

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: N 2301 Strojní inženýrství
Studijní zaměření: Průmyslové inženýrství a management

DIPLOMOVÁ PRÁCE

Plánování výroby s využitím systémů APS
(Advanced Planning and Scheduling)

Autor: **Bc. Jan Průcha**

Vedoucí práce: **Ing. Petr Hořejší, Ph.D.**

Akademický rok 2018/2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan PRŮCHA**

Osobní číslo: **S17N0082P**

Studijní program: **N2301 Strojní inženýrství**

Studijní obor: **Průmyslové inženýrství a management**

Název tématu: **Plánování výroby s využitím systémů APS (Advanced Planning and Scheduling)**

Zadávací katedra: **Katedra průmyslového inženýrství a managementu**

Z á s a d y p r o v y p r a c o v á n í :

1. Definice APS systémů
2. Pokročilé plánování a rozvrhování - využití, možnosti
3. Popis vybraného algoritmu
4. Realizace databázového systému s daty pro plánování
5. Realizace vybraného algoritmu

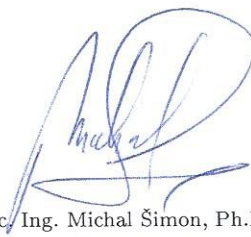
Rozsah grafických prací: 0 výkresů
Rozsah kvalifikační práce: 50 - 70 stran
Forma zpracování diplomové práce: tištěná
Seznam odborné literatury:

1. MAUERGAUZ Y. *Advanced Planning and Scheduling in Manufacturing and Supply Chains*. Springer, 2016. ISBN 978-3319275215
2. EL-BERISHY N. *An Advanced Production Planning and Scheduling System: for Batch Process Industry*. LAP LAMBERT Academic Publishing, 2011. ISBN 978-3846558058
3. BASL, J., VELKOBORSKÝ J. *Přehled českého trhu softwarových nástrojů APS a CRM*. Computerworld 32/2000.

Vedoucí diplomové práce: Ing. Petr Hořejší, Ph.D.
Katedra průmyslového inženýrství a managementu
Konzultant diplomové práce: Doc. Ing. Pavel Kopeček, CSc.
Katedra průmyslového inženýrství a managementu
Datum zadání diplomové práce: 24. září 2018
Termín odevzdání diplomové práce: 24. května 2019



Doc. Ing. Milan Edl, Ph.D.
děkan



Doc. Ing. Michal Šimon, Ph.D.
vedoucí katedry

V Plzni dne 24. září 2018

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této diplomové práce.

V Plzni dne:

.....

podpis autora

ANOTAČNÍ LIST DIPLOMOVÉ PRÁCE

AUTOR	Příjmení Bc. Průcha	Jméno Jan	
STUDIJNÍ OBOR	Průmyslové inženýrství a management		
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Ing. Hořejší, Ph.D.	Jméno Petr	
PRACOVIŠTĚ	ZČU - FST - KPV		
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ	Nehodící se škrtněte
NÁZEV PRÁCE	Plánování výroby s využitím systémů APS (Advanced Planning and Scheduling)		

FAKULTA	strojní	KATEDRA	KPV	ROK ODEVZD.	2019
----------------	---------	----------------	-----	------------------------	------

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	68	TEXTOVÁ ČÁST	56	GRAFICKÁ ČÁST	12
---------------	-----------	---------------------	-----------	--------------------------	-----------

STRUČNÝ POPIS (MAX 10 ŘÁDEK) ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY	Cílem této práce je seznámením s APS systémy a nahlédnutí do současného stavu používání APS systémů. Dalšími cíli jsou analýza vybrané metody pro APS, návržení vlastního algoritmu ze skupiny APS fungujícího na datech z vlastní databáze.
KLÍČOVÁ SLOVA ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE	APS, výrobní plán, plánování výroby, rozvrhování výroby, aplikace

SUMMARY OF DIPLOMA (BACHELOR) SHEET

AUTHOR	Surname Bc. Průcha	Name Jan	
FIELD OF STUDY	Industrial Engineering and Management		
SUPERVISOR	Surname (Inclusive of Degrees) Ing. Hořejší, Ph.D.	Name Petr	
INSTITUTION	ZČU - FST - KPV		
TYPE OF WORK	DIPLOMA	BACHELOR	Delete when not applicable
TITLE OF THE WORK	Manufacturing Planning Using APS (Advanced Planning and Scheduling) systems		

FACULTY	Mechanical Engineering	DEPARTMENT	Industrial Engineering and Management	SUBMITTED IN	2019
----------------	------------------------	-------------------	---------------------------------------	---------------------	------

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	68	TEXT PART	56	GRAPHICAL PART	12
----------------	----	------------------	----	-----------------------	----

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	Aim of this thesis is introduction of APS systems. Other aims are the analysis of the chosen method for APS and design an own algorithm from APS operating on data from own database.
KEY WORDS	APS, production management, planning, capacities, algorithms, application

Poděkování

Na tomto místě bych rád poděkoval Ing. Petru Hořejšímu, Ph.D. a Doc. Ing. Pavlovi Kopečkovi, CSc. za cenné připomínky a odborné rady při zpracovávání diplomové práce.

Obsah

1	Úvod.....	11
1.1	Problematika definice APS.....	11
1.1.1	Tři směry definování APS.....	12
1.1.2	Definice klíčových slov.....	15
1.2	Vznik APS.....	16
2	Počátky APS systémů.....	17
2.1	MRP II.....	17
2.1.1	Zpětné plánování dle MRP II.....	18
2.1.2	Dopředné plánování dle MRP II.....	18
2.2	APS jako následník MRP II.....	19
2.2.1	Dopředné plánování APS.....	20
2.2.2	APS s kombinovaným plánováním.....	20
3	Použití pokročilého plánování a rozvrhování.....	22
3.1	Použití APS systémů v S&OP procesech.....	22
4	Analýza algoritmu.....	24
4.1	Metoda programovacího omezení pro pokročilý plánovací a rozvrhovací systém s víceúrovňovými strukturovanými produkty.....	24
4.1.1	Model CP.....	25
4.1.2	Řešení přístupu.....	28
4.1.3	Šíření omezení.....	29
4.1.4	Strategie větvení.....	30
4.1.5	Výpočetní studie.....	31
4.1.6	Porovnání s metodou MIP.....	32
5	Aplikace s databází.....	34
5.1	Databázová analýza.....	34
5.1.1	Funkční analýza aplikace.....	34
5.1.2	Datová analýza aplikace.....	35
5.1.3	Techniky databáze.....	37
5.1.4	Prezentační vrstva.....	43
5.1.5	Algoritmus plánování.....	44

5.1.6	Fungování algoritmu	44
5.2	Uživatelský pohled	48
5.2.1	Databáze v D2APS.....	48
5.2.2	Zobrazení plánu v D2APS.....	53
5.2.3	Příklad fungování algoritmu.....	58
5.3	Časová validace algoritmu.....	65
6	Závěr.....	67
7	Seznam použitých zdrojů	68

Seznam obrázků

Obr. 1 APS moduly jako matrice dodavatelského řetězce (upraveno z [1]).....	14
Obr. 2 S&OP proces.....	22
Obr. 3 Struktura produktů [5].....	25
Obr. 4 Proces CSPs (upraveno z [5])	29
Obr. 5 Ganttův diagram zobrazující daný příklad [5]	32
Obr. 6 Výskytové diagramy databáze	36
Obr. 7 E - R diagram databáze [11].....	36
Obr. 8 Model-View-ViewModel [7]	37
Obr. 9 Ukázka tabulky Operace	41
Obr. 10 Ukázka tabulky Zakazky.....	42
Obr. 11 Úvodní stránka aplikace.....	48
Obr. 12 Přehled výrobních operací	49
Obr. 13 Části stránky.....	49
Obr. 14 Nový stroj.....	50
Obr. 15 Přidání nového stroje	51
Obr. 16 Přidělení Id.....	51
Obr. 17 Mazání stroje.....	52
Obr. 18 Režim Prohlížení strojů.....	52
Obr. 19 Editační režim	53
Obr. 20 Spuštění plánování v D2APS	54
Obr. 21 D2APS plán pro den	55
Obr. 22 D2APS plán pro týden	56
Obr. 23 D2APS plán pro měsíc.....	57
Obr. 24 Časová osa.....	58
Obr. 25 Algoritmus na příkladu	58
Obr. 26 Data operací k příkladu	59
Obr. 27 Data strojů k příkladu.....	59

Seznam tabulek

Tabulka 1 Rozdíly mezi systémy APS a ERP (upraveno z [2]).....	12
Tabulka 2 Základní rozdíly mezi APS a MRP II (upraveno z [1]a [2]).....	17
Tabulka 3 Informace o výrobcích (upraveno z [5])	31
Tabulka 4 Informace o problémech (upraveno z [5])	32
Tabulka 5 Porovnání modelů CP a MIT (upraveno z [5])	33
Tabulka 6 Informace o problémech (upraveno z [5])	65
Tabulka 7 Časové výsledky zadaných problémů	65
Tabulka 8 Časové výsledky algoritmu	66

Seznam použitých zkratk

APS	Advanced Planning and Scheduling
BT	Backtracking
COP	Constraint optimization problem
CP	Constraint Programming
CPSs	Constraint Satisfaction Problems
CRP	Capacity requirements planning
DBR	Drum Buffer Rope
EDM	Entity Data Model
ERP	Enterprise Resource Planning
GT	Generate and Test
JIT	Just In Time
TOC	Theory of Constraints
MIP	Mixed-Integer Programming
MPS	Master production schedule
MRP II	Manufacturing Resource Planning
MVVM	Model-View-ViewModel
RAM	Random Access Memory
SCP	Supply Chain Planning
S&OP	Sales and operation planning
WPF	Windows Presentation Foundation

1 Úvod

Tato práce se zabývá pokročilým plánováním a rozvrhováním. V první části bude nastíněna problematika definice APS, dále je zpracován vznik APS systémů. V práci je dále zpracováno použití APS v S&OP procesech a představení algoritmické metody pro APS systémy. Nakonec je popsána vytvořená aplikace.

Pokročilé plánování a rozvrhování užívá zkratku APS odvozenou od anglického *Advanced Planning and Scheduling*. Občas se používá i kratší verze názvu a to „Pokročilé plánování“.

Systém pokročilého plánování a rozvrhování může být v podstatě jakýkoliv program, který užívá pokročilé matematické nebo logické algoritmy pro optimalizování a případně i simulování kapacitního rozvrhování, kapacitního plánování, plánování zdrojů a poptávek a vytváření prognóz blízké budoucnosti.

Teoretické benefity z používání APS jsou značné, nicméně stejně jako u obrábění, kde pro dosažení přesnějších rozměrů je potřeba lepší stroj, tak i u APS je potřeba lepší přístup k informacím a celkově program více „opečovávat“, než jiné, „hrubější“, plánovací nástroje. Proto pro mnoho podniků je jednodušší spokojit se s méně přesnějším plánovacím programem, než je APS. Problém APS tedy spočívá v náročnosti jeho údržby i implementace. Přestože se mluví o Průmyslu 4.0, většina malých a středních podniků na tuto úroveň zdaleka nedosáhne. Mnoho úkonů se vykonává ručně, jejich digitální zaznamenání je obtížné a vyskytují se v něm často chyby, které programy jako APS mohou přivést ke katastrofálním závěrům. I to pak vede ke skutečnosti, že poměrně hodně implementací Pokročilého plánování končí neúspěchem.

Proč se tedy vůbec s APS zaobírat? Protože, i když velká skupina podniků má k Průmyslu 4.0 velmi daleko, časem bude nucena se k němu alespoň přiblížit. A digitalizace APS programům velmi nahrává, protože znamená přesná data. Lze tedy očekávat s postupem času velký rozvoj APS programů a to i v menších podnicích. Pokročilé plánování je logickým krokem po zavedení ERP systému do podniku. Výsledky fungujícího Pokročilého plánování totiž stojí za investování do jeho údržby.

1.1 Problematika definice APS

Dnes se APS používá většinou jako pomocný software ke komplexnějším ERP systémům. Zde pak obstarává plánování výroby, nebo plní pouze podpůrnou roli v plánování pomocí svých prognóz a simulací.

Právě kvůli časté implementaci APS systému do ERP systému se těmito dvěma systémům přisuzují stejné vlastnosti a schopnosti (někdy úmyslně). Nicméně mezi APS a ERP je několik podstatných rozdílů. Předně ERP systém je zpravidla podstatně větší program, který zajišťuje celou administrativu podniku. Má tedy v podniku velkou skupinu uživatelů. APS systém naopak bývá menší a věnuje se především plánování výroby, případně podpoře plánovacích a rozvrhovacích rozhodnutí. Uživatelé APS systému tvoří v podniku velice malou skupinu. Pro uživatele se APS skládá z několika kroků (většinou jdoucích v tomto pořadí):

- vložení potřebných dat, případně stáhnutí dat z ERP
- úprava dat v APS systému
- definování krajních podmínek plánování a rozvrhování
- zpracování získaných informací od APS

Problém rozeznávání APS a ERP také do jisté míry způsobuje modulární sestavení obou systémů. Oba systémy se skládají z různých modulů, které po složení vytvoří požadovaný systém. Nicméně u některých modulů je problém určit, zda patří k APS či ERP. Toho využívá mnoho prodejců zabývajících se prodejem těchto systémů. APS systémy od různých výrobců se tedy mohou podstatně lišit ve funkcionalitě. U některých systémů, prezentujících se jako APS je opravdu těžké najít něco z pokročilého plánování a rozvrhování, za to obsahují spoustu administrátorských nástrojů. Naopak velcí výrobci ERP systémů si od APS vypůjčili a integrovali několik vhodných modulů a tím si pojistili svoji pozici na trhu. Akademická představa o APS systému a produkt APS se tedy může velice lišit. Není tedy divu, že i definice APS systémů se někdy liší. [1]

Tabulka 1 Rozdíly mezi systémy APS a ERP (upraveno z [2])

Oblast	APS systém	ERP systém
Plánování	Reálné plány zahrnující omezené kapacity	Plánování do neomezených kapacit
	Cíl: optimální plány	Cíl: proveditelné plány
	Tažný i tlačný systém	Tlačný systém
Oblast řízení	Řízení celého výrobního řetězce	Řízení výroby
Hlavní oblasti zaměření	Plánování poptávky, výrob, logistiky, dodavatelského řetězce	Transakční systém: finance, controlling, výroba
Tok informací	Oběma směry	Shora dolů
Schopnost simulací	Vysoká	Nízká až žádná
Schopnost optimalizace nákladů,	Vysoká	Není k dispozici
Doba výroby	Flexibilní	Fixní
Postupné plánování	K dispozici	Není k dispozici
Rychlost plánování a přeplánování	Vysoká, možné po každé změně	Nízká, obvykle jednou týdně až měsíčně
Zpracování dat	Analytické zpracování dat	Transakční systém
Uchování dat	Memory-resident	Databáze

1.1.1 Tři směry definování APS

Definicí APS je hodně. Celkově je lze rozdělit na tři myšlenkové směry. První porovnává APS s MRP II, druhý porovnává APS s ERP a třetí definuje APS popsáním jeho typických charakteristik. Pro komplexní přehled o APS je ale vhodné definovat APS ze všech tří směrů.

Akademické Pokročilé plánování a rozvrhování se často srovnává se MRP II, *Manufacturing Resource Planning*. MRP II se zároveň bere jako přímý předchůdce APS (viz kap. 2 Počátky APS systémů).

Definování APS pomocí popsání typických vlastností APS zkoušelo mnoho autorů. Stadler a Kilger definují APS podle jeho třech hlavních vlastností:

1. integrální plánování celého dodavatelského řetězce od dodavatelů až po zákazníky podniku
2. skutečná optimalizace správným určením alternativ, cílů a omezení pro různé problémy s plánováním a to pomocí optimalizačních plánovacích metod – exaktních nebo heuristických
3. hierarchický plánovací systém, který vytváří prostředí vhodné pro kombinaci předchozích dvou vlastností

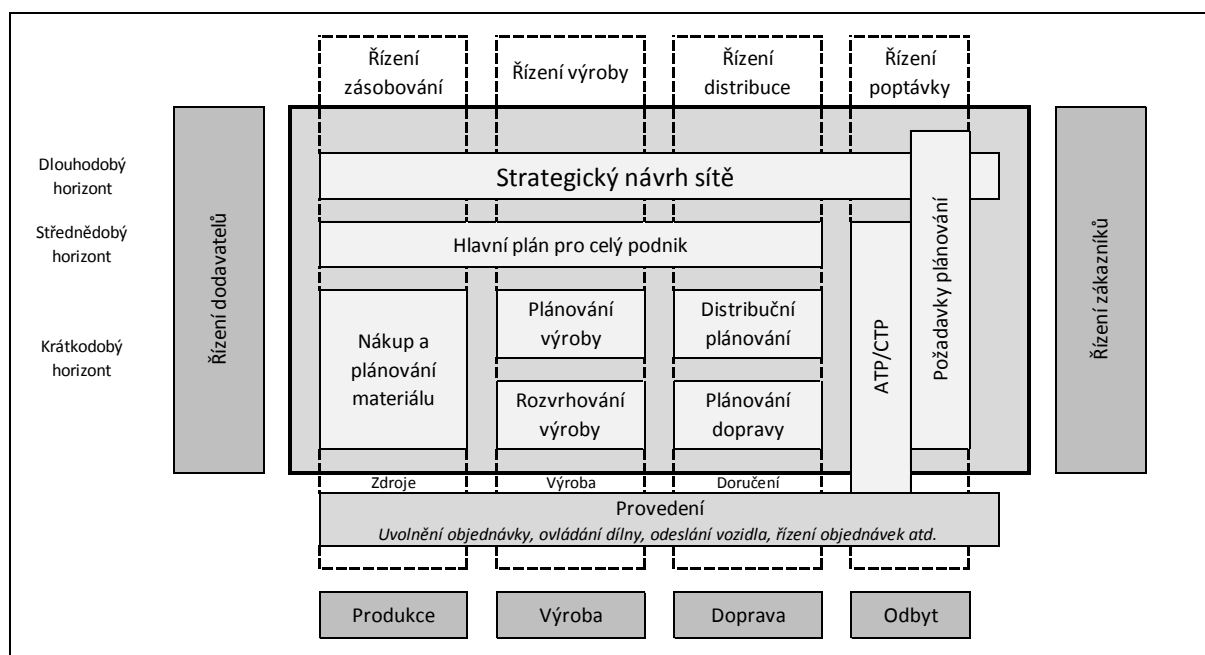
Helo a Szekely popisují APS podle pěti hlavních funkcí:

1. inventarizace dodavatelského řetězce a optimalizace velikosti dodávky
2. kalkulace splnitelnosti plánu
3. optimalizace zásob a dopravy
4. snížení nákladů na skladování
5. analýza toku materiálu

Wiers identifikuje tři prvky APS systémů:

1. je založen na modelu systému, který má být naplánován nebo rozvržen
2. obsahuje podpůrné nebo automatizované funkce pro generování plánů a rozvrhů
3. poskytuje grafické rozhraní pro prezentaci výsledného plánu či rozvrhu a dává uživateli možnost plán či rozvrh předělat [1]

Lze tedy vidět, že mnohé definice se velmi liší a nemuselo by být vůbec zřejmé, že definují stejný pojem. I z tohoto důvodu Stadler a Kilger navrhuje společnou strukturu komerčních APS systémů. A to na základě takzvaného SCP, tedy Plánování dodavatelského řetězce, anglicky *supply chain planning*. SCP určuje plánovací úkoly podle dvou parametrů. A to podle plánovacího horizontu (strategický, taktický, provozní) a podle procesu dodavatelského řetězce (nákup, výroba, distribuce, prodej). Obr. 1 popisuje strukturu modulů systému APS podporující procesy plánování v SCP.[1]



Obr. 1 APS moduly jako matrice dodavatelského řetězce (upraveno z [1])

Rozlišujeme tyto moduly:

- Strategické plánování sítě
- Plánování poptávky
- Hlavní plánování
- Plánování a rozvrhování výroby
- Plánování distribuce a dopravy
- Potenciální splnitelnost zakázky
- Plánování nákupních a materiálových požadavků

Strategické plánování sítě se zabývá kvantitativní částí strategického plánování. Patří sem otázky jako rozmístění výrobních a skladovacích prostor a zařízení nebo výběr nákupních a distribučních kanálů. Modul není zodpovědný za celkové zodpovězení těchto otázek, pouze za zodpovězení z kvantitativního pohledu.

Modul plánování poptávky řeší jak dlouhodobé – strategické odhady poptávek, tak i plánování odbytu ve střednědobém časovém horizontu. Výstupy z tohoto modulu obvykle obsahují vstupní údaje pro ostatní moduly.

Hlavní plánování koordinuje materiálový tok dodavatelského řetězce ve střednědobém horizontu.

Modul plánování a rozvrhování výroby má na starosti velikosti dávky, přiřazení strojů, rozvrhování a posloupnosti ve výrobě. APS je většinou určováno jedním až dvěma moduly pro plánování a rozvrhování výrobních úkolů.

Jak už napovídá název modulu, modul plánování distribuce a dopravy řeší omezení v distribučním systému podniku a to v střednědobém – taktickém časovém horizontu. Například se jedná o pravidelné dopravní spoje, dodávky do skladů nebo přidělování zákazníků ke zdrojům a využívání poskytovatelů služeb.

Modul potenciální splnitelnosti zakázky se stará o příchozí objednávky od zákazníků. U objednávek kontroluje jejich splnitelnost v zadaném časovém termínu, pomocí dat o dostupnosti materiálů, vytížení strojů. Někdy také počítá i s tím, zda se vyplatí přijmout objednávku a dodat jí se zpožděním.

Poslední modul, plánování nákupních a materiálových požadavků, je spojen s krátkodobými a střednědobými nákupními procesy. Tento modul není v APS systémech tak častý, kvůli tomu, že je již obsažen v ERP systémech, ke kterým se APS přidružují.

Jaká je tedy správná definice APS systému? Je obtížné definovat APS pomocí jeho procedur tak, aby to nezasvěcený člověk pochopil. Je rovněž obtížné porovnávat funkce APS a funkce ERP, protože mnoho ERP systémů přijalo některé funkce APS za své. Definování APS pomocí specifických modulů SCP má tu nevýhodu, že moduly mohou být pokaždé nazývány jinak. Nehledě na to, že vždy budou vznikat nové a alternativní moduly a funkčnost jednotlivých modulů se může značně lišit v závislosti na dodavateli. Asi nejlepší možností je tedy definování APS systémů podle svých jedinečných vlastností. [1]

1.1.2 Definice klíčových slov

Zcela první, zřejmou, vlastností je, že systém APS je počítačový program, který podporuje plánování na různých úrovních. APS většinou získává řídicí a transakční data od ERP nebo podobného informačního systému. APS data zpracuje, navrhne rozhodnutí, které pošle zpět do informačního systému, kde se provede konečné rozhodnutí. Řídicí data jsou základní charakteristiky podnikatelských subjektů, tedy zákazníci, produkty, služby a dodavatelé. Transakční údaje popisují události ve firmě, například faktury, platby a dodávky. Protože úspěšné rozhodování je často spjaté se schopností rychle vytvářet nové plány, tak APS systém používá RAM, tedy *Random-access memory*. Data jsou tedy zpracovávána jen v paměti počítače, což vede k rychlému zpracování a rychlé změně plánu. Poslední vlastností je simulování scénářů, které představují vygenerované varianty rozhodnutí.

Čím si APS zasloužilo první část názvu, tedy *pokročilý*? Podle Bermudeze je důsledek přidání slova „pokročilý“ před plánování a rozhodování jeho „současným zvážením omezení, které mají zlepšit rozvrhnutí a plán výroby“. Právě použití omezení modelování je často bráno jako další jedinečná vlastnost systému APS. Obecně lze říct, že omezení představují rozbor pravidel, limitů a cílů, které řídí oblast možností pro splnění podnikatelského plánu. [1]

Konkrétně tedy omezení může mít spoustu aspektů. Pravidla mohou být třeba výrobní nařízení, že po určitých strojních hodinách musí proběhnout vyčištění stroje. Nebo mohou být pravidla ekonomická, například specifikace, že objednávky zákazníků jsou brány v úvahu před nebo do očekávané poptávky. Co se týče limitů, může to být něco obecného, jako třeba dostupnost materiálu nebo kapacita stroje, nebo to může být třeba podrobně popsána potřeba minimální pracovní dovednosti u stroje pro určitou část výrobního postupu.

Dalším z důležitých vlastností APS systému je optimalizace, tedy systematický přístup ke zlepšování výrobního plánu. Optimalizace probíhá pomocí rozdělení omezení na měkká a tvrdá. Tvrdá omezení jsou obvykle fyzická omezení jako kapacita stroje nebo množství materiálu k dispozici. Měkká omezení fyzickou podstatu nemají, jedná se většinou o cíle jako minimalizace nákladů, eliminace přesčasů. Zatímco tvrdá omezení je třeba vždy nutně splnit,

měkká omezení není bezpodmínečně nutné plnit. Nicméně při nesplnění měkkého omezení většinou následuje nějaký postih pro podnik (např. zvýšené náklady). Optimalizace se obvykle řeší pomocí lineárního programování, genetických nebo heuristických algoritmů.

Pokud tedy se tedy výše popsané vlastnosti shrnou dohromady, vyjde následující definice APS systémů. APS systémy:

- poskytují podporu v plánování a to jak v krátkodobém tak i ve střednědobém a dlouhodobém časovém období
- vytváří a zároveň vyhodnocuje různé scénáře
- uvažuje omezení
- používá matematických algoritmů k řešení optimalizačních úloh
- pět hlavních úkolů systému APS je plánování poptávky, plánování výroby, rozvrhování výroby, plánování distribuce a plánování dopravy [1]

1.2 Vznik APS

Vznik APS jako programu se dělí na 2 hlavní proudy. Prvním je vzniknutí komerčního systému, který je pak nabízen podnikům jako celek. Druhý způsob je systém na zakázku, který je založený na algoritmech dělaných na míru zákazníkovi. Tyto algoritmy mohou řešit speciální plánovací problém, případně mohou být použity jako podpůrné algoritmy pro plánování v různých plánovacích úrovních, nebo pro plánování speciálního výrobku. APS systémy tedy mohou vycházet také z interně vyvinutých systémů, které slouží pro podporu rozhodování. [1]

Pokročilé plánování se stále řadí k těm mladším technologiím, přestože myšlenka i pojem Pokročilé plánování se objevily už v 90. letech minulého století.

2 Počátky APS systémů

Pokročilé plánování a rozhodování vzniklo v důsledku snahy ulehčit práci plánovačům a udělat plány více reálné, ne „přibližně přibližné“, jak tomu bylo často před vznikem APS (a bohužel to lze vidět i dnes). Jednoduché plánovací metody, nejčastěji založené na programu Microsoft Excel, nejsou schopny reagovat opakovanou situaci, kdy má firma rozpracovanou zakázku s termínem až ke konci měsíce a přijde další zakázka od jiného zákazníka, který by ji chtěl mít dodanou do konce týdne. Samozřejmě zde bude tendence přijmout i tuto objednávku a přejít z rozpracované první objednávky na druhou. To znamená zcela změnit výrobní plány pracovišť. Programování jednoduchými metodami je velice zdlouhavé a tak i reakční doba podniku se značně prodlužuje.

Ve zkratce se dá říci, že APS systémy jsou vhodné tam, kde jednodušší metody plánování nezvládají vícekritériální rozhodování.

2.1 MRP II

Aby se funkčnost APS lépe pochopila, bude nejprve představeno plánování pomocí předchůdce Pokročilého plánování a rozvrhování a to MRP II. MRP II je zkratka anglického *Manufacturing Resource Planning*, česky se překládá jako plánování podnikových zdrojů.

Tabulka 2 Základní rozdíly mezi APS a MRP II (upraveno z [1]a [2])

APS systém	MRP II systém
Celková racionalizace	Lokální racionalizace výrobního plánu
Simultární plánování	Sekvenční plánování
Plánování s omezeními	Velmi omezení možnosti při zahrnutí omezení do výrobního plánu.
Sledování vícefaktorových změn, vysoká rychlost	Sledování změn pouze v jednom směru, vícenásobné opakování
Přesné zobrazení dodavatelského řetězce	Omezené zobrazení reality produkce
Priorita zakázek se může lišit v závislosti na obchodním významu zákazníka	Všichni zákazníci mají v systému stejnou prioritu
Realizační doba zakázky se může dynamicky posouvat kontaktováním	Realizační doba je daná a neměnná
Dynamicky přepočítá plán i rozvrhování během několika po nějaké změně	Plány se dělají dávkově (například na následující den)
Identifikace vyjímek, která ulehčuje prohledávání zpráv ze systémů	Detailní zprávy
Podporuje nadstandartní rozhodování pomocí analýzy "what if" a pomocí	Neobsahuje žádné podpůrné rozhodovací prvky
Rozložení materiálu podle dostupnosti a zadaných kritérií	Materiál se rozděluje od prvních operací a dokud je k dispozici

2.1.1 Zpětné plánování dle MRP II

Jako první se u plánování dle MRP II vytvoří hrubý plán, který se nazývá MPS, tedy *Master production schedule*. Hrubý plán se sestává z finálních položek kusovníku, tedy většinou hotových výrobků, který chce firma vyrobit a to v střednědobém časovém horizontu. MPS se vyznačuje tím, že zcela ignoruje rozpracovanost výroby a kapacity strojů či pracovišť.

MPS je sám o sobě velmi nedostatečný, ale slouží jako základní bod, od kterého se odvíjí plánování dál. Dalším krokem je tedy plán hlavní, označovaný jako MRP z anglického *Manufacturing Resources Plan*. MRP se zaměřuje na části výrobků, tedy jednotlivé úrovně kusovníku a na materiál, který je pro ně potřeba. Dále hlavní plán počítá s časy. U nakupovaných materiálů, částí výrobků a polotovarů počítá s dobou nákupu. U vyráběných položek pak řeší čas potřebný pro jejich vyrobení. Hlavní plán zanedbává možnost nedostupnosti materiálu.

Po hlavním plánu přijde na řadu kapacitní plánování, tedy CRP, *Capacity requirements planning*. Kapacitní plánování prochází hlavní plán a hledá možnost přetížení kapacity u některého z výrobních zdrojů. V případě detekování takovéto možnosti musí plánovač upravit hlavní a případně i hrubý plán. Nicméně zasahování do těchto plánů je problematické, protože přesunutí práce z přetíženého stroje na druhý často způsobuje přetíženost onoho druhého stroje. Plánovač tedy musí zkoušet různé kombinace a tento úkol má tím těžší, čím víc výrobních zdrojů podnik má. Rychlost vytvoření plánu tak s velkou mírou souvisí se zkušenostmi plánovače.

Velká výhoda zpětného plánování MRP je jeho jednoduchost a velká tolerance k přesnosti dat, na kterých pracuje. Z toho vyplývá i další výhoda a to levná cena provozování aplikace založené na MRP II. Nicméně pak jsou zde i nevýhody: zanedbání dostupnosti i omezení kapacit v hlavním plánu, naprostá absence vazeb mezi jednotlivými zakázkami. Výsledkem je bohužel to, že použití zpětného plánování MRP II nepředkládá, i při jednoduché výrobě, realistický hlavní plán. I po zásazích plánovače, které často zaberou hodiny, se plán k reálné výrobě jen přiblíží. Čím větší bude množství objednávek, typů výrobků a vytíženost pracovišť, tím hůř půjde vytvořit použitelný hlavní plán, i když se do něj bude zpětně zasahovat.

Použití MRP II tedy nemůže v dnešní době zajistit funkční a splnitelný plán s jasným určením data ukončení jednotlivých zakázek. Tento důvod vedl k tomu, že se zavedlo dopředné plánování s uvažováním omezených kapacit. [2]

2.1.2 Dopředné plánování dle MRP II

Dopředné plánování MRP II, občas označované jako Post MRP II, je odpověď na nedokonalost plánování zpětného. Na rozdíl od zpětného plánování, plánování dopředné plánuje do omezených zdrojů. Vstupní data jsou stejně jako v zpětném plánování hrubý a hlavní plán.

Hlavní rozdíl mezi dopředným a zpětným plánováním je, že dopředné plánování plánuje od současnosti, tedy časová osa dopředného plánování má začátek v současnosti. Zpětné

plánování má začátek časové osy v budoucnosti. Druhá odlišnost je v plánování kapacitním. V dopředném plánování je tendence zatěžovat pracoviště na jeho maximální kapacitní výkon. Tedy v praxi práce je naplánována na pracoviště hned, jak je k dispozici materiál z předchozí operace. Tato tendence se vyplatí pro silně zatížené pracoviště, ale pracoviště s malým vytížením tak produkují příliš brzy rozpracovaný materiál, který na výrobě překáží a zvyšuje nákladovost.

Tato tendence plánování tedy umožňuje plánovat se skutečnými kapacitami zdrojů a umožňuje případnou optimalizaci jednotlivých zdrojů. Nevýhodou je značná pracnost plánování v případě, že některé pracoviště není schopné zvládnout všechnu práci a musí se část práce přesunout na pozdější dobu a rozpracovaná výroba. Dopředné plánování dle MRP II vyžaduje opravdu přesná data, což je pro velkou část podniků v České Republice problém.

Dopředné plánování dle MRP II, nevyužívá metodu TOC (teorie omezení) a proto se operace na kritických pracovištích zpožďují. Díky dopřednému plánování se sice podnik vyhne přetížení kritických pracovišť, nicméně při přesouvání práce na pozdější dobu u kritických zdrojů nedochází ke stejnému přesunutí dodání materiálu. Takže se materiál objedná zbytečně brzy, čímž se zvýší náklady na skladování. Další problém je, že při posunutí práce na pozdější dobu se nepřesune výroba, která této práci předchází. Zvyšuje se tak rozpracovanost výroby. A další problém vznikne, když je dopředné plánování použito u více produktové výroby. V případě, kdy jedno pracoviště má podle plánu provést operace z různých zakázek, dopředné plánování požadavky všech zakázek prostě sečte dohromady, místo aby s nimi nějak pracovalo dále. Výsledek je samozřejmě přetížení daného pracoviště.

Lze tedy říct, že nadstavba MRP II, Post MRP II, příliš nezlepšila plánování výroby jako takové. Nepodařilo se odstranit zpoždění zakázek, kvůli přesouvání výroby do budoucna dochází k velké rozpracovanosti výroby a propustnost výroby se stejně nezvýšila. Post MRP II nedala moc možností více produktovým podnikům bez linkových pracovišť. Výrobní plán tedy opět není reálný a slouží jen jako určitá osnova výroby. [2]

2.2 APS jako následník MRP II

Jak bylo výše ukázáno, klasické MRP II systémy tedy postupně procházejí jednotlivé technologické operace a přiřazují jim potřebný materiál a výrobní kapacitu. Nicméně tyto systémy nejsou přizpůsobené k rychlým změnám v požadavcích na výrobní zdroje a nepočítají s reálnou kapacitou materiálu. Systémy APS vznikly právě proto, aby tyto nedostatky pokryly. Jsou tedy vyvíjeny s důrazem právě na plánování s pravidly, omezeními podniku, s požadavky na materiál a dostupnou kapacitou výrobních pracovišť. Velkou výhodou pro APS systémy je, že používají optimalizační metody pro tvorbu výrobního plánu tak, aby zohlednil aktuální stav priorit v plánu. V praxi tedy použití APS několikanásobně zkrátí plánovači čas a umožní mu více se zaměřit na úzká místa a efektivitu plánování.

Pokud by se mělo shrnout, v čem APS vyčnívá nad MRP II do několika bodů, vypadalo by to následovně:

- APS systémy pracují s dostupností veškerých výrobních zdrojů
- APS systémy podporují metodu TOC a zohledňují kritické výrobní zdroje
- Výrobní dávka není fixní, ale je tvořena dynamicky podle aktuální potřeby
- Výrobní zakázky jsou plánovány všechny najednou a se zohledněním na jejich priority
- Průběžná doba výroby není fixní, ale je výsledkem výrobního plánu
- APS systémy umožňují plánování alternativních postupů

Jelikož důvody vzniku APS a rozdíly mezi APS a MRP II jsou tímto vysvětleny, zbývá pouze lehce nastínit samotný vývoj APS systémů. [2]

2.2.1 Dopředné plánování APS

Jako první byly APS systémy s dopředným plánováním. Tyto systémy počítali s veškerými druhy omezení v podniku. Plánovalo se s dostupností materiálu, s kapacitami jednotlivých strojů, lidí i jakýchkoli evidovaných přípravků a s prioritami jednotlivých zakázek. APS systémy kontrolovaly u každé operace dostupnost všech potřebných zdrojů. Pokud nějaký zdroj nebyl dostupný, operace nebyla zaplánována.

Díky tomu se podstatně zvýšila propustnost systému, plány byly reálně proveditelné a tím pádem i termíny dodávek byly mnohem přesnější. Nicméně stále zde byly problémy, které zůstaly nevyřešené, jako roztržitost výroby a velké skladové náklady. Chybí zde element ze zpětného plánování, který by zajišťoval synchronizaci nákupů a výdejů materiálu. [2]

2.2.2 APS s kombinovaným plánováním

Potom, co vyšly najevo nedostatky dopředného plánování APS systému, byla vyvinuta kombinace tohoto dopředného plánování a starého zpětného plánování dle MRP II. Tvorba plánu zde probíhá odlišně od předešlých principů plánování.

Nejprve proběhne dopředné plánování, které seřadí operace podle kritických zdrojů, zadaných priorit a všech omezení. Případně bere v potaz i plánovačem nastavené faktory, kterými mohou být mimo jiné velikost výrobních a dodacích dávek, velikost časových bufferů nebo hranice maximální rozpracovanosti. Tím by měla být zajištěna propustnost systému.

Po zajištění propustnosti přichází na řadu zpětné plánování. To taktuje nekritické pracoviště podle pracovišť kritických, aby se zajistila synchronizace operací a snížila se rozpracovanost. V tomto kroku jsou využívány metody JIT, TOC a DBR. Operace u nekritických pracovišť se plánují tak, aby výstup dané operace mohl ihned pokračovat na následující kritické pracoviště.

Ve výsledku plán vzniklý kombinací dopředného a zpětného plánování není roztržitý, ale je synchronizovaný a účelně taktovaný. Náklady i rozpracovanost jsou sníženy o několik desítek procent. Tento plán je navíc skutečně realizovatelný.

Tato metoda byla vyvinuta hlavně pro výrobu na objednávku, kde je poptávka časově proměnná. Cílená skupina byly podniky s velkým portfoliem výrobků, které se vyrábí na stejných pracovištích a podniky se složitými výrobky skládajícími se z mnoha komponent.

Nicméně schopnost reagovat na častou změnu je málokdy na škodu, takže se metoda uplatní ve většině výrobních prostředí.

Vývoj se pak rozdělil APS systémy na 2 skupiny. Do první patří APS systémy, které jsou používány samostatně a do druhé patří APS řešení, které jsou použity jako moduly nějakého ERP systému. Obě dvě skupiny mají nicméně stejný cíl a být užitečným nástrojem pro podporu plánování a rozvrhování výroby. [2]

3 Použití pokročilého plánování a rozvrhování

V této kapitole bude popsáno použití APS systémů v jednom ze základních použití a to v S&OP procesech.

S&OP je zkratka od anglického *sales and operation planning*, tedy česky obchodně-provozní plánování. Jedná se o proces, který shromažďuje všechny plány v podniku (například z výroby, vývoje, marketingu) a z informace z nich poskytuje manažerské části podniku. Hlavním cílem procesu je vytvořit celkový podnikový plán, který by umožnil vrcholovému vedení odhalit kritické zdroje a ty co nejlépe využít. Proto, aby mohl S&OP proces podávat tyto zprávy, musí využívat zástupce různých funkcí z odvětví, odkud plány bere. Tímto způsobem také využívá systém APS a jeho výstupy. S&OP proces se skládá z 5 kroků, jak lze vidět na Obr. 2. [4]



Obr. 2 S&OP proces

3.1 Použití APS systémů v S&OP procesech

Nejprve něco o použití APS u prvního a druhého kroku v S&OP procesu. Typicky systém APS generuje statistické prognózy pomocí různých metod, které uživatel může následně dopravit ručně. Většina APS systémů podporuje metodu společné prognózy, kde je možné shromáždit vstupy od všech zúčastněných aspektů a to od částí podniků až po zákazníky. Důvod tohoto postupu je snaha použít co nejvíce příslušných informací.

V některých procesech se bere v úvahu statistická prognóza, nicméně v řadě dalších jsou prováděny ruční úpravy (například od prodejních manažerů nebo rovnou zákazníků). Obvykle plánovací oddělení, případně hlavní plánovač, odpovídá za vytvoření předběžného plánu dodání v systému APS. To v praxi znamená vytváření agregované prognózy a následně její

upravování dle připomínek zákazníků a prodejních manažerů. Následně se zpracovaná prognóza přenesou do plánovacího programu. APS systémy mají možnost prognózy shromažďovat a rozdělovat na základě různých aspektů, jako třeba skupiny produktů, skupiny zákazníků, časové přímky nebo interních organizačních funkcí. To poskytuje přehled o prognóze na požadovaném rozměru a úrovni, které je možné použít během domlouvání termínů dalších zakázek, případně při uvažování o možném zvýšení prodeje.

Zároveň se zde uplatní simulace, kde uživatel APS systémů může zkoumat dopady různých rozhodnutí a činností, jako jsou například propagace a slevy, na poptávku zákazníků, na úroveň zásob nebo na přepravu. Tyto metody jsou při plánování velmi cenné, pokud se podnik nechce spoléhat na pouhé odhady.

K podpoře kroků 3 až 5 v S&OP procesu, tedy vyrovnání poptávky s dostupnými kapacitami a přidělení poptávky výrobním místům, systémy APS pomáhají s rozvržením výrobních plánů. Systémy APS mohou vytvářet plány pro celé dodavatelské řetězce, ne jen pro jednotlivé podniky. APS tedy zahrnou několik výrobních míst, subdodavatelů a distributorů do modelu plánování. Vstup do plánovacího programu je výše zmíněný předběžný plán dodávek spolu s hlavními údaji, které se většinou čerpají z ERP systému. Většinou se data z ERP systému přenesou do APS systému zcela automaticky, ale někdy je třeba data zadat ručně. Plány se generují automaticky, po spuštění příkazu uživatelem.

Po vygenerování plánu je možné provést ručně nějaké testy a změny, zde záleží na potřebách uživatele. Uživatel může pro tento plán přenastavit měkké a tvrdé omezení, nebo některá omezení úplně deaktivovat. Může samozřejmě provádět i změny jako zvýšení/snížení kapacity nebo poptávky. Následně se vytvoří nový plán, který se bude porovnávat s tím původním. Většinou APS systémy používají seznam porušení omezení, které uživateli ukážou dopady změn. Výsledné návrhy plánů pak mohou být základním materiálem pro plánovací schůze, kde je výrobní plán dokončen. V menších podnicích plánovač rovnou provede potřebné změny a dokončí plán sám. Ve větších podnicích výrobní a dodací plány, které APS vygeneruje, jsou často zaslány plánovačům na výrobních místech a nákupním oddělením. První skupina rozvrhne plán na vlastním místě a druhá skupina, tedy nákupní oddělení, má informace o zaplnění kapacity výroby. [4]

4 Analýza algoritmu

V této kapitole je představena algoritmická metoda pro APS systémy. Metody a algoritmy pro APS systémy jsou obecně chráněná tajemství výrobce, který tyto algoritmy považuje za své know-how a podle toho si je i chrání. Následující metoda je výjimkou potvrzující pravidlo a je veřejně dostupná.

4.1 Metoda programovacího omezení pro pokročilý plánovací a rozvrhovací systém s víceúrovňovými strukturovanými produkty

Pánové Y. Peng, D. Lu z Šanghajske univerzity a Y. Chen z Wenzhou univerzity navrhli zajímavý model pro APS systémy pracujícími s víceúrovňovými strukturovanými produkty. Tento model pracuje s následujícími omezeními:

- kapacita
- datum splatnosti
- tzv. „deadline“ – čas, do kterého musí být výrobek hotov
- priority

Dále je v tomto modelu navržena nová metoda programování omezení pro minimalizaci nákladů. Tato metoda je označována zkratkou CP z anglického *constraint programming*. Metoda je řešena pomocí optimalizační metody větví a mezí.

CP byl původně vyvinut pro řešení problému CSPs, tedy *Constraint satisfaction problems*. Česky by se CSPs dalo přeložit jako problémy spokojenosti s omezením. CSPs jsou matematické otázky, které se definují jako objekty, které musí splnit řadu omezení. Klasický příklad, na který lze CSPs implementovat, je křížovka. CP zde měla nalézt hodnotu pro každou proměnou, kde omezení specifikují, že některé podmnožiny hodnot nemůžou být použity dohromady. Tedy řešení se měla větvit na varianty. Pak byla metoda rozšířena o cílovou funkci, takže řešila problémy s optimalizovaným omezením. Cílová funkce může být například nákladovost. Optimalizované řešení CSPs je prováděno tak, že cílové funkci je vnučována nová hodnota. Takže výsledná metoda CP spoléhá na řešení problému pomocí CSPs s cílovou funkcí s vhodnou kombinací s metodou větví a mezí.

Nejprve tedy bude uvedena definice *Constraint satisfaction problems*. CSPs je trojnásobná matice $P=(X,D,C)$, kde X je n -tice proměnných – $X=(x_1,x_2,\dots,x_n)$, D je odpovídající n -tice domén – $D=(D_1,D_2,\dots,D_n)$ a C je n -tice omezení – $C=(C_1,C_2,\dots,C_1)$. Zároveň platí, že $x_i \in D_i$. Řešení představuje n -tice $A=\{a_1,a_2,\dots,a_n\}$, kde platí, že $a_i \in D_i$ a zároveň každé C_j je uspokojené.[5]

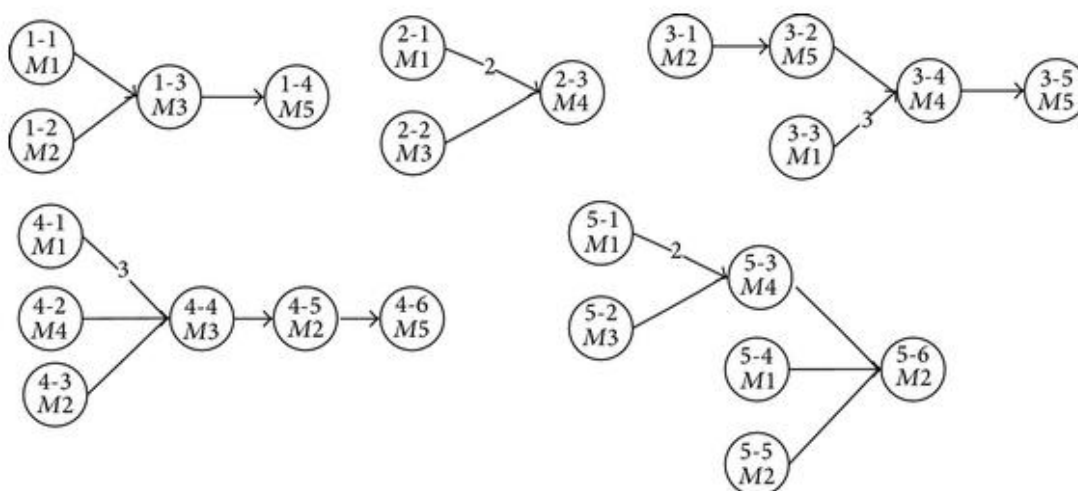
Algoritmus může být rozdělen do dvou funkčních částí – hledání a interference. Interferenční část algoritmu může eliminovat velké podprostory pomocí metody omezení lokálního šíření. Druhá část, vyhledávací, systematicky zkoumá řešení a může eliminovat podprostory s jediným selháním. Tyto dvě strategie eliminací jsou velice často kombinovány.

CSPs poskytuje řešení splňující všechna zadaná omezení, nicméně nenajde řešení optimální. Proto se použije cílová funkce. Možný rozptyl hodnoty cílové funkce je definován horní a

dolní hranicí. Ty jsou pomocí výpočtů postupně omezovány. Cílová funkce je přepsána jako omezení, které nutí onu funkci být rovna nové hodnotě (nebo být větší nebo menší než nová hodnota – zde záleží na nastavení). Řešení je iterativní. Tedy pokaždé, když je nalezeno možné řešení, tak se příslušná hranice (horní nebo dolní) sníží směrem ke středu rozptylu hodnoty cílové funkce. Proces skončí, až když se horní a dolní hranice rovnají nebo jsou všechny možnosti vyčerpány. Toto vylepšení CSPs pak dostalo nový název a to *Constraint optimization problem*, zkráceně COP. COP by do češtiny šlo přeložit jako optimalizované problémy s omezením. [5]

4.1.1 Model CP

Model je ukázán na konkrétním příkladu produktu, který má víceúrovňový kusovník. Strukturu produktů, se kterými je pracováno lze vidět na Obr. 3.



Obr. 3 Struktura produktů [5]

Z Obr. 3 lze vidět, že se jedná o 5 finálních výrobků. Informace v kroužku jsou číslo operace (nahore) a stroj potřebný pro tuto operaci (dole). Například 4-5 značí pátou operaci u čtvrtého výrobku a M2 znamená, že na tuto operaci je třeba stroj 2. Číslo u šipky znamená potřebu počtu dílů vyrobeného z předchozí operace. Absence čísla u šipky znamená potřebu jednoho dílu. Například na operaci 5-3 je potřeba 2x materiál z operace 5-1 a 1x materiál z operace 5-2.

Pro znázornění modelu CP je třeba uvést několik indexů a parametrů.

Indexy:

- i index produktu
- m index stroje
- j index operace

Parametry:

I	počet produktů
M	počet strojů
N_i	číslo operace pro i -tý výrobek
$task(i,j)$	j -tá operace i -tého výrobku
$machine(i,j)$	stroj pro j -tou operaci i -tého výrobku
q_{ij}	počet dílů, které $task(i,j)$ potřebuje
t_{ij}	čas zpracování operace $task(i,j)$
v_i	požadované množství výrobku i
r_i	doba uvolnění i -tého výrobku z výroby
d_i	datum splatnosti i -tého produktu
c_i	celkový výrobní čas i -tého produktu
z_{ijk}	nabývá hodnot 0 a 1, hodnotu 1 má tehdy, pokud operaci $task(i,j)$ předchází operace $task(i,k)$, jinak má hodnotu 0
H	efektivní pracovní doba za den
TC	poplatky za zpožděnou výrobu za den za práci
EC	poplatky za příliš rychle vyrobenou výrobu za den za práci

Proměnné:

C_{max}	celkový výrobní čas
e_i	brzké dodání produktu (čísla v oboru reálných hodnot)
l_i	pozdní dodání produktu (čísla v oboru reálných hodnot)
E_i	brzké dodání produktu (celá čísla)
L_i	pozdní dodání produktu (celá čísla)

Základní proměnná modelu programování omezení je počáteční čas každé operace ($task(i,j).start$). Další, navazující, proměnné jsou čas trvání operace ($task(i,j).duration$) a čas ukončení operace ($task(i,j).end$). Přičemž platí rovnice:

$$task(i,j).start + task(i,j).duration = task(i,j).end$$

Zde je matematické popsání problému:

$$Min Z = \sum_{i=1}^I (TC * L_i + EC * E_i) \quad (1)$$

s následujícími omezeními:

$$task(i,j).start \geq r_i, \text{ pro } \forall i,j \quad (2)$$

$$task(i, j).duration = t_{ij} * q_{ij} * v_i, \text{ pro } \forall i, j \quad (3)$$

$$task(i, n(i)).end \leq C_{max}, \text{ pro } \forall i \quad (4)$$

$$task(i, k).start + task(i, k).duration \leq task(j, p).start$$

nebo

$$task(j, p).start + task(j, p).duration \leq task(i, k).start,$$

pro $\forall i, j, k, p \in \{i, j, k, p \mid task(i, k) \neq task(j, p), machine(i, k) = machine(j, p)\}$

$$(5)$$

$$task(i, j).start + task(i, j).duration \leq task(i, k).start,$$

pro $\forall i, j, k \in \{i, j, k \mid z_{ijk} = 1\}$

$$(6)$$

$$\frac{c_i}{H} - d_i \leq l_i, \text{ pro } \forall i \quad (7)$$

$$d_i - \frac{c_i}{H} \leq e_i, \text{ pro } \forall i \quad (8)$$

$$L_i \geq l_i, \text{ pro } \forall i \quad (9)$$

$$E_i \geq e_i - 0,99, \text{ pro } \forall i \quad (10)$$

$$L_i, E_i, l_i, e_i \geq 0, \text{ pro } \forall i \quad (11)$$

Cílem je minimalizace celkových nákladů, které zahrnují i sankce za zpoždění a naopak i sankce za příliš časnou výrobu a to, že se musí někde skladovat, případně doručení příliš brzy také zvyšuje náklady.

Víceúrovňová struktura produktů je definována pomocí binárního parametru, která může vyjadřovat všechny stromové struktury. Nerovnice (2) vyjadřuje omezení počátečního času, který nesmí být menší než je doba uvolnění výrobku z výroby. Rovnice (3) definuje dobu trvání operace. Nerovnice (4) ukazuje, že doba dokončení každého produktu by měla být menší nebo rovna než celkový výrobní čas. Omezení (5) znamená, že pokud 2 operace požadují stejný stroj, tak druhá nemůže začít předtím, než se ukončí ta první. Nerovnice (6) omezuje začátek operace za pomoci předchozí operace. Nerovnice (7) a (8) definují brzký a pozdní čas každého produktu v reálných číslech. Nerovnice (9) a (10) převádí hodnoty

přechozích nerovnic do celočíselného stavu (kvůli tomu, že poplatky jsou počítány na celé dny). Nerovnice (11) omezuje proměnné na kladné hodnoty a nulu.[5]

4.1.2 Řešení přístupu

Cílová funkce (1) se vymaže a přidá se nerovnice (12), která znamená omezení pro strukturu CSPs. Optimální řešení COP může být generováno iteračně pomocí řešení CSPs a omezením (12).

$$Z \leq C \quad (12)$$

Následně je třeba se řídit postupem:

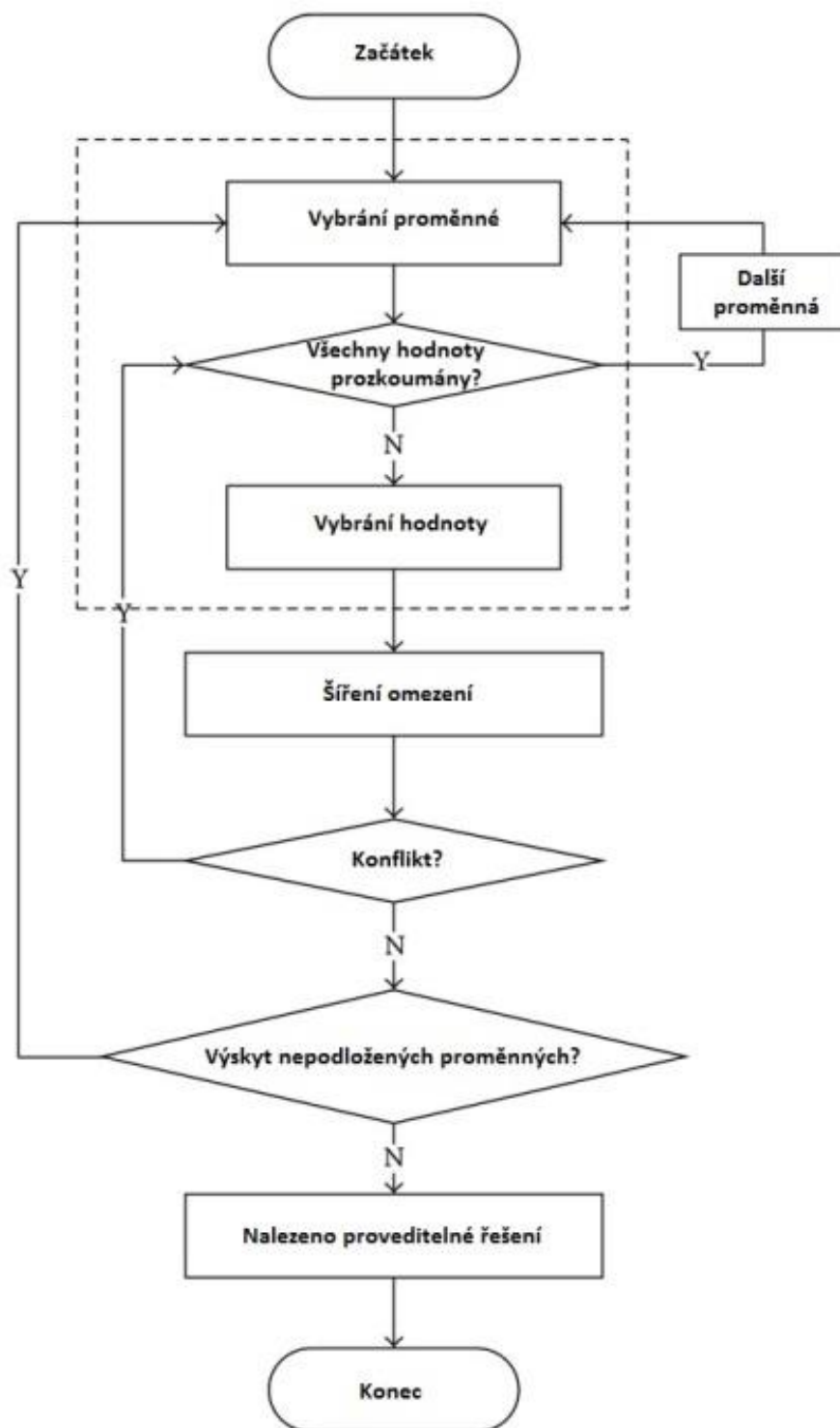
1. Vypočítat horní (UB) a dolní hranici (LB) z cílové funkce (1).
2. Přidat omezení (12), kde $C = (LB + UB) / 2$ do CSPs.
3. Vyřešit CSPs a nastavit pevný časový limit. Pokud je nalezeno proveditelné řešení S , je třeba snížit horní hranici UB na hodnotu $Z(S)$. V případě, že řešení nalezeno není, získá dolní hranice LB hodnotu $C+I$.
4. Opakovat kroky 2 a 3 dokud $UB=LB$.

Při zjištění prvního proveditelného řešení S_0 se horní hranice zjistí pomocí rovnice $UB=Z(S_0)$. Dolní hranici zjistíme následovně:

$$LB_z = TC * \sum_{i=1}^I \max\{0, \left\lceil \frac{C_{i \min}}{H} - d_i \right\rceil\} + EC * \sum_{i=1}^I \max\{0, \left\lfloor d_i - \frac{C_{i \min}}{H} - 0,99 \right\rfloor\} \quad (13)$$

kde $C_{i \min}$ je rovna nejdřívější době zahájení poslední operace $task(i, n_i)$ + době zpracování operace. A $C_{i \max}$ se rovná nejpozdnějšímu startu poslední operace $task(i, n_i)$ + době zpracování operace. Hodnoty v hranatých závorkách zde představují nejmenší možnou hodnotu. Vzhledem k tomu, že počáteční horní hranice je často velice nízká, se provádí zmenšení vzniklé mezery provedením dichotomického vyhledávání.

Proces řešení CSP probíhá pomocí hloubkového průzkumu vyhledávacího algoritmu. Jak lze vidět na Obr. 4, tak v každém uzlu je spuštěna fáze šíření pro zjištění nesrovnalostí a snížení vyhledávacího objemu dat. Pokud se zjistí nějaká nesrovnalost, proces odstraní účinky předchozího rozhodnutí. Pokud v hledaném prostoru žádná nesrovnalost není nalezena, tak se rekurzivně aplikuje podproces rozvětvení (opět na Obr. 4) pro uzly potomků. To se opakuje tak dlouho, dokud se nenalezne řešení, nebo dokud nejsou prohledány všechny prostory.[5]



Obr. 4 Proces CSPs (upraveno z [5])

4.1.3 Šíření omezení

Fáze šíření je velice důležitá pro snižování prohledávaného prostoru. Fáze šíření má za úkol zabránit prohledávání prostoru s exponenciální velikostí. Šíření omezení může být založené na dvou faktorech: na čase nebo na zdrojích. Šíření omezení založené na čase se nazývá

metoda rozvrhu (*timetable metod*). Metoda rozvrhu většinou používá časové okno. Časové okno času zahájení operace $task(i,j)$ je $[task(i,j).est, task(i,j).lst]$. $Task(i,j).est$ zde znamená nejdřívější čas zahájení operace a $task(i,j).lst$ znamená nejpozdější čas zahájení operace $task(i,j)$. Časové okno během procesu postupně zmenšuje svůj obor hodnot, dalo by se říct, že se ztenčuje. Jakmile se změní časové okno operace $task(i,j)$, změní se také časová okna následujících a předchozích operací. Pravidla šíření omezení pro metodu rozvrhu jsou následující:

$$task(i,j).est = r_i, \text{ pro } \forall i,j \in \{i,j | z_{ijk} = 0 \forall k\} \quad (14)$$

$$task(i,k).est = task(i,j).est + task(i,j).duration, \text{ pro } \forall i,j,k \in \{i,j,k | z_{ijk} = 1\} \quad (15)$$

$$task(i,j).lst = C_{max}(S_o) - task(i,j).duration, \text{ pro } \forall i,j \in \{i,j | z_{ijk} = 0 \forall k\} \quad (16)$$

$$task(i,j).lst = task(i,k).lst - task(i,k).duration, \text{ pro } \forall i,j,k \in \{i,j,k | z_{ijk} = 1\} \quad (17)$$

Nejdřívější čas každé operace lze vypočítat pomocí rovnice (14) a (15). Přičemž rovnice (14) se týká prvních operací, které nemají žádnou předchozí operaci a nepracují tedy s žádnými díly vyrobenými v jiných operacích, na které by museli čekat. Vzorec (15) pak platí pro nejdřívější časy následujících operací, tedy těch, které mají nějakou předchozí operaci a jsou závislé na jejich výrobcích. Obdobně lze vypočítat z vzorců (16) a (17) nejpozdější čas operací. Rovnice (16) je pro poslední operace, po kterých už žádné operace nenásledují a rovnice (17) je pro nejpozdější čas operací, které mají za sebou ještě další operace.[5]

4.1.4 Strategie větvení

Nejčastější vyhledávací metoda používaná v algoritmu CP je GT algoritmus, zkratka je odvozená od anglického *generate and test*. Česky by šel GT algoritmus přeložit jako generování a testování. Nicméně jeho účinnost je docela slabá a to kvůli neinformovanému generátoru a pozdnímu odhalování nesrovnalostí. Proto byl navržen algoritmus zpětné vazby – BT (z anglického *backtracking*). BT algoritmus je základní prohledávací metoda pro řešení problémů s omezeními. BT vytváří částečné řešení výběrem hodnot pro proměnné, dokud nedojde k nějaké slepé uličce, kde ono částečné řešení nemůže být rozšířeno.

Několik strategií založených na větvení bylo navrženo pro standartní řešení pro malé manufaktury. Strategie větvení určuje tvar prohledávacího stromu, který pak přímo ovlivňuje rychlost s jakou je prostor prohledáván. Zde se představí tři takové strategie.[5]

Strategie 1

První strategie (heuristické binární omezení) vytváří binární prohledávací strom rozdělením na dvě možnosti (větvě) definované disjunkcí. Omezení z rovnice (5) definuje takovéto dvě

možnosti. Za předpokladu, že dvě operace o_{ik} a o_{jp} sdílí stejný stroj m , omezení $task(i,k).start + task(i,k).duration \leq task(j,p).start$ je zařazeno na jednu větev stromu a omezení $task(j,p).start + task(j,p).duration \leq task(i,k).start$ je dáno na druhou větev.

Strategie 2

Druhá strategie (heuristické proměnné) se používá pro výběr proměnných s nejmenší velikostí domény. Doména proměnné je v intervalu od nejdřívějšího času zahájení operace po nejpozdější čas zahájení. Je vybrána nejmenší doména a hodnoty jsou seřazeny ve vzestupném pořadí.

Strategie 3

Třetí strategie (heuristika založená na úkolu) spočívá v definici strategie pro výběr úkolů a v heuristickém výběru hodnoty pro čas zahájení úkolu. Je vybrána operace s nejmenším posledním časem dokončení a je vybrána hodnota v sestupném pořadí.[5]

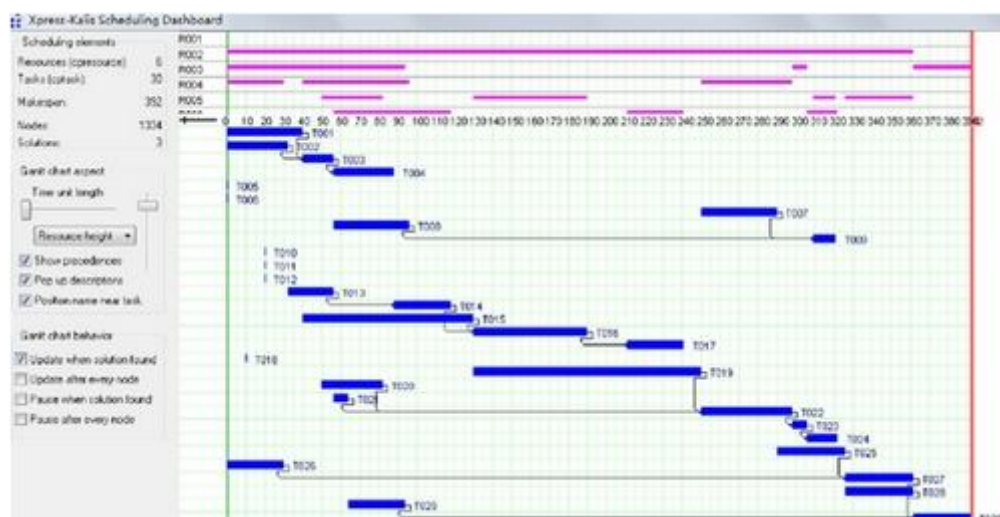
4.1.5 Výpočetní studie

Pro názornou ukázkou jak metoda CP funguje, je tato metoda uváděna na příkladu. V příkladu se vyskytuje 5 typů výrobků a 5 typů strojů. Na Obr. 3 bylo vidět stromové struktury výrobků včetně strojů, které výrobky vyrábějí. V tabulce 3 jsou uvedeny potřebné parametry produktů pro metodu CP. Tři čísla v závorkách znamenají v následujícím pořadí: požadované množství, dobu uvolnění a datum splatnosti. Dále jsou zde uvedené doby zpracování u jednotlivých operací. Co už v tabulce není, jsou informace, že efektivní pracovní doba je 80 za den, poplatky za zpoždění je 25 jednotek za den a poplatky za příliš brzké dodání jsou 5 jednotek za den.

Tabulka 3 Informace o výrobcích (upraveno z [5])

Výrobky	(v_i, r_i, d_i)	O1	O2	O3	O4	O5	O6
1	(8,0,2)	5	4	2	4	-	-
2	(4,20,5)	5	10	3	-	-	-
3	(6,10,4)	4	5	5	10	5	-
4	(8,50,3)	5	4	1	6	1	2
5	(6,0,5)	3	5	6	6	5	5

Výsledek metody CP je plán, ve kterém se produkt 4 zpozdí o 2 dny. Náklady na tento plán jsou tedy 50 jednotek a celkový pracovní čas je 392. Na Obr. 5 lze vidět Ganttův diagram, který znázorňuje tento plán.[5]



Obr. 5 Ganttův diagram zobrazující daný příklad [5]

4.1.6 Porovnání s metodou MIP

Nicméně bez porovnání s jinou metodou nemá tato ukázka velkou hodnotu. Proto zde bude ukázáno srovnání metody CP s metodou MIP (smíšeně-celočíselné programování). Jelikož chování metod je odlišné při každé struktuře podniku, je třeba zavést alespoň tři základní typy struktury, které tyto struktury podniku budou charakterizovat a to velký, střední a malý podnik. V každém typu struktur budou řešeny tři instance. Značení pro typy struktury je následující: malý podnik (a), střední podnik (b), velký podnik (c). Tabulka 4 uvádí počty produktů a operací pro tyto varianty.

Tabulka 4 Informace o problémech (upraveno z [5])

Typ výroby	Počet výrobků	Max. počet operací	Počet strojů
a	5	6	5
b	10	6	5
c	5	21	5

Tabulka 5 Porovnání modelů CP a MIT (upraveno z [5])

	Instance	Počet proměnných	Počet omezení	Čas zpracování (s)	Náklady
CP	a-1			0,1	50
	a-2	182	539	0,1	50
	a-3			0,1	150
	b-1			0,5	50
	b-2	337	1964	1000+	75
	b-3			3,6	150
	c-1			16,5	75
	c-2	407	13937	5,7	125
	c-3			1000+	300
MIP	a-1			0,2	50
	a-2	182	301	0,2	75
	a-3			0,1	150
	b-1			37,4	50
	b-2	657	1190	1000+	150
	b-3			1000+	175
	c-1			44,4	75
	c-2	4693	9188	1000+	125
	c-3			1000+	275

Tabulka 5 ukazuje počty omezení, proměnných, čas zpracování softwarem a hodnotu cílové funkce pro každou metodu. Zatímco u metody MIP je mnohem více proměnných než u metody CP, metoda CP má zase více omezení. Při větších problémech prudce roste čas řešení pro metodu MIP. Maximální doba výpočtu byla nastavena na tisíc sekund a MIP u větších problémů nestihá najít do tohoto limitu optimální řešení. Stalo se tak čtyřikrát z devíti zkoumaných problémů. Nicméně i metoda CP v zadaném limitu nestihla ve dvou případech z devíti optimálních řešení najít.

Zároveň byly na příkladu testovány strategie větvení, kde si nejlépe vedla strategie 1.[5]

5 Aplikace s databází

Praktickou částí práce je realizace databázového systému s algoritmem pro generování testovacích dat. Následná data z vytvořené databáze jsou pak použita ve vybraném algoritmu plánování. Databáze i plánovací algoritmus jsou propojené v jedné aplikaci pod názvem D2APS. Programování obou částí bylo prováděno v Microsoft Visual Studio 2017 v jazyku C#. Grafické zobrazení výsledků plánovacího algoritmu pak bylo vytvořeno s využitím platformy Telerik.

5.1 Databázová analýza

V této kapitole bude představena vnitřní stavba aplikace. Vnitřní stavba aplikace představuje základ, na kterém se staví dále. Stejně jako u ledovců i vnitřní stavba aplikací může být mnohem mohutnější, než část, kterou člověk - uživatel vidí nad vodou.

5.1.1 Funkční analýza aplikace

Nejprve je třeba stanovit základní funkce, které se od aplikace očekávají. Od aplikace se očekávají zejména tyto stěžejní funkce:

- zobrazovat data
- uchovávat data v databázi
- editovat data
- přidávat a odebírat data
- ověřovat správnost zadávaných a měněných dat
- plánovat výrobu dle aktuálních dat
- možnost zobrazit více plánů výroby současně

Data, se kterými aplikace pracuje:

- data charakterizující zadané zakázky pro plánování
- data charakterizující zadané operace pro plánování
- data charakterizující stroje pro plánování

Aplikace se dělí na čtyři části, jednu hlavní a tři vedlejší: správa strojů, správa operací a plánovací část.

Hlavní část obstarává následující funkce:

- rychlý a jednoduchý přechod mezi vedlejšími částmi
- ukládání a kontrola dat
- úvodní uvítání uživatele

Správa strojů je věnována strojům. Její funkce jsou:

- zobrazení dat o strojích
- možnost seřazení dat
- možnost editace dat strojů

- možnost odebírání stávajících strojů
- možnost přidání nových strojů
- ukládání dat do databáze
- kontrola správnosti dat

Správa operací je věnována operacím. Jedná se o jednoduchý přehled operací, a proto zde není potřeba mnoho funkcí:

- zobrazení dat týkajících se operací
- možnost seřazení dat

Plánovací část je pro plánování výrobního plánu. Její funkce jsou následující:

- plánování dle aktuálních dat (nemusí být uložené v databázi)
- zobrazení plánu pro konkrétní den
- zobrazení plánu pro konkrétní týden
- zobrazení plánu pro konkrétní měsíc
- zobrazení plánu jako časovou osu pro jednotlivé stroje
- možnost pohodlného přepínání mezi jednotlivými dny, týdny i měsíci

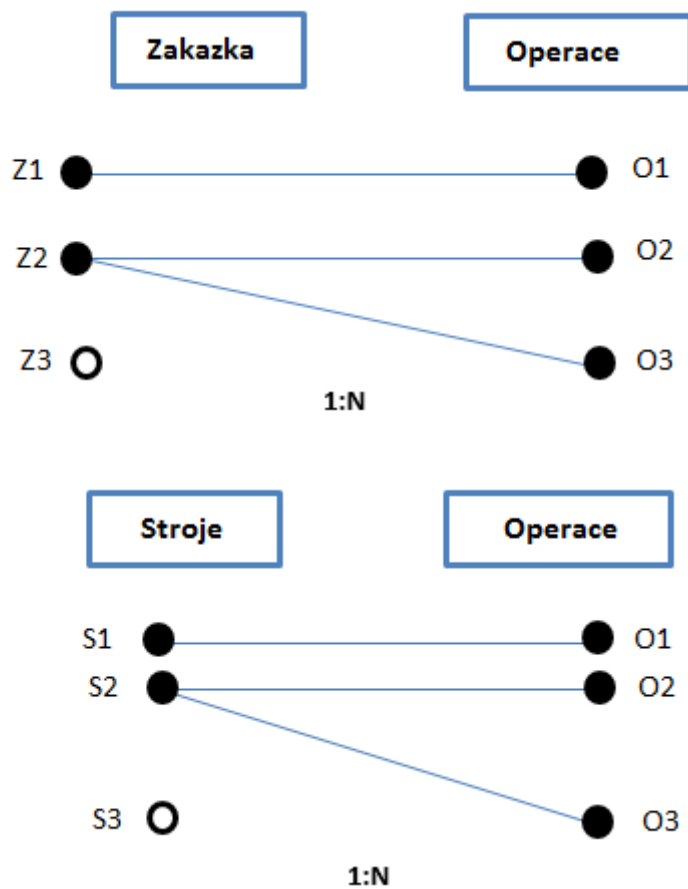
5.1.2 Datová analýza aplikace

Samotná databáze pro algoritmus obsahuje tři navzájem propojené relační tabulky: Stroje, Operace a Zakázky.

Nejprve je třeba uvést integritní omezení databáze:

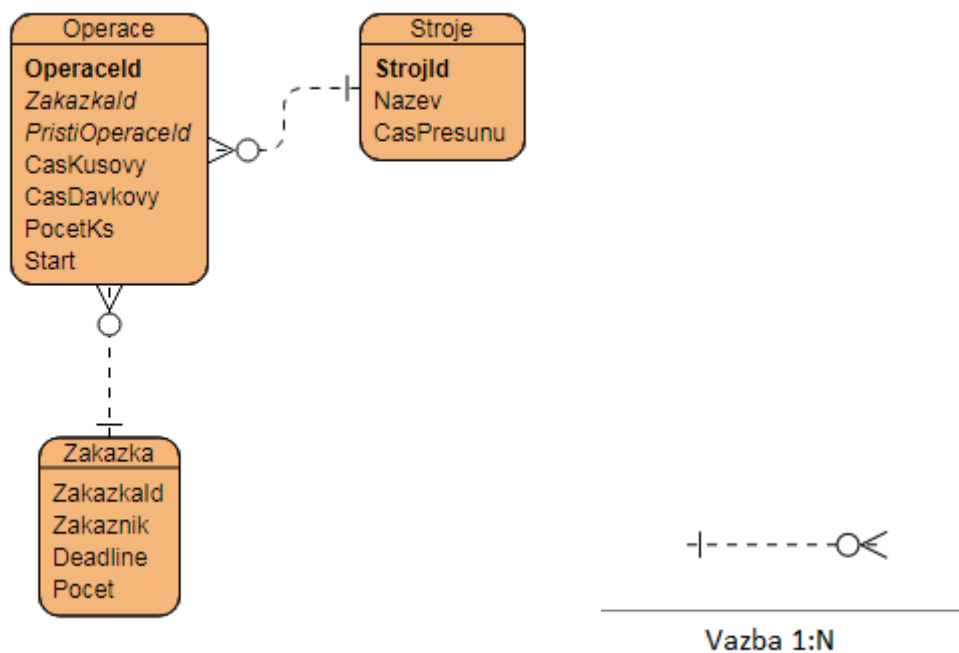
- Jedna výrobní operace se musí dělat na základě právě jedné zakázky
- Jedna zakázka může mít libovolný počet operací
- Jedna výrobní operace se musí provádět na jednom
- Jeden stroj může být použit na libovolný počet operací

V závislosti na integritních omezeních pak platí následující výskytové diagramy:



Obr. 6 Výskytové diagramy databáze

Pro úplný přehled je pak uveden E-R diagram databáze.



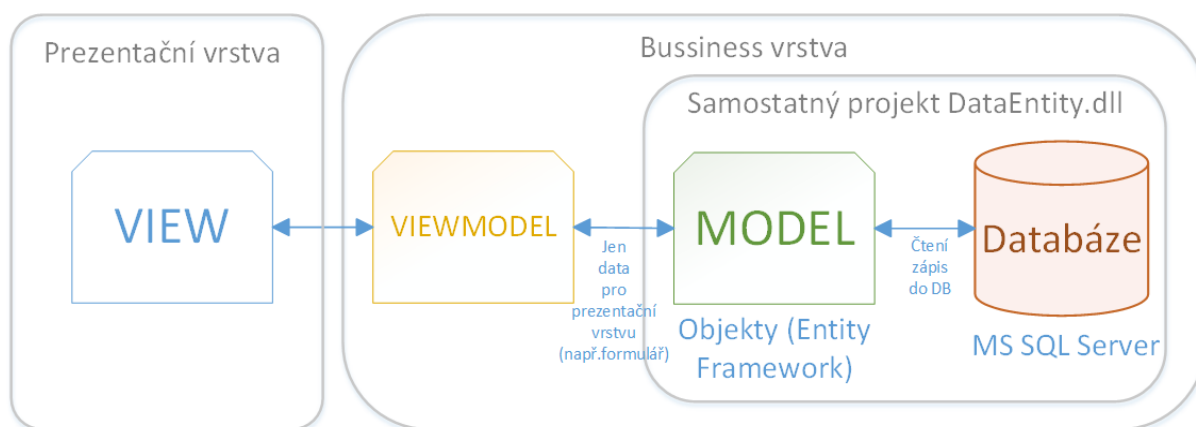
Obr. 7 E - R diagram databáze [11]

5.1.3 Techniky databáze

Nejprve byla vytvořena databázová aplikace, která je schopná poskytovat data pro plánovací algoritmus. Jedná se o jednoduchou databázovou aplikaci vytvořenou pomocí jazyku C# ve Visual Studiu 2017. S daty se pracuje pomocí lokální databáze (LocalDB) a pomocí balíčku Entity Framework. Entity Framework využívá model zvaný Entity Data Model (EDM) a z toho důvodu je datový model na straně uživatele. EDM, potažmo Entity Framework, také převádí relační databázi na objekty. Informace z databáze pak Entity Framework načítá uživateli pomocí techniky *lazy loadingu*. Tato technika zobrazování dat z databáze zobrazuje takové databázové údaje, které jsou v daný moment opravdu potřeba.

Jsou tři hlavní přístupy k tomu, jak vytvářet databázi. První je *Database First*, kde pracujeme s již existující databází nebo klademe velký důraz na kontrolu nad samotnou databází. Druhý přístup je *Model First*. Jedná se o přístup, který odděluje kód a databázi pomocí deklarativního popisu. Třetí a poslední možností je *Code First*. Zde se většina pozornosti zaměřuje na samotný kód. Databázový model je vytvořený pomocí kódu - vytváří objekty a z těch se automaticky vytvoří databáze. A právě *Code First* byl použit pro tuto databázi.[6]

Z přístupu *Code First* se pak dostáváme k architektuře psaní kódu a tou je Model-View-ViewModel (MVVM). Jedná se o rozdělení aplikace na vrstvy dle jejich funkčnosti. Jedna vrstva má na starost komunikaci s uživatelem. Obstarává tedy grafickou stránku aplikace a předává požadavky uživatele hlouběji do aplikace. Tato vrstva se nazývá Prezentační vrstva. Druhou vrstvu samotný uživatel neuvidí. Jedná se Bussiness vrstvu, která zajišťuje samotný chod aplikace. [7]



Obr. 8 Model-View-ViewModel [7]

V případě databáze aplikace D2APS byly vytvořeny dva samostatné projekty. První projekt, se stejným pojmenováním jako aplikace, je pro prezentační vrstvu, Viewmodel a plánovací algoritmus. Druhý projekt, který se nazývá DataEntity, zajišťuje komunikaci aplikace s databází. Projekt DataEntity tedy představuje Model na Obr. 8.

Nejprve jsou představeny techniky v projektu DataEntity. Pro lepší orientaci se zde uvádí kód každé třídy i s vysvětlením jednotlivých fragmentů. Nejprve je nutné uvést základní třídu, která sama o sobě negeneruje žádnou tabulku. Základní třída, zvaná `BaseModel`, obsahuje údaje

nebo vlastnosti, které by měly mít všechny ostatní třídy. Ostatní třídy získají tyto údaje nebo vlastnosti tak, že základní třídu budou dědit.

```
public abstract class BaseModel : ViewModelBase, IDataErrorInfo
{
    "validace"

    [Timestamp]
    public byte[] RowVersion { get; set; }

}
```

Klíčové slovo "validace" označuje skrytý blok kódu věnovaný funkčním příkazům, které zajišťují ověření pomocí datových anotací, jako například omezení délky textového řetězce u atributu `Nazev` v tabulce strojů (viz níže).

Důležitá v základní třídě je vlastnost `RowVersion`, která dědí všechny tabulkové třídy. Tento atribut je typu `Timestamp`, tedy časová značka. Díky této vlastnosti bude každý záznam opatřen časovou značkou, což je zásadní pro víceuživatelský přístup k databázi. [7] Ten sice není implementován, ale vlastnost je k dispozici pro případné rozšíření aplikace tímto směrem.

Další kód je určen pro nejjednodušší tabulku a to tabulku strojů.

```
public partial class Stroj : BaseModel
{
    [Key]
    [Column(Order = 1)]

    public int StrojId { get; set; }

    [Required(ErrorMessage = "Název stroje je povinné pole")]
    [StringLength(50)]

    public string Nazev { get; set; }

    public int CasPresunu { get; set; }

    public virtual ObservableCollection<Operace> Operace { get; set; }

}
```

Hned v hlavičce je vidět, že třída `Stroj` dědí vlastnosti od základní třídy `BaseModel`. Třída `Stroj` tedy získá funkční příkazy a vlastnost `RowVersion`. Stejně tak se bude dít i u tříd `Operace` a `Zakazka`.

Jako první vlastnost lze vidět `StrojId`, což představuje primární celočíselný klíč pro danou tabulku – identifikaci pro databázi. Skutečnost, že se jedná o klíč, potvrzuje `[Key]`. `[Column(Order = 1)]` zajišťuje pozici klíče v prvním sloupci tabulky.

Další vlastnost v řadě je `Nazev`. Jedná se o vlastnost pro název stroje, která pomáhá uživateli, aby daný stroj identifikoval. Jedná se samozřejmě o textovou vlastnost. Jedná se o povinný

údaj, který může být dlouhý maximálně 50 znaků. V případě nevyplnění údaje v aplikaci se zobrazí varovná hláška "Název stroje je povinné pole".

Třetí vlastnost CasPresunu charakterizuje průměrnou dobu přesunu materiálu od tohoto stroje k jinému. Jedná se o celočíselnou vlastnost a hodnoty jsou v sekundách.

Poslední vlastnost představuje vazbu na tabulku Operace.

V hlavičce třídy `Stroj` můžeme vidět klíčové slovo `partial`, které nám říká, že jde o částečnou třídu. Tedy třídu, která může být rozdělena mezi dva nebo více zdrojových souborů. Rozdělení se většinou provádí v rámci přehlednosti, pokud ve třídě koexistuje více logických úrovní.[8] Třída `Stroj` je rozdělena na dvě části. První část definuje tabulku a druhá část řeší grafické zobrazení výsledků plánovacího algoritmu. Se samotnou databází souvisí okrajově, a proto zde druhá část nebude uváděna.

Další kód je určen pro tabulku operací:

```
public partial class Operace : BaseModel
{
    [Key]
    [Column(Order = 1)]

    public int OperaceId { get; set; }

    public int ZakazkaId { get; set; }

    [ForeignKey("ZakazkaId")]
    public virtual Zakazka Zakazka { get; set; }

    public int? PristiOperaceId { get; set; }
    [ForeignKey("PristiOperaceId")]
    public virtual Operace PristiOperace { get; set; }
    [InverseProperty("PristiOperace")]
    public virtual ObservableCollection<Operace> PodVyrobniPrikazy { get; set; }

    public int StrojId { get; set; }

    [ForeignKey("StrojId")]
    public virtual Stroj Stroje { get; set; }

    [Column(TypeName = "time")]
    public TimeSpan CasKusovy { get; set; }

    [Column(TypeName = "time")]
    public TimeSpan CasDavkovy { get; set; }

    public int PocetKs { get; set; }

    public double Trvani
    {
        get
        {
            return CasDavkovy.TotalSeconds + (CasKusovy.TotalSeconds * PocetKs);
        }
    }
}
```

```
public DateTime Start { get; set; }

public DateTime Konec
{
    get
    {
        return Start.AddSeconds(Trvani);
    }
}
```

První vlastnost `OperaceId` je opět primární celočíselný klíč tabulky, což potvrzuje příkaz `[Key]`. Příkaz `Column(Order = 1)` zajišťuje, že `OperaceId` bude umístěna v prvním sloupci v databázi.

Jako druhá vlastnost je `ZakazkaId`, která představuje cestu spojení pro tabulky operací a zakázek, tedy cizí klíč. `ZakazkaId` nabývá celočíselných hodnot a určuje u každé operace zakázku, pro kterou operace vznikla.

Třetí vlastnost je `PristiOperaceId`. Jedná se o nepovinný celočíselný cizí klíč, který poukazuje na atribut `OperaceId` u jiné operace. Tedy propojuje různé záznamy z téže tabulky. Důvod vzniku takového cizího klíče lze odhadnout z jeho jména – jednoznačně určuje id operace, která čeká na výrobek této operace.

I čtvrtá vlastnost je cizí klíč. Jedná se o `StrojId`, který propojuje tabulky operací a strojů. Jeho úkolem je určit pro každou operaci, na kterém stroji bude prováděna.

Pátá vlastnost je `CasKusovy`. Tato vlastnost nabývá časových hodnot, je tedy zadávána ve formátu „00:00:00“. `CasKusovy` charakterizuje čas potřebný na výrobu jednoho kusu (`tA`).

`CasDavkovy`, šestá vlastnost tabulky `Operace`, funguje na stejném principu jako `CasKusovy`. Je to časová vlastnost s formátem „00:00:00“. Zastupuje čas potřebný při přípravě celé dávky kusů (`tB`).

Sedmá vlastnost tabulky pro operace je `PocetKs`. `PocetKs` je celočíselné vyjádření, kolik kusů se v operaci bude zpracovávat.

Osmá vlastnost se neukládá do databáze, ale vždy pokud je potřeba, se vypočítá. Jedná se o `Trvani`. Nabývá číselných hodnot s dvoumístnou přesností za desetinou čárkou. Výpočet atributu je podle vzorce:

$$T = tB + tA * ks$$

kde: T = čas trvání operace

tB = čas dávkový

tA = čas kusový

ks = počet zpracovávaných kusů

Devátá vlastnost je *start*. Je to datum a je zapisován ve formátu „MM/dd/yy HH:mm:ss“. Jeho defaultní hodnota není důležitá, skutečný význam získává tato vlastnost až po použití plánovacího algoritmu.

Desátá vlastnost je *konec*. Jedná se opět o datum ve stejném formátu. Stejně jako u vlastnosti *Trvani*, ani *konec* se neukládá do databáze, ale v případě potřeby se vypočte přičtením trvání operace k času zahájení operce (*Start*).

Hlavička třídy *Operace* obsahuje klíčové slovo *partial* stejně jako u třídy *Stroj*. Je to i ze stejného důvodu. Druhá část třídy *Operace* je věnována grafickému zobrazení výsledků plánovacího algoritmu.

	Operaceld	Zakazkald	PristiOperaceld	Strojld	CasKusovy	CasDavkovy	PocetKs	Start	RowVersion
1	1	1	NULL	3	00:02:10	00:01:30	21	1.1.1970 0:00:00	0x000000000000007D6
2	1	1	1	1	00:01:17	00:04:29	21	1.1.1970 0:00:00	0x000000000000007D7
3	1	1	1	1	00:02:39	00:03:49	21	1.1.1970 0:00:00	0x000000000000007D8
4	1	2	2	4	00:02:00	00:06:07	21	1.1.1970 0:00:00	0x000000000000007D9
5	1	1	1	4	00:02:32	00:02:07	21	1.1.1970 0:00:00	0x000000000000007DA
6	1	2	2	1	00:03:09	00:02:45	21	1.1.1970 0:00:00	0x000000000000007DB
7	1	3	3	4	00:03:18	00:02:50	21	1.1.1970 0:00:00	0x000000000000007DC
8	1	6	6	4	00:01:07	00:07:06	21	1.1.1970 0:00:00	0x000000000000007DD
9	1	4	4	3	00:02:20	00:02:39	21	1.1.1970 0:00:00	0x000000000000007DE
10	1	1	1	1	00:02:29	00:04:12	21	1.1.1970 0:00:00	0x000000000000007DF
11	2	2	NULL	3	00:02:10	00:01:30	12	1.1.1970 0:00:00	0x000000000000007E1
12	2	2	11	1	00:00:37	00:05:43	12	1.1.1970 0:00:00	0x000000000000007E2
13	2	2	11	1	00:02:20	00:05:18	12	1.1.1970 0:00:00	0x000000000000007E3
14	2	2	12	1	00:01:03	00:05:13	12	1.1.1970 0:00:00	0x000000000000007E4
15	2	2	11	2	00:01:26	00:06:49	12	1.1.1970 0:00:00	0x000000000000007E5
16	2	2	11	3	00:02:47	00:03:25	12	1.1.1970 0:00:00	0x000000000000007E6
17	2	2	12	2	00:00:56	00:02:24	12	1.1.1970 0:00:00	0x000000000000007E7
18	2	2	12	2	00:03:01	00:06:01	12	1.1.1970 0:00:00	0x000000000000007E8
19	2	2	12	2	00:02:15	00:05:19	12	1.1.1970 0:00:00	0x000000000000007E9
20	2	2	13	1	00:03:09	00:03:01	12	1.1.1970 0:00:00	0x000000000000007EA
21	2	2	11	1	00:01:08	00:06:21	12	1.1.1970 0:00:00	0x000000000000007EB
22	2	2	12	3	00:00:51	00:05:32	12	1.1.1970 0:00:00	0x000000000000007EC
23	2	2	19	4	00:02:25	00:02:28	12	1.1.1970 0:00:00	0x000000000000007ED
24	2	2	11	4	00:01:44	00:07:35	12	1.1.1970 0:00:00	0x000000000000007EE
25	2	2	19	4	00:01:45	00:02:30	12	1.1.1970 0:00:00	0x000000000000007EF
26	2	2	17	2	00:02:25	00:01:27	12	1.1.1970 0:00:00	0x000000000000007F0
27	2	2	14	2	00:01:55	00:07:13	12	1.1.1970 0:00:00	0x000000000000007F1
28	2	2	12	2	00:01:29	00:05:29	12	1.1.1970 0:00:00	0x000000000000007F2
29	3	3	NULL	3	00:02:10	00:01:30	19	1.1.1970 0:00:00	0x000000000000007F4
30	3	3	29	1	00:01:58	00:05:17	19	1.1.1970 0:00:00	0x000000000000007F5
31	3	3	29	2	00:00:37	00:01:22	19	1.1.1970 0:00:00	0x000000000000007F6
32	3	3	29	4	00:02:24	00:03:47	19	1.1.1970 0:00:00	0x000000000000007F7
..

Obr. 9 Ukázka tabulky Operace

Na Obr. 9 si lze všimnout jednotného data ve sloupci *Start*. Jedná se o výše zmíněné defaultní nastavení pro každou operaci před spuštěním plánovacího algoritmu. Ten pak tento údaj přepisuje, na aktuální hodnoty. Důvodem zavedení této defaultní hodnoty je skutečnost, že proměnná je deklarovaná jako *DateTime*, který nepovoluje prázdnou hodnotu data. Alternativa deklarovat *Start* jako *DateTime?*, která prázdnou hodnotu data povoluje není možná, protože *DateTime?* nepodporuje stejné funkce jako *DateTime*. Defaultní datum bylo zvoleno tak, aby bylo jasně rozeznatelné od skutečných začátků operací.

Následující kód je pro tabulku zakázek.

```
public class Zakazka : BaseModel
{
    [Key]
    [Column(Order = 1)]
```

```
public int ZakazkaId { get; set; }  
public string Zakaznik { get; set; }  
public DateTime Deadline { get; set; }  
public int Pocet { get; set; }  
public virtual ObservableCollection<Operace> Operace { get; set; }  
}
```

První vlastnost `ZakazkaId` je opět primární celočíselný klíč tabulky. Jako předchozí primární klíče mu jeho místo zajišťují `[Key]` a `[Column(Order = 1)]`.

Druhá vlastnost `zakaznik` je ve formě textového řetězce. Přenáší informaci o zákazníkovi (jméno zákazníka nebo jeho označení), převážně pro lepší orientaci uživatele.

V pořadí třetí vlastnost tabulky zakázek je `Deadline`, který je typu `DateTime`, tedy datum. Určuje čas, kdy nejpozději musí být vyrobený, zabalený nebo zkontrolovaný (záleží na typu poslední operace) poslední výrobek zakázky, aby se stihla zakázka doručit zákazníkovi včas.

Čtvrtá vlastnost je `Pocet`. Jedná se o počet kusů finálního výrobku, který zákazník požaduje.

Poslední vlastnost vytváří vazbu mezi tabulkou zakázek a tabulkou strojů.

	ZakazkaId	Zakaznik	Deadline	Pocet	RowVersion
	1	A	11.4.2019 17:25:00	1	0x00000000000007D5
	2	B	10.4.2019 12:35:00	1	0x00000000000007E0
	3	C	11.4.2019 13:25:00	2	0x00000000000007F3
	4	D	10.4.2019 12:10:00	1	0x0000000000000801
	5	E	12.4.2019 20:30:00	1	0x0000000000000809
	6	F	11.4.2019 17:55:00	1	0x0000000000000819
	7	G	12.4.2019 12:35:00	1	0x0000000000000831
	8	H	11.4.2019 8:15:00	1	0x0000000000000842
	9	I	12.4.2019 2:35:00	1	0x000000000000084D
	10	J	12.4.2019 13:25:00	1	0x0000000000000854
	11	K	11.4.2019 15:25:00	1	0x0000000000000868
	12	L	12.4.2019 9:35:00	1	0x0000000000000870
	13	M	11.4.2019 13:55:00	2	0x000000000000087D
	14	N	11.4.2019 12:40:00	1	0x000000000000088E
	15	O	11.4.2019 20:00:00	1	0x000000000000089D
	16	P	11.4.2019 17:45:00	1	0x00000000000008A8
	17	Q	11.4.2019 12:15:00	1	0x00000000000008C0
	18	R	11.4.2019 8:35:00	1	0x00000000000008CB
	19	S	11.4.2019 4:35:00	1	0x00000000000008DB

Obr. 10 Ukázka tabulky Zakazky

5.1.4 Prezentací vrstva

V prvním projektu D2APS je řešena prezentací vrstva aplikace a k tomu odpovídající Viewmodel. Prezentací vrstva je rozdělena na čtyři části, které jsou odvozené od funkční analýzy (kap. 5.1.1 Funkční analýza aplikace). Každá část má vlastní třídu pro Viewmodel (které všechny dědí vlastnosti ze základní třídy pro Viewmodel).

První a hlavní část je okno aplikace, které obsahuje základní ovládání aplikace a prostor pro zobrazování dalších částí: strojí a operační. Tyto části jsou ve formě stránek, které se pak vloží do prostoru v okně.

Strojní stránka zobrazuje údaje o strojích a umožňuje uživateli editovat a mazat stávající stroje či vytvářet stroje nové.

Operační stránka pak pouze zobrazuje údaje o operacích.

Poslední část je zobrazení plánování. Plánování se zobrazuje v samostatném okně. To zajišťuje, že může být vygenerován libovolný počet plánů a ty pak mezi sebou porovnávány. Oken totiž může mít aplikace více, ale stránka je vždy aktivní pouze jedna.

Všechny části jsou naprogramované ve WPF (Windows Presentation Foundation), což je způsob vytváření okenních aplikací pro C# v prostředí Visual Studio. WPF je programováno pomocí značkovacího jazyku XAML. Pro ukázkou je zde část kódu pro stránku operací:

```
...
<DataGrid Name="dgOperace" ItemsSource="{Binding OperaceCol}"
    HeadersVisibility="Column" SelectionMode="Single" AutoGenerateColumns="False"
    IsReadOnly="True">
  <DataGrid.Columns>
    <DataGridTextColumn Header="ID" Width="Auto"
        Binding="{Binding OperaceId}"/>
    <DataGridTextColumn Header="Počet kusů" Width="Auto"
        Binding="{Binding PocetKs}"/>
    <DataGridTextColumn Header="Zakázka" Width="Auto"
        Binding="{Binding Zakazka.Zakaznik}"/>
  </DataGrid.Columns>
</DataGrid>
...
```

`DataGrid` je způsob jak zobrazit data do mřížky. `DataGrid.Columns` je označení, že následující kód se bude týkat sloupců pro komponentu `DataGrid`. `DataGridTextColumn` je pak sloupec pro `DataGrid`, jehož obsah je text. Ostatní část kódu jsou vlastnosti, z kterých nejdůležitější je `ItemsSource`. Tato vlastnost říká, že veškerá data pro daný `DataGrid` se mají čerpat z `OperaceCol`, což je kolekce operací.

Tato kolekce nicméně není přítomná v samotné prezentací vrstvě. Kód k vytvoření této kolekce je ve třídě, která představuje Viewmodel pro stránku operací. Tento kód naplní kolekci `OperaceCol` daty z tabulky `Operace`, která byla ukázána výše.

Pokud tedy uživatel chce zobrazit stránku operací, tak si vytvoří instanci třídy pro Viewmodel, která naplní kolekci operací daty z tabulky `Operace`, která tato data čerpá z databáze. Tedy do zobrazení jsou zapojeny všechny části z Obr. 8.

5.1.5 Algoritmus plánování

Realizovaný algoritmus plánování probíhá podle kombinovaného plánování APS. Algoritmus pracuje s následujícími omezeními:

- kapacita
- tzv. „deadline“ – čas, do kterého musí být výrobek hotov
- priority

Z důvodu časové náročnosti zanedbává algoritmus některé klasické aspekty plánování. Mezi nejmarkantnější zjednodušení patří zanedbání dostupnosti materiálu a rozpracovanosti výroby. Je zde také zanedbání směnnosti. Předpokládá se nepřetržitá výroba 24 hodin denně. Algoritmus není ani speciálně vyladěn na nejvyšší možnou rychlost. Což ovšem neznamená, že by plánování tisíce operací trvalo neúměrně dlouho dobu. Rychlost zpracování se obvykle provádí v řádu setin vteřin až jednotek vteřin. Algoritmus byl také navržen tak, aby byl adaptabilní v počtu zadávaných parametrů, jako jsou počty strojů, operací, zakázek nebo třeba trvání operací. Dalším omezením je, že algoritmus pracuje se zakázkami, které požadují jen 1 typ produktu. Ovšem jeho množství může být jakékoli. V případě zakázky na 2 různé produkty by pak bylo třeba vytvořit v databázi 2 zakázky se stejným deadline. Algoritmus také zanedbává možnost spojení 2 operací, které by vyráběli totéž, do jedné větší operace.

5.1.6 Fungování algoritmu

Jak již bylo výše zmíněno, algoritmus probíhá dle kombinovaného plánování, které bylo popsáno v kap. 2.2.2 APS s kombinovaným plánováním. Kombinované plánování v algoritmu funguje tak, že dopředné plánování má dvě části a zpětné plánování má jednu část.

První část dopředného plánování je určení priorit zakázek, podle jejich naléhavosti. Naléhavost je vyjádřena deadline každé zakázky. Aplikace ponechává prostor pro jednoduché doprogramování možnosti zadat priority zakázek ručně.

Druhá část dopředného plánování je dynamické určení priorit jednotlivých operací uvnitř právě plánované zakázky. Algoritmus vyhledává nejvhodnější operaci k naplánování tím, že jednotlivých operacím přeřazuje priority. Priority jsou opět přiřazovány dle naléhavosti jednotlivých operací. Jelikož následujícím krokem je provedení zpětného plánování, algoritmus bere v potaz pouze operace, jejichž předchůdci v plánovacím systému už byli naplánováni.

Zde je třeba upozornit na skutečnost, že pokud máme operaci A a jejího předchůdce v plánovacím systému, operaci B, pak v reálné výrobě musí být nejdříve provedena operace A, jejíž výrobek pak bude nadále zpracováván v operaci B.

Poslední částí algoritmu je zpětné plánování, kdy vybrané operaci vybrané zakázky je přiřazen adekvátní čas na pracovišti a do kapacitního plánu pracoviště je zaznamenán tento čas jako využitý.

Pro popsání funkčnosti algoritmu je nutné stanovit několik indexů, parametrů a proměnných.

Indexy:

i, k index produktu

j, l index operace

Parametry:

$task(i, j)$ j -tá operace i -tého výrobku

$machine(i, j)$ stroj pro j -tou operace i -tého výrobku

$t_{ks}(i, j)$ čas kusový operace $task(i, j)$

$t_{da}(i, j)$ čas dávkový operace $task(i, j)$

T_b časový buffer

q_i požadované množství výrobku i

d_i deadline i -tého produktu

d_{last} deadline produktu s nejmenší prioritou

C_i celkový výrobní čas i -tého produktu

C_{ij} výrobní čas j -té operace i -tého produktu dohromady se všemi následujícími závislými operacemi

p_i priorita i -tého produktu

p_{ij} priorita j -té operace i -tého produktu

z_{ijl} nabývá hodnot 0 a 1, hodnotu 1 má tehdy, pokud operaci $task(i, j)$ předchází operace $task(i, l)$, jinak má hodnotu 0

Proměnné:

r_{ij} doba uvolnění j -tého výrobku i -tého produktu z výroby

C_{max} časový úsek od začátku plánu po konec plánu

L_i pozdní dodání produktu

E_i brzké zhotovení produktu

Pro znázornění algoritmu je třeba dodat, že čas, od kterého se začíná plán rozvrhovat, má na časové ose hodnotu 0. Pak tedy například tvrzení, že

$$task(i, j).start \geq task(k, l).start$$

znamená, že $task(i, j).start$ je na časové ose dál od nuly, než $task(k, l).start$, neboli $task(i, j).start$ nastává později než $task(k, l).start$.

Přičemž základní proměnná je $task(i, j).start$, počáteční čas operace. Následuje trvání operace ($task(i, j).duration$). Čas ukončení operace je pak označováno jako $task(i, j).end$. Zatímco počáteční časy operací jsou výstupní data z algoritmu, trvání operací a čas ukončení jsou vypočítávány během procesu algoritmu, takže mezi výstupní data nepatří.

Zde je matematické popsání problému:

$$\text{Min } Z = \sum_{i=1} L_i + E_i \quad (1)$$

s následujícími omezeními:

$$r_{ij} = \text{task}(i, l).end + \text{machine}(i, l).travel, \text{ pro } \forall i, j, l \in \{i, j, l \mid z_{ijl} = 1\} \quad (2)$$

$$\text{task}(i, j).start \geq r_{ij}, \text{ pro } \forall i, j \quad (3)$$

$$\text{task}(i, j).duration = t_{ks}(i, j) * q_i + t_{da}(i, j), \text{ pro } \forall i, j \quad (4)$$

$$\text{task}(i, j).start = \text{task}(i, j).end - \text{task}(i, j).duration \text{ pro } \forall i, j \quad (5)$$

$$\text{task}(i, j).end \leq C_{max}, \text{ pro } \forall i, j \quad (6)$$

$$d_{last} \geq C_{max}, \text{ pro } \forall i \quad (7)$$

$$\text{task}(i, j).start + \text{task}(i, j).duration \leq \text{task}(k, l).start$$

nebo

$$\text{task}(k, l).start + \text{task}(k, l).duration \leq \text{task}(i, j).start, \\ \text{pro } \forall i, j, k, l \in \{i, j, k, p \mid \text{task}(i, j) \neq \text{task}(k, l), \text{ machine}(i, j) = \text{machine}(k, l)\} \quad (8)$$

$$p_i > p_k \quad \text{pro}$$

$$\forall i, j, k, l \in \{i, j, k, l \mid d_i - T_b - \sum_{j=1} \text{task}(i, j).duration \leq \\ d_k - T_b - \sum_{l=1} \text{task}(k, l).duration\} \quad (9)$$

$$p_{ij} > p_{il}, \text{ pro } \forall i, j, l \in \{i, j, l \mid C_{ij} > C_{il}\} \quad (10)$$

$$L_i = d_i - T_b - \text{task}(i, last).end \\ \text{pro } \forall i \in \{i \mid d_i - T_b < \text{task}(i, last).end\} \quad (11)$$

$$E_i = d_i - T_b - \text{task}(i, \text{last}).\text{end}$$

$$\text{pro } \forall i \in \{i | d_i - T_b > \text{task}(i, \text{last}).\text{end}\} \quad (12)$$

Cíl algoritmu (1) je eliminovat pozdní dodávky produktů a zároveň snížit skladování hotových výrobků na minimum.

Rovnice (2) vyjadřuje výpočet doby uvolnění pro j-tou operaci ($\text{task}(i,j)$), které v plánovacím systému předchází l-tá operace ($\text{task}(l,j)$).

Nerovnice (3) pak říká, že čas počátku operace nesmí být menší, než doba uvolnění do výroby materiálu potřebného k této operaci.

Rovnice (4) definuje výpočet trvání operace.

Rovnice (5) uvádí výpočet času zahájení operace.

Nerovnice (6) ukazuje, že doba dokončení každé operace by měla být menší nebo rovna než celkový výrobní čas.

Omezení (7) pak dodává, že celková výroba by měla skončit dříve, než je deadline produktu s nejmenší prioritou.

Soustava nerovnic (8) poukazuje na to, že žádné dvě různé operace nemohou být zpracovávány na stejném stroji ve stejnou dobu.

Nerovnice (9) ukazuje, že ze dvou produktů má větší prioritu ten, jenž má méně manipulačního času. Manipulačním časem se rozumí čas, který získáme, když odečteme od deadline časový buffer a součet trvání všech operací na produktu.

Nerovnice (10) říká, že operace s větším celkovým operačním výrobním časem má větší prioritu.

Rovnice (11) definuje velikost pozdního dodání produktu. Pozdní dodání produktu nabývá hodnoty pouze tehdy pokud deadline zkrácený o časový buffer je menší než čas ukončení finální operace produktu (v případě více finálních operací se jedná o tu finální operaci, která končí nejpozději).

Na druhou stranu rovnice (12) pak definuje velikost brzkého dodání produktu, který nabývá hodnoty tehdy, pokud deadline zkrácený o časový buffer je větší než čas ukončení finální operace produktu.

5.2 Uživatelský pohled

Po představení vnitřní stavby je čas představit prezentační vrstvu z pohledu uživatele. Prezentační vrstva aplikace představuje komunikační kanál uživatele s aplikací, proto výsledek musí být intuitivní a přehledné uživatelské rozhraní.

Konkrétně se jedná o zobrazení dat z jednotlivých databázových tabulek, zobrazení výsledných plánů výroby a prezentační vrstva editace tabulky Stroje.

5.2.1 Databáze v D2APS



Obr. 11 Úvodní stránka aplikace

Na Obr. 11 lze vidět úvodní obrazovku. Nahoře v záhlaví aplikace se nachází čtyři tlačítka. Stisknutím prvního tlačítka zleva se uživatel dostane k sekci strojů, druhé tlačítko je pro přehled operací, třetí je pro spuštění plánovacího algoritmu a grafického znázornění plánu. Poslední tlačítko je na druhé straně než ostatní. Slouží pro uložení provedených změn, například pro uložení naplánovaných operací, kterým se změnil čas startu.

Jen na okraj - grafické logo vychází ze jména aplikace D2APS a s mottem „*Plánování s D2APS je jistější nežli smrt*“.

ID	POČET KUSŮ	ZAKÁZKA	ČAS JEDNOTKOVÝ	ČAS DÁVKOVÝ	TRVÁNÍ	PŘÍSTÍ	ZAČÁTEK	KONEC	STROJ
1	21	A	00:02:10	00:01:30	2820		4/10/2019 2:08:35 AM	4/10/2019 2:55:35 AM	Vrtačka R12
2	21	A	00:01:17	00:04:29	1886	1	4/10/2019 1:36:19 AM	4/10/2019 2:07:45 AM	Soustruh KLX
3	21	A	00:02:39	00:03:49	3568	1	4/10/2019 12:36:51 AM	4/10/2019 1:36:19 AM	Soustruh KLX
4	21	A	00:02:00	00:06:07	2887	2	4/10/2019 12:46:32 AM	4/10/2019 1:34:39 AM	Obrážka Hivo
5	21	A	00:02:32	00:02:07	3319	1	4/9/2019 10:27:44 PM	4/9/2019 11:23:03 PM	Obrážka Hivo
6	21	A	00:03:09	00:02:45	4134	2	4/9/2019 11:27:57 PM	4/10/2019 12:36:51 AM	Soustruh KLX
7	21	A	00:03:18	00:02:50	4328	3	4/9/2019 11:23:03 PM	4/10/2019 12:35:11 AM	Obrážka Hivo
8	21	A	00:01:07	00:07:06	1833	6	4/9/2019 8:57:46 PM	4/9/2019 9:28:19 PM	Obrážka Hivo
9	21	A	00:02:20	00:02:39	3099	4	4/9/2019 11:52:43 PM	4/10/2019 12:44:22 AM	Vrtačka R12
10	21	A	00:02:29	00:04:12	3381	1	4/9/2019 10:31:36 PM	4/9/2019 11:27:57 PM	Soustruh KLX
11	12	B	00:02:10	00:01:30	1650		4/10/2019 11:05:20 AM	4/10/2019 11:32:50 AM	Vrtačka R12
12	12	B	00:00:37	00:05:43	787	11	4/10/2019 10:51:23 AM	4/10/2019 11:04:30 AM	Soustruh KLX
13	12	B	00:02:20	00:05:18	1998	11	4/10/2019 10:18:05 AM	4/10/2019 10:51:23 AM	Soustruh KLX
14	12	B	00:01:03	00:05:13	1069	12	4/10/2019 10:00:16 AM	4/10/2019 10:18:05 AM	Soustruh KLX
15	12	B	00:01:26	00:06:49	1441	11	4/10/2019 7:56:34 AM	4/10/2019 8:20:35 AM	Frézka HB 204
16	12	B	00:02:47	00:03:25	2209	11	4/10/2019 10:26:21 AM	4/10/2019 11:03:10 AM	Vrtačka R12
17	12	B	00:00:56	00:02:24	816	12	4/10/2019 10:03:28 AM	4/10/2019 10:17:04 AM	Frézka HB 204
18	12	B	00:03:01	00:06:01	2533	12	4/10/2019 9:21:15 AM	4/10/2019 10:03:28 AM	Frézka HB 204
19	12	B	00:02:15	00:05:19	1939	12	4/10/2019 10:17:04 AM	4/10/2019 10:49:23 AM	Frézka HB 204

Obr. 12 Přehled výrobních operací

Na Obr. 12 je vidět přehled výrobních operací připravených pro plánování.

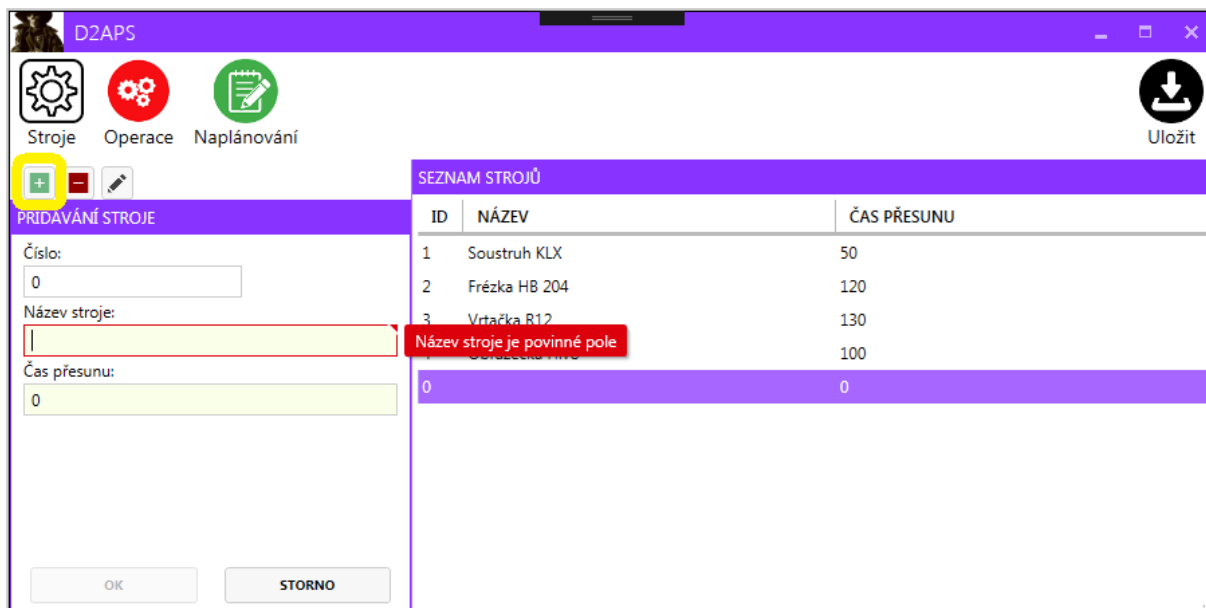
ID	NÁZEV	ČAS PŘESUNU
1	Soustruh KLX	50
2	Frézka HB 204	120
3	Vrtačka R12	130
4	Obrážka Hivo	100

Obr. 13 Části stránky

Na Obr. 13 je vidět první pohled na stránku strojů. Takto stránka strojů vypadá, pokud na ní uživatel právě přešel.

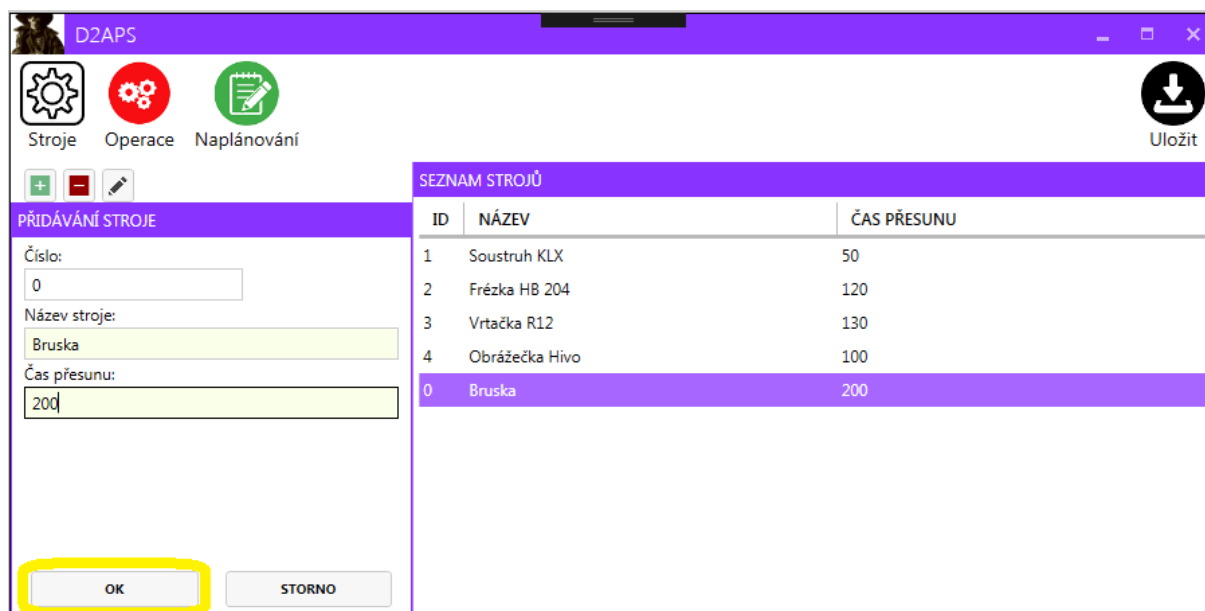
Stránka strojů je rozdělena na dvě části, jelikož stroje je možné v aplikaci upravovat. První část je pro editaci tabulky strojů. Druhá část, nazývaná *Seznam strojů*, zobrazuje současný stav databázové tabulky. Aplikace tak demonstruje svoji možnost vytvářet, upravovat a mazat data z databáze. Na Obr. 13 je první část zvýrazněna žlutým rámečkem a druhá červeným rámečkem.

Stránka zdrojů funguje ve třech módech *Prohlížení strojů*, *Přidávání stroje* a *Editace stroje*, které ovlivňují hlavně první část stránky strojů, které zde budou blíže představeny.



Obr. 14 Nový stroj

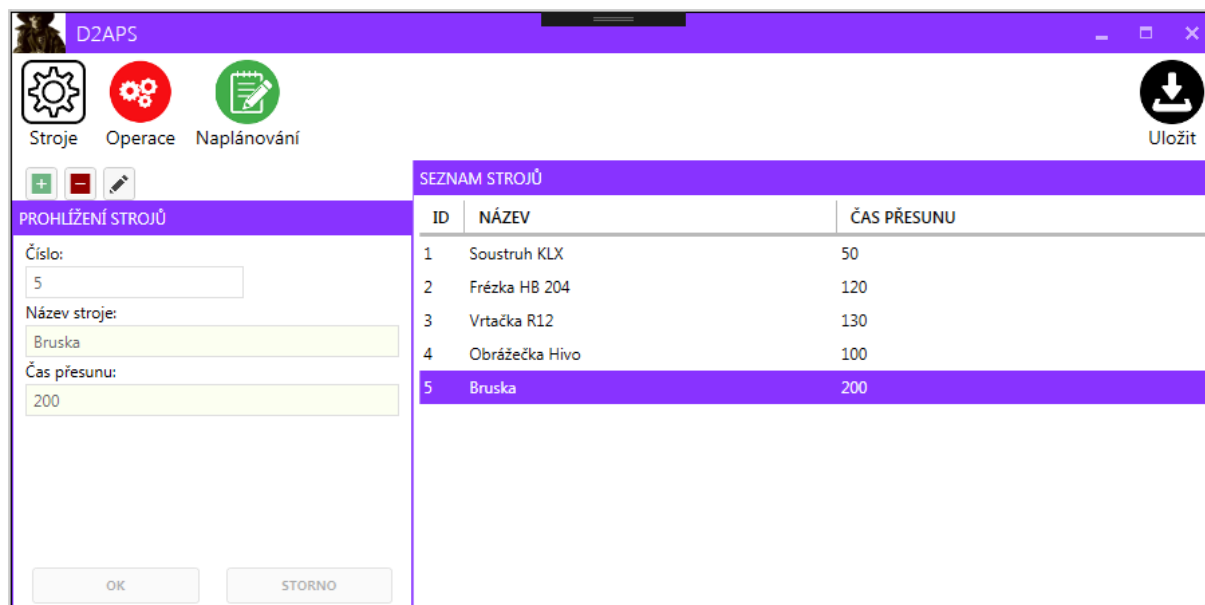
Základní mód je *Prohlížení*. Tento mód je aktivní pokud uživatel neprovádí žádné změny v databázi nebo pokud ty změny právě dokončil. Pokud chce uživatel přidat další stroj, musí stisknout ikonu přidání, tedy malé „+“ v zeleném poli. Na Obr. 14 je ikona zvýrazněna žlutým rámečkem. Tím se mu umožní vstoupit do módu *Přidávání stroje* a zpřístupní se mu *Editace stroje* část stránky. Na Obr. 14 můžete zároveň vidět upozornění, které bylo zadáno v kódu pro tabulku strojů v kapitole 5.1.3 *Techniky databáze*.



Obr. 15 Přidání nového stroje

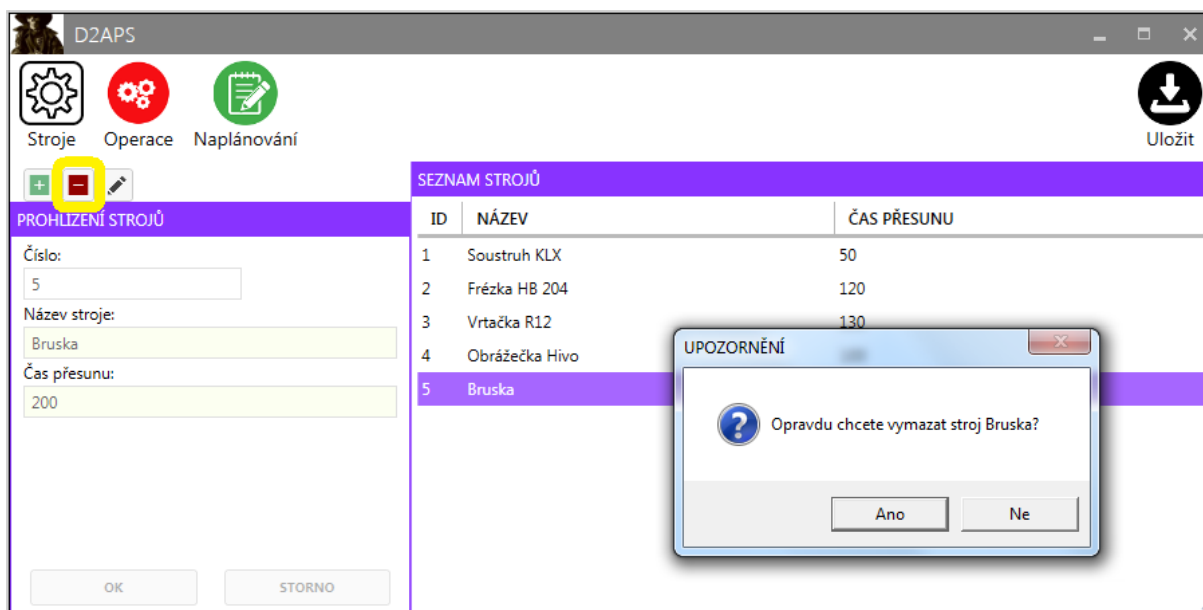
Na Obr. 15 lze vidět, že po vyplnění požadovaných informací o novém stroji, které odpovídají kontrole zabudované v kódu, se zvýrazní tlačítko *OK*. Zvýraznění signalizuje, že už je možné nový stroj potvrdit a uložit stisknutím tlačítkem *OK* do databáze. Zadávané parametry zde jsou název stroje a čas přesunu. Název stroje musí uživatel zadat, čas přesunu je pak defaultně stanoven na nulovou hodnotu. Uživatel ho může i nemusí měnit. Identifikační číslo, které při zadávání nové položky zobrazuje nulu, je uživatelsky nepřepisovatelné a aplikace přiřazuje následně číslo id sama.

Pokud by si uživatel přidání materiálu rozmyslel, může se vrátit do režimu *Prohlížení strojů* tlačítkem *Storno*.



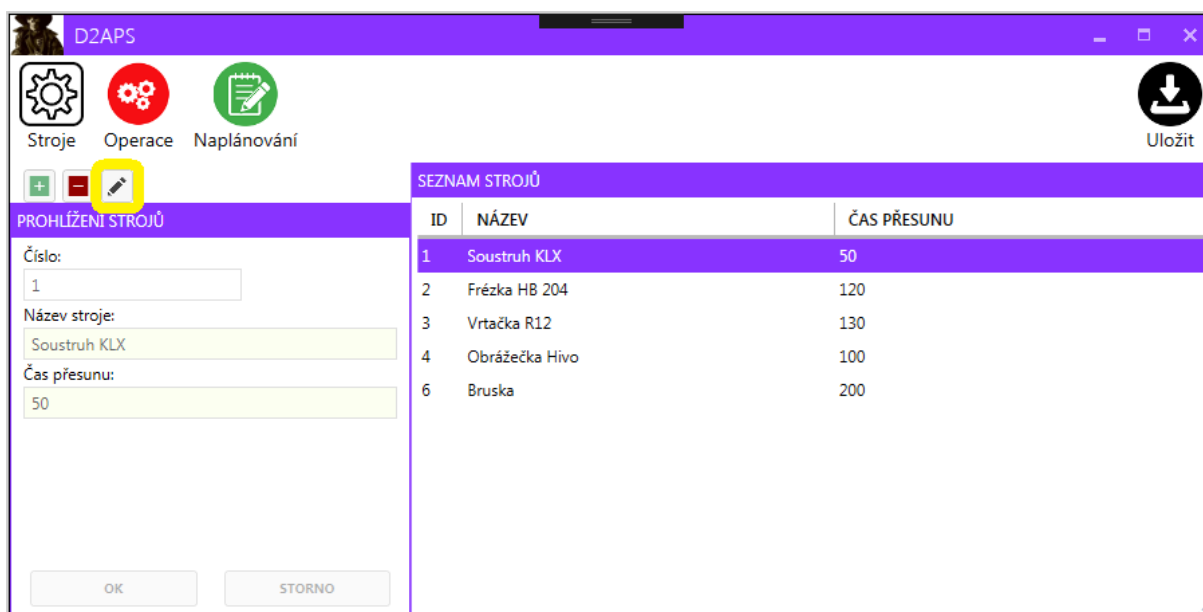
Obr. 16 Přidělení Id

Po stisknutí tlačítka *OK* se do databáze uloží nová položka a přisoudí se jí identifikační číslo, jak lze vidět na Obr. 16. Zároveň se přejde zpět do módu *Prohlížení strojů*.



Obr. 17 Mazání stroje

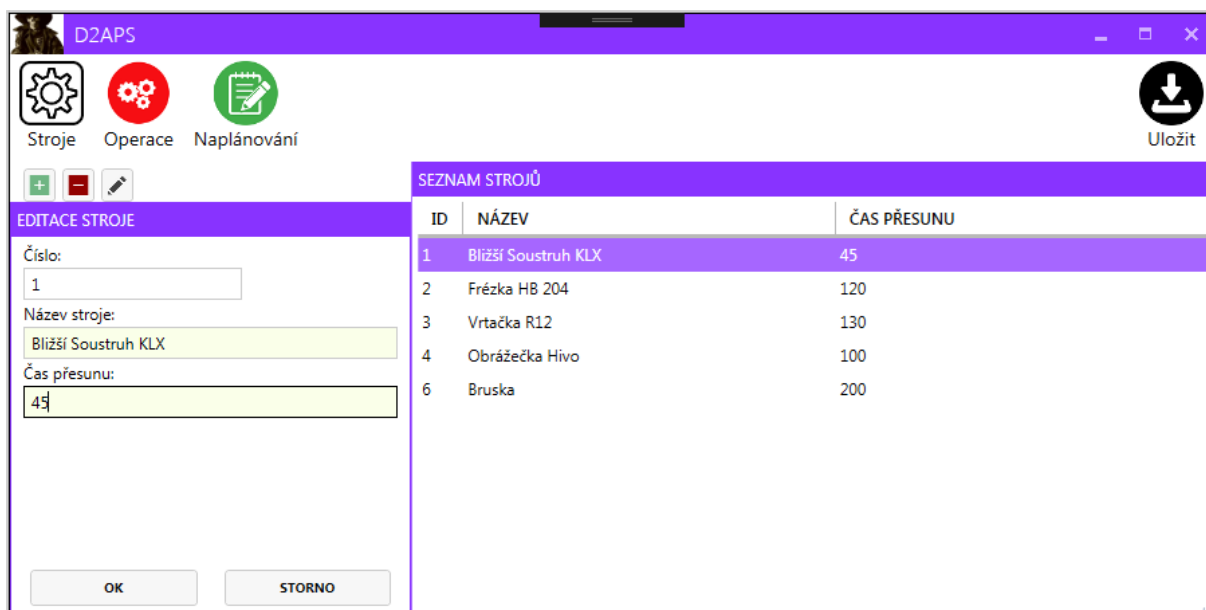
Pokud chce uživatel některý stroj smazat, stiskne tlačítko s ikonou „-“, na červeném pozadí. Na Obr. 17 je tlačítko zvýrazněno žlutým rámečkem. Po stisknutí tlačítka se objeví žádost o potvrzení smazání materiálu. Jelikož se často stává, že uživatel automaticky potvrdí, že chce smazat položku a pak zjistí, že smazal jinou položku, než chtěl, musí se smazání potvrdit kliknutím na tlačítko *Uložit* v záhlaví aplikace. Po uložení je odebraný stroj nenávratně smazán z databáze.



Obr. 18 Režim Prohlížení strojů

Druhá část stránky strojů, *Seznam strojů*, je interaktivní. Pokud uživatel klikne na některý ze strojů, zobrazí se mu údaje vybraného stroje v editační části. Pokud by se uživateli některý

údaj nelíbil, může kliknout na tlačítko s ikonou tužky, kterým přejde do *Editačního režimu*. Tlačítko je zvýrazněné žlutým rámečkem v Obr. 18.



Obr. 19 Editací režim

Změny v provedené v režimu editace se okamžitě zobrazí v *Seznamu strojů*. Jakmile je uživatel v *Editačním režimu*, může editovat jakýkoli stroj, který si v *Seznamu strojů* vybere. Klidně může provést editaci i u více strojů. Pro uložení hodnot do databáze je třeba stisknout tlačítko *OK* nebo tlačítko *Uložit*. Tlačítko *Storno* vrátí všechny neuložené editační změny zpět.

Pokud by uživatel chtěl otestovat, jaké změny by nastaly při změně času přesunu některého nebo některých strojů, nemusí kvůli tomu ukládat změny do databáze. Stačí v *Editačním režimu* změnit časy přesunu a pak nechat generovat plán zeleného ikonou *Naplánování* (generování plánu je vysvětleno v kapitole 5.1.5 Algoritmus plánování). Plán bude počítat se změněnou verzí. Díky tomu, že zobrazení plánu se generuje do samostatných oken, může vygenerovat více plánů a ty pak mezi sebou porovnávat. Nakonec může zmáčknout tlačítko *Storno* a vrátit se k původním hodnotám.

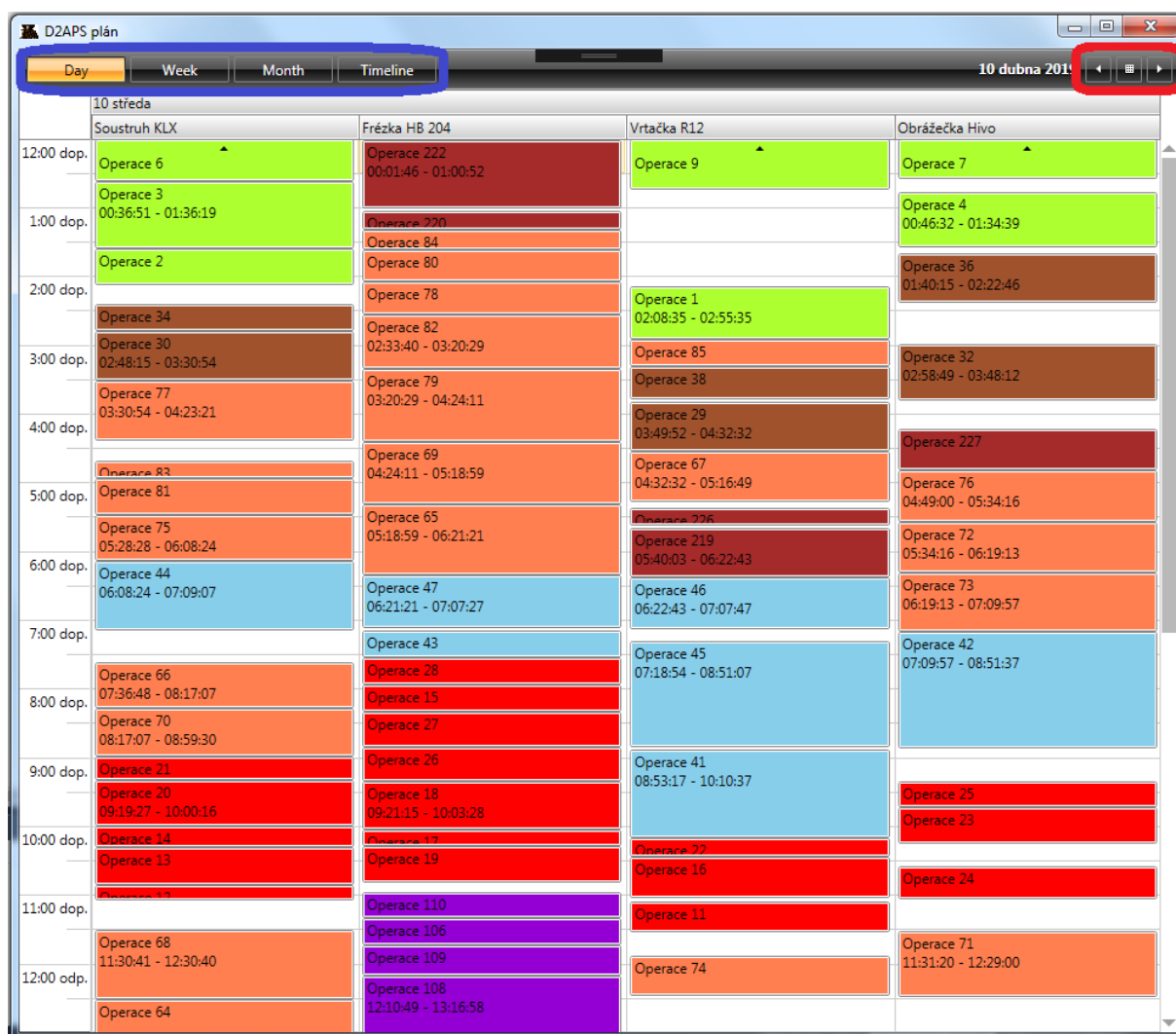
5.2.2 Zobrazení plánu v D2APS

Po předvedení fungování algoritmu ještě zbývá předvést, jak je zaimplementován do aplikace D2APS.



Obr. 20 Spuštění plánování v D2APS

Pro spuštění algoritmu je nutné zmáčknout tlačítko *Naplánování* v záhlaví aplikace. Tím se spustí algoritmus a po něm jeho grafické znázornění zvané *D2APS plán*.



Obr. 21 D2APS plán pro den

Grafické znázornění plánu je schopné zobrazovat plán v několika úrovních. Uživatel si může zobrazit plán pro kterýkoli den, týden nebo měsíc. Případně se může kouknout na plán na časové ose. Vybrané časové rozpětí je vidět na Obr. 21 vlevo nahoře (zvýrazněné modře).

Na Obr. 21 je tedy vidět D2APS plán pro zvolený den (na Obr. 21 je vidět pouze třináct hodin). Posouvání vybraného časového rozpětí je možné vpravo nahoře (zvýrazněné červeně). Mezi šipkami se nachází tlačítko s ikonou kalendáře, přes který je možné najít vzdálenější den, týden nebo měsíc.

Plán je rozdělen na sloupce, kde každý sloupec patří jednomu stroji, jehož název má v hlavičce. Pomyslné řádky zde nahrazuje časová osa.

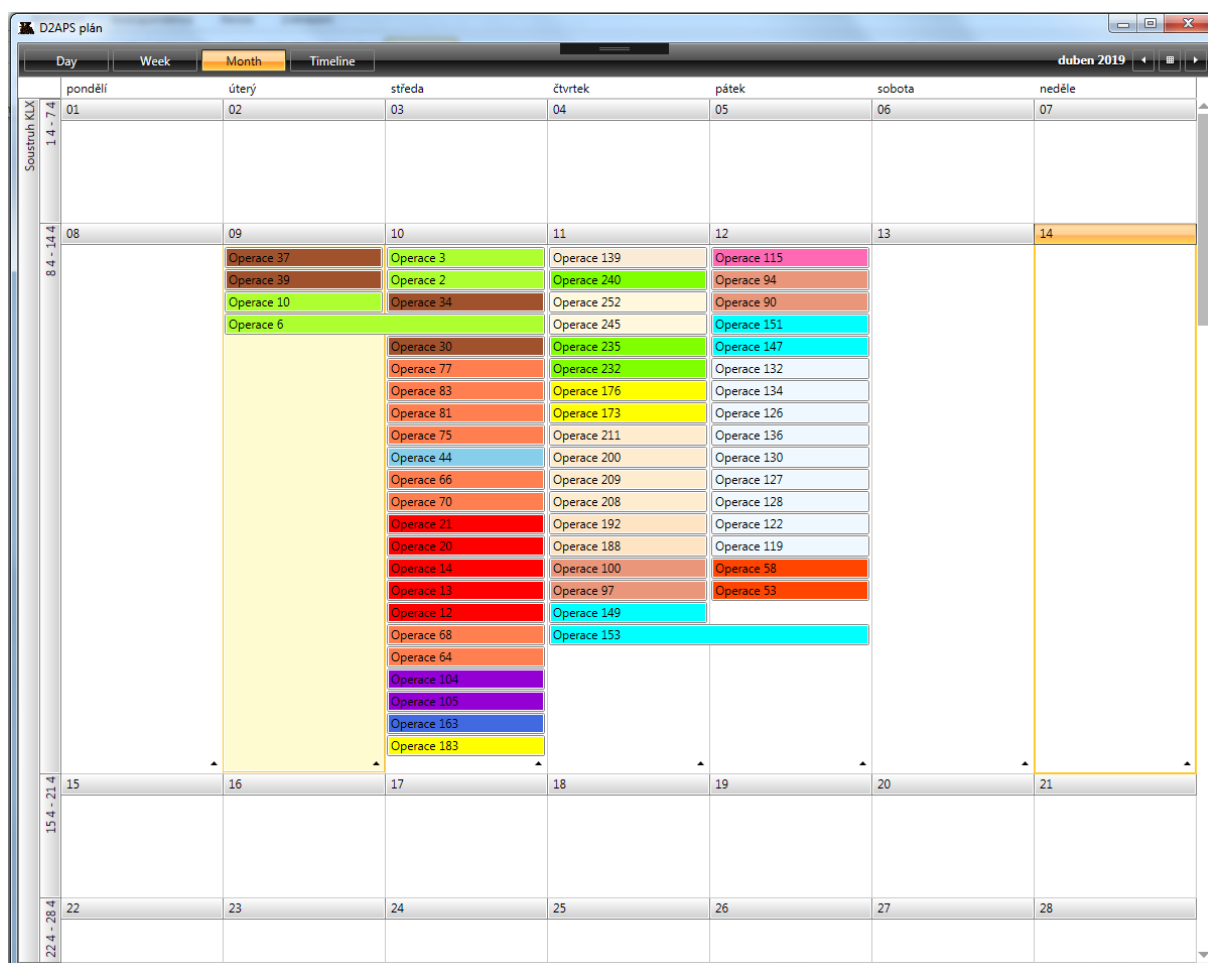
Jednotlivé operace jsou od sebe dobře rozeznatelné. Aplikace má k dispozici dvacet barevných rozlišení. Operace pro stejnou zakázku mají stejnou barvu. Velikost zobrazení operace je přímo závislá na trvání operace. Tedy čím větší operace, tím větší bude její znázornění. Pokud je operace dostatečně velká, lze z ní přečíst identifikační číslo, její začátek a konec. V případě, že operace trvá příliš krátce a tudíž se tyto informace do jejího grafického

zobrazení nevejdu, může tyto informace získat uživatel najetím kurzoru na tuto operaci. Operace, které přesahují do dalšího dne, jsou opatřeny malou šipkou.



Obr. 22 D2APS plán pro týden

Na Obr. 22 lze vidět plán v zobrazení pro celý týden. Zobrazení je velmi podobné jako u zobrazení dne.



Obr. 23 D2APS plán pro měsíc

Obr. 23 ukazuje, jak vypadá plán zobrazený pro měsíc. Zde došlo ke změnám vertikálních a horizontálních popisků, které jsou seřazeny do klasického kalendářního řazení. Horizontálně jsou zobrazeny dny v týdnu a vertikálně stroje a jednotlivé týdny. V tomto zobrazení už neplatí přímá závislost velikost zobrazení operace na trvání a zobrazení operací jsou všechny stejně velké. V případě většího množství strojů i operací za den se může toto zobrazení stát poněkud nepřehledné.

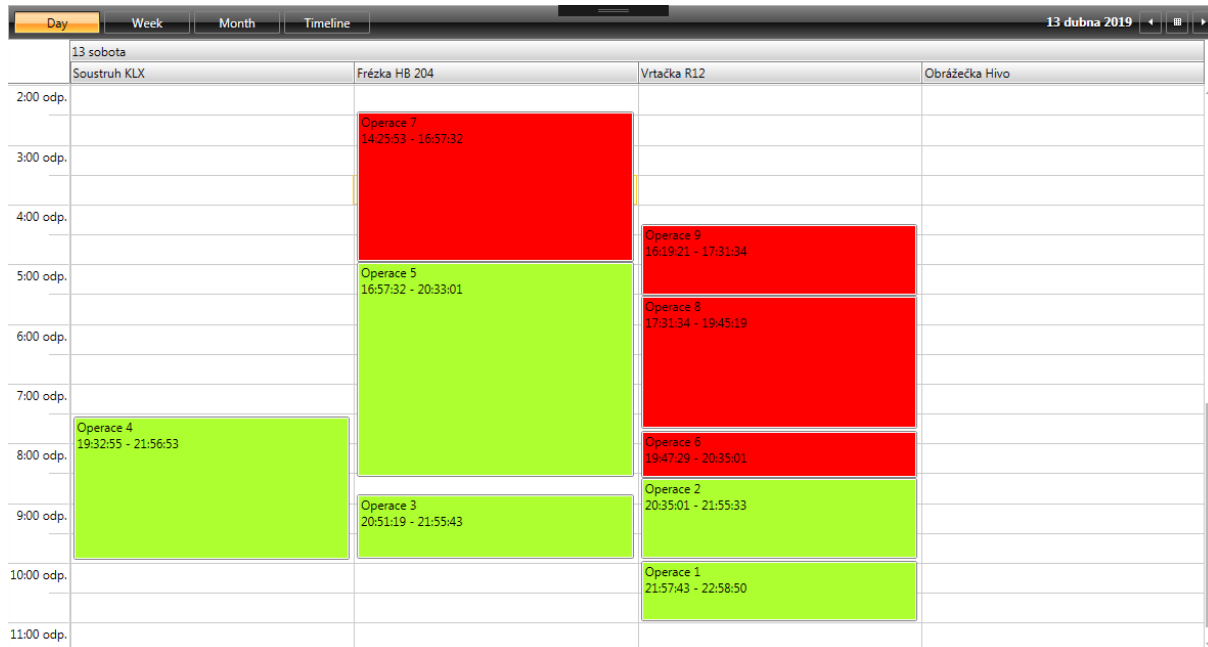


Obr. 24 Časová osa

Posledním možným zobrazením je zobrazení v časové ose, které lze vidět na Obr. 24. Jedná se o čistě vedlejší zobrazení, které může pomoci uživateli udělat si celkový přehled o využitelnosti strojů ve vybraném týdnu.

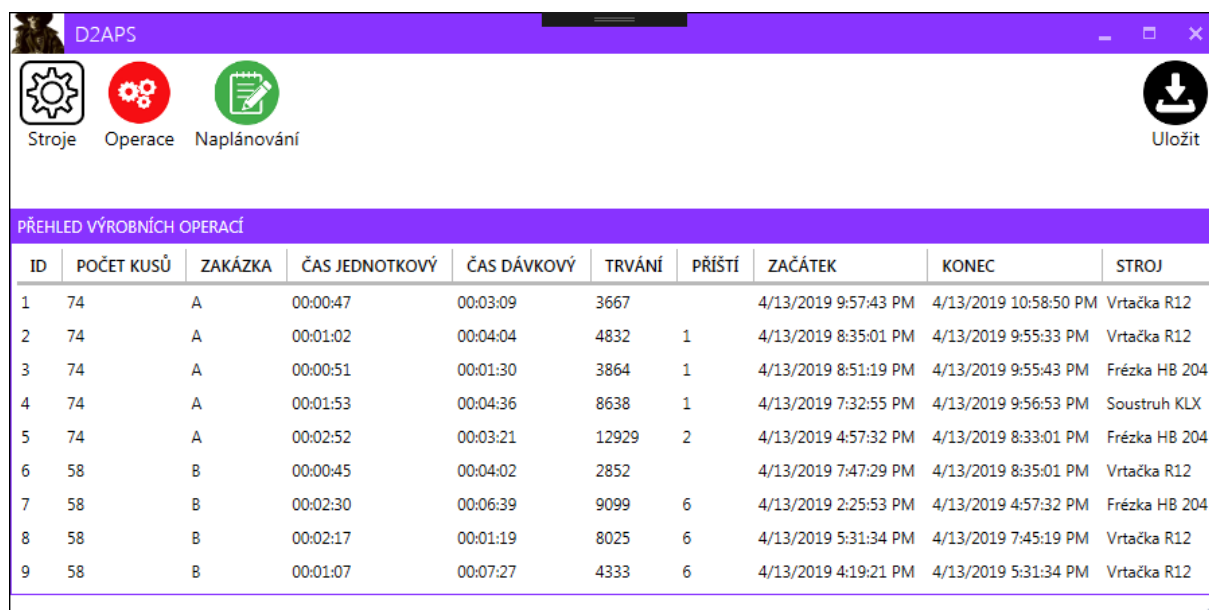
5.2.3 Příklad fungování algoritmu

Po seznámení s fungováním i zobrazením je možné přistoupit k ukázce některých funkcí algoritmu přímo na příkladu.



Obr. 25 Algoritmus na příkladu

Na Obr. 25 můžeme vidět pět zeleně zbarvených operací s čísly 1 až 5 pro zakázku A a čtyři červené operace s čísly 6 až 9 pro zakázku B. Finální operace jsou operace 1 a operace 6.

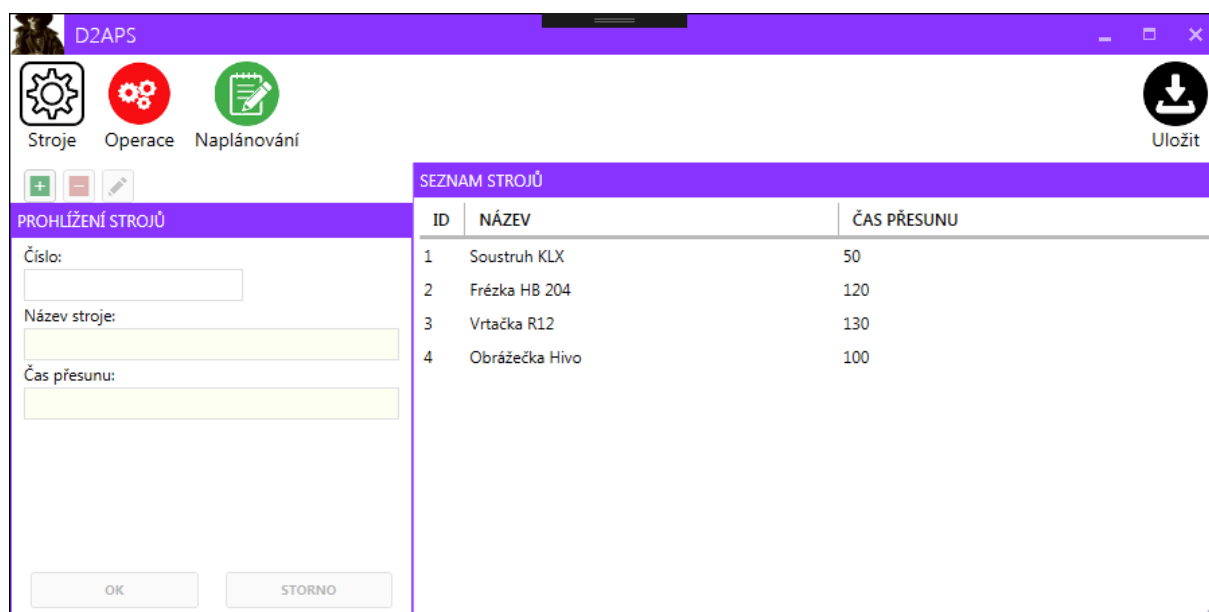


The screenshot shows the 'D2APS' software interface. At the top, there are three icons: 'Stroje' (Machines), 'Operace' (Operations), and 'Naplánování' (Scheduling). Below these is a table titled 'PŘEHLED VÝROBNÍCH OPERACÍ' (Overview of Production Operations). The table has the following columns: ID, POČET KUSŮ (Number of Pieces), ZAKÁZKA (Order), ČAS JEDNOTKOVÝ (Unit Time), ČAS DÁVKOVÝ (Batch Time), TRVÁNÍ (Duration), PŘÍŠTÍ (Next), ZAČÁTEK (Start), KONEC (End), and STROJ (Machine). The table contains 9 rows of data.

ID	POČET KUSŮ	ZAKÁZKA	ČAS JEDNOTKOVÝ	ČAS DÁVKOVÝ	TRVÁNÍ	PŘÍŠTÍ	ZAČÁTEK	KONEC	STROJ
1	74	A	00:00:47	00:03:09	3667		4/13/2019 9:57:43 PM	4/13/2019 10:58:50 PM	Vrtačka R12
2	74	A	00:01:02	00:04:04	4832	1	4/13/2019 8:35:01 PM	4/13/2019 9:55:33 PM	Vrtačka R12
3	74	A	00:00:51	00:01:30	3864	1	4/13/2019 8:51:19 PM	4/13/2019 9:55:43 PM	Frézka HB 204
4	74	A	00:01:53	00:04:36	8638	1	4/13/2019 7:32:55 PM	4/13/2019 9:56:53 PM	Soustruh KLX
5	74	A	00:02:52	00:03:21	12929	2	4/13/2019 4:57:32 PM	4/13/2019 8:33:01 PM	Frézka HB 204
6	58	B	00:00:45	00:04:02	2852		4/13/2019 7:47:29 PM	4/13/2019 8:35:01 PM	Vrtačka R12
7	58	B	00:02:30	00:06:39	9099	6	4/13/2019 2:25:53 PM	4/13/2019 4:57:32 PM	Frézka HB 204
8	58	B	00:02:17	00:01:19	8025	6	4/13/2019 5:31:34 PM	4/13/2019 7:45:19 PM	Vrtačka R12
9	58	B	00:01:07	00:07:27	4333	6	4/13/2019 4:19:21 PM	4/13/2019 5:31:34 PM	Vrtačka R12

Obr. 26 Data operací k příkladu

Na Obr. 26 jsou vidět příslušná data k operacím.



The screenshot shows the 'D2APS' software interface with a dialog box titled 'SEZNAM STROJŮ' (Machine List). The dialog box has a 'PROHLÍŽENÍ STROJŮ' (View Machines) section on the left with input fields for 'Číslo:' (Number), 'Název stroje:' (Machine Name), and 'Čas přesunu:' (Transfer Time). On the right, there is a table with the following columns: ID, NÁZEV (Name), and ČAS PŘESUNU (Transfer Time). The table contains 4 rows of data.

ID	NÁZEV	ČAS PŘESUNU
1	Soustruh KLX	50
2	Frézka HB 204	120
3	Vrtačka R12	130
4	Obrážka Hivo	100

Obr. 27 Data strojů k příkladu

Na Obr. 27 je možné vidět časy přesunů potřebné pro příklad.

Samotné plánování probíhá následovně:

Nejprve se určí prioritizace zakázek. Deadline obou zakázek je stejný, neděle 14.4.2019 v 00:01:00. Časový buffer je také nastaven pro obě zakázky stejně a to na jednu hodinu. Vyšší prioritu získá zelená zakázka A, kvůli větší časové náročnosti.

Plánování zakázky A

Jelikož zakázka A má větší prioritu, bude také jako první naplánována, jelikož algoritmus plánuje podle kritických zakázek.

Algoritmus tedy vyhledá všechny operace, které patří k plánované zakázce A a začne mezi nimi hledat vhodné operace k naplánování, které si dá do zvláštní skupiny. Pro lepší orientaci bude tato skupina dále nazývána jako *Operace k plánování*.

Jelikož zatím není naplánována žádná operace, jediná vhodná operace k naplánování je finální operace 1. Operace 1 se tedy rovnou naplánuje. Ideální konec operace se vypočte dle vzorce:

$$\text{ideální konec operace 1} = \text{deadline} - \text{buffer} - \text{čas přesunu}$$

$$\text{ideální konec operace 1} = 14.04.19\ 00:01:00 - 3600\ s - 130\ s = 13.04.19\ 22:58:50$$

Čas přesunu Vrtačky R12 je 130 sekund, výsledný konec tedy vyjde na 22:58:50 o den dříve, než je deadline, tedy na sobotu 13.4.2019. Začátek operace se vypočítá dle rovnice (5) v kapitole 5.1.6 Fungování algoritmu:

$$\text{ideální začátek operace 1} = \text{ideální konec operace 1} - \text{trvání operace 1}$$

$$\text{ideální začátek operace 1} = 13.04.19\ 22:58:50 - 3667\ s = 13.04.19\ 21:57:43$$

Poté, co algoritmus získá ideální časový interval, do které by měla být operace naplánována, proběhne kontrola, zda tento časový interval je na Vrtačce R12 volný. Jelikož je to první plánovaná operace, proběhne kontrola s kladnou odezvou a operace 1 je zaplánována.

Potom co operace 1 je naplánovaná, vymaže se ze skupiny *Operace k plánování*. Poté se do *Operací k plánování* přidají další vhodné operace. Další vhodné operace jsou vždy ty, které předchází právě naplánované operaci, tedy v tomto případě operace 2, operace 3 a operace 4.

Ve skupině *Operace k plánování* se tedy nacházejí tři operace a je třeba vybrat z nich tu nejkritičtější, aby se naplánovala nejdříve. Dojde tedy na určování priorit operací v rámci skupiny *Operace k plánování*.

Priorita operací se určuje podobně jako u priority zakázek. U každé operace dojde k součtu trvání operace a trvání všech operací, které této operaci předchází. Tím se získá *celkové trvání operace*. *Celkové trvání operace* pak charakterizuje jistý přehled o důležitosti operace. Samozřejmě čím větší číslo vyjde, tím je operace důležitější. Nejdůležitější operace má logicky nejvyšší prioritu.

Nyní tedy proběhne výpočet u operací 2, 3 a 4.

Operace 2 má jednu předchozí operaci a to operaci 5. Operace 5 již žádnou předchozí operaci nemá, takže výpočet *celkového trvání pro operaci 2* je:

$$\begin{aligned} \text{celkové trvání operace 2} &= \text{trvání operace 2} + \text{trvání operace 5} \\ \text{celkové trvání operace 2} &= 4832 + 12929 = 17761 \text{ s} \end{aligned}$$

Operace 3 nemá žádnou předchozí operaci.

$$\text{celkové trvání operace 3} = \text{trvání operace 3} = 3864 \text{ s}$$

I operace 4 je bez předchozí operace.

$$\text{celkové trvání operace 4} = \text{trvání operace 4} = 8638 \text{ s}$$

Nejvyšší prioritu získá operace 2, druhou prioritu operace 4 a třetí prioritu operace 3.

Dojde tedy k naplánování operace 2. Protože operace 2 je předchozí operací jiné operace (konkrétně operace 1), proběhne výpočet odlišně od výpočtu pro operaci 1. Konec operace 2 musí skončit nejméně tak brzy, aby bylo možné dopravit výrobky operace 2 z pracoviště pro operaci 2 na pracoviště pro operaci 1.

$$\begin{aligned} \text{ideální konec operace 2} &= \text{začátek operace 1} - \text{čas přesunu} \\ \text{ideální konec operace 2} &= 13.04.19 \text{ 21:57:43} - 130 \text{ s} = 13.04.19 \text{ 21:55:33} \end{aligned}$$

$$\begin{aligned} \text{ideální začátek operace 2} &= \text{ideální konec operace 2} - \text{trvání operace 2} \\ \text{ideální začátek operace 2} &= 13.04.19 \text{ 21:55:33} - 4832 \text{ s} = 13.04.19 \text{ 20:35:01} \end{aligned}$$

Zde je třeba malá odbočka. Shodou okolností byly operace 1 i 2 zpracovávány na stejném stroji Vrtačka R12. Přesto se počítalo s tím, že bude probíhat přesun mezi pracovišti, protože čas přesunu je vypočtená hodnota, která představuje průměrný čas přepravy mezi operacemi. A do tohoto průměru jsou zahrnuty i operace, které jsou na stejném pracovišti.

Ideální konec operace 2 je tedy stanoven na 21:55:33. Ideální začátek operace je pak vypočten na 20:35:01. Algoritmus zkontroluje, že v tento časový interval na stroji Vrtačka R12 ještě neprobíhají žádné operace a operaci zaplňuje.

Po naplánování operace 2, je operace 2 odebrána ze skupiny *Operace k plánování* a místo ní je do skupiny přidána operace 5, jako její předchozí operace.

Proběhne priorita operací a nejvyšší prioritu získá operace 5. Proběhne výpočet ideálního konce operace a ideálního začátku operace.

$$\text{ideální konec operace 5} = \text{začátek operace 2} - \text{čas přesunu}$$

$$\text{ideální konec operace 5} = 13.04.19\ 20:35:01 - 120\ s = 13.04.19\ 20:33:01$$

$$\text{ideální začátek operace 5} = \text{ideální konec operace 5} - \text{trvání operace 5}$$

$$\text{ideální začátek operace 5} = 13.04.19\ 20:33:01 - 12929\ s = 13.04.19\ 16:57:32$$

Algoritmus zjistí, že v tu dobu není na stroji Frézce HB 204 žádná jiná operace a tak bude operace 5 zaplánována.

Ze skupiny *Operace k plánování* je odebrána operace 5. A protože nemá žádnou předchozí operaci, zůstanou ve skupině jen operace 3 a 4.

Opět proběhne priorita operací a nejvyšší prioritu získá operace 4. Jelikož je to jediná operace na stroji Soustruh KLX, je naplánována bez jakékoli překážky na 19:32:55 až 21:55:53.

Operace 4 také nemá žádnou předchozí operaci, proto ve skupině *Operace k plánování* zbyde pouze operace 3. Ta je naplánována na Frézku HB 204 v čase 20:51:19 až 21:55:43 opět bez jakéhokoli problému.

Po naplánování je odebrána ze skupiny *Operace k plánování*, a jelikož nemá žádnou předchozí operaci, zůstane skupina prázdná a plánování zakázky A skončí.

Plánování zakázky B

Začne tedy plánování zakázky B. Algoritmus vyhledá všechny operace zakázky B a vybere z nich finální operaci 6 a tu dá do skupiny *Operace k plánování*.

Jako jedinou operaci skupiny jí pak algoritmus rovnou začne plánovat. Jako první proběhne výpočet ideálního konce operace.

$$\text{ideální konec operace 6} = \text{deadline} - \text{buffer} - \text{čas přesunu}$$

$$\text{ideální konec operace 6} = 14.04.19\ 00:01:00 - 3600\ s - 130\ s = 13.04.19\ 22:58:50$$

$$\text{ideální začátek operace 6} = \text{ideální konec operace 6} - \text{trvání operace 6}$$

$$\text{ideální začátek operace 6} = 13.04.19\ 22:58:50 - 2852\ s = 13.04.19\ 22:11:18$$

Ideální konec operace je tedy v 22:58:50 a začátek v 22:11:18. Nicméně algoritmus zjistí, že na pracovišti Vrtačka R12 se během této doby už pracuje. Je proto nalezen nejbližší možný termín a to od 19:47:29 do 20:35:01 a do tohoto termínu je operace 6 zaplánována.

Operace 6 je z *Operací k plánování* odebrána a nahradí jí její předchozí operace: operace 7, 8 a 9. Proběhne priorita úrovní. Protože žádná z operací nemá předchozí operace, platí pro operace 7, 8 a 9:

$$\text{celkové trvání operace} = \text{trvání operace}$$

Nejvyšší prioritu tak získává operace 7 s trváním 9099 sekund, prostřední prioritu získává operace 8 s trváním 8025 sekund a nejnižší prioritu dostává operace 9 s trváním 4333 sekund.

Proběhlo tedy plánování operace 7.

$$\text{ideální konec operace 7} = \text{začátek operace 6} - \text{čas přesunu}$$

$$\text{ideální konec operace 7} = 13.04.19\ 19:47:29 - 120\ \text{s} = 13.04.19\ 19:45:29$$

$$\text{ideální začátek operace 7} = \text{ideální konec operace 7} - \text{trvání operace 7}$$

$$\text{ideální začátek operace 7} = 13.04.19\ 19:45:29 - 9099\ \text{s} = 13.04.19\ 17:13:50$$

Nicméně na stroji Frézka HB 204, kde má být operace 7 provedena, už pro tuto dobu je naplánována jiná operace. Proto algoritmus najde nejbližší možný termín na 14:25:53 až 16:57:32 a tam operaci 7 zaplánuje.

Z *Operací k plánování* je odebrána operace 7, žádná operace není přidána. Proto proběhne priorita operací pouze s operacemi 8 a 9. Vyšší operaci získá operace 8.

$$\text{ideální konec operace 8} = \text{začátek operace 6} - \text{čas přesunu}$$

$$\text{ideální konec operace 8} = 13.04.19\ 19:47:29 - 130\ \text{s} = 13.04.19\ 19:45:19$$

$$\text{ideální začátek operace 8} = \text{ideální konec operace 8} - \text{trvání operace 8}$$

$$\text{ideální začátek operace 8} = 13.04.19\ 19:45:19 - 8025\ \text{s} = 13.04.19\ 17:31:34$$

Algoritmus zjistí, že tento čas je pro Vrtačku R12 volný a operaci 8 zaplánuje.

Ze skupiny *Operací k plánování* se odebere operace 8, a jelikož operace 8 nemá žádnou předchozí operaci, zůstane ve skupině pouze operace 9. Takže dojde k naplánování operace 9.

$$\textit{ideální konec operace 9} = \textit{začátek operace 6} - \textit{čas přesunu}$$

$$\textit{ideální konec operace 9} = 13.04.19\ 19:47:29 - 130\ \text{s} = 13.04.19\ 19:45:19$$

$$\textit{ideální začátek operace 9} = \textit{ideální konec operace 9} - \textit{trvání operace 9}$$

$$\textit{ideální začátek operace 9} = 13.04.19\ 19:45:19 - 4333\ \text{s} = 13.04.19\ 18:33:06$$

Nicméně algoritmus zjistí, že během tohoto časového intervalu už práce na Vrtačce R12 probíhá. Naplánuje proto operaci 9 na nejbližší možný čas a to od 16:19:21 do 17:31:34.

Operace 9 se odebere ze skupiny *Operace k plánování* a tím zůstane skupina prázdná. Skončí tedy plánování zakázky B, a jelikož zakázka B je poslední plánovanou zakázkou, skončí i celé plánování.

5.3 Časová validace algoritmu

U plánovacích algoritmů je vyžadováno, aby plánovali po dobu únosnou pro uživatele. Algoritmus, který plánuje denní plán půl dne je zbytečný. Proto jsou zde uvedeny některé časové zkoušky plánovacího algoritmu.

Nejprve byly provedeny časové zkoušky algoritmu dle vzoru validace algoritmu CP (kapitola 4.1.6 Porovnání s metodou MIP).

Tabulka 6 Informace o problémech (upraveno z [5])

Typ výroby	Počet výrobků	Max. počet operací	Počet strojů
a	5	6	5
b	10	6	5
c	5	21	5

Byly vytvořeny 3 sady problémů, jejichž parametry lze vidět v Tabulka 6. Zadání výrobních podmínek (počet výrobků, strojů a operací) je identické jako zadání problémů u algoritmu CP. U každého problému proběhly 3 sady testování. Výsledky jsou vidět v Tabulka 7.

Tabulka 7 Časové výsledky zadaných problémů

	Instance	Čas zpracování (s/1000)	Operace
D2APS	a-1	28,25	10
	a-2	32,04	15
	a-3	41,47	25
	b-1	45,85	25
	b-2	42,88	40
	b-3	57,13	50
	c-1	34,98	25
	c-2	69,28	50
	c-3	85,72	100

Nicméně je třeba upozornit, že srovnání CP a plánovacího algoritmu D2APS nemá velkou vypovídající hodnotu, protože CP plánuje mnohem komplexněji než plánovací algoritmus. Plánovací algoritmus je pouhou částí skupiny APS algoritmů, které se musí provázat, aby byl plán optimalizovaný na nejvyšší možnou úroveň.

Proto byl pro všeobecnou představu algoritmus podroben dalšímu časovému měření, jehož výsledky jsou v Tabulka 8.

Tabulka 8 Časové výsledky algoritmu

	Operace	Zakázky	Čas zpracování (s)
DZAPS	10	10	0,047
	100	10	0,092
	1000	10	4,753
	10000	100	185,858

Měření rychlosti pro jednu instanci spočívalo v deseti pokusných měřeních. Výsledná hodnota je průměr osmi středních hodnot měření.

6 Závěr

Kvalitní informační systém je dnes nepostradatelnou součástí každé moderní organizace a význam informačních technologií v současné době exponenciálně narůstá. Kromě komplexních řešení se stávají oblíbené právě aplikace na řízení výroby. Mnoho podniků má velmi složité a rozmanité výrobní prostředky a plánování výroby bez odpovídajících softwarových nástrojů se pro ně stává velkým problémem. Přestože mají tyto aplikace své odpůrce, s příchodem Průmyslu 4.0 a celkovou digitalizací výroby se postavení těchto aplikací bude jen a jen upevňovat.

První část práce je pojata jako studie pokročilého plánování. Cílem je, aby čtenář měl po přečtení hrubý přehled o obecné funkčnosti APS systémů a určitou představu o problematice vytváření algoritmů pro tyto systémy. Čtenář je seznámen s problematikou přesné definice APS a s důvody vzniku pokročilého plánování a rozvrhování. Dále je pak čtenář seznámen s metodou programovacího omezení pro pokročilý plánovací a rozvrhovací systém s víceúrovňovými strukturovanými produkty.

Druhá část práce je věnována praktické části diplomové práce. Byla vytvořena aplikace D2APS se schopností plánování a rozvrhování dle vlastního algoritmu. Zde je cílem, aby čtenář měl pro přečtení přehled o fungování vytvořené aplikace. Čtenář je seznámen s databází aplikace, včetně funkcí aplikace, které se s databází vážou. Dále je čtenář seznámen s fungováním naprogramovaného algoritmu plánování, které završí ukážka na příkladu.

7 Seznam použitých zdrojů

- [1] IVERT, LINEA KJELLSDOTTER. *Use of Advanced Planning and Scheduling (APS) systems to support manufacturing planning and control processes*. Göteborg, Sweden, 2012. Chalmers University Of Technology.
- [2] PRŮCHA, JAN. *Plánování výroby pomocí Advanced Planning Scheduling*. Plzeň, Česká republika, 2017. Západočeská univerzita.
- [3] VELKOBORSKÝ, Jan. *Analýza úspěšnosti implementace APS systému za podpory V-A-T analýzy*. Západočeská univerzita, 2001
- [4] MILLIKEN, Allan. What is S&OP?. *Demand-planning.com* [online]. 2018 [cit. 2018-12-03]. Dostupné z: <http://demand-planning.com/2018/02/01/what-is-sop-and-how-does-it-work/>
- [5] PENG, Yunfang, Dandan LU a Yarong CHEN. A Constraint Programming Method for Advanced Planning and Scheduling System with Multilevel Structured Products. *Discrete Dynamics in Nature and Society* [online]. 2014, 2014, **2014**(917685) [cit. 2018-12-03]. Dostupné z: <https://www.hindawi.com/journals/ddns/2014/917685/>
- [6] *Dokumentace Entity Framework: Vytvoření modelu* [online]. Microsoft, 2018 [cit. 2019-04-16]. Dostupné z: <https://docs.microsoft.com/cs-cz/ef/ef6/modeling/>
- [7] ING. HOŘEJŠÍ, PH.D., Petr. *Tvorba databázové aplikace Sklad*. Plzeň, Česká Republika, 2017. ZČU.
- [8] *Průvodce programováním v C#: Částečné třídy a metody* [online]. Microsoft, 2015 [cit. 2019-04-16]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/csharp/programming-guide/classes-and-structs/partial-classes-and-methods>
- [9] Basl, J., Velkoborský J.: Přehled českého trhu softwarových nástrojů APS a CRM. *Computerworld* 32/2000
- [10] KOPEČEK, P., PINTE, M. Optimization Heuristics for Supplies of Assembly Lines. *Periodica Polytechnica Transportation Engineering*, 2014, roč. 42, č. 1, s. 49-52. ISSN: 0303-7800
- [11] *Online ERD Tool* [online]. visual-paradigm.com, 2018 [cit. 2019-05-07]. Dostupné z: <https://online.visual-paradigm.com/>