMS WITH NO MACHINE IDLE-TIME: In: *6th International Conference on Informatics in Control, Automation and Robotics: Proceedings of the 6th International Conference on Informatics in Control, Automation and Robotics* [online]. Milan, Italy: SciTePress - Science and and Technology Publications, 2009, s. 89–96 [vid. 2024-02-24]. ISBN 978-989-674-000-9. Dostupné z: doi:10.5220/0002213500890096
- [13] PANWALKAR, S. S. a Wafik ISKANDER. A Survey of Scheduling Rules. *Operations Research* [online]. 1977, **25**(1), 45–61. ISSN 0030-364X, 1526-5463. Dostupné z: doi:10.1287/opre.25.1.45
- [14] LU, C., J.A. STANKOVIC, G. TAO a S.H. SON. Design and evaluation of a feedback control EDF scheduling algorithm. In: *20th IEEE Real-Time Systems Symposium: Proceedings 20th IEEE Real-Time Systems Symposium (Cat. No.99CB37054)* [online]. Phoenix, AZ, USA: IEEE Comput. Soc, 1999, s. 56–67 [vid. 2024-02-24]. ISBN 978-0-7695-0475-9. Dostupné z: doi:10.1109/REAL.1999.818828
- [15] KANADE, Vijay. Genetic Algorithms - Meaning, Working, and Applications. *Spiceworks* [online]. 6. září 2023 [vid. 2024-02-26]. Dostupné z: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/>
- [16] Genetic Algorithms. *GeeksforGeeks* [online]. 29. červen 2017 [vid. 2024-02-27]. Dostupné z: <https://www.geeksforgeeks.org/genetic-algorithms/>
- [17] *pymoo - NSGA-II: Non-dominated Sorting Genetic Algorithm* [online]. [vid. 2024-05-21]. Dostupné z: <https://pymoo.org/algorithms/moo/nsga2.html>

- [18] *NSGA-II Optimization: Understand fast how it works [complete explanation]* [online]. 2018 [vid. 2024-05-21]. Dostupné z: https://www.youtube.com/watch?v=SL-u_7hIqjA
- [19] DEB, K., A. PRATAP, S. AGARWAL a T. MEYARIVAN. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* [online]. 2002, **6**(2), 182–197. ISSN 1089778X. Dostupné z: doi:10.1109/4235.996017
- [20] DAGLI, C. a R. HUGGAHALLI. A neural network architecture for faster dynamic scheduling in manufacturing systems. In: *1991 IEEE International Conference on Robotics and Automation: Proceedings. 1991 IEEE International Conference on Robotics and Automation* [online]. Sacramento, CA, USA: IEEE Comput. Soc. Press, 1991, s. 2408–2413 [vid. 2023-09-05]. ISBN 978-0-8186-2163-5. Dostupné z: doi:10.1109/ROBOT.1991.131983
- [21] PRIORE, Paolo, Alberto GÓMEZ, Raúl PINO a Rafael ROSILLO. Dynamic scheduling of manufacturing systems using machine learning: An updated review. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* [online]. 2014, **28**(1), 83–97. ISSN 0890-0604, 1469-1760. Dostupné z: doi:10.1017/S0890060413000516
- [22] HU, Yanhai, Feifan YE, Ziqing GUO a Zizhao ZHENG. Multi-shop Rescheduling Problem under Rush Orders. In: *2009 Fifth International Conference on Natural Computation: 2009 Fifth International Conference on Natural Computation* [online]. Tianjian, China: IEEE, 2009, s. 40–44 [vid. 2023-09-06]. ISBN 978-0-7695-3736-8. Dostupné z: doi:10.1109/ICNC.2009.330
- [23] FOX, Mark S. a Stephen F. SMITH. ISIS? a knowledge-based system for factory scheduling. *Expert Systems* [online]. 1984, **1**(1), 25–49. ISSN 0266-4720, 1468-0394. Dostupné z: doi:10.1111/j.1468-0394.1984.tb00424.x
- [24] KIRAN, Ali S. a Milton L. SMITH. Simulation studies in job shop scheduling—I a survey. *Computers & Industrial Engineering* [online]. 1984, **8**(2), 87–93. ISSN 03608352. Dostupné z: doi:10.1016/0360-8352(84)90001-9
- [25] BANERJEE, Abhishek, Dharmpal SINGH, Sudipta SAHANA a Ira NATH. Impacts of metaheuristic and swarm intelligence approach in optimization. In: *Cognitive Big*

Data Intelligence with a Metaheuristic Approach [online]. B.m.: Elsevier, 2022 [vid. 2024-01-31], s. 71–99. ISBN 978-0-323-85117-6. Dostupné z: doi:10.1016/B978-0-323-85117-6.00008-X

[26] LEE, Key K. Fuzzy rule generation for adaptive scheduling in a dynamic manufacturing environment. *Applied Soft Computing* [online]. 2008, **8**(4), 1295–1304. ISSN 15684946. Dostupné z: doi:10.1016/j.asoc.2007.11.005

[27] SCHWORM, Philipp, Xiangqian WU, Moritz GLATT a Jan C. AURICH. *Responsiveness to sudden disturbances in manufacturing through dynamic job shop scheduling using Quantum Annealing* [online]. preprint. B.m.: In Review. 2023 [vid. 2023-09-14]. Dostupné z: doi:10.21203/rs.3.rs-3070804/v1

[28] PAPROCKA, Iwona a Bożena SKOŁUD. A hybrid multi-objective immune algorithm for predictive and reactive scheduling. *Journal of Scheduling* [online]. 2017, **20**(2), 165–182. ISSN 1094-6136, 1099-1425. Dostupné z: doi:10.1007/s10951-016-0494-9

[29] WANG, Zhen, Jihui ZHANG a Shengxiang YANG. An improved particle swarm optimization algorithm for dynamic job shop scheduling problems with random job arrivals. *Swarm and Evolutionary Computation* [online]. 2019, **51**, 100594. ISSN 22106502. Dostupné z: doi:10.1016/j.swevo.2019.100594

[30] AMIN, M. Miftakul a Yevi DWITAYANTI. The Best Academic Administration Personnel Selection Model Using the Weighted Sum Model (WSM): In: *5th FIRST T1 T2 2021 International Conference (FIRST-T1-T2 2021)* [online]. 2022 [vid. 2024-02-27]. Dostupné z: doi:10.2991/ahe.k.220205.057

[31] TRIANTAPHYLLOU, Evangelos a Alfonso SÁNCHEZ. A Sensitivity Analysis Approach for Some Deterministic Multi-Criteria Decision-Making Methods*. *Decision Sciences* [online]. 1997, **28**(1), 151–194. ISSN 0011-7315, 1540-5915. Dostupné z: doi:10.1111/j.1540-5915.1997.tb01306.x

[32] VAFAEI, Nazanin, Rita A. RIBEIRO a Luis M. CAMARINHA-MATOS. Selection of Normalization Technique for Weighted Average Multi-criteria Decision Making. In: Luis M. CAMARINHA-MATOS, Kankam O. ADU-KANKAM a Mohammad JULASHOKRI, ed. *Technological Innovation for Resilient Systems* [online]. Cham: Springer International Publishing, 2018 [vid. 2024-03-02], IFIP Advances in Information

and Communication Technology, s. 43–52. ISBN 978-3-319-78573-8. Dostupné z: doi:10.1007/978-3-319-78574-5_4

[33] S.R.O, OptimWeb. *O nás | TOS VARNSDORF a. s.* [online]. [vid. 2024-03-31]. Dostupné z: <https://www.tosvarnsdorf.cz/cs/o-nas>

[34] RCMT. *Systém správy stroje*. Interní dokument. V-19–060. B.m.: TOS VARNSDORF a.s. 2019. *CK-SVT-WP09*.

[35] BRANDIMARTE, Paolo. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research* [online]. 1993, **41**(3), 157–183. ISSN 1572-9338. Dostupné z: doi:10.1007/BF02023073

10 Seznam obrázků

Obrázek 1 Otevřený rozvrhovací problém [1].....	15
Obrázek 2 Proudový rozvrhovací problém [1]	16
Obrázek 3 Flexibilní sekvenční rozvrhovací problém[1]	17
Obrázek 4 Třídy rozvrhovacích modelů [1]	19
Obrázek 5 Názvosloví používané v genetických algoritmech[16]	28
Obrázek 6 Příklad křížení chromozomů	29
Obrázek 7 Příklad mutace chromozomů.....	29
Obrázek 8 Barevné rozlišení jedinců do jednotlivých front[18].....	30
Obrázek 9 Výběr další generace jedinců[17].....	31
Obrázek 10 Vzdálenost řešení [19].....	32
Obrázek 11 Algoritmus strojového učení pro výběr prioritních pravidel [21].....	33
Obrázek 12 Architektura algoritmu pro dynamické rozvrhování v práci Elmaraghyho a Jaina[4].....	36
Obrázek 13 Mapa dceřiných společností korporátu TOS Varnsdorf a.s. [33].....	42
Obrázek 14 Základní rozhraní TOScontrol [34].....	47
Obrázek 15 Aplikace Status[34]	49
Obrázek 16 Aplikace Kalendář[34]	49
Obrázek 17 Aplikace Dokumentace[34].....	50
Obrázek 18 Aplikace IP kamera[34]	50
Obrázek 19 Aplikace Manažer zakázek[34].....	51
Obrázek 20 Stávající forma náhledu operace v Manažeru zakázek[34].....	51
Obrázek 21 Struktura statického rozvrhování	57
Obrázek 22 Struktura rozvrhovacího algoritmu	58
Obrázek 23 Modifikovaný algoritmus Gifflera a Thompsona pro vytvoření rozvrhu bez zpoždění	60
Obrázek 24 Výběr z konfliktní množiny pomocí prioritních pravidel SPT a LPT.....	61
Obrázek 25 Struktura genetického algoritmu	63
Obrázek 26 Opravný algoritmus genů	64
Obrázek 27 Algoritmus vytvoření a ohodnocení jedince populace	65
Obrázek 28 Algoritmus řešení směnnosti.....	66
Obrázek 29 Příklad Ganttova diagramu vytvořený algoritmem.....	67
Obrázek 30 Graf zachycující nejlepšího jedince (červeně) a průměr populace (modře) 67	

Obrázek 31 Diagram dynamického algoritmu.....	68
Obrázek 32 Algoritmus pro přerozvrhování výrobního rozvrhu při příchodu nové zakázky	69
Obrázek 33 Algoritmus pro přerozvrhování výrobního rozvrhu při poruše stroje.....	70
Obrázek 34 Algoritmus pro přerozvrhování výrobního rozvrhu při změně přesnosti výrobních strojů (dynamická událost – vlevo, změna v odvádění zakázek – vpravo) ...	71
Obrázek 35 Prioritní pravidla použitá pro dynamické rozvrhování	73
Obrázek 36 Ganttův diagram vygenerovaný genetickým algoritmem.....	78
Obrázek 37 Výsledky dynamických testů pro kritérium průběžné doby výroby	81
Obrázek 38 Výsledky dynamických testů pro kritérium průměrnou odchylku od termínu	82
Obrázek 39 Výsledky dynamických testů pro kritérium průměrné využití strojů.....	82
Obrázek 40 Výsledky dynamických testů pro kritérium počet dynamických seřízení...	83
Obrázek 41 Graf srovnání průměrných rozvrhů vyprodukovaných jednotlivými algoritmy	84

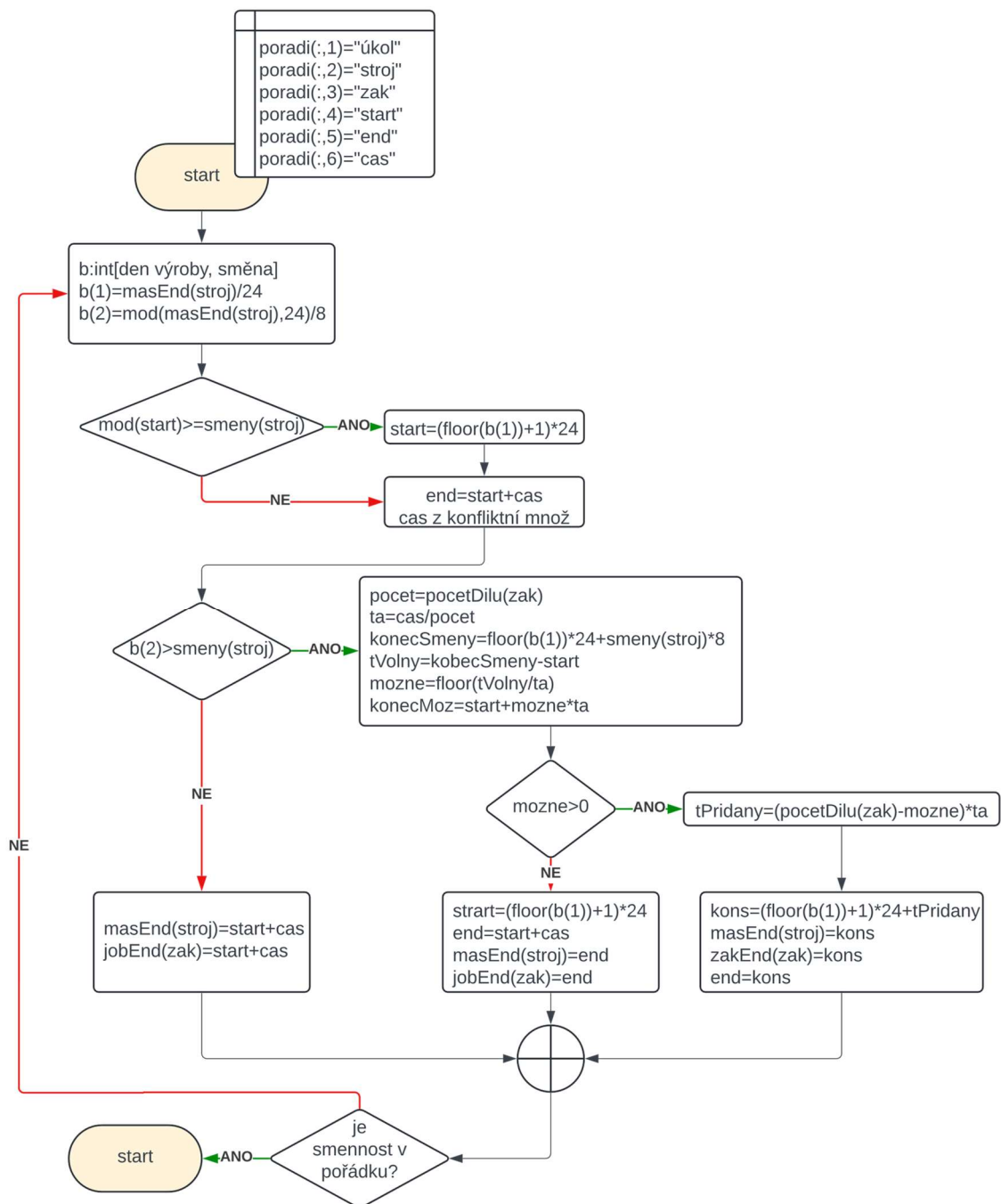
11 Seznam tabulek

Tabulka 1 Přehled prioritních pravidel	27
Tabulka 2 Přehled modelů MK01 – MK10	53
Tabulka 3 Směnnost strojů na středisku těžké mechaniky	55
Tabulka 4 Výsledky rozvrhů bez zpoždění vytvořené genetickým algoritmem	75
Tabulka 5 Výsledky rozvrhů bez zpoždění vytvořené prioritním pravidlem SPT	76
Tabulka 6 Výsledky rozvrhů bez zpoždění vytvořené prioritním pravidlem LPT	76
Tabulka 7 výpočet vah porovnáním jednotlivých kritérií.....	79
Tabulka 8 hodnoty ÚF pomocí váženého součinu normalizovaných kritérií.....	85

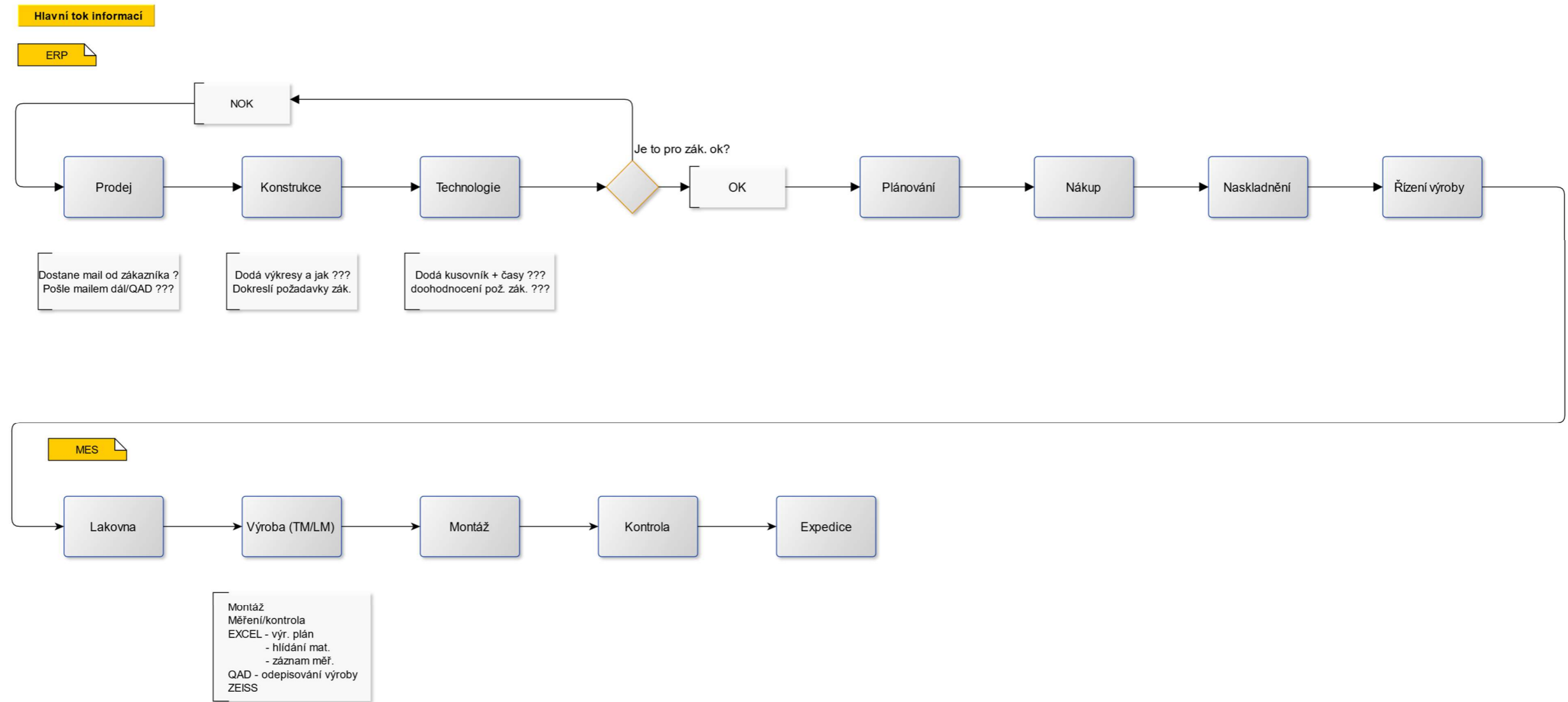
12 Seznam příloh

Příloha 1/ 1 Algoritmus řešení směnnosti dělbou výrobní dávky „po kusech“	98
Příloha 1/ 2 Hlavní tok informací	99
Příloha 1/ 3 Tok informací pro zakázku opravy obráběcí hlavy.....	100
Příloha 1/ 4 Tok informací pro zakázku výroby obráběcí hlavy	101
Příloha 1/ 5 Tok informací na středisku těžké mechaniky.....	102
Příloha 1/ 6 Příklad optimalizace toku informací	103

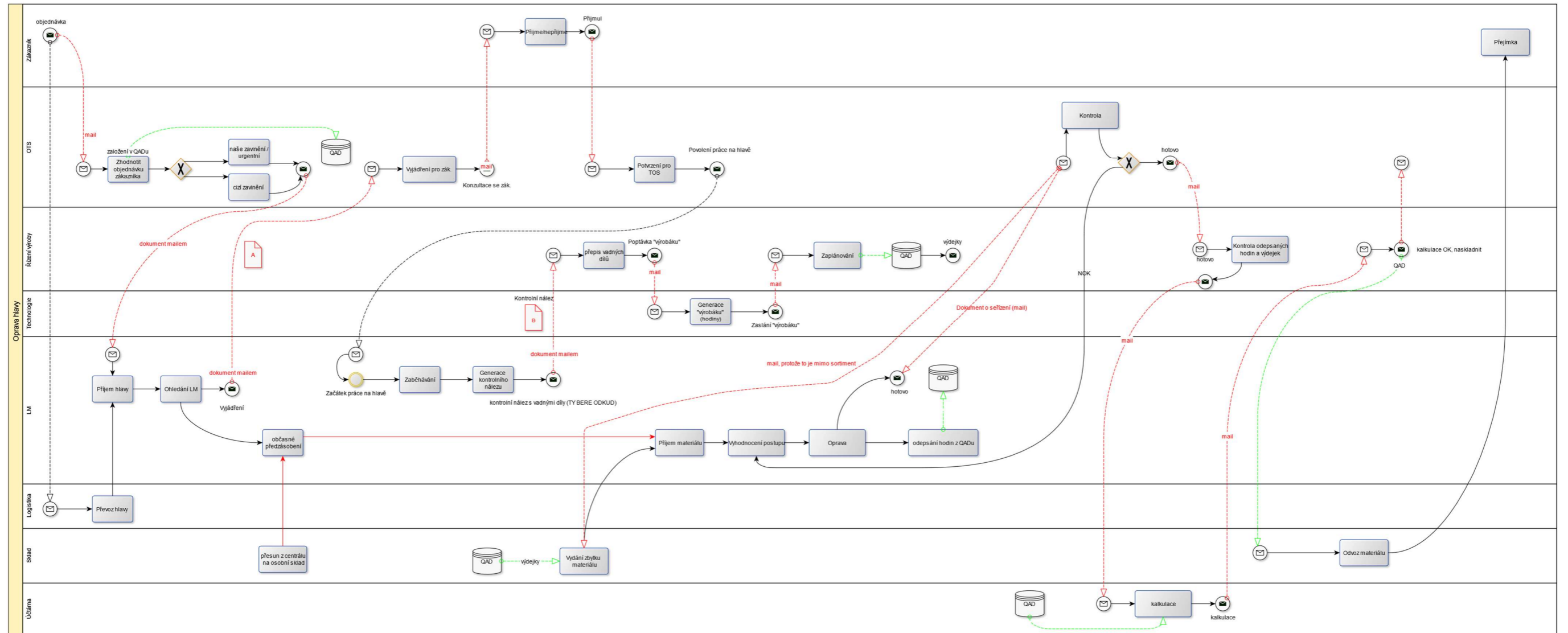
Příloha 1/ 1 Algoritmus řešení směnnosti dělbou výrobní dávky „po kusech“



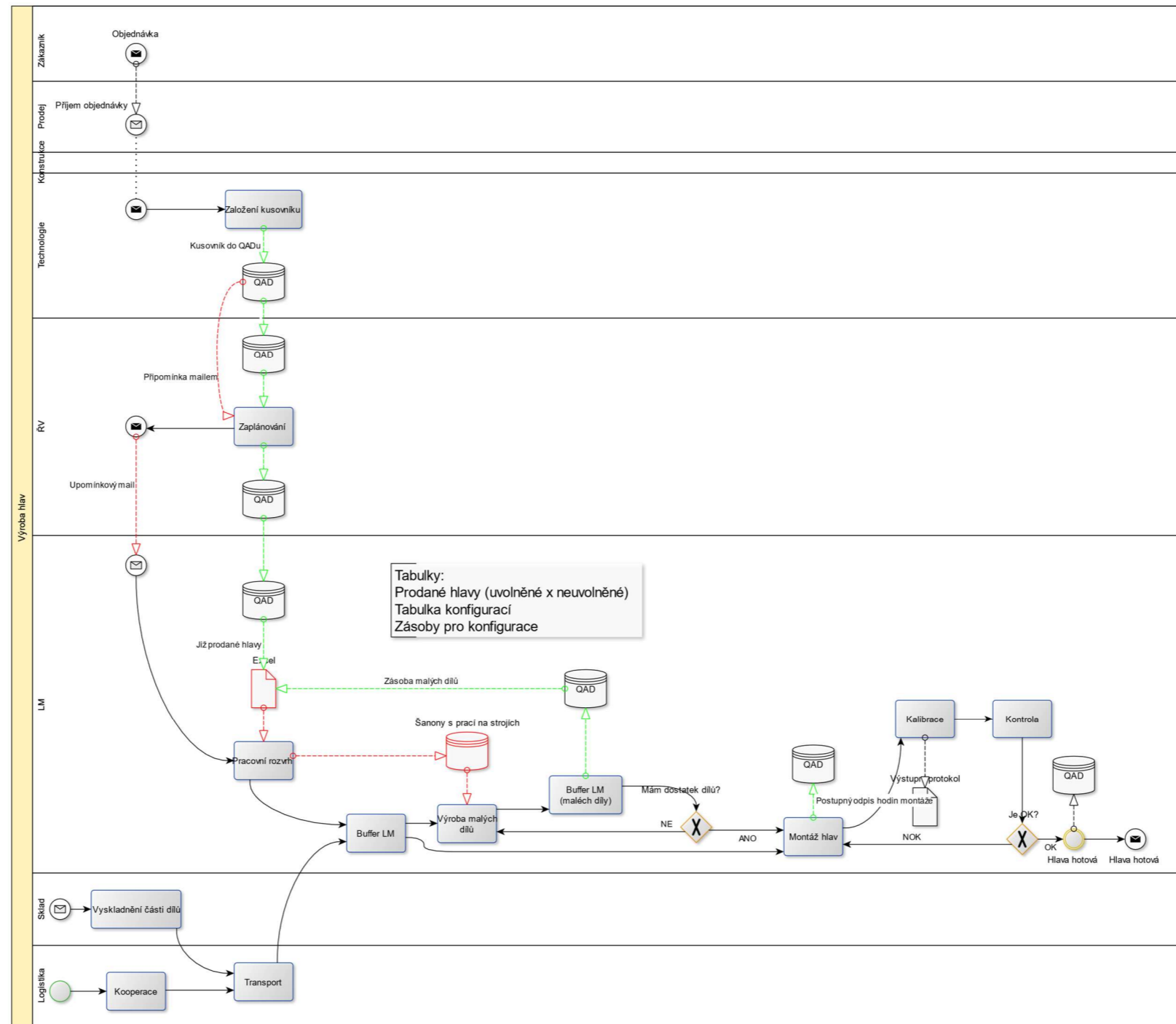
Příloha 1/ 2 Hlavní tok informací



Příloha 1/ 3 Tok informací pro zakázku opravy obráběcí hlavy

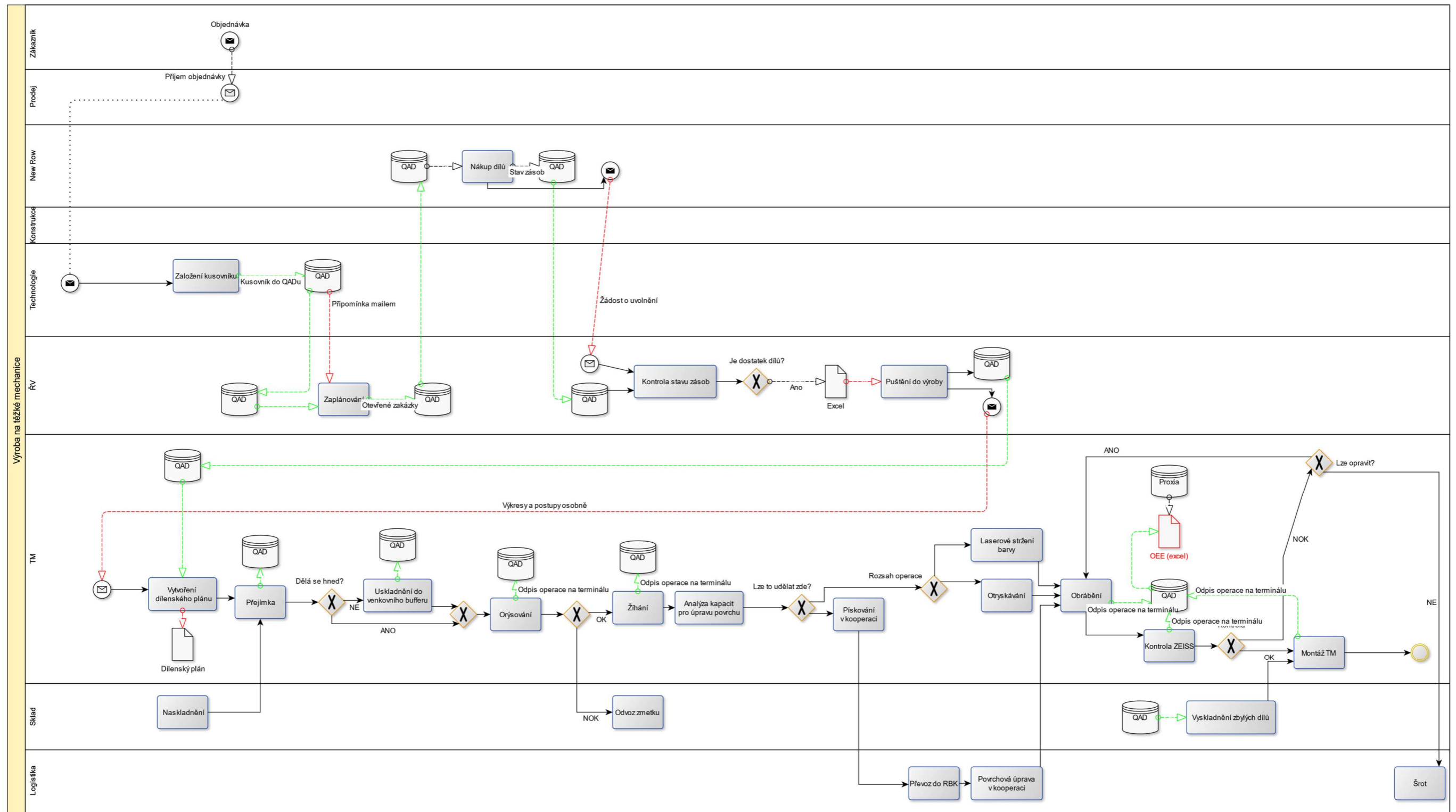


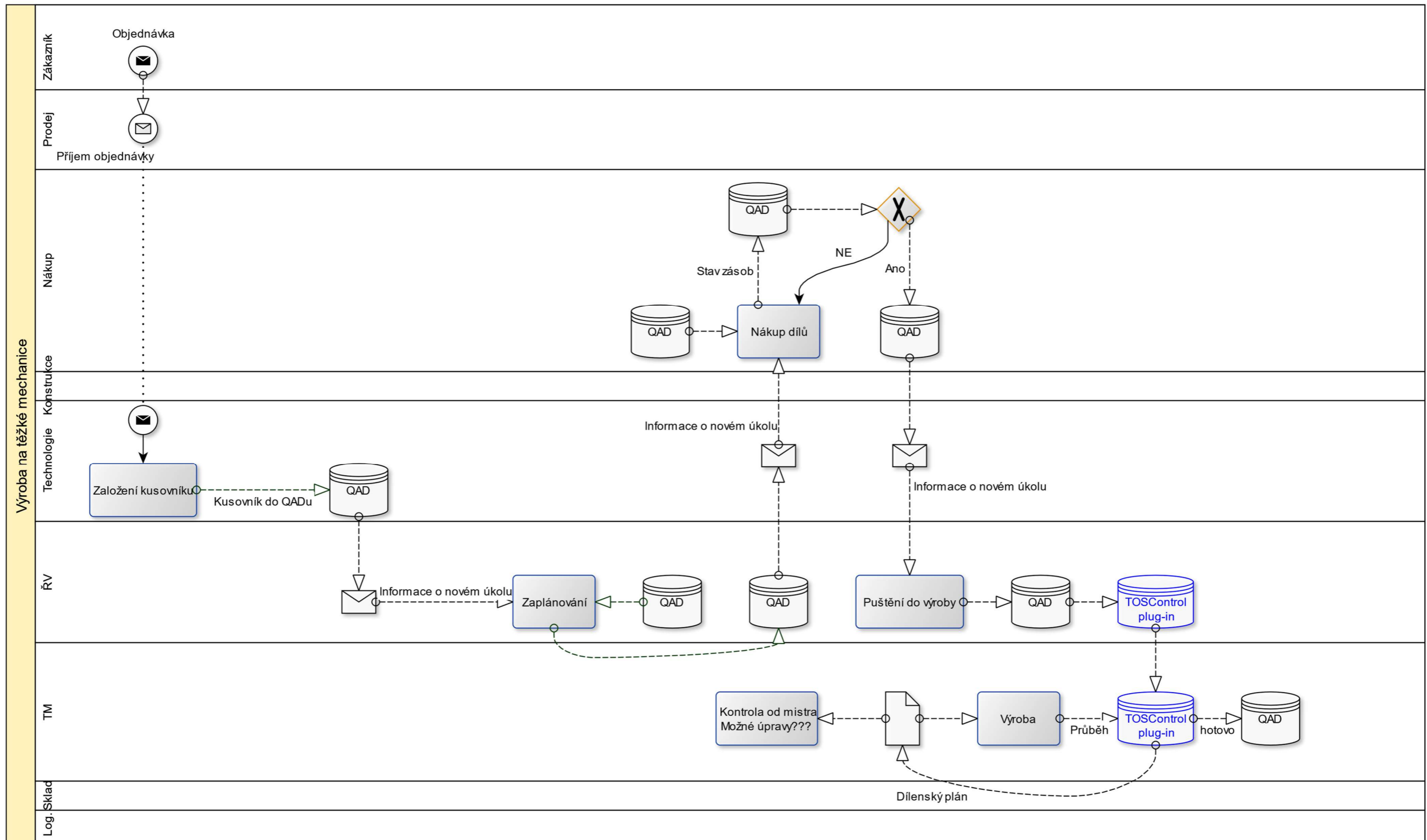
Příloha 1/ 4 Tok informací pro zakázku výroby obráběcí hlavy



- Odepisování vždy zpětně po dokončení části postupu
- Mistr vytváří svůj vlastní excel, kde vidí co bude v budoucnu potřebovat za díly na montáž
- Podle něj si vytváří předzásobu malých dílů, které zůstávají na skladě LM

Příloha 1/ 5 Tok informací na středisku těžké mechaniky





- Momentální upomínky skrze QAD nejsou využívány