

ŠKODA AUTO VYSOKÁ ŠKOLA o.p.s.

Studijní program: N0413A050001 Ekonomika a management

Studijní obor/specializace: Specializace Řízení mezinárodních dodavatelských řetězců

Využití modelů hromadné obsluhy s prioritou

Diplomová práce

Bc. Michal Vild

Vedoucí práce: doc. Ing. Jan Fábry, Ph.D.



ŠKODA AUTO Vysoká škola

ZADÁNÍ DIPLOMOVÉ PRÁCE

Zpracovatel: **Bc. Michal Vild**

Studijní program: Ekonomika a management

Specializace: Řízení mezinárodních dodavatelských řetězců

Název tématu: **Využití modelů hromadné obsluhy s prioritou**

Cíl: Tématem této práce jsou modely hromadné obsluhy se zaměřením na modely s prioritními frontami. V rámci práce bude provedena analýza použití těchto modelů v reálných systémech. Cílem práce je analýza a následná aplikace modelů s prioritními frontami v oblasti dobíjecích stanic pro elektromobily.

Rámcový obsah:

1. Provedte rešerši odborné literatury se zaměřením na modely hromadné obsluhy, a především modely s prioritními frontami.
2. Analyzujte aktuální využívání front s prioritou v reálných systémech hromadné obsluhy.
3. Popište současný stav elektromobility.
4. Provedte aplikaci teorie front s prioritou v oblasti dobíjecích stanic pro elektromobily.

Rozsah práce: 55 – 65 stran

Seznam odborné literatury:

1. BHAT, U N. *An introduction to queueing theory: modeling and analysis in applications*. USA: Birkhäuser, 2015. 339 s. ISBN 978-0-8176-8421-1.
2. CHAN, W C. *An elementary introduction to queueing systems*. Singapore: World Scientific, 2014. 116 s. ISBN 978-981-4612-00-5.
3. SAID, D., CHERKAoui, S. and L. KHOUKHI, 2015, Multi-priority queueing for electric vehicles charging at public supply stations with price variation. *Wirel. Commun. Mob. Comput.* 2015, vol. 15, s. 1049–1065. doi: 10.1002/wcm.2508
4. SHONE, R., GLAZEBROOK, K. and K. G. ZOGRAFOS. Resource allocation in congested queueing systems with time-varying demand: An application to airport operations. *European Journal of Operational Research*. 2019, vol. 276, no. 2, s. 566-581. ISSN 0377-2217
5. SCHWARZ, J. A., SELINKA, G. and R. STOLLETZ. Performance analysis of timedependent queueing systems: Survey and classification. *Omega Oxford*. 2016, vol. 63, s. 170-189. ISSN 0305-0483

Datum zadání diplomové práce: únor 2022

Termín odevzdání diplomové práce: leden 2023

L. S.

Elektronicky schváleno dne 8. 2. 2022

Bc. Michal Vild

Autor práce

Elektronicky schváleno dne 8. 2. 2022

doc. Ing. Jan Fábry, Ph.D.

Vedoucí práce

Elektronicky schváleno dne 8. 2. 2022

doc. Ing. Jan Fábry, Ph.D.

Garant studijní specializace

Elektronicky schváleno dne 8. 2. 2022

doc. Ing. Pavel Mertlík, CSc.

Rektor

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a použité zdroje uvádím v seznamu literatury. Prohlašuji, že jsem se při vypracování řídil vnitřním předpisem ŠKODA AUTO Vysoké školy o.p.s. (dále jen ŠAVŠ) směrnicí Vypracování závěrečné práce.

Jsem si vědom, že se na tuto závěrečnou práci vztahuje zákon č. 121/2000 Sb., autorský zákon, že se jedná ve smyslu § 60 o školní dílo a že podle § 35 odst. 3 je ŠAVŠ oprávněna mou práci využít k výuce nebo k vlastní vnitřní potřebě. Souhlasím, aby moje práce byla zveřejněna podle § 47b zákona č. 111/1998 Sb., o vysokých školách.

Beru na vědomí, že ŠAVŠ má právo na uzavření licenční smlouvy k této práci za obvyklých podmínek. Užiji-li tuto práci, nebo poskytnu-li licenci k jejímu využití, mám povinnost o této skutečnosti informovat ŠAVŠ. V takovém případě má ŠAVŠ právo ode mne požadovat příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to až do jejich skutečné výše.

V Mladé Boleslavi dne 28.12.2022



V první řadě děkuji doc. Ing. Janu Fábrymu, Ph.D. za odborné vedení diplomové práce, za jeho pomoc a v neposlední řadě za cenné rady, podněty a také za čas strávený u konzultací.

Dále děkuji Ing. Radku Vařechovi za praktické rady diskuse nad praktickou částí této práce.

Dále bych rád poděkoval rodině a také přátelům za jejich podporu.

Obsah

Úvod.....	8
1 Systémy hromadné obsluhy.....	9
1.1 Základní části systémů hromadné obsluhy	10
1.2 Režimy front	14
1.3 Birth-death proces	16
1.4 Klasifikace modelů hromadné obsluhy	16
1.5 Stochastické procesy	17
1.6 Markovovy procesy v systémech hromadné obsluhy	19
2 Využití softwarových nástrojů a heuristických metod	22
2.1 Simulační software Tecnomatix Plant Simulation	23
2.2 Software MPL.....	25
2.3 Heuristické algoritmy	26
3 Modely prioritního řazení front	27
3.1 Obecný prioritní model fronty	28
3.2 Prioritní model Head-of-the-Line	30
3.3 Prioritní model Time-Dependent Priority	31
3.4 Využití bribingu pro pozici ve frontě.....	31
3.5 Prioritní fronta s využitím haldy	32
3.6 Binární halda	34
3.7 Využití haldy v prioritních frontách	36
4 Modely rozvrhování	38
5 Aplikace modelů hromadné obsluhy v reálném prostředí	40
5.1 Model pro dobíjení elektromobilů	43
6 Aplikace heuristického algoritmu pro rozvrhování dávek	51
6.1 Řešení pomocí základního heuristického algoritmu	51
6.2 Využití priority pro vybrané vozy	54
6.3 Využití priority podle nejvyššího nabití	56
7 Vyhodnocení výsledků	59
Závěr	62

Seznam literatury	63
Seznam obrázků a tabulek	68
Seznam příloh	70

Seznam použitých zkratk a symbolů

PS	Plant Simulation
CAD	Computer Aided Design
HOL	Head of the Line
MPL	Mathematical Programming Language
SFJ	Shortest-Job-First
LFJ	Longest-Job-First
MS	Microsoft
NP	Nedeterministicky polynomiální obtížnost

Úvod

Systemy hromadné obsluhy jsou v současné době naprosto běžnou součástí každodenního života. Stejně tomu tak bylo v minulosti a s velkou pravděpodobností lze říci, že stejně tomu bude i v budoucnosti. Pokud však v nějakém systému fronta vznikne, je vždy žádoucí s frontou pracovat, aby nebylo nutné trávit příliš mnoho času nebo lépe, aby vůbec nevznikla. Stejně je tomu v případě nabíjení elektromobilů. Téma elektromobility je v současné době velmi aktuální, kdy je řešen systém nabíjení těchto vozů. Nabíjecí stanice a elektromobily společně tvoří systém hromadné obsluhy, kde se přirozeně vyskytuje řada proměnných. Mezi ně patří například doba nabíjení každého vozu, rychlost nabíjení a s tím související potřebný počet nabíjecích stanic. Cílem je poté tento systém optimalizovat takovým způsobem, kdy jsou všechny elektromobily v přijatelném čase nabitý, u nabíjecích stanic nedochází ke vzniku front a zároveň vše probíhá s akceptovatelnými náklady.

Tato práce je rozdělena do dvou tematických částí. Teoretická část, skládající se ze třech kapitol, se věnuje nejprve obecně problematice systémů hromadné obsluhy, kdy jsou popisovány jednotlivé části systému a vztahy mezi nimi. Další kapitoly se zaměřují nejprve na možnosti, jakými lze systémy hromadné obsluhy modelovat pomocí softwarových nástrojů i pomocí heuristických metod. Následná kapitola už více popisuje speciální případy front, jakým jsou fronty prioritní, kdy jsou popsána jejich specifika a také práce s nimi.

Praktická část je věnována výzkumu systémů hromadné obsluhy v návaznosti na téma elektromobility. Zde je vytvořen model reprezentující provoz elektrických vozidel a jejich nabíjení na omezeném počtu nabíjecích stanic. Tento model je řešen pomocí softwarového nástroje MPL i heuristické metody, popsaných v teoretické části. Model je poté různými způsoby modifikován. Hlavním cílem je vždy dokončit celý proces nabíjení ve stanoveném čase.

1 Systémy hromadné obsluhy

Dle Sundarapandiana (2009) jsou systémy hromadné obsluhy obsahem oboru aplikované matematiky označovaného jako teorie front, jinak také teorie hromadné obsluhy. Hlavní činností je analýza systémů, do kterých v určitém intervalu přichází požadavky, a cílem je tyto požadavky efektivně obsloužit, resp. vytvořit efektivně fungující prostředí. Model hromadné obsluhy je zároveň konstruován takovým způsobem, aby bylo možné předvídat délku fronty i čekací dobu.

Jak dále uvádí Queuing-it (2022) představují teorie front velmi účinný nástroj pro analýzu každodenního jevu čekání ve frontě. Bhat (2015) dále dodává, že systémy hromadné obsluhy a fronty obecně pomáhají efektivně zpracovat a zvládat tok požadavků, pokud systém disponuje omezenou kapacitou. Teorie front zkoumá holistickým pojetím systém čekání ve frontě, včetně prvků, jakými jsou:

- intenzita příchodů požadavků a intervaly mezi příchody,
- počet obslužných stanovišť a jejich fungování,
- počet zákazníků čekajících ve frontě,
- kapacita čekací zóny,
- průměrná doba obsluhy,
- režim fronty.

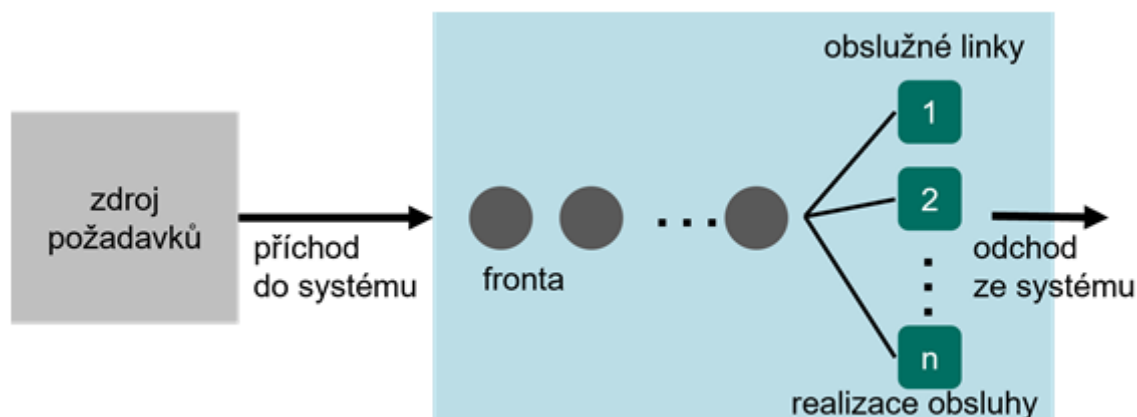
Režim fronty odkazuje na daná pravidla fronty, například zda se chová na principu první v pořadí, poslední v pořadí, obsloužení v náhodném pořadí nebo zda v systému existuje priorita.

Za zakladatele tohoto vědního oboru je považován dánský inženýr matematik Agner Krarup Erlang, který se na počátku 20. století věnoval problematice teorie pravděpodobnosti v telefonní společnosti *Copenhagen Telephone Exchange* (Sundarapandian, 2009). Zde publikoval první článek k problematice teorie front, kde řešil příchozí hovory přicházející na ústřednu. Od té doby se tato teorie neustále vyvíjela až do podoby, v jaké je známa dnes. V současné době jsou systémy hromadné obsluhy běžně a nedomyslitelně používány v oblasti plánování logistiky (Šeda, 2011). Jablonský (2011) dále popisuje systém hromadné obsluhy jako vše, co se nachází mezi příchodem požadavku do systému a odchodem ze systému.

Obecně platí, že systémy hromadné obsluhy jsou v současné době naprosto běžnou součástí každodenního života.

1.1 Základní části systémů hromadné obsluhy

Jak uvádí Jablonský (2011), v systémech hromadné obsluhy se vyskytují jednak požadavky a jednak obslužná zařízení (nebo také obslužné linky). Systémy hromadné obsluhy jsou charakterizovány realizací obsluhy příchozích požadavků. Pokud požadavky přichází do systému s velkou intenzitou, případně pokud je v daném okamžiku obslužné zařízení plně obsazeno, dochází ke vzniku front. Požadavky se tak při příchodu do systému hromadí před obslužným pracovištěm a tím vytvoří frontu (Fábry, 2019). Příklad systému hromadné obsluhy je znázorněn na Obr. 1.



Zdroj: (Jablonský, 2011)

Obr. 1 Schéma modelu hromadné obsluhy

1.1.1 Příchod požadavků do systému

Požadavky jsou prvky systému hromadné obsluhy, které systémem prochází a jsou obsluhovány. Mají tedy dynamickou povahu. Požadavky tvoří frontu ve chvíli, kdy je obslužné pracoviště obsazeno a nemůže v daném čase přijímat další požadavky. Ty se následně hromadí a vyčkávají ve frontě do chvíle, kdy se obslužné pracoviště opět uvolní. V reálném prostředí mohou požadavky představovat například lidé, předměty, informace apod. Požadavkem tedy mohou být například zákazníci v supermarketu, vozidla projíždějící křižovatkami nebo pacienti u lékaře.

Ve výrobním procesu mohou být požadavky karoserie v lakovně nebo přístrojové panely na montážní lince.

Jak uvádí Chan (2014), příchod požadavků do systému je důležitou charakteristikou systému. Příchozí požadavky lze popsat následujícími dvěma způsoby pomocí:

1. Počtu příchodů, tedy počet požadavků, které vstoupí do systému za jednotku času.
2. Intervalů mezi příchody po sobě příchozích požadavků, jinými slovy časem mezi dvěma po sobě jdoucími příchody.

Jak dále uvádí Jablonský (2011), oba zmíněné způsoby mohou být dvojího typu, a to konkrétně:

- Deterministické, kdy jsou intervaly mezi příchody požadavků pevně dané. S tímto případem se lze v praxi setkat například na výrobní lince, kdy v pevně daném intervalu (taktu) přichází požadavky v podobě rozpracované výroby k dalšímu zpracování.
- Stochastické, kdy jsou intervaly mezi příchody proměnlivé v čase. V praxi se s tímto případem lze setkat častěji, protože v mnoha systémech, jako úřady, dopravní křižovatky, čerpací stanice aj., nelze předpokládat pravidelný příchod požadavků. Intervaly jsou v tomto případě popisovány některým z pravděpodobnostních rozdělení. Určení konkrétního pravděpodobnostního rozdělení závisí na statistické analýze. Pro intervaly mezi příchody požadavků se podle Jablonského (2011) často vyskytuje rozdělení exponenciální s jedním parametrem λ , kdy střední hodnota je dána následujícím vztahem (1):

$$E(X) = \frac{1}{\lambda} \quad (1)$$

Parametr λ je interpretován jako střední počtu požadavků, které vstoupí do systému za časovou jednotku, a nazývá intenzita příchodů požadavků.

Pravděpodobnost, že interval mezi příchody bude kratší než t , což značí hodnotu distribuční funkce exponenciálního rozdělení v bodě t , je potom dána vzorcem (2):

$$P(X \leq t) = F(t) = 1 - e^{-2\lambda}, \quad (2)$$

kde hodnota e jako základu přirozeného logaritmu je 2,7183.

1.1.2 Obsluha

Pojem obsluha, jinak nazývána také jako obslužné zařízení nebo obslužná linka, popisuje Jablonský (2011) jako zařízení, jehož cílem je obsloužit požadavky. Obsluha je tím, proč požadavky vstupují do systému. Nejdůležitějším parametrem obsluhy a zároveň charakteristikou systému je čas, za který jsou obslužná pracoviště schopna obsloužit příchozí požadavky. Jablonský (2011) dále zmiňuje, že tato veličina může být, stejně jako v případě příchodu požadavků, deterministická nebo stochastická. V případě stochastické veličiny se opět používá nejčastěji exponenciální rozdělení s parametrem μ , který se nazývá intenzita obsluhy. Intenzita obsluhy je charakterizována jako průměrný počet obslužených požadavků za danou časovou jednotku. Střední hodnota doby obsluhy je dána vztahem (3) (Jablonský, 2011):

$$E(X) = \frac{1}{\mu}. \quad (3)$$

Jak znázorňuje Obr. 2, v případě jednoduchého systému existuje pouze jedna obslužná linka.



Zdroj: (Jablonský, 2011)

Obr. 2 Systém hromadné obsluhy s jednou obslužnou linkou

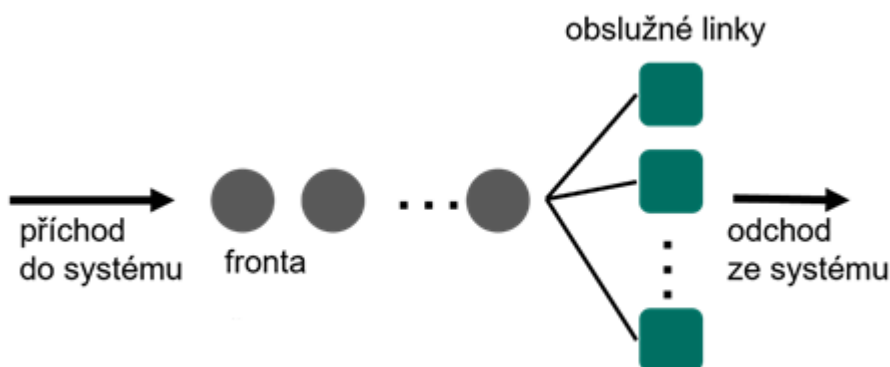
V praxi se ale v systémech hromadné obsluhy vyskytuje obslužných linek i více. V případě jejich většího počtu se rozlišují typy podle rozmístění linek:

- Sériové uspořádání (viz Obr. 3), kdy požadavky prochází postupně všemi obslužnými linkami v sérii. Praktickým příkladem může být výrobní linka, kdy požadavky prochází postupně různými pracovišti.
- Paralelní uspořádání (viz Obr. 4), kdy se požadavky rozdělují k jednotlivým obslužným linkám. Fronta může vznikat buď zvlášť před každou linkou nebo v podobě jediné fronty, odkud požadavky přechází k jednotlivým obslužným linkám. Zároveň oproti sériovému uspořádání požadavky prochází pouze jedinou obslužnou linkou. V praxi se lze setkat s tímto uspořádáním na čerpacích stanicích.



Zdroj: (Jablonský, 2011)

Obr. 3 Sériové uspořádání obslužných linek



Zdroj: (Jablonský, 2011)

Obr. 4 Paralelní uspořádání obslužných linek

V reálném prostředí dále existují i systémy, které kombinují typy uspořádání podle požadavků a potřeb daného systému.

Jablonský (2011) dále uvádí i příklady analýzy systémů hromadné obsluhy. Při analýze jsou zásadní charakteristiky, díky kterým lze popsat fungování celého systému.

Zpravidla se jedná o tyto charakteristiky:

- průměrná doba, jakou stráví požadavek čekáním ve frontě,
- průměrná doba čekání v systému jako celku,
- průměrná délka fronty,
- průměrný počet požadavků v systému.

Dále se zjišťují charakteristiky související s pravděpodobnostmi určitých situací, jako například pravděpodobnost, že linka obsluhuje nebo neobsluhuje, pravděpodobnost, že požadavek bude muset čekat ve frontě apod.

1.1.3 Proces vzniku fronty

Systémy hromadné obsluhy lze rozdělovat také podle toho, zda fronta vzniká či nevzniká. V případě vzniku fronty požadavky přichází do systému a postupně obsazují všechny dostupné obslužné linky. Další příchozí požadavky musí následně čekat na uvolnění obslužné linky, což vede ke vzniku fronty. Fronta může být s omezenou i neomezenou délkou. Omezená délka fronty znamená, že po dosažení maximální délky fronty, tedy maximálního počtu požadavků ve frontě, další příchozí požadavky již na obsluhu nečekají a odchází bez obsluhy. K obdobné situaci dochází i v druhém případě, kdy fronta nevzniká. Požadavky přichází do systému a pokud je v tomto čase obsluha obsazena, požadavky přímo systém opouští, aniž by byly obslouženy nebo založily frontu. Systémy, ve kterých dochází k odchodu požadavků bez poskytnutí obsluhy, se nazývají systémy se ztrátami (Kořenář, 2010).

1.2 Režimy front

Jak popisuje Šeda (2011), fronty mohou obecně fungovat v různých režimech, které jsou založené na pořadí příchodu a následném odchodu požadavků ze systémů.

Na zvoleném režimu fronty závisí i efektivnost systému jako celku. Režim fronty tedy určuje pořadí, v jaké budou příchozí požadavky obslouženy.

FIFO

Režim FIFO (First-In, First-Out) se v praxi objevuje velmi často. Lze ho ukázat na příkladu zákazníků v obchodě, kteří jsou obsluhováni v pořadí, v jakém přišli. Jinými slovy tedy kdo dříve přijde, dříve také odejde.

LIFO

Režim LIFO (Last-In, First-Out) je opakem k režimu FIFO, kdy jsou požadavky obsluhovány v opačném pořadí, než v jakém do systému vstoupily. V praxi se s ním lze setkat například ve skladu zboží. Zboží, které bylo dodáno jako první je uloženo vzadu, je tedy nejhůře přístupné a bude vyskladněno jako poslední.

SIRO

Oproti předchozím režimům, režim SIRO (Selection In Random Order) je charakterizován náhodným výběrem požadavků z fronty. Jak uvádí Fábry (2019), všechny požadavky mají stejnou pravděpodobnost výběru. V praxi lze tento režim vidět u výstupu cestujících z autobusu na konečné zastávce, kdy se náhodně dostávají ke dveřím a vystupují.

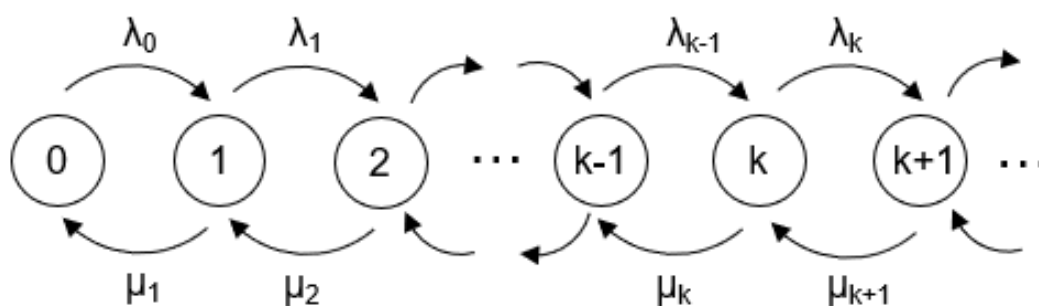
PRI

PRI (Priority system) je režim fronty, kdy jsou požadavky z fronty vybírány na základě určité přiřazené priority. Ta je přiřazována s ohledem na charakter systému. S tímto režimem se lze setkat například v nemocnici na urgentním příjmu, kdy nejvyšší prioritu obdrží pacienti s nejzávažnějším zraněním. Priorita může být ale přiřazována i na jiných základech. Tomuto tématu budou věnovány následující kapitoly této práce. Modely s prioritou se řadí do dvou kategorií dle toho, zda může či nemůže být obsluha přerušena kvůli příchodu požadavku s vyšší prioritou (Harchol-Balter, 2012).

Dle Fábryho (2019) je i odchod požadavků (podobně jako příchod) ze systému určitým rozhraním systému hromadné obsluhy a okolí systému. Z hlediska analýzy nejsou však odchody nijak důležité.

1.3 Birth-death proces

Proces birth-death (označovaný také jako birth-and-death proces) je speciálním případem spojitého Markovova procesu (Nowak, 2006). Vyznačuje se dvěma různými změnami stavu. V prvním případě *birth* vstoupí jeden požadavek do systému, čímž zvýší celkový počet požadavků v systému. V druhém případě *death* je proces přesně opačný, kdy požadavek po obslužení ze systému vystoupí a celkový počet požadavků se tedy sníží. V praxi se uplatňuje například v demografii nebo v biologii. V teorii hromadné obsluhy je tento proces také velmi často využíván, zejména pro fronty s příchody s Poissonovým rozdělením o nekonečné kapacitě (Tavaré, 2018). Na Obr. 5 je schematicky znázorněn birth death proces. Počet vstupů (*birth rates*) je zde dán jako $\{\lambda_i\}_{i=0 \dots \infty}$, počet výstupů (*death rates*) jako $\{\mu_i\}_{i=1 \dots \infty}$.



Zdroj: (Tavaré, 2018)

Obr. 5 Schéma procesu birth-death

1.4 Klasifikace modelů hromadné obsluhy

Pro klasifikaci modelů hromadné obsluhy se používá Kendallova klasifikace, kterou již v 50. letech 20. století zavedl anglický matematik D. G. Kendall (Šeda, 2011). Ta je založena na šestimístném kódu A/B/C/D/E/F. Jak dále uvádí Fábry (2019), tato klasifikace je používána pro modely s paralelním upořádáním obslužných linek a zároveň nese i určitá omezení. Použít ji lze pouze tehdy, kdy všechny obslužné linky mají stejný typ pravděpodobnostního rozdělení, které charakterizuje dobu obsluhy. Tab. 1 následně zobrazuje vysvětlení Kendallovy klasifikace.

Tab. 1 Kendallova klasifikace

Pozice	Význam
A	<p>Typy pravděpodobnostního rozdělení intervalů mezi příchody. Nejčastěji jsou používány následující:</p> <ul style="list-style-type: none"> • M = exponenciální rozdělení, • E = Erlangovo rozdělení, • U = rovnoměrné rozdělení, • N = normální rozdělení, • D = deterministická (konstantní) hodnota, • G = blíže nespecifikované rozdělení.
B	<p>Typ pravděpodobnostního rozdělení doby obsluhy (dle zkratek stejně jako u předchozího případu).</p>
C	<p>Počet paralelně uspořádaných obslužných linek (číslo 1 značí jednoduchý systém hromadné obsluhy).</p>
D	<p>Režim fronty, tzn. FIFO, LIFO, PRI nebo SIRO.</p>
E	<p>Maximální počet požadavků ve frontě.</p>
F	<p>Velikost zdroje požadavků.</p>

Zdroj: (Fábry, 2019)

Příkladem může být model M/M/1, kde je kód ve zjednodušené třímístné podobě. Jde o jednoduchý, běžně užívaný, model, kde jedno obslužné zařízení obsluhuje požadavky, které přicházejí podle Poissonova procesu (kde mají intervaly mezi příchody exponenciální rozdělení) a mají exponenciálně distribuované doby obsluhy (M označuje Markovův proces). Podobným příkladem je model fronty M/G/1, kde G označuje libovolné rozdělení pravděpodobnosti pro dobu obsluhy.

1.5 Stochastické procesy

V teorii pravděpodobnosti a také v jiných příbuzných oborech je stochastický, jinak také náhodný, proces matematickým objektem definovaným jako množina náhodných proměnných, jinak také v případě prvků a vazeb mezi nimi existují náhodné procesy. Obdobně i prvky a vazby samotné mohou mít charakter náhodných prvků, resp. náhodných jevů (Skalská, 2006). Stochastické procesy jsou

často využívané v matematických systémech, zejména v těch, kde se určitá náhodná veličina přímo vyskytuje. Lze je proto uplatnit v širokém spektru oborů, jako například biologie, informatika, fyzika a další (Parzen, 2015). Stochastické procesy se začaly uplatňovat i ve finančním sektoru, z důvodu zjištění náhodných procesů právě ve finanční problematice. Jak dále uvádí Skalská (2006), řešení pomocí stochastických metod je vhodné v případech, kde dochází k hromadění jednotek procházejících určitým systémem. Typicky se jedná o problematiku vzniku front, čekání ve frontě a modely hromadné obsluhy obecně. Tento model je v zásadě velmi blízký reálným dějům, kde se náhodná proměnná zpravidla vyskytuje.

Jak dále Parzen (2015) uvádí, stochastický proces lze definovat jako soubor náhodných proměnných, který je indexován nějakou matematickou množinou. To znamená, že každá náhodná proměnná daného stochastického procesu je jednoznačně spojena s prvkem v množině. V systému zároveň existuje určitá pravděpodobnost, s jakou dochází ke změnám v systému. Množina využívaná k indexování náhodných proměnných se nazývá množina indexů.

Billingsley (2008) uvádí, že pokud se za náhodnou veličinu uvažuje čas, může mít indexová množina konečný počet prvků a stochastický proces je potom dán diskrétním časem. V opačném případě, kdy je indexová množina z určitého intervalu reálných čísel, stochastický proces je dán spojitým časem.

Zvláštním typem stochastického procesu je proces Poissonových bodů, zjednodušeně nazýván jako Poissonův proces. Lze ho interpretovat jako matematický model s náhodnou posloupností jevů. Je často využíván v teorii hromadné obsluhy, kde slouží k modelování náhodných událostí v systému, jako například příchod zákazníků do systému (Matematicka biologie, 2022).

Jak dále uvádí portal.matematickabiologie.cz (2022), Poissonův proces se vyznačuje následujícími vlastnostmi:

- Nezávislost – počty vstupujících požadavků jsou v různých časových intervalech na sobě nezávislé. Pokud nejsou současné statistické vlastnosti procesu závislé na hodnotách z minulosti, jedná se o Markovův proces, o který je zmiňován v další kapitole.
- Stacionárnost – intenzita vstupu požadavků do systému je konstantní. Pokud platí tato vlastnost, je daný Poissonův proces homogenní.

- Regulérnost – pravděpodobnost příchodu požadavku do systému není ovlivňována dobou od předchozího příchodu.

Kořenář (2010) charakterizuje stochastický proces jako množinu náhodných veličin $\{X_t\}_{t \in T}$, kde T reprezentuje množinu parametrů t . Množina T je chápána jako časová množina a zároveň tvoří definiční obor. Obor funkčních hodnot je nazýván jako prostor stavů stochastického procesu R . Tyto stavy jsou brány jako číselné hodnoty, proto i obor funkčních hodnot je dán jako číselná množina

1.6 Markovovy procesy v systémech hromadné obsluhy

Bellman (1957) definuje Markovovy procesy jako rozhodovací stochastické procesy. Poskytuje zároveň vhodný matematický nástroj pro modelování rozhodovacích procesů v situacích, kdy jsou výstupy z procesu z části náhodné a z části pod kontrolou. Markovovy procesy lze také charakterizovat jejich vlastností, kdy se s danou pravděpodobností v okamžiku n nachází v jednom z možných stavů. Vhodné využití je také v při řešení optimalizačních problémů. Bellman (1957) dále označuje Markovovy procesy jako takové procesy, které splňují Markovovu vlastnost, označovanou také jako bezpaměťovost. Z této vlastnosti proto vyplývá, že existuje nezávislost mezi stavy minulými, současnými i budoucími.

Markovovy procesy jsou využívány již od 50. let 20. století, přičemž jejich název pochází od ruského matematika Andreje Markova. Gagniuc (2017) dále popisuje první Markovovy studie, které se vážou již k počátku 20. století. Od té doby se tento zavedený model dále rozvíjel, zejména v různých odvětvích aplikované matematiky. V současné době nachází uplatnění v řadě oborů, jakými jsou například robotika, ekonomie a finance, řízení výrobních systémů, ale také v jednoduchých systémech hromadné obsluhy. Právě Markovovy řetězce tvoří základ pro analytické zpracování front v rámci teorie front

Jak popisuje Müller (2008), nejjednodušším případem Markovových procesů jsou Markovovy řetězce. Jsou používány pro popis systémů, které se nachází v určitém stavu, kdy počet těchto stavů je konečný.

Jeho vlastností je, že pravděpodobnost, s jakou přechází do následujícího stavu, závisí pouze na aktuálním stavu, a naopak nikterak nezáleží na stavu předchozím. Tuto vlastnost Gagniuc (2017) popisuje jako sled událostí, ve kterém pravděpodobnost jednotlivých událostí závisí pouze na aktuálním stavu, tzn. na tom

dosaženém z předchozí události. Jak uvádí Fačevićová a kol. (2018), tato vlastnost je obecně nazývána markovská.

Z hlediska času se Markovovy řetězce dělí na dva typy:

- Jako diskrétní veličina, kdy se čas mění jen v určitém momentu – často uváděným příkladem je házení hrací kostkou, kdy se výsledek zjišťuje následně po každém hodu.
- Jako spojitá veličina, kdy se čas mění libovolně. Příkladem pro tuto situace může být příchod zákazníků k pokladnám, kdy tito zákazníci přichází kontinuálně v čase. Spojitý čas je typický pro Markovovy řetězce v teorii front.

Jak dále uvádí Kořenář (2002), Markovovy řetězce jsou popisovány pomocí:

1. vektoru absolutních pravděpodobností dle vzorce (4):

$$p(n) = [p_1(n), p_2(n), \dots, p_N(n)], \quad (4)$$

kde $n = 0, 1, 2, \dots, N$ a $p_i(n)$ označuje pravděpodobnost toho, že daný proces je v okamžiku n ve stavu i ,

2. maticí pravděpodobnostní přechodu podle vzorce (5):

$$P(n) = [p_{ij}(n)], \quad (5)$$

kde $i, j = 1, 2, \dots, n$.

Dále se Markovovy řetězce dělí na:

- Homogenní, kde $p_{ij}(n)$ nezávisí na n ,
- Nehomogenní, kde naopak $p_{ij}(n)$ na n závisí.

Typickým užitím Markovových procesů jsou metody Monte Carlo. Jedná se o širokou skupinu výpočetních algoritmů, které při získávání numerických výsledků využívají opakované náhodné výběry.

Tyto metody jsou postavené na konceptu využívání náhodnosti k řešení problémů, které by ale mohly být v principu deterministické. V zásadě lze metody Monte Carlo využívat k řešení problémů s pravděpodobnostní interpretací.

Své praktické uplatnění nalézají například ve statistice, v oblastech fyziky a dále také ve finančních oborech a ekonomii. Obecně je možné říci, že jsou často

používány k řešení problémů, u kterých je téměř nebo zcela nemožné použít jiné přístupy. Je zřejmé, že ve všech zmíněných příkladech hraje roli náhodná veličina. Metoda Monte Carlo, potažmo samotné Markovovy procesy, jsou v oblasti teorie front uplatňovány při studiu front zákazníků v obchodech, kde figuruje náhodná veličina, například u příchodu požadavků do systému. Dalším případem může být vývoj tempomatů v automobilovém průmyslu (Meyn a Tweedie, 2009).

2 Využití softwarových nástrojů a heuristických metod

U modelů hromadné obsluhy, které se v praxi vyskytují, bývá často zapotřebí tyto systémy určitým způsobem řídit, což také znamená znát jejich způsob chování a fungování. K tomu slouží modelování, jehož cílem je dle Cartwright (1983) právě usnadnit pochopení, definování, kvantifikaci, vizualizaci nebo simulaci určité části nebo vlastnosti zkoumaného systému tím, že se odkazuje na existující a obvykle obecně přijímané znalosti. Vyžaduje výběr a identifikaci relevantních aspektů situace v reálném prostředí a následné vytvoření modelu, který by replikoval systém s těmito vlastnostmi. Různé typy modelů mohou být použity pro různé účely, například konceptuální modely pro lepší pochopení, operační modely pro operacionalizaci, matematické modely pro kvantifikaci, výpočetní modely pro simulaci a grafické modely pro vizualizaci zkoumaného systému nebo dílčího procesu.

K tomu dále Sokolowski a Banks (2009) uvádí, že simulace je v praxi využívána zejména v případech, kdy je model, ze kterého simulace vychází, příliš složitý pro analytické řešení. Díky simulaci je možné napodobit fungování zkoumaného reálného systému v čase. Předtím, než je možné přistoupit k samotné simulaci, je zapotřebí vytvořit tzv. pojmový model. Ten představuje popis reálného systému. Je klíčové, aby pojmový model obsahoval všechny vlastnosti, které jsou pro daný systém podstatné. Na druhé straně je důležité zachovat jeho srozumitelnost a vypovídací schopnost díky vynechání aspektů, které nejsou pro danou simulaci důležité. Jde tak o zjednodušený obraz reality s určitým stupněm abstrakce. Pojmový model tvoří základ pro model simulační, kdy je provedeno jeho převedení do virtuálního prostředí s využitím softwaru. Následně je nutné provést kontrolu, zda se vytvořený virtuální model shoduje s pojmovým modelem. Tento proces se nazývá verifikace. Zároveň je nutné ověřit i shodu s reálným systémem, tedy „předlohou“ pro daný model. Tím se rozumí proces validace.

Cílem počítačové simulace je modelování reálné, případně hypotetické situace za využití softwaru tak, aby bylo možné zkoumat a zjišťovat, jak systém funguje. Změnou proměnných v simulaci lze předpovědět chování systému stejně, jak by se se změnila reálné podmínky ovlivňující systém. Simulace lze definovat jako nástroj, který umožňuje virtuálně zkoumat chování zkoumaného systému (Banks, 2001).

Počítačová simulace se stala užitečnou součástí modelování řady různých produkčních systémů a dále mnoha přírodních systémů v oborech, jakými jsou například fyzika, chemie nebo biologie. Uplatnění nalézá i v případě systémů v ekonomii a společenských vědách (např. počítačová sociologie). Banks (2001) dále popisuje příklad užitečnosti počítačové simulace lze v oblasti síťového provozu. Při takových simulacích se chování modelu mění při každé simulaci v závislosti na souboru počátečních parametrů, které se předpokládají pro dané prostředí.

Obvykle se formální modelování systémů provádí prostřednictvím matematického modelu, kde je cílem nalezení analytického řešení umožňující předpovědět chování systému ze souboru parametrů a počátečních podmínek. V praxi existuje mnoho různých typů počítačové simulace, společným znakem všech je snaha vytvořit vzorek reprezentativních scénářů pro model, u něhož by byl úplný výčet všech možných stavů neúnosný nebo nemožný, například z technologického nebo ekonomického hlediska.

Existuje několik softwarových nástrojů pro praktické provádění počítačového simulačního modelování. Dle Bankse (2001) jde například o simulaci Monte Carlo (popisovanou v následující kapitole), stochastické modelování, vícemetodové modelování a další. Díky těmto nástrojům má veškeré modelování téměř hladký průběh. Moderní používání pojmu "počítačová simulace" může zahrnovat prakticky jakoukoli počítačovou reprezentaci libovolného systému.

2.1 Simulační software Tecnomatix Plant Simulation

Tecnomatix Plant Simulation (dále jen PS) je softwarový nástroj vyvinutý společností Siemens Digital Industries Software. Využívá se modelování simulací a dále také k různým vizualizacím, jejichž cílem je optimalizace zpravidla výrobních a logistických systémů nebo dílčích procesů v nich. Pomocí něj lze také optimalizovat, zkoumat a plánovat materiálové toky nebo vytvářet layouty pro výrobní pracoviště (Siemens Software, 2011).

Pomocí PS lze optimalizovat tok materiálu, využití zdrojů a také logistické činnosti na všech úrovních plánování výrobního závodu od globálních výrobních zařízení, přes místní závody až po jednotlivé linky. PS je využíván pro individuální plánování výroby i v rámci nadnárodních společností ke strategickému plánování uspořádání,

logiky řízení a dimenzí velkých, komplexních výrobních investic. Jedná se o produkt, s výrazným obsazením na trhu (v tomto oboru).

Z hlediska vizualizace je PS schopný zobrazovat modelované sekvence jak ve 2D, tak i ve 3D formátu. Zejména formát 3D je vhodný k prezentaci výsledků simulace ve vnitropodnikovém prostředí, například při řešení nového rozložení výrobní linky. Jak uvádí Jallas (2009), 3D zobrazení je vhodné (a často využívané) pro prezentaci osobám, které nepatří mezi experty v této oblasti, a kterým 3D zobrazení pomáhá ve správné interpretaci modelu. V PS se k tomuto účelu využívají defaultně dané grafické objekty nebo je možné použít vlastní personalizované objekty. K vytvoření 3D modelu lze dojít tak, že se nejprve vytvoří model ve 2D a ten se následně převede do 3D. Druhým způsobem je vytváření 3D modelu ihned od počátku. Tento způsob však nese s sebou komplikace v podobě ztráty přehlednosti. 3D zobrazení je charakterizováno formátem JT (Jupiter Tessellation), což je formát standardizovaný dle norem ISO a běžně se využívá pro 3D projekci CAD dat (Bangsow, 2010).

PS pracuje s řadou objektů, které může uživatel při simulaci využít. Mezi ně patří především:

- EventController (pro nastavení času),
- Station (pracoviště),
- Connector (pro propojení objektů, dopravníky apod.),
- AssemblyStation (montážní pracoviště měnící vstupní objekty),
- Source a Drain (vstup a výstup),
- a další.

Tyto zmíněné objekty slouží pro modelování nepříliš složitých modelů, kdy uživatel pracuje pouze se základní charakteristikou objektů. Pro modelování složitějších systémů dobře reflektujících reálné systémy nabízí PS objekt *Method*. Ten je spojen s programovacím jazykem SimTalk, pomocí něhož je možné programovat chování modelu. Objekt *Method* funguje s deklaračními sekcemi: *Is*, *to* a *end*. Ze softwarového hlediska využívá PS 64bitového prostředí, a to pouze s operačním systémem Windows.

2.2 Software MPL

Jak popisuje MPL Modeling System (2016), Software Mathematical Programming Language for Windows, dále jen MPL, je pokročilý modelovací systém, který umožňuje tvůrcům modelů formulovat složité optimalizační modely jasným, stručným a efektivním způsobem. Modely vytvořené v MPL lze poté řešit pomocí některého z mnoha komerčních optimalizátorů, které jsou dnes na trhu k dispozici.

MPL pracuje na bázi algebraického modelovacího jazyku, který umožňuje tvůrci modelu vytvářet optimalizační modely pomocí algebraických rovnic. Model slouží jako základ pro vytvoření matematické matice, kterou lze přenést přímo do řešitele optimalizace, kterým může být například vhodně připravený soubor MS excel. Celý tento proces probíhá na pozadí, takže je zapotřebí soustředit se pouze na formulaci modelu. Algebraické modelovací jazyky, jako je MPL, se v průběhu let osvědčily jako nejefektivnější metoda vývoje a údržby optimalizačních modelů, protože se snáze učí, rychleji se formulují a vyžadují méně programování.

Modelovací jazyk MPL má podobu přirozeného algebraického zápisu. Díky tomu zde existuje možnost formulovat i složité optimalizační modely stručným a přehledným způsobem bez předchozí znalosti některého z programovacích jazyků. Obecně lze říci, že modelovací jazyk MPL nabízí dobré prostředí v oblasti vyjadřování a obecně v uživatelské přívětivosti. Modelovací jazyk MPL byl navržen tak, aby používal a měl přehlednou syntaxi, díky níž je proces formulace modelů v MPL efektivní a produktivní. Je zároveň flexibilní a lze jej použít k formulaci modelů v různých oblastech optimalizace, například v plánování výroby, jejím rozvrhování, v distribučních úlohách až po optimalizaci celého dodavatelského řetězce

MPL Modeling System (2016) dále uvádí vybrané významné vlastnosti jazyka MPL:

- oddělení dat od formulace modelu,
- import dat z různých zdrojů dat,
- nezávislost na konkrétních řešitelských softwarech,
- možnost použití maker pro opakující se části modelu,
- vyloučení částí modelu pomocí nastavení podmínek modelu,
- podmínky WHERE/IF pro řešení zvláštních případů,

- čitelná a užitečná hlášení v případě výskytu chyby.

2.3 Heuristické algoritmy

Heuristika představuje techniku, díky které je v rámci matematických a informačních optimalizačních úloh možné nalézt řešení. Uplatňuje se zejména v situacích, kdy klasické exaktní metody jsou příliš pomalé nebo nedokážou řešení nalézt. Dále se používají tehdy, kdy není možné vytvořit optimalizační matematický model. Její výhodou je možnost programovat ji v některém z programovacích jazyků včetně různých přizpůsobení pro specifické úlohy. Cílem heuristiky je nalézt v přiměřeném čase řešení, které je pro řešitele dostatečně dobré vzhledem k povaze daného problému. Shone, Glazenbrook a Zografos (2019) dále zmiňují numerické experimenty ukazující, že heuristicky založené metody vyžadují jen malé výpočetní úsilí a zároveň nabízí lepší řešení než jiné naivnější rozhodovací metody. Důležitou vlastností heuristických metod je to, že nalezené řešení nemusí být tím nejlepším ze všech řešení daného problému, může se ale přiblížit optimálnímu řešení. Řešení pomocí heuristické metody je řešení přípustné, nelze ho však s jistotou označit za optimální. Mezi heuristické metody patří například metoda severozápadního rohu, metoda maticového minima nebo metoda obchodního cestujícího.

3 Modely prioritního řazení front

Fronta je něčím, s čím se lze setkávat denně a je přirozené, že existuje snaha snižovat čekací čas nebo se frontě vyhnout zcela. V předchozích případech záleželo pouze na pořadí, v jakém požadavky vstoupily do systému a na samotném režimu fronty. Model hromadné obsluhy s prioritním řazením je oproti základnímu modelu specifický z hlediska různé váhy, jakou mají požadavky ve frontě. V prioritním systému hromadné obsluhy se požadavky řadí do různých kategorií dle přiřazené priority s číselným označením od 1 do n . S klesajícím číslem roste priorita. Jinými slovy, pořadí, v jakém budou požadavky obsluhovány, závisí na přiřazené prioritě. Problém nastává ale ve chvíli, kdy jeden požadavek je již obsluhován, zatímco do systému vstoupí požadavek s vyšší prioritou. Jak popisuje Kleinrock (1976), řešení existuje dvojího druhu:

- U preemptivní priority platí pravidlo, že v případě příchodu požadavku s vyšší prioritou se již probíhající obsluha požadavku s nižší prioritou přeruší. Požadavek s vyšší prioritou začne být obsluhován a ten s nižší se vrátí do fronty. Nevýhodou může být nutnost začít přerušenou obsluhu od počátku, pokud na ni nelze navázat. Jedná se o absolutní prioritu.
- V případě nepreemptivní priority se v popsaném případě obsluha požadavku s nižší prioritou nepřeruší. Po jejím dokončení začne obsluha požadavku s vyšší prioritou (ten do té doby čeká). Jedná se o relativní prioritu.

Jak dále uvádí Kleinrock (1976), principem teorie front je výběr, který z požadavků ve frontě bude obslužen nejdříve a který následně. Tento výběr může být založený na některém z režimů fronty v závislosti na tom, jakým způsobem se priorita projevuje. Dále může být výběr založený na členství v určité privilegované skupině nebo na atributu daného požadavku. Takový výběr je potom podstatou teorie front s prioritou. Priorita může být určena podle různých hledisek.

V praxi se lze setkat například s prioritou dle času zpracování požadavku v systému:

- shortest-job-first (SFJ), kdy jsou nejprve přijaty požadavky s nejkratším časem zpracování,
- longest-job-first (LFJ), tedy opačně, kdy jsou nejprve přijaty požadavky s nejdelším časem zpracování,
- a dále obdobná pravidla pracující například s průměrnými hodnotami apod.

Jako příklad prioritního modelu bývá uváděn příklad zakázek ve výrobě. Přijaté zakázky jsou zadávány do výroby na základě jejich důležitosti. To znamená, že místo běžného zadávání těchto zakázek metodou FIFO je zakázkám přidělována priorita, díky které jsou obsluhovány dříve.

Prioritou lze nazvat vlastnost prvků používanou v systémech hromadné obsluhy, kdy je některým prvkům přiřazena určitá preference oproti ostatním prvkům. Priorita je obvykle znázorněna pomocí přirozeného čísla. Číslo jedna je dáno prvků s nejvyšší prioritou a se zvyšujícím se číslem klesá priorita prvku vůči ostatním prvkům (Dolores a kol., 2009).

Systémy hromadné obsluhy, kde existuje priorita, patří k systémům, u nichž nelze odvodit jejich základní charakteristiky analytickým způsobem. Analytické odvození je možné pouze u velmi jednoduchých systémů. U složitějších systémů, kam se řadí i systémy s prioritou, lze získat jejich charakteristiky pouze pomocí experimentů. Tyto experimenty se v současné době provádí s využitím vhodného softwaru, jakým je například zmiňovaný PS (Jablonský, 2011).

3.1 Obecný prioritní model fronty

Pro obecný prioritní model platí, že je předpokládán příchod požadavku (někdy nazývaného zákazník) do systému, který má dané parametry. Ty determinují jeho pozici ve frontě. Zároveň tyto parametry odlišují daný prvek od jiných prvků ve frontě.

Parametry jsou následující:

- Čas t , který je dán jako funkce přiřazených parametrů času obsluhy a času stráveného v systému.
- Parametr p nabývající hodnot $p = 1, 2, \dots, P$, který určuje prioritní skupinu a zároveň numericky vstupuje do prioritní funkce $q_p(t)$ v čase t . Vyšší hodnota této funkce určuje vyšší prioritu. To znamená, že jako první je z fronty vybrán prvek s nejvyšší hodnotou $q_p(t)$ (Kleinrock, 1976).

Je uvažován model ve tvaru M/M/1, případně v obecnějším tvaru G/G/1. Potom existuje předpoklad, že zákazníci z prioritní třídy P (jednotlivé prioritní skupiny s indexem p) přichází do systému s intenzitou λ za předpokladu Poissonova rozdělení, každý požadavek z této skupiny má svůj čas obsluhy daný nezávisle z rozdělení $B_p(x)$. Proto existuje následující vztah vyjádřený vzorci (6) až (9):

$$\lambda = \sum_{p=1}^P \lambda_p, \quad (6)$$

$$\bar{x} = \sum_{p=1}^P \frac{\lambda_p}{\lambda} \bar{x}_p, \quad (7)$$

$$\rho_p = \lambda_p \bar{x}_p, \quad p = 1, 2, \dots, P, \quad (8)$$

$$\rho = \lambda \bar{x} = \sum_{p=1}^P \rho_p, \quad (9)$$

kde ρ zde značí obsazenost obslužného zařízení v daném okamžiku. Pro zjednodušení může být uvažován model, ve kterém se vyskytují pouze dva požadavky. První z nich má nepreemptivní (relativní) prioritu. V případě započetí obsluhy druhého požadavku tedy nebude v případě příchodu požadavku s prioritou tato obsluha přerušena. Rozložení počtu příchozích požadavků je dáno

Poissonovým rozdělením, intenzita příchodů je i nyní značena λ . Požadavky s prioritou přichází s intenzitou λ_1 , požadavky bez priority s intenzitou λ_2 .

Jak uvádí Slaninová (2015), platí tedy, že $\lambda = \lambda_1 + \lambda_2$. Dále je dán index (relativní) priority pro požadavky s prioritou $r = 1$, pro požadavky bez priority $r = 2$.

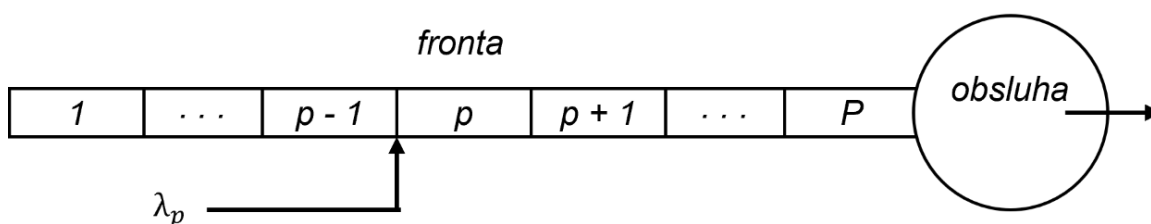
3.2 Prioritní model Head-of-the-Line

Ve srovnání s ostatními modely využívajícími prioritu je model head-of-the line (dále HOL) používaný nejčastěji a lze to také označit za nejpřirozenější. Tento model byl poprvé popsán v roce 1954 a je také nazýván jako fronta s pevně danou prioritou (Kleinrock, 1976).

Princip tohoto modelu je následující. Jak je znázorněno na Zdroj: (Kleinrock, 1976)

Obr. 6, v systému jsou vytvořeny skupiny, které spojují požadavky se stejnou prioritou. Požadavky, které přichází do systému, jsou striktně oddělovány na základě toho, do jaké prioritní skupiny patří. Příchod požadavku ze skupiny p se poté zařadí do fronty za všechny požadavky ze skupiny p a vyšší, a zároveň před všechny požadavky ze skupiny $p - 1$ a nižší. Hodnota přidělené priority se v čase nemění, je tedy možné ji zapsat pomocí funkce dle vzorce (10):

$$q_p(t) = p, \quad p = 1, 2, \dots, P. \quad (10)$$



Zdroj: (Kleinrock, 1976)

Obr. 6 Model Head-of-the-Line

3.3 Prioritní model Time-Dependent Priority

Další způsob, jakým může být priorita implementována do daného systému, představuje model Time-Dependent Priority. Podstatu tohoto modelu lze ukázat na studii autorů Min a Yih (2014). Ti se zabývají teorií front v systému nemocničního příjmu pacientů. Vzhledem ke zdravotním problémům a úrazům, které mají přítomní pacienti (v roli požadavků), je jim přidělována priorita. Způsob, jakým je priorita přiřazována různým pacientům, se snaží zohlednit jednat urgentnost daného zranění a také to, aby pacienti strávili čekáním jen nezbytně dlouhou dobu. Studie tedy bere v úvahu úroveň urgency a čekací dobu, z čehož skládá prioritní skóre. Na základě získaných dat dochází k závěru, že právě využití modelu s Time-Dependent prioritou je vhodným nástrojem, jak prioritu využít. V tomto případě tak i pacienti s nízkou prioritou (tzn. s nenaléhavým problémem) stráví ve frontě méně času, než v případě užití HOL modelu, kde by byli neustále odsunováni příchozími pacienty s naléhavějším problémem. Je to díky tomu, že jejich neurgentní problém s postupujícím časem vyvažuje vyšší hodnota čekací doby.

Jak popisuje Kleinrock (1976), dalším případem využití HOL modelu je pracoviště servisu automobilů. Pokud se v daném čase nahromadí větší počet automobilů čekajících na opravu, servisní technik vybere k servisu vůz, u kterého je čas potřebný k servisu nejkratší. I v tomto případě se priorita mění v čase.

3.4 Využití bribingu pro pozici ve frontě

Výše popisované modely prioritních front mají nejméně jednu společnou vlastnost. Tou je skutečnost, že požadavky, které si lze v tomto příkladu zaměnit za zákazníky, nemají možnost jakkoliv ovlivnit, do jaké prioritní skupiny budou zařazeny. Pasivně tedy přijímají svou pozici.

Teorie bribingu se zabývá problémem, jak umožnit zákazníkům (prvkům ve frontě) ovlivnit svou pozici ve frontě neboli jak využít úplatky. Výše úplatku závisí jednak na bohatství daného prvku, ale také na jeho neochotě čekat ve frontě.

Jak uvádí Kleinrock (1976) pomocí rovnice (11), je uvažován systém M/G/1, volitelné funkce pravděpodobnostního rozdělení časů trvání obsluhy $B_{(y)}$. Dále je dán požadavek nabízející úplatek označovaný jako Y a požadavek nenabízející úplatek y , z funkce libovolného rozdělení.

$$\beta_{(y)} = P[Y \leq y] \quad (11)$$

Předpokladem je, že intervaly mezi příchody, časy trvání obsluhy a také úplatky jsou náhodné veličiny pro každého zákazníka. Systém funguje následovně: Zákazník vstupující do systému nabízí úplatek Y . Jako příjemce úplatku si lze představit někoho, kdo zde figuruje jako organizátor fronty. Důležitým faktem je, že zákazník poskytuje úplatek ještě předtím, než zjistí délku fronty. Z tohoto důvodu je výše úplatku nezávislá na délce fronty. Zákazník je následně umístěn do fronty na pozici, která odpovídá výši úplatku Y . Zákazníci před ním poskytli úplatek $Y' \geq Y$, naopak zákazníci za ním poskytli úplatek $Y' < Y$. Podle těchto vztahů budou umístování zákazníci, kteří přijdou později. V případě stejné výše úplatku je uplatňováno pravidlo FIFO.

Úplatek zmiňovaný v této kapitole je zapotřebí brát nikoliv jako negativní prvek. Systém využívající tento druh úplatku je stabilní a udržitelný pouze za předpokladu, že existuje určitá vytvořená politika, která užívání úplatků reguluje. Předmětem této regulace je zejména to, aby zákazníci nevyužívaly úplatky k odsunutí jiného zákazníka z fronty nebo také ke zhoršení jeho pozice. Předpokladem je i nepreemptivní (relativní) priorita a nelze tedy poskytnout takový úplatek, který by zastavil obsluhu již obsluhovaného zákazníka.

3.5 Prioritní fronta s využitím haldy

Dle Schierreicha (2018), prioritní fronty se často vyskytují v informatice, kdy se s prvky fronty pracuje jako s daty. Využívá se zde tzv. halda. Pojem halda představuje v informatice specializovanou stromovou datovou strukturu se zvláštní vlastností. Tato vlastnost je dána vztahem $x(A) \geq x(B)$, kde B je potomek A . Jak poté uvádí algoritmy.net (2015), všechny prvky v prioritní frontě musí nést klíč, který určuje jejich prioritu. Klíče společně tvoří uspořádanou množinu. V závislosti na chápání klíče lze rozlišit dvě různé varianty řazení. První případ uvažuje na začátku fronty, tedy v kořeni stromu, klíč s nejvyšší hodnotou. Tato halda je označována

jako $heap_{max}$. Ve druhém případě je na začátku fronty naopak klíč s nejnižší hodnotou a halda je označována jako $heap_{min}$. Halda je ve spojení s prioritní frontou využívána zejména pro zefektivnění jejího fungování a obecně ke zlepšení jejího výkonu.

Cormen (2009) popisuje haldu jako jednu z nejefektivnějších implementací datového typu představujícího prioritní fronty. V haldě je vždy prvek s nejvyšší nebo naopak nejnižší prioritou uložen v kořeni této stromové datové struktury. Zároveň však halda není seřazená datová struktura, lze ji považovat pouze za částečně organizovanou. Důvodem je, že není určeno pořadí potomků, kteří jsou sourozenci. Oproti tomu ale umožňuje efektivně a opakovaně odstraňovat prvky s nejvyšší (nebo i nejnižší) prioritou.

V případě hald a operací s nimi se uvažuje tzv. asymptotická složitost. Ta udává, kolik elementárních operací je daný algoritmus schopen provést. Společně s operační náročností programu představují nástroje, kterými lze srovnávat rychlost a také efektivitu různých algoritmů. Díky asymptotické složitosti lze zjišťovat, jak se změní chování daného algoritmu při změně vstupních dat. Dle složitosti se algoritmy dělí do tříd, kdy platí, že algoritmus náležící do dané třídy je vždy pomalejší než algoritmus z předchozí třídy. Na výkonu počítače přitom nezáleží. Zápis se provádí pomocí Landauovy notace, například v podobě $O(n)$, kde n je veličina udávající velikost vstupních dat (Kleinberg a Tardos, 2014). V tomto případě zápis říká, že doba trvání algoritmu závisí lineárně na velikosti vstupních dat. Oproti tomu zápis z třídy $O(n^2)$ znamená, že se doba trvání algoritmu zvyšuje kvadraticky vůči změně velikosti vstupních dat. Pro přehled lze uvést příklad, kdy existují dva algoritmy $O(n)$ a $O(2n)$. Oba náležejí do stejné třídy složitosti a při využití počítače o dvojnásobném výpočetním výkonu u $O(2n)$ bude čas výpočtu totožný. Pokud by šlo ale o různé třídy, například $O(n)$ a $O(n^2)$, bude algoritmus se složitostí $O(n^2)$ trvat déle i při použití výkonnějšího počítače kvůli kvadratickému růstu času vůči růstu množství vstupních dat (algoritmy.net, 2015).

Algoritmy.net (2015) dále uvádí rozdělení hald dle svého uspořádání například na:

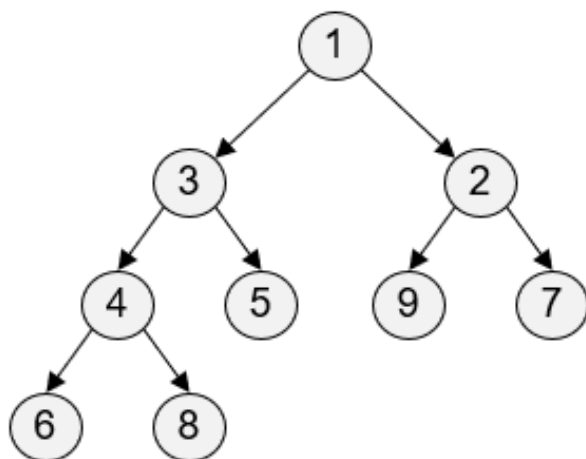
- binární haldu, která se využívá u prioritních front,
- binomiální haldu,
- Fibonacciho haldu a další.

Binomiální halda je svými vlastnostmi podobná binární haldě. Je vhodná zejména pro Dijkstrův algoritmus pro hledání nejkratší cesty v grafu. Z binomiální haldy vychází Fibonacciho halda, jejíž výhodou je nízká asymptotická složitost. Jarníkův a Dijkstrův algoritmus jsou nejčastějšími aplikace Fibonacciho haldy (Cormen, 1990).

3.6 Binární halda

Jak uvádí Kamp (2010), binární halda je jednoduchý typ haldy, kdy strom má binární podobu. Byla uvedena v roce 1964 J. W. J. Williamsem a v současnosti je běžně využívána pro implementace prioritních front. Binární strom může být buď perfektně vyvážený, kdy jsou všechny jeho listy na stejné úrovni, nebo nekompletní, kdy poslední úroveň haldy není zcela zaplněná a uzly tak naplňují strom zleva doprava. Příklad binárního stromu znázorňuje Zdroj: (Algoritmy.net, 2015)

Obr. 7.



Zdroj: (Algoritmy.net, 2015)

Obr. 7 Příklad binárního stromu

Vyskytuje se zde rekurzivní vlastnost, tedy že haldami jsou i všechny podstromy původního stromu (algoritmy.net, 2015). Tato vlastnost se projevuje při použití řadícího algoritmu *heapsort*, který pracuje na principu porovnávání dvou prvků. Platí také základní vlastnost obecné haldy, že klíč uzlu je vždy větší nebo roven klíčům uzlů jeho potomků (v případě *heap_{max}*). Binární strom B_k je charakterizován:

- výškou k , kde k je hodnota klíče,
- obsahem 2^k vrcholů,
- obsahem $\binom{k}{i}$ vrcholů na i -té úrovni, kdy $i = 0, 1, \dots, k$.

V rámci prioritní fronty lze provádět následující operace:

- přidání prvku x do množiny M ,
- výběr prvku z množiny M s minimálním klíčem,
- výběr prvku s minimálním klíčem a jeho následné odebrání z množiny M ,

kdy množinou M se zde rozumí soubor prvků (požadavků) ve frontě. Jak uvádí Algoritmy.net (2015), mimo tyto operace existuje dále řada operací, které se využívají při programování.

Mezi programovací příkazy patří například:

- *heapify* – zkonstruovat v zadaném poli haldy,
- *insert* – přidání prvku na konec haldy,
- *top* – operace vracející hodnotu prvku s nejvyšší prioritou,
- *return top* – stejně jako *top*, navíc ale tento prvek z haldy odebere,
- *merge* – sloučení dvou hald do jedné.

Následující Tab. 2 zobrazuje porovnání asymptotické složitosti vybraných operací pro výše uvedené typy hald.

Tab. 2 Vybrané typy hald a jejich asymptotické složitosti

Typ haldy	Příkaz		
	createHeap	insert	merge
Binární	$\theta(1)$	$O(\log n)$	$\theta(n)$
Binomiální	$\theta(1)$	$\theta(\log n)$	$\theta(\log n)$
Fibonacciho	$\theta(1)$	$\theta(1)$	$\theta(1)$

Zdroj: (Algoritmy.net, 2015)

V Tab. 2 se vyskytují dva případy interpretace asymptotické složitosti. Obecně platí, že předpis $O(f(x))$ značí, že daný algoritmus běží stejně rychle nebo rychleji než $f(x)$. Ve druhém případě $\theta(f(x))$ běží algoritmus stejně rychle jako $f(x)$. Dále existuje i třetí případ $\Omega(f(x))$, kdy algoritmus běží stejnou nebo nižší rychlostí než $f(x)$ (Algoritmy.net, 2015).

3.7 Využití haldy v prioritních frontách

Jak uvádí Ioannou a Katevenis (2007), halda jako struktura dat slouží především k jednomu účelu, a to nalezení maxima nebo minima dané sady čísel. To je zároveň i cílem prioritních front, kdy se prvky tvořící frontu řadí dle daného pravidla. Využití haldy je vhodné zejména v případech, kdy je fronta tvořena velkým množstvím prvků, například v řádu tisíců. V těchto případech je zapotřebí využití daných softwarových nástrojů, které vyžadují práci s frontou, jako se specifickou strukturou dat. Obecně je více využíván typ min heap.

Sedgewick (2003) popisuje, že z hlediska využití haldy a programování obecně je implementace prioritní fronty možná různými způsoby. Z pohledu datových struktur lze použít spojovací seznamy, netříděná pole nebo haldy. Využití haldy s sebou nese výhodu v podobě nižší asymptotické složitosti, oproti tomu je však náročnější na programování.

Jak bylo zmíněno, nejčastěji je využívána halda binární, lze však použít i haldu binomiální nebo Fibonacciho, které mohou být v konkrétních případech vhodnější, především z hlediska asymptotické složitosti.

4 Modely rozvrhování

Tématem této kapitoly je formulace modelu rozvrhování dávek na procesory. V tomto případě se jedná o procesory uspořádané paralelně. Je proto možné jednotlivé dávky obsluhovat současně. Existuje řada různých variant tohoto modelu, proto se tato kapitola zaměřuje na konkrétní model formulovaný zápisem $P_m || F_{max}$. Dle Fialy (2002) zde F_{max} představuje proměnnou, která je dána jako nejdelší doba pobytu dávky v systému, kdy cílem je tuto proměnnou minimalizovat. Součástí tohoto modelu je m paralelně uspořádaných procesorů a n jednotlivých dávek. Cílem úlohy je veškeré dávky zrealizovat na některém z procesorů. Předpokladem je zde nemožnost přerušení dávek. Zároveň neexistují mezi dávkami žádné precedenční relace, procesory jsou totožné a neexistuje ani omezení v podobě nejdříve možných začátků obsluhy.

V následujícím matematickém modelu je předpokládána znalost t_j , což je doba trvání j -té dávky. Binární proměnná x_{ij} nabývá hodnoty 1 v případě, že j -tá dávka je zpracována na i -tém procesoru.

Účelová funkce (12) pro tento model představuje cíl úlohy minimalizovat hodnotu F_{max} :

$$z = F_{max} \rightarrow \min. \quad (12)$$

V modelu jsou dále uvedeny následující omezující podmínky:

$$\sum_{j=1}^n x_{ij} t_j \leq F_{max}, \quad i = 1, 2, \dots, m, \quad (13)$$

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, 2, \dots, n. \quad (14)$$

Nerovnice (13) zaručují, že součet dob trvání dávek přiřazených na daný procesor nepřekročí nejdelší dobu pobytu dávky v systému. Soustava rovnic (14) zajistí, že

každá dávka bude přiřazena právě na jeden procesor. Tento matematický model bude v následující kapitole upraven pro úlohu dobíjení elektromobilů.

Jak dále uvádí Fiala (2002), tato úloha se vzhledem k její výpočetní složitosti řadí mezi tzv. úlohy s NP-obtížností. Z toho důvodu je zde formulován heuristický algoritmus, díky kterému lze v některých případech nalézt i optimální řešení. U běžných úloh z reálného prostředí se ale výraznější problémy s řešitelností obvykle nevyskytují.

Heuristický algoritmus pro tuto úlohu se skládá z následujících tří fází:

Fáze 1

V první fázi jsou jednotlivé dávky seřazeny v následujícím pořadí (15). Hodnota indexu $[k]$ zde obecně představuje index dávky, která je v daném rozvrhu na k -tém místě v pořadí:

$$t_{[1]} \geq t_{[2]} \geq \dots \geq t_{[n]}. \quad (15)$$

Fáze 2

V druhé fázi se ze sestaveného pořadí (15) vybere prvních m dávek. Každá dávka je umístěna na samostatný procesor, začátek realizace je v čase 0.

Fáze 3

Poté, co jsou první dávky rozvrženy podle fáze 2, jsou vybírány další dávky v pořadí. Ty jsou řazeny postupně za poslední již umístěnou dávku na procesor, který se uvolní nejdříve. To znamená, že tato dávka bude umístěna za tu dávku, která bude nejdříve dokončena. Takto se postupuje až do rozmístění všech dávek.

Dále v práci bude tato metoda modifikována pro úlohu dobíjení elektromobilů. Zároveň je v této modifikaci využita priorita vybraných dávek.

5 Aplikace modelů hromadné obsluhy v reálném prostředí

Již několik let probíhá na evropském a také asijském a americkém trhu k poměrně silnému trendu rozvoje elektromobility, především v oblasti automobilového průmyslu. Tento trend je dán zejména stále významnějším důrazem na ochranu klimatu a s tím související legislativou. Typickým příkladem je právě evropský automobilový trh, který podléhá zmiňovaným emisním normám EURO. Tyto normy jsou pro evropské prodejce závazné, resp. platí v rámci celého Evropského hospodářského prostoru, a stanovují limitní hodnoty pro škodliviny ve výfukových exhalacích benzinových a dieselových motorů osobních a dalších vozů. Aktuálně je pro osobní vozy platná emisní norma EURO 6, přičemž je očekáváno schválení normy Euro 7, která by ještě mnohem více upřednostnila alternativní pohony (Stratstone, 2022).

Trendy v automobilovém průmyslu tedy aktuálně směřují k elektřině jako k součásti budoucnosti. Dnes se můžeme setkat jak s modely čistě na elektrický pohon, tak i s modely hybridními, využívajícími kombinaci dvou různých pohonů, například elektrického a spalovacího motoru. Zároveň lze jen těžko najít výrobce, který by ve svém portfoliu neměl některý z těchto vozů. Jak uvádí Acatech (2022), situace z pohledu evropských zákazníků je ale jiná. Dle jejich průzkumů provedených v Německu vychází, že zájem o koupi elektromobilu má zhruba jen 25 % dotazovaných, v případě hybridních vozů 31 % dotazovaných. Dle E15 (2022), je právě toto zjištění protichůdné oproti krokům Evropské komise. Ta mimo zavedení emisní normy EURO 7 směřuje i k úplnému zákazu prodeje vozů se spalovacím motorem. Jinými slovy jsou evropští producenti v situaci, kdy musí přesvědčit zákazníky o výhodách elektromobilů. Zároveň ale ceny elektromobilů jsou pro značnou část zákazníků neakceptovatelné, jak ukazuje zmiňovaný průzkum. Pro představu lze v Tab. 3 uvést ceny některých elektromobilů prodávaných v České republice.

Tab. 3 Srovnání cen vybraných elektromobilů

Model	Základní cena v Kč
Dacia Spring	464 900
Renault Zoe	745 000
Nissan Leaf	779 000
Hyundai Kona electric	789 990
Peugeot e-208	800 000
Hyundai Ioniq 5	1 169 900
Škoda Enyaq iV	1 179 900

Zdroj: (Aktuálně.cz, 2022)

Podle zmiňovaného průzkumu ceny elektromobilů nevyhovují 69 % respondentů. Otázkou v oblasti elektromobility zůstává i reálný vliv na životní prostředí. Zde záleží především na energetickém mixu pro danou oblast či danou zemi. Z toho vyplývá, že velmi záleží, kde daný uživatel vozu vůz nabíjí. Může to být například na veřejné nabíjecí stanici, kde může elektrická energie pocházet z uhelné elektrárny. Na druhé straně uživatelé vozy nabíjí i na vlastních nabíječkách, kde elektrická energie pochází z fotovoltaických panelů umístěných například na střeších. Zároveň lze čekat i postupné zlepšování z pohledu energetického mixu. Velmi důležitým faktorem je skutečnost, že elektromobily neprodukují emise tam, kde jezdí. To v praxi znamená snížení emisí na exponovaných místech, hlavně tedy ve městech (E15, 2022).

S elektromobilitou je úzce spojena i problematika nabíjení. S tím se v praxi vyskytuje i řada mýtů a nepodložených tvrzení, například, že nabíjení trvá několik hodin, na jedno nabití je dojezd příliš nízký nebo že nabíjecích stanic je málo a je složité nějakou najít, navíc aktuálně neobsazenou. Vývoj, jakým elektromobilita prošla a stále prochází, dokázal řadu těchto tvrzení vyvrátit.

Existují ale i nabíjecí stanice, kde se výkon pohybuje od 150 do 300 kW. Při tomto výkonu lze elektromobil dobít v řádu několika minut při nabíjení na 80 % kapacity baterie. Nabíjení na tuto úroveň je doporučováno přímo výrobcí elektromobilů

vzhledem k šetrnějšímu zacházení s baterií a také z důvodu, kdy nabíjení na plných 100 % kapacity baterie velmi prodlužuje čas nabíjení. Důvodem je ale i fakt, že nabíjení na plných 100 % výrazně prodlužuje čas dobíjení s ohledem na křivku průběhu nabíjení lithiových baterií. Nejefektivnější způsob z pohledu rychlosti nabíjení je proto takový, kdy se baterie dobíjí v rozmezí od 20 do 80 %. Společně s nabíjecími stanicemi se rozvíjí i systém plánování nabíjení pomocí řady mobilních aplikací. Obdobně funguje i způsob placení, kde trendem je sjednocování pod jeden platební prostředek. Jde tak o výrazný posun, kdy uživatel již nemusí mít čip, aplikaci nebo ještě další prostředek pro nabíjení u různých poskytovatelů. Za příklad lze uvést službu Powerpass, kterou poskytuje společnost ŠKODA AUTO, a.s. Jedná se o tarifní kartu, která umožňuje nabíjení na většině dobíjecích stanic v České republice i Evropě (ŠKODA AUTO, 2022)

Nejvýraznější transformací na elektromobilitu představuje rychlý nárůst dobíjecích stanic. Příkladem v české republice je společnost ČEZ, která provozuje více než 400 dobíjecích stanic (Emobilita, 2022). S rostoucím počtem elektromobilů se budou rozšiřovat i sítě veřejných dobíjecích stanic pro tato vozidla. S tím souvisí i stále větší zatěžování elektrické sítě. Said, Cherkaoui a Khoukhi (2015) ve své práci navrhuje modely, které mají vyřešit přetížení elektrické sítě. Tyto modely pracují s myšlenkami, v nichž vozidla komunikují se sítí a předávají si informace o své potřebě nabíjení a poloze. Ve své podstatě se jedná o modely hromadné obsluhy, kde se pracuje jak s klasickými frontami v režimu FIFO, tak i s prioritními frontami. Zároveň s růstem počtu dobíjecích stanic roste postupně i jejich výkon. Nabíjecí stanice o výkonu 50 kW lze již označit za určitý standard.

Systémy hromadné obsluhy v oblasti elektromobility

Systémy hromadné obsluhy se v reálném prostředí vyskytují velmi často. V celé řadě případů dochází k obsluze v režimu FIFO, kdy mají všechny požadavky stejnou váhu. Oproti tomu ale dochází i k situacím, kdy je přinejmenším žádoucí některé vybrané požadavky upřednostnit na úkor ostatních, tzn. udělit jim prioritu. Priorita v případech provozu elektromobilů může být přiřazována na základě jejich stavu nabití, potřebné doby odjezdu po nabití apod. Pro optimalizaci procesů v rámci hromadné obsluhy je zapotřebí vytvořit model, který prezentuje vybrané procesy a pomocí kterého je možné tyto procesy dále řešit.

5.1 Model pro dobíjení elektromobilů

S výše uvedeným souvisí i změna infrastruktury. Společně s nárůstem počtu vozů na elektrický pohon roste i počet nabíjecích stanic. Protože ale v současné době není často počet nabíjecích stanic dostatečný a zároveň současná technologie neumožňuje nabíjení stejně rychlé jako tankování paliva, jako například automobilového benzínu nebo motorové nafty, dochází logicky ke vzniku front u nabíjecích stanic. Na celý tento systém lze proto nahlížet jako na systém hromadné obsluhy. Přijíždějící vozy představují příchozí požadavky vstupující do systému, nabíjecí stanice poté představují obslužná pracoviště. Aby bylo možné systém hromadné obsluhy zkoumat, řídit a také optimalizovat, je nutné za tímto účelem vytvořit model reprezentující zkoumaný systém. U dále popisovaného modelu i u jakéhokoliv obecného modelu platí vlastnost, že se jedná o zjednodušený obraz reality s určitým stupněm abstrakce.

Tento systém může představovat stochastický proces, kdy intenzita příchodů i intervaly mezi příchody jsou v čase proměnlivé a neexistuje mezi nimi žádná pravidelnost. Provoz vozů není v tomto případě nikterak pravidelný a jejich vybíjení probíhá zcela nahodile.

5.1.1 Pojmový model

Je uvažován systém hromadné obsluhy, kde probíhá nabíjení vozů u nabíjecích stanic. Tento systém je reprezentován vytvořeným modelem. Pro zajištění reálného výstupu (čas nabíjení a obsazenost stanic) z tohoto modelu jsou vybrána reálná data, i když model jako takový je uvažován obecně. V tomto systému se nachází celkem 4 nabíjecí stanice a 20 vozidel na elektrický pohon. Nabíjecí stanice jsou totožné a jsou typu DC, tedy fungují na bázi stejnosměrného proudu, o výkonu 50 kW. Vozy jsou v tomto případě reprezentovány modelem ŠKODA ENYAQ iV 80, jehož technické parametry vztahující se k modelu jsou uvedeny v Tab. 4. V systému tedy dochází k provozu těchto vozů, čímž dochází k jejich postupnému vybíjení. Požadovaný stav nabití baterií vozů je 80 %. V případě, že je stav nabití nižší, je vůz přistaven na parkoviště a čeká na nabíjení (pokud jsou stanice obsazeny). Pokud je některá z nabíjecích stanic v tomto čase volná, vůz je nabíjen ihned. K nabíjení je určen časový úsek daný pracovní dobou 8 hodin, tedy 480 minut, zkrácenou o 15 minut na začátku i na konci pracovní doby.

Časové omezení a relativně vyšší počet vozidel oproti počtu nabíjecích stanic vytváří společně problém. Ten vyžaduje řešení, díky kterému bude možné zrealizovat nabíjení všech vybitých vozidel v rámci jedné pracovní doby.

Tab. 4 Technické specifikace ENYAQ iV 80 vztahující se k modelu

Technické specifikace vozu	ENYAQ iV 80
Celková kapacita baterie (kWh)	82
Využitelná kapacita baterie (kWh)	77
Nabíjení – DC (5–80 %) – výkon (kWh)/čas (h/min)	125 kW/38 min

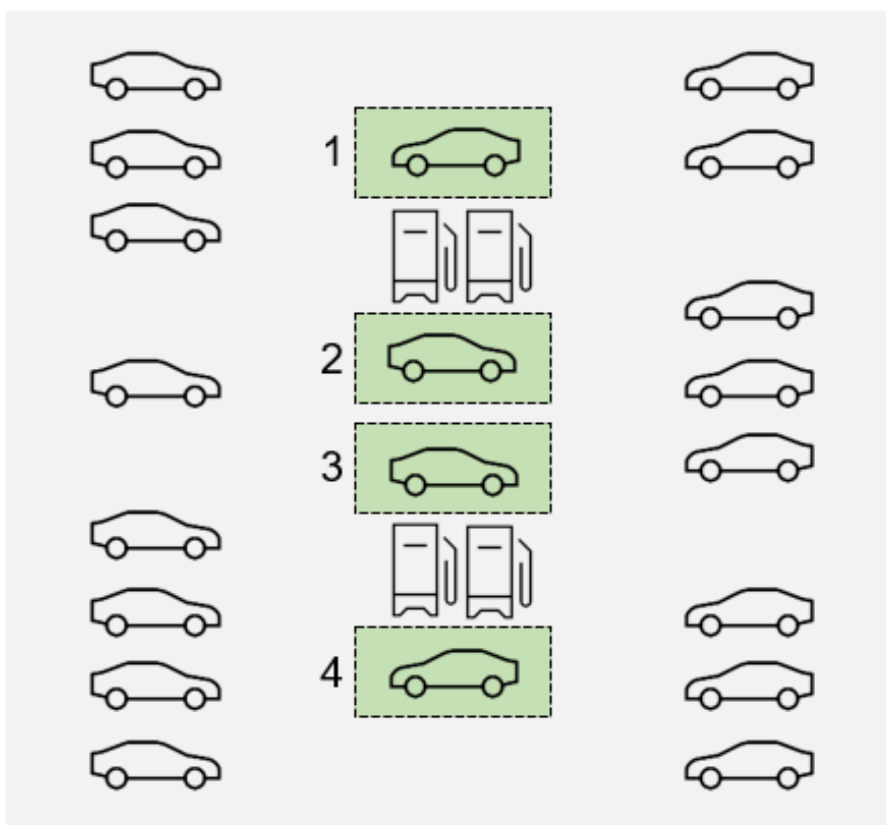
Zdroj: (ŠKODA AUTO, 2022)

Obr. 8 zobrazuje layout tohoto modelu. Jak bylo popsáno, nachází se v něm 4 dobíjecí stanice. Ty jsou umístěny uprostřed plochy parkoviště. Místa, kde stojí vozy během nabíjení jsou značena zeleně. Vozy, které se právě nenabíjí, se nachází po stranách parkoviště, 10 vozů vlevo i vpravo. V praxi to znamená, že při potřebě nabíjení jednotlivých vozů je zapotřebí počítat i s časem manipulace. Ten zahrnuje pohyb vozu k nabíjecí stanici, jeho následné připojení a po skončení nabíjení naopak odpojení a pohyb zpět na parkovací místo. Při nabíjení vozů neexistuje žádné omezení, které by bránilo nabíjení vozu na kterékoliv stanici. Každý vůz může být proto umístěn na kteroukoliv z aktuálně volných stanic.

Tab. 4 zobrazuje údaj k nabíjení z 5 % na 80 %. Pro potřeby tohoto modelu je ale uvažováno nabíjení z počáteční hodnoty 20% a vyšší. Horní hranice zůstává na 80 % kapacity baterie. Z toho vyplývá, že je vždy zapotřebí nabít maximálně 60 % kapacity. Vzhledem ke skutečnosti, že model nezkoumá problematiku nabíjení baterií, je zde průběh nabíjení v pásmu 20-80 % brán jako lineární. Pro výše uvedený model ENYAQ představuje 60% kapacity baterie 49,2 kWh. Při nabíjení z nabíjecí stanice o výkonu 50 kW bude proces nabíjení trvat při zaokrouhlení 60 minut, tedy za každou minutu se baterie nabije o 1 % z celkové kapacity. Model zároveň nepředpokládá a nezohledňuje výrazné klimatické vlivy (hluboké mrazy,

vysoké teploty apod.), které by ovlivňovaly model a měnily dobu potřebnou k nabití každého vozu.

Cílem tohoto modelu je optimalizace provozu dobíjecích stanic a dále výzkum toho, jakým způsobem se bude systém chovat při vysoké poptávce vozů po nabíjecích stanicích při jejich vybití. Cílem je i zjistit, zda je počet nabíjecích stanic vůči počtu vozů dostatečný.



Obr. 8 Layout systému

5.1.2 Matematický model

Následující model, který pracuje s parametry uvedenými v Tab. 5, je modifikací matematického modelu uvedeného v kapitole 4. Modifikace spočívá v přidání dalších parametrů s cílem získat reálný výstup.

Tab. 5 Parametry modelu

Označení	Význam
n	počet vozidel v systému
m	počet dobíjecích stanic v systému
T	pracovní doba (min)
d_i	stav nabití baterie u i -tého vozidla, kde $i = 1, 2, \dots, n$
s_i	požadovaný stav nabití i -tého vozidla, $i = 1, 2, \dots, n$
a_j	rychlost nabíjení j -té dobíjecí stanice (tzn. jaký podíl kapacity baterie se dobije za 1 minutu), kde $j = 1, 2, \dots, m$
r_i	Doba manipulace s i -tým vozidlem (tzn. čas strávený přesunem vozu mezi parkovištěm a nabíjecí stanicí), kde $i = 1, 2, \dots, n$

Model mimo výše uvedené parametry pracuje i s proměnnými veličinami. Jedná se o proměnné týkající se obsazenosti jednotlivých stanic, času nabíjení jednotlivých vozů a maximálního času nabíjení na stanici. Tyto proměnné jsou dále uvedeny v Tab. 6.

Účelová funkce z je dána dle vztahu z kapitoly 4 (16):

$$z = R \rightarrow \min \quad (16)$$

Z účelové funkce vyplývá, že cílem modelu je minimalizovat veličinu F_{max} , resp. R . Jinými slovy je cílem minimalizovat hodnotu nejdelší doby nabíjení způsobem, aby nebyl překročen celkový čas T .

Tab. 6 Proměnné modelu

Proměnná	Význam
x_{ij}	Binární proměnná nabývající hodnoty 1 tehdy, když se bude i -té vozidlo nabíjet na j -té dobíjecí stanici, hodnoty 0 jinak.
t_{ij}	Doba nabíjení i -tého vozidla na j -té dobíjecí stanici. Jedná se o reálnou proměnnou.
R	Nejdelší celková doba nabíjení na stanici. Jedná se o reálnou proměnnou.

Omezující podmínky

Jsou stanoveny podmínky omezující model vyplývající z reálných předpokladů pro tuto úlohu.

$$\sum_{i=1}^n (t_{ij} + r_i) \leq R, \quad j = 1, 2, \dots, m, \quad (17)$$

$$d_i + \sum_{j=1}^m a_j t_{ij} \geq s_i, \quad i = 1, 2, \dots, n, \quad (18)$$

$$\frac{1}{M} x_{ij} \leq t_{ij} \leq x_{ij} M, \quad i = 1, 2, \dots, n, j = 1, 2, \dots, m, \quad (19)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (20)$$

$$R \leq T. \quad (21)$$

Podmínky (17) souvisí s předpokladem, že celkový součet časů nabíjení a manipulace na každé stanici nepřekročí hodnotu R . Nerovnosti (18) zaručují dobití vozidla na požadovanou úroveň. Omezující podmínky (19) definují, že pokud vozidlo i nebude dobíjeno na stanici j , dobíjecí čas t_{ij} bude roven nule. Podle rovnic (20), každé vozidlo bude dobíjeno právě na jedné stanici. Nerovnost (21) definují, že doba dobíjení na každé stanici bude maximálně R . Tato hodnota zároveň nesmí překročit délku pracovní doby, což zaručuje podmínka (22).

Výsledky

Výše uvedený modifikovaný matematický model je zde aplikován na ukázkový příklad, kdy k jednotlivým vozům je přiřazen stav nabití, který je dán náhodným číslem v intervalu $(0,2; 0,8)$. Tato čísla reprezentují zbývající kapacitu baterie v podílovém vyjádření. Dále je dán požadovaný stav nabití baterie, který má pro každý vůz hodnotou 0,8, opět v podílovém vyjádření z celkové kapacity baterie. Čas manipulace r_i , tedy čas potřebný k přesunu jednoho vozu k libovolné nabíjecí stanici a zpět, byl určen jako 8 minut. V reálném prostředí byl tento čas vyměřen jako 4 minuty na odjezd vozu ke stanici a následné zapojení, 4 minuty naopak k odpojení a odjezdu zpět na parkovací stání. V souhrnu je tento čas roven 160 minutám (při zapojení 20 vozů). Tento model předpokládá např. firemní garáže, v nichž se odpovědná osoba stará o dobíjení automobilů všech zaměstnanců.

Pro řešení této úlohy byl zvolen výpočetní software MPL, který je popsán v kapitole 2. Ten v tomto případě pracuje ve spojení se zdrojovým souborem MS Excel, který obsahuje veškeré parametry a proměnné uvedené v předchozí kapitole. Společně tvoří matice, které následně slouží jako vstupní data pro výpočet v programu MPL. Příloha 1 této práce zobrazuje zmíněný soubor MS Excel. Zeleně jsou podbarveny oblasti, ve kterých se nalézají proměnné. Zároveň tyto zeleně podbarvené oblasti jsou určeny pro

zobrazení výsledků získaných výpočtem pomocí MPL. Zápis matematického modelu v programu MPL je součástí přílohy 2.

Levá matice v příloze 1 představuje matici vstupních parametrů, ze kterých vychází výpočty pro MPL. Jedná se o data, která jsou nastavena uživatelem modelu a lze je v průběhu měnit a následně zkoumat, jakým způsobem tato změna ovlivní konečný výsledek. Zbývající matice tvoří oblast, kam se po provedení výpočtu zobrazí výsledky (zeleně podbarveno). Prostřední matice ukazuje, která vozidla se budou nabíjet na které nabíjecí stanici. Jak již bylo zmíněno, model za tímto účelem obsahuje binární proměnou. Ta v této matici vychází jako hodnota 0 nebo 1. Hodnota 1 vychází pro případy, kdy se bude vůz nabíjet na dané stanici, hodnota 0 v opačném případě, tedy vozidlo se na dané nabíjecí stanici nabíjet nebude. Zároveň platí vlastnost, že každé vozidlo se dobije právě na jedné nabíjecí stanici, tzn. hodnota 1 se v každém řádku nachází právě jednou. V tomto případě není proto možné proces nabíjení předčasně ukončit a pokračovat v něm na jiné nabíjecí stanici. Poslední matice poskytuje výsledky ohledně časů strávených nabíjením pro jednotlivé vozy.

Časy jsou logicky závislé na počátečním stavu nabití baterie d . Pracovní doba T byla nastavena na 450 minut. To znamená, že hodnota nejdelší doby nabíjení na stanici R dosahující v součtu 398 minut, nepřekračuje danou pracovní dobu T . Navržené řešení tedy vzhledem ke splnění všech podmínek lze označit jako optimální. Zároveň bylo díky tomuto výpočtu zjištěno, že systém nabíjení má stále rezervu. To znamená, že zde pracovní doba výrazně překračuje čas R , tedy existuje zde rezerva 52 minut. Na základě dalších výpočtů se nabízí možnost prodloužit dobu manipulace r_i u všech vozů z původních 8 minut až na 10 minut. Pokud by se ale doba manipulace zkrátila na 7 minut u každého vozu, je zde reálná možnost nabít všechny vozy z 20 na 80 % během stanovené pracovní doby.

Za tohoto předpokladu je stále možné plnit podmínku, kdy hodnota R nepřekročí hodnotu T . Z výsledků také vyplývá zásadní skutečnost, že při daném počtu vozů systém nevyžaduje přidání dalších nabíjecích stanic, což by s sebou neslo finanční náklady na jejich pořízení a následný provoz.

6 Aplikace heuristického algoritmu pro rozvrhování dávek

V této kapitole je v různých variantách řešena úloha hromadné obsluhy jako problému rozvrhování dávek, představeného v kapitole 4. Z této kapitoly vychází i heuristický algoritmus, pomocí kterého je úloha řešena. Algoritmus je zde aplikován přímo na konkrétní úlohu, která vychází ze stejného pojmového modelu jako modifikovaná úloha v předchozím případě. Zde je nejprve představeno řešení pomocí heuristického algoritmu bez jakékoliv úpravy. Následně je do úlohy aplikován princip priority, která je vybraným vozům přiřazována dvěma různými způsoby. Výsledky jednotlivých variant jsou následně interpretovány a je také provedeno jejich porovnání a vyhodnocení použití principu priority.

6.1 Řešení pomocí základního heuristického algoritmu

Uvedený algoritmus je aplikován na úlohu s uvedenými parametry z předchozí kapitoly. Metoda poskytuje přípustné řešení, ale zároveň nelze vyloučit ani to, že algoritmus může poskytnout i řešení optimální. Úloha s využitím paralelních procesorů je zde řešena v několika variantách. První varianta pracuje s uvedeným heuristickým algoritmem, kdy dávky mají své pořadí podle velikosti parametru. Další varianty pracují také s tímto algoritmem, ale zároveň se zaměřují na upřednostňování vybraných dávek, resp. vozů, na základě potřeb jejich uživatelů. Zmíněný algoritmus využívá stejný postup jako v kapitole 4, zde je ale navíc aplikován přímo na tento konkrétní model.

Fáze 1

Na začátku algoritmu jsou vozy seřazeny sestupně podle času (v minutách) potřebnému k nabití na 80 % kapacity baterie. Data vychází z předchozího případu, kde však bylo počítáno s podílovým vyjádřením stavu nabití. V tomto modelu je také počítáno se zmíněným předpokladem nabíjení rychlostí jedno procento za minutu. Pro lepší sestavení rozvrhu jsou zde uvažovány časy t_j v minutách, vzniklé z rovnice (22).

$$t_j = (0,8 - d_j) * 100 \quad (22)$$

Z tohoto vztahu poté vychází hodnoty časů potřebných k nabití na 80 % kapacity baterie vyjádřených v Tab. 7:

Tab. 7 Časy potřebné k nabití

Vozy	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
t_j (min)	54	51	50	59	51	26	40	58	52	31
Vozy	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20
t_j (min)	52	24	54	60	57	31	55	60	54	30

Podle hodnot t_j v minutách je následně vytvořeno pořadí. Toto pořadí prezentuje Tab. 8 **Chyba! Nenalezen zdroj odkazů..**

Tab. 8 Pořadí vozů

Vůz	V14	V18	V4	V8	V15	V17	V1	V13	V19	V9
t_j	60	60	59	58	57	55	54	54	54	52
Vůz	V11	V2	V5	V3	V7	V16	V10	V20	V6	V12
t_j	52	51	51	50	40	31	31	30	26	24

Fáze 2

Východiskem pro druhou fázi algoritmu je vytvořené pořadí. Jak bylo uvedeno, model obsahuje celkem 4 procesory, tedy v tomto konkrétním případě 4 nabíjecí

stanice. Následně je proto proveden výběr 4 prvních vozů, v tomto případě vozy V14, V18, V4 a V8, které se zařadí postupně na nabíjecí stanice (dále značených S) v čase 0, v pořadí od S1 až po S4, kdy budou všechny stanice obsazeny.

Fáze 3

Z fáze 2 vyplývá obsazenost všech stanic po umístění prvních 4 vozů. V momentě, kdy se některá ze stanic uvolní (tzn. po nabití prvního vozu ze čtyřech umístěných na stanicích), vybere se z pořadí další vůz a obsadí nově uvolněnou stanici. V rozvrhu je tento vůz umístěn hned za předchozí vůz dobíjený na stejné stanici. Další vůz z pořadí bude umístěn v momentě dalšího uvolnění některé ze stanic. Tento postup je následně aplikován, dokud není rozmístěno všech 20 vozů.

Poté, co jsou všechny vozy rozmístěny a nabity, je sestaven rozvrh, který reprezentuje tento algoritmus rozmisťování vozů.

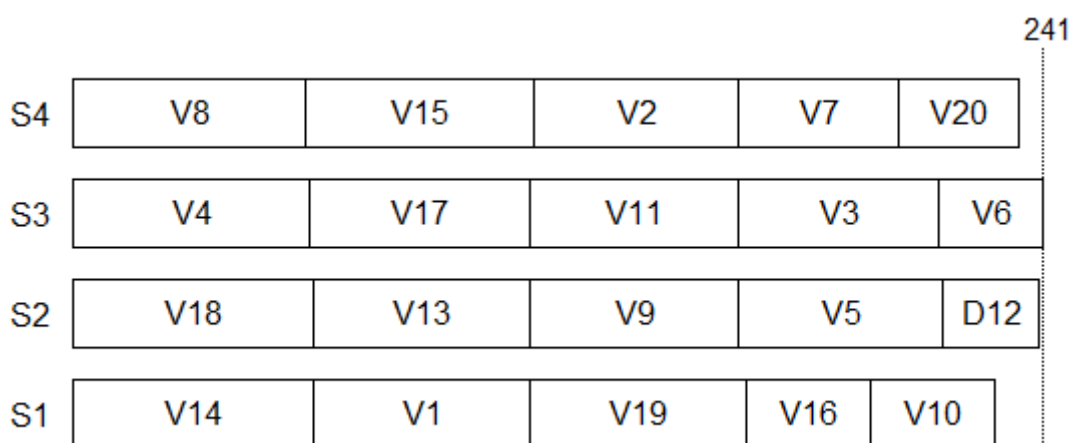
V prvním případě je model řešen pouze pomocí uvedeného heuristického algoritmu, kdy pořadí nabíjení vychází z Tab. 8. S využitím tohoto pořadí je následně sestaven rozvrh včetně časů, kdy bude každý vůz dobit. Tento rozvrh je znázorněn v tabulkové podobě v Tab. 9 společně s časy dokončení nabíjení u jednotlivých vozů. Proces nabíjení začíná vozem V14, který bude nabíjen 60 minut, a končí vozem V12, který se bude nabíjet už pouze 24 minut.

Tab. 9 Nabíjecí schéma s časy dokončení

Vůz	V14	V18	V4	V8	V15	V17	V1	V13	V19	V9
t_j	60	60	59	58	57	55	54	54	54	52
Čas dokončení	60	60	59	58	115	114	114	114	168	166
Vůz	V11	V2	V5	V3	V7	V16	V10	V20	V6	V12
t_j	52	51	51	50	40	31	31	30	26	24
Čas dokončení	166	166	217	216	206	199	230	236	242	241

Celý proces nabíjení končí po 241 minutách. Jak je dále zřejmé z nabíjecího rozvrhu na Obr. 9, nabíjecí stanice 1 a 4 ukončily nabíjení výrazně dříve než stanice 2 a 3, což znamená nerovnoměrné vytížení jednotlivých stanic. Výsledný čas 241 minut

představuje teoretickou hodnotu, kdy je respektován pouze čas nabíjení. Stejně jako v modifikovaném modelu v předchozí kapitole, je i v tomto případě uvažován čas r_i představující čas manipulace. Pod manipulací je opět chápán přejezd každého vozu z parkoviště k jedné z nabíjecích stanic a po skončení nabíjení zpět na parkoviště. Vyměřen byl na 8 minut, tedy 4 minuty na přejezd ke stanici a 4 minuty zpět. V souhrnu je tento čas roven 160 minutám. To znamená, že čas potřebný k nabití všech 20 vozů je 401 minut. V tomto případě lze čas manipulace rozložit rovnoměrně na všechny stanice, protože jsou všechny postupně obsazeny 5 vozy. V porovnání s pracovní dobou zkrácenou o přestávky je možné konstatovat, že celý proces nabíjení je při daném stavu nabití jednotlivých vozů možné realizovat v rámci pracovní doby 450 minut, včetně rezervy 49 minut.



Obr. 9 Nabíjecí schéma při použití heuristického algoritmu

6.2 Využití priority pro vybrané vozy

V druhé variantě této úlohy je v modelu využita priorita. Přiřazení této priority lze ilustrovat na situaci, kdy jsou některé vozy v určitém ohledu lepší nebo vhodnější než ostatní a jsou proto upřednostněny (například na delší cesty apod.). V praxi může být priorita přiřazena například na základě vyšší výbavy nebo motorizace. V této úloze, resp. v této variantě, je priorita přiřazena vybraným vozům V1 až V10 (v Tab. 10 označených žlutě). U těchto vozů jejich uživatelé požadují, aby byly vozy v určený čas připraveny, tzn. aby byly dobity na požadovaných 80 %. Prvních 10 vozů s prioritou je proto seřazeno podle požadavků uživatelů na odjezd.

To znamená, že čím dřívější odjezd je požadován, tím dříve bude daný vůz nabíjen. Zbylé vozy žádnou prioritu nemají, jsou tedy seřazeny také podle postupu z fáze 1, ale celkově se zařadí až za prvních 10 prioritních vozů.

Tab. 10 *Prioritní nabíjecí schéma podle přiřazené priority daným vozům*

Vůz	V2	V4	V1	V5	V8	V9	V3	V7	V10	V6
t_j	51	59	54	51	58	52	50	40	31	26
Čas dokončení	51	59	54	51	109	103	104	99	130	129
Vůz	V14	V18	V15	V17	V13	V19	V11	V16	V20	V12
t_j	60	60	57	55	54	54	52	31	30	24
Čas dokončení	164	169	186	185	218	223	237	217	247	242

Schématický rozvrh je znázorněn na Obr. 10. Stejně jako v předchozím případě, i zde je vidět poměrně výrazný rozdíl ve vytíženosti jednotlivých stanic, zejména v případě stanice S1. Ta, jak je zřejmé, končí v 223. minutě, tedy o 19 minut dříve než nejvytíženější stanice S4, která končí až v čase F_{max} , tedy v 242. minutě. K výslednému času je opět nutné připočítat manipulační čas r_i . Oproti předchozímu případu ale zde není možné ho rovnoměrně rozdělit na všechny stanice. Stanice S2 a S3 zde také obsluhují 5 vozů výsledný čas se zvýší o manipulační čas 40 minut z celkových 160 minut. Stanice S1 zde obsluhuje pouze 4 vozy a náleží jí proto manipulační čas 32 minut. Nejvíce, tedy 6 vozů bude nabíjeno na stanici S4. Náleží jí manipulační čas 48 minut. Výsledný čas R je 402 minut. Oproti předchozí variantě je tedy čas delší o 1 minutu.

S4	V5	V9	V6	V15	V16	V20
S3	V1	V3	V14	V13	V12	
S2	V4	V7	V10	V17	V11	
S1	V2	V8	V18	V19		

Obr. 10 Nabíjecí schéma s využitím priority

Cílem této úlohy tedy bylo připravit 10 prioritních vozů. Jak je ale zřejmé z Tab. 11, požadavky u prioritních vozů nebyly ve většině splněny. Hodnoty požadavků jsou zde ve formátu minut od počátku. Ke splnění požadovaného času odjezdu došlo pouze u první čtveřice vozů, které byly nabíjeny od času 0. Při zaokrouhlení vychází průměrné zpoždění červeně vyznačených vozů 20 minut. Ačkoliv celkový čas procesu nabíjení je splněn, požadavky prioritních vozů jsou za těchto předpokladů nemožné. Z toho důvodu je nutné u všech deseti vozů snížit požadovaný stav nabití z 80 %.

Tab. 11 Vyhodnocení přiřazení priority vybraným vozům

Vůz	V2	V4	V1	V5	V8	V9	V3	V7	V10	V6
t_j	51	59	54	51	58	52	50	40	31	26
Čas dokončení	51	59	54	51	109	103	104	99	130	129
Požadavek	52	61	63	67	83	83	87	88	104	108

6.3 Využití priority podle nejvyššího nabití

V této variantě je opět využita priorita v procesu nabíjení. Ta navazuje na předchozí situace, kdy byl dán požadavek na odjezd 10 vozů v určitých časech. V předchozím případě ale šlo o vozy, jejichž čas nabíjení ve většině přesahoval hranici 50 minut, což vedlo k nesplnění tohoto požadavku. V tomto případě neexistuje předpoklad, že

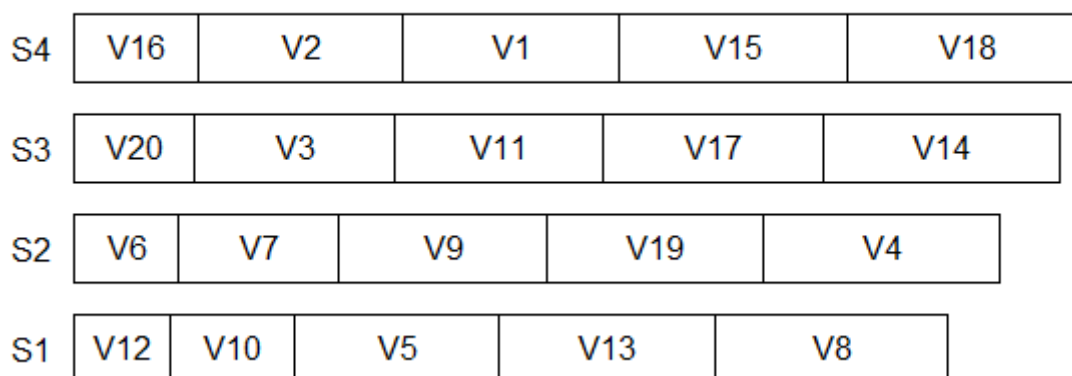
některé vozy jsou vhodnější. Priorita zde vyplývá ze stavu nabití. Čím více je tedy daný vůz nabitý, tím dříve bude dobíjen.

První místa v pořadí proto obsadí vozy, které budou z výchozího stavu nabití dobity na výsledných 80 % nejdříve. Požadavek na odjezd zůstává stejný, pouze se neváže k jiným vozům. Tab. 12 zobrazuje pořadí na základě uvedeného předpokladu. Prakticky jde o opačné pořadí, jaké vyplývalo z běžného využití heuristického algoritmu.

Tab. 12 Prioritní nabíjecí schéma podle nejnižších časů nabití

Vůz	V12	V6	V20	V16	V10	V7	V3	V2	V5	V9
t_j	24	26	30	31	31	40	50	51	51	52
Čas ukončení	24	26	30	31	55	66	80	82	106	118
Vůz	V11	V1	V13	V19	V17	V15	V8	V4	V14	V18
t_j	52	54	54	54	55	57	58	59	60	60
Čas ukončení	132	136	160	172	187	193	218	231	247	253

Následující Obr. 11 zobrazuje schéma nabíjení vozů uvedeným způsobem. Je zde zřejmé, že poměrně brzy bude nabito více vozů než v předchozích případech. Oproti ostatním variantám ale stanice S4 končí nabíjení až v čase 253 minut, tedy o 12 a 11 minut později oproti prvnímu a druhému případu. I zde se ale vyskytuje nerovnoměrné vytížení stanic, kdy stanice S1 končí v čase 218 minut. Zároveň je nutné, stejně jako v předchozích variantách, počítat s manipulačním časem r_i .



Obr. 11 Prioritní nabíjecí schéma

Tab. 13 zobrazuje porovnání časů dokončení první desítky prioritních vozů s požadovaným časem odjezdu. Zatímco v předchozích případech bylo z 10 prioritních vozů 6, které nebyly včas nabity, zde jsou pouze 2 takové vozy. Jedná se o vozy D5 a D9, které vzhledem ke stavu nabití na počátku byly dobíjeny až jako poslední z desítky prioritních. Nedochozí ale k výraznému zpoždění, kdy vůz D5 je zpožděn o 2 minuty a vůz D9 o 10 minut. Samozřejmě při tomto způsobu přiřazování priority existuje značná závislost na tom, jak budou vozy dobity na začátku. I zde proběhne nabíjení všech vozů v rámci pracovní doby, včetně započítání času manipulace r_i .

Tab. 13 Vyhodnocení přiřazení priority nejvíce dobítým vozům

Vůz	V12	V6	V20	V16	V10	V7	V3	V2	V5	V9
t_j	24	26	30	31	31	40	50	51	51	52
Čas dokončení	24	26	30	31	55	66	80	82	106	118
Požadavek	52	61	63	67	83	83	87	88	104	108

V praxi může dojít k situaci, kdy vlivem priority pro nejvíce nabité vozy nebude z hlediska času možné dobít i zbylé vozy, které budou dobíjeny delší dobu. Tuto hrozbu je nutné zvážit již při rozvrhování priority jednotlivým vozům. Toto rozhodnutí je dáno tím, zda uživatel systému trvá na dobití všech vozů, zda jsou nutné takto brzké odjezdy a samozřejmě má vliv také výchozí stav nabití jednotlivých vozů. Ve zde uvedeném případě je možné nabít vozy s prioritou i ostatní vozy.

7 Vyhodnocení výsledků

Při provedení uvedených řešení pomocí různých metod byly zjištěny také různé výsledky. Tyto výsledky se liší především v rozvrhu, podle kterého budou vozy nabíjeny. Za účelem srovnání byla u všech modelů vypočtena hodnota R . Ta, jak bylo uvedeno, představuje nejdelší dobu strávenou na stanici. V praxi tato hodnota udává čas v minutách od počátku, kdy poslední stanice končí proces nabíjení.

V modifikovaném modelu je hodnota R vypočtena přímo pomocí řešení v MPL. Ta je zde rovna 398 minutám. Díky využití MPL jsou zároveň všechny stanice rovnoměrně vytíženy, kdy každá stanice postupně nabije 5 vozů.

Průběh nabíjení, resp. pořadí vozů během nabíjení, zobrazuje následující schéma na Obr. 12.

Modifikovaný model s řešením v MPL						R	398
S1	54	51	24	55	54		
S4	50	26	58	52	52		
S2	59	60	57	31	30		
S3	51	40	31	54	60		

Obr. 12 Schématický rozvrh pro modifikovaný model

Obdobně jsou vyjádřeny i jednotlivé varianty úlohy s paralelními procesory v dále uvedených schématech, ze kterých vychází i hodnota R . Při využití heuristického algoritmu je opět docíleno rovnoměrného vytížení všech stanic, kdy každá nabije 5 vozů, jak zobrazuje Obr. 13. Oproti předchozímu modelu je ale hodnota R rovna 420 minutám. To stačí na nabití všech vozů během pracovní doby, zároveň je ale v porovnání s předchozím případem prostor pro zlepšení.

Úloha s paralel. procesory – heuristický algoritmus						R	420
S3	59	55	52	50	26		
S2	60	54	52	51	24		
S4	58	57	51	40	30		
S1	60	54	54	31	31		

Obr. 13 Schématický rozvrh pro úlohu s paralel. procesory – algoritmus

Další varianta na Obr. 14, která pracuje s prioritou přiřazenou desítce vybraných vozů, už postrádá rovnoměrné rozložení vozů ke stanicím. Zároveň ani hodnota R 411 minut není nejlepší možnou. Z toho vyplývá, že udílení priority vozům dle vlastního uvážení nelze v tomto případě označit za vhodnou koncepci vzhledem k minimalizaci R .

Úloha s paralel. procesory – prioritá výběru TOP10						R	411
S4	51	52	26	57	31	30	
S3	54	50	60	54	24		
S2	59	40	31	55	52		
S1	51	58	60	54			

Obr. 14 Schématický rozvrh pro úlohu s paralel. procesory – prioritá TOP10

V případě přiřazení priority desítce nejvíce nabitých vozů vychází hodnota R absolutně nejlépe, kdy je rovna 333 minutám. Zároveň všechny stanice obsluhují shodně 5 vozů. Tato varianta poskytuje výhodu v možnosti rychlého nabití pro 10 vozů, kdy nesplnění požadavku na čas odjezdu nebyl splněn pouze ve dvou případech, a to v řádu jednotek minut. Jak ale bylo zmíněno, je zde silná závislost na počátečním stavu dobití vybraných vozů. Pro zbylé vozy už tato varianta není příliš vhodná. Schéma je pro tuto variantu zobrazuje Obr. 15.

Úloha s paralelními procesory – prioritá dle nabití						R	333
S4	31	51	54	57	60		
S3	30	50	52	55	60		
S2	26	40	52	54	59		
S1	24	31	51	54	58		

Obr. 15 Schématický rozvrh pro úlohu s paralel. procesory – prioritá nabitých vozů

Ve vztahu k výsledkům všech řešení lze konstatovat splnění základního požadavku na dobití všech 20 vozů během pracovní doby T . Všechna uvedená řešení jsou tedy minimálně přípustná. Optimální řešení vzhledem k hodnotám v Tab. 14 nabízí řešení pro modifikovaný model, které využívá software MPL. Ve všech variantách se vyskytuje i nevyužitý čas, který může být využit různými způsoby dle potřeby uživatele systému. Jako příklad využití tohoto času lze uvést nabíjení dalších vozů, které mohou být v budoucnu pořízeny. Dále je možné v systému pracovat i s více

vybitými vozy, než jaké jsou uvedeny v uvedených případech, popřípadě lze zabezpečit provoz systému i při výpadku některé ze stanic. Následující Tab. 14 uvádí srovnání všech řešených variant podle hodnot R a také podle časů nejdelšího provozu dané stanice. Z tohoto srovnání vyplývá, že z hlediska hodnoty R vychází nejlépe varianta úlohy s paralelními procesory s využitím priority pro nejvíce nabité vozy. Z druhého hlediska vychází nejlépe varianta modifikovaného modelu řešeného pomocí MPL.

Tab. 14 Shrnutí výsledků

Metoda	hodnota R [min]	Nejdelší provoz stanice [min]
Modifikovaný model - MPL	398	238
Paralelní procesory – využití algoritmu	420	242
Paralelní procesory – priorita vybraných vozů	333	247
Paralelní procesory – priorita podle stavu nabití	411	253

Ve všech variantách se vyskytují vozy, jejichž stav nabití se pohybuje vždy pod 60 % kapacity baterie. Zároveň je ale potřeba uvažovat situaci, kdy se v systému bude nacházet vůz se stavem nabití například 65 % nebo více. V tom případě je nutné určit hranici stavu nabití, nad kterou daný vůz již nebude dobíjen vzhledem k dostatečnému dojezdu. Jako příklad může být uveden vůz nabitý na 75 %. Ten by byl nabíjen na konečnou hodnotu 80 % celkové kapacity v řádu jednotek minut a dojezd by se zvýšil pouze nepatrně. Oproti tomu je nutné i u tohoto vozu počítat s časem manipulace, překračujícího délkou dobu nabíjení. Tímto způsobem tak může dojít k uvolnění nabíjecích stanic právě o vozy s již dostatečným stavem nabití.

Závěr

Cílem této práce byla analýza systému hromadné obsluhy. Práce se soustřeďuje zejména na systémy hromadné obsluhy v oblasti elektromobility, ve kterých se vyskytují prioritní prvky. Za tímto účelem byly zpracovány různé varianty úloh postavené na modelu reálného systému hromadné obsluhy. Vypracování této práce proběhlo na základě podrobné literární rešerše týkající se jak obecných systémů hromadné obsluhy, tak i systémů hromadné obsluhy s prioritou. Rešerše zahrnuje také heuristické a softwarové nástroje vhodné pro operace s uvedenými modely.

Teoretická část se věnuje nejprve obecnému popisu systému hromadné obsluhy. Následující kapitoly popisují jednotlivé části, které systémy hromadné obsluhy obsahují. Dále jsou zmíněny stochastické procesy, a zvláště Markovovy procesy v souvislosti s tématem této práce. Další kapitoly jsou věnovány vybraným softwarovým nástrojům pro práci s modely hromadné obsluhy a také heuristickým metodám. V neposlední řadě jsou popsány systémy hromadné obsluhy s výskytem priority. Tématem teoretické části je dále úloha s paralelními procesory a také rešerše aktuálního stavu elektromobility, což slouží jako úvod pro část praktickou.

V praktické části je představen model úlohy s paralelními procesory v modifikaci pro jeho řešení pomocí softwaru MPL. Dále je tento model řešen v již nemodifikované variantě pomocí představené heuristické metody. Model je následně řešen i ve dvou dalších variantách, které využívají dva způsoby přidělení priority vybraným prvkům řešeného systému. Závěr práce je věnován srovnání všech uvedených variant řešeného modelu a vyhodnocení jejich dopadu.

Seznam literatury

BANGSOW, S. Manufacturing Simulation with Plant Simulation and SimTalk.

Zwickau: Springer International Publishing, 2010. ISBN 978-3-642-05073-2.

BANKS, J. Discrete-event system simulation. New Jersey: Prentice-Hall, 2001. ISBN 978-0-13-088702-3.

BELLMAN, R. A Markovian Decision Process. Indiana University Mathematics Journal, 6, 679-684, 1957. JSTOR 24900506

BHAT, U N. *An introduction to queueing theory: modeling and analysis in applications*. USA: Birkhäuser, 2015. 339 s. ISBN 978-0-8176-8421-1.

BILLINGSLEY, P. Probability and measure. 3rd ed. Wiley India Pvt. Limited, 2008. ISBN 9788126517718.

Algoritmy.net. Binární halda. Algoritmus [online]. Copyright © 2015 [cit. 17.09.2022]. Dostupné z: <https://www.algoritmy.net/article/15/Binarni-halda>

CARTWRIGHT, N. How the Laws of Physics Lie. Oxford, England: Oxford University Press, 1983. ISBN 9780198247043.

CHAN, W C. *An elementary introduction to queueing systems*. Singapore: World Scientific, 2014. 116 s. ISBN 978-981-4612-00-5.

CORMEN, T. H., LEISERSON, CH. E., RIVEST, R. L., STEIN, C. Introduction to algorithms. 2nd ed. Cambridge, Mass.: MIT Press, 2001. ISBN 0-262-03293-7

CORMEN, T. H. Introduction to algorithms. 3rd ed. Cambridge, Mass.: MIT Press, c2009. ISBN 978-0-262-03384-8.

DOLORES, M. M., M^a. SOLEDAD, J. F. SÁNCHEZ GARCÍA y J. J BERNAL GARCÍA. La hoja de cálculo como apoyo a la simulación de los fenómenos de espera con prioridades. Una aplicación a la sanidad. Rect@. Revista Electrónica De Comunicaciones y Trabajos De ASEPUMA. 2009, vol. 10, no. 1, s. 77-104. ISSN 1575-605X.

E15. Elektromobilita: Jaká je její budoucnost? Porostou prodeje vozů? | E15.cz. E15.cz - Byznys, politika, ekonomika, finance, události [online] [cit. 2022-11-17]. Dostupné z: <https://www.e15.cz/elektromobilita>

Emobilita: Skupina ČEZ [online], 2022 [cit. 18.11.2022]. Dostupné z: <https://www.elektromobilita.cz/>

FÁBRY, J. Operační výzkum pro prezenční a kombinovanou formu studia. ŠKODA AUTO Vysoká škola, 2019. ISBN 978-80-87042-84-7.

FAČEVIČOVÁ K., K. HRON a P. KUNDEROVÁ. Markovovy řetězce a jejich aplikace. 2., doplněné vydání, Univerzita Palackého v Olomouci, 2018. ISBN 978-80-244-5432-0.

FIALA, P. Modelování a analýza produkčních systémů. Praha: Professional Publishing, 2002. ISBN 80-86419-19-3.

GAGNIUC, P. A. Markov chains: from theory to implementation and experimentation. Hoboken, NJ: John Wiley, 2017. ISBN 978-1-119-38755-8.

HARCHOL-BALTER, M. Scheduling: Non-Preemptive, Size-Based Policies. Performance Modeling and Design of Computer Systems [online]. Cambridge: Cambridge University Press, 2012, s. 499-507 [cit. 2022-04-09]. ISBN 9781139226424. Dostupné z: doi:10.1017/CBO9781139226424.039

IOANNOU, A. a M. G. KATEVINIS. "Pipelined Heap (Priority Queue) Management for Advanced Scheduling in High-Speed Networks," in IEEE/ACM Transactions on Networking, vol. 15, no. 2, pp. 450-461, April 2007, doi: 10.1109/TNET.2007.892882.

JABLONSKÝ, J. Operační výzkum: kvantitativní modely pro ekonomické rozhodování. 3. vyd. Praha: Professional Publishing, 2011. ISBN 978-80-86946-44-3.

JALLAS, E., R. SEQUEIRA and P. MARTIN et al. Mechanistic Virtual Modeling: Coupling a Plant Simulation Model with a Three-dimensional Plant Architecture Component. Environ Model Assess **14**, 29–45 2009.

KAMP, P. H. You're Doing It Wrong.: Think you've mastered the art of server performance? Think again. [online], vol. 8, 2010. [cit. 2022-09-18]. ISSN 1542-7730. Dostupné z: <https://dl.acm.org/doi/10.1145/1810226.1814327>

KLEINBERG, J., TARDOS, É. Algorithm design. Thirteenth impression. Noida, India: Pearson, c2014. ISBN 9789332518643.

KLEINROCK, L. Queueing systems. New York: John Wiley, 1976. ISBN 0-471-49111-x.

KOŘENÁŘ, V. *Stochastické procesy*. Praha: Vysoká škola ekonomická, 2002. ISBN 80-245-0311-5.

KOŘENÁŘ, V. *Stochastické procesy*. 2., přeprac. vyd. Praha: Oeconomica, 2010. ISBN 978-80-245-1646-2.

Matematická biologie. Matematická biologie učebnice: Definice Poissonova procesu. Matematická biologie učebnice: Úvod [online], 2022. [cit. 2022-09-24] Dostupné z: [https://portal.matematickabiologie.cz/index.php?pg=analiza-a-](https://portal.matematickabiologie.cz/index.php?pg=analiza-a)

modelovani-dynamickych-biologickych-dat--vybrane-kapitoly-z-matematickeho-modelovani--poissonuv-proces--definice-poissonova-procesu

MEYN, S. and R. L. TWEEDIE. Markov chains and stochastic stability. 2nd ed. Cambridge: Cambridge University Press, 2009. ISBN 978-0-521-73182-9.

MIN, D. and Y. YIH. Managing a patient waiting list with time-dependent priority and adverse events. ResearchGate | Find and share research [online]. Copyright © 2014 Springer Science [cit. 16.04.2022]. Dostupné z: https://www.researchgate.net/publication/264944019_Managing_a_patient_waiting_list_with_time-dependent_priority_and_adverse_events.

MÜLLER, Jan. Markovské rozhodovací procesy a jejich řešení v prostředí MS Excel [online]. Praha, 2008 [cit. 2022-10-22]. Dostupné z: <https://theses.cz/id/wuw1pz/>. Bakalářská práce. Vysoká škola ekonomická v Praze. Vedoucí práce Václav Kořenář.

MPL Modeling System. Maximal Software - Optimization Modeling [online], 2016 [cit. 08.11.2022]. Dostupné z: <https://www.maximalsoftware.com/mpl/>

Aktuálně.cz. Nejlevnější auta do zásuvky. Dostupnějších elektromobilů přibývá, nejsou jen do města - Aktuálně.cz. Zprávy - Aktuálně.cz [online]. Copyright © [cit. 28.11.2022]. Dostupné z: <https://zpravy.aktualne.cz/ekonomika/auto/nejlevnejsi-elektromobily/r~dd980ff08d9a11ecbdb0ac1f6b220ee8/>

NOWAK, M. A. Evolutionary dynamics: exploring the equations of life. Cambridge, Mass.: Belknap Press of Harvard University Press, 2006. ISBN 0-674-02338-2.

PARZEN, E. Stochastic processes. Dover edition. Mineola, New York: Dover Publications, 2015. ISBN 978-0-486-79688-8.

PEARL, J. Heuristics: Intelligent search strategies for computer problem solving. United States: Addison-Wesley Pub. Co., Inc., Reading, MA (1984). ISBN 978-0201055948.

Queuing Theory: Definition, History & Real-Life Applications. Queue-it Virtual Waiting Room | Deliver Online Fairness [online]. 2022 [cit. 20.09.2022]. Dostupné z: <https://queue-it.com/blog/queuing-theory/>

SAID, D., CHERKAOUI, S. and L. KHOUKHI, 2015, Multi-priority queuing for electric vehicles charging at public supply stations with price variation. Wirel. Commun. Mob. Comput. 2015, vol. 15, s. 1049–1065. doi: 10.1002/wcm.2508

SCHIERREICH, Š. Praktická výkonnost různých implementací prioritní fronty [online]. Praha, 2018. [cit. 2022-09-18]. Dostupné z <https://dspace.cvut.cz/bitstream/handle/10467/76639/F8-BP-2018-Schierreich->

Simon-thesis.pdf?sequence=-1&isAllowed=y. Bakalářská práce. České vysoké učení technické, fakulta informačních technologií. Doc. Ing. Ivan Šimeček, Ph.D.

SEdgeWICK, R. Algoritmy v C. Praha: SoftPress, 2003. ISBN 80-86497-56-9.

SHONE, R., GLAZEBROOK, K. and K. G. ZOGRAFOS. Resource allocation in congested queueing systems with time-varying demand: An application to airport operations. European Journal of Operational Research. 2019, vol. 276, no. 2, s. 566-581. ISSN 0377-2217

SLANINOVÁ, M. *Stochastické modely v ekonomii* [online]. Hradec Králové, 2015 [cit. 2022-04-16]. Dostupné z: <https://theses.cz/id/dzrjw4/STAG74240.pdf>. Bakalářská práce. Univerzita Hradec Králové, Přírodovědecká fakulta. Mgr. Jitka Kühnová, Ph.D.

SKALSKÁ, H. Stochastické modelování. Vyd. 2., rozšířeno a upraveno. Hradec Králové: Gaudeamus, 2006. 162 s. ISBN 80-7041-488-x.

SOKOLOWSKI, J. A. a BANKS, C. M. Principles of modeling and simulation: a multidisciplinary approach. Hoboken: John Wiley, 2009. ISBN 978-0-470-28943-3.

SUNDARAPANDIAN, V. Probability and Queueing Theory. PHI, 2009. ISBN 978-8120338449.

ŠEDA, M. Modely hromadné obsluhy [online]. Článek, Vysoké učení technické v Brně, Fakulta strojního inženýrství. [cit. 2022-04-02]. Dostupné z: http://web2.vslg.cz/fotogalerie/acta_logistica/2011/2_cislo/3_seda.pdf

ŠKODA AUTO. ŠKODA AUTO Česká republika | Oficiální web ŠKODA AUTO a.s. [online]. Copyright © [cit. 05.11.2022]. Dostupné z: https://www.skoda-auto.cz/_doc/36507de5-aba6-46d0-a9f4-134e8f45ed56

Themenseite Mobility - acatech - National Academy of Science and Engineering. acatech – National Academy of Science and Engineering [online]. Copyright © acatech [cit. 17.11.2022]. Dostupné z: <https://en.acatech.de/topic/mobility/>

VUILLEMIN, J. A data structure for manipulation priority queues. Comm. ACM 21, April 1978, 309-315.

TAVARÉ, S. The linear birth-death process: An inferential retrospective. Advances in Applied Probability, 50(A), 253-269 [online]. 2018 [cit. 25.09.2022]. Dostupné z: <https://www.cambridge.org/core/journals/advances-in-applied-probability/article/linear-birthdeath-process-an-inferential-retrospective/52799690050C62D855EE9A10E30128DF#metrics>

Siemens Software. 301 Moved Permanently [online]. Copyright © Siemens 2022 [cit. 08.04.2022]. Dostupné z: <https://www.plm.automation.siemens.com/global/en/products/tecnomatix/>.

What are the Euro Emissions Standards?. Stratstone | New and Used | Aftercare | Servicing [online]. Copyright © 2022 Stratstone. All rights reserved. [cit. 05.11.2022]. Dostupné z: <https://www.stratstone.com/blog/european-car-emissions-standards/>

Emobilita. Emobilita: Skupina ČEZ [online], 2022 [cit. 18.11.2022]. Dostupné z: <https://www.elektromobilita.cz/>

Seznam obrázků a tabulek

Seznam obrázků

Obr. 1 Schéma modelu hromadné obsluhy	10
Obr. 2 Systém hromadné obsluhy s jednou obslužnou linkou	12
Obr. 3 Sériové uspořádání obslužných linek	13
Obr. 4 Paralelní uspořádání obslužných linek	13
Obr. 5 Schéma procesu birth-death	16
Obr. 6 Model Head-of-the-Line.....	30
Obr. 7 Příklad binárního stromu	34
Obr. 8 Layout systému	45
Obr. 9 Nabíjecí schéma při použití heuristického algoritmu	54
Obr. 10 Nabíjecí schéma s využitím priority	56
Obr. 11 Prioritní nabíjecí schéma.....	58
Obr. 12 Schématický rozvrh pro modifikovaný model	59
Obr. 13 Schématický rozvrh pro úlohu s paralel. procesory – algoritmus.....	59
Obr. 14 Schém. rozvrh pro úlohu s paralel. procesory – priorita TOP10	60
Obr. 15 Schém. rozvrh pro úlohu s paralel. procesory – priorita nabitých vozů ...	60

Seznam tabulek

Tab. 1 Kendallova klasifikace	17
Tab. 2 Vybrané typy hald a jejich asymptotické složitosti	36
Tab. 3 Srovnání cen vybraných elektromobilů.....	41
Tab. 4 Technické specifikace ENYAQ iV 80 vztahující se k modelu	44
Tab. 5 Parametry modelu	46
Tab. 6 Proměnné modelu	47
Tab. 7 Časy potřebné k nabití	52
Tab. 8 Pořadí vozů	52
Tab. 9 Nabíjecí schéma s časy dokončení.....	53
Tab. 10 Prioritní nabíjecí schéma podle přiřazené priority daným vozům	55
Tab. 11 Vyhodnocení přiřazení priority vybraným vozům.....	56
Tab. 12 Prioritní nabíjecí schéma podle nejnižších časů nabití	57
Tab. 13 Vyhodnocení přiřazení priority nejvíce dobitým vozům	58
Tab. 14 Shrnutí výsledků.....	61

Seznam příloh

Příloha 1 Výchozí data modifikovaného modelu pro MPL	71
Příloha 2 Zápis matematického modelu ze softwaru MPL.....	72

Příloha 1 Výchozí data modifikovaného modelu pro MPL

T	450 min	pracovní doba	
R	398 min		
	d	s	r
	stav baterie	požadovaný stav	doba manipulace
V1	0,26	0,8	8
V2	0,29	0,8	8
V3	0,3	0,8	8
V4	0,21	0,8	8
V5	0,29	0,8	8
V6	0,54	0,8	8
V7	0,4	0,8	8
V8	0,22	0,8	8
V9	0,28	0,8	8
V10	0,49	0,8	8
V11	0,28	0,8	8
V12	0,56	0,8	8
V13	0,26	0,8	8
V14	0,2	0,8	8
V15	0,23	0,8	8
V16	0,49	0,8	8
V17	0,25	0,8	8
V18	0,2	0,8	8
V19	0,26	0,8	8
V20	0,5	0,8	8
Průměr	0,3255		160

Dobijení na j-té stanici	a _j			
	a ₁	a ₂	a ₃	a ₄
Rychlost dobíjení / hod	S1	S2	S3	S4
	0,01	0,01	0,01	0,01
V1	1	0	0	0
V2	1	0	0	0
V3	0	0	0	1
V4	0	1	0	0
V5	0	0	1	0
V6	0	0	0	1
V7	0	0	1	0
V8	0	0	0	1
V9	0	0	0	1
V10	0	0	1	0
V11	0	0	0	1
V12	1	0	0	0
V13	0	0	1	0
V14	0	1	0	0
V15	0	1	0	0
V16	0	1	0	0
V17	1	0	0	0
V18	0	0	1	0
V19	1	0	0	0
V20	0	1	0	0

Ano	1
Ne	0

Proměnné

	Jak dlouho se bude dobíjet			
	S1	S2	S3	S4
V1	54	0	0	0
V2	51	0	0	0
V3	0	0	0	50
V4	0	59	0	0
V5	0	0	51	0
V6	0	0	0	26
V7	0	0	40	0
V8	0	0	0	58
V9	0	0	0	52
V10	0	0	31	0
V11	0	0	0	52
V12	24	0	0	0
V13	0	0	54	0
V14	0	60	0	0
V15	0	57	0	0
V16	0	31	0	0
V17	55	0	0	0
V18	0	0	60	0
V19	54	0	0	0
V20	0	30	0	0
Suma	238	237	236	238

Příloha 2 Zázpis matematického modelu ze softwaru MPL

```
TITLE Nabijeni;
? DP;

OPTIONS
ExcelWorkBook="Model.xlsx";
ExcelSheetName="MPL";

INDEX
i:=EXCELRange("vozidlo");
j:=EXCELRange("Stanice");

DATA
T:= EXCELRange("PracovniDoba");
d[i]:= EXCELRange("Stav");
s[i]:= EXCELRange("Pozadavek");
r[i]:= EXCELRange("Manipulace");
a[j]:= EXCELRange("Rychlost");

BINARY VARIABLES
x[i,j] EXPORT TO EXCELRange("Prirazeni");

VARIABLES
t[i,j] EXPORT TO EXCELRange("Cas");
R EXPORT TO EXCELRange("CasMax");

MODEL
MIN R;

SUBJECT TO
minmax[j]:          sum(i:t[i,j]+r[i])<=R;
nabijeni[i]:        d[i]+sum(j:a[j]*t[i,j])>=s[i];
prirazeni1[i,j]:    1/10000*x[i,j]<=t[i,j];
prirazeni2[i,j]:    t[i,j]<=10000*x[i,j];
jednastanice[i]:    sum(j:x[i,j])=1;
pracovnidoba:       R<=T;

?pracovnidoba[j]:   sum(i:t[i,j]+r[i])<=T;

END
```

ANOTAČNÍ ZÁZNAM

AUTOR	Bc. Michal Vild		
STUDIJNÍ PROGRAM/OBOR/SPECIALIZACE	specializace Řízení mezinárodních dodavatelských řetězců		
NÁZEV PRÁCE	Využití modelů hromadné obsluhy s prioritou		
VEDOUCÍ PRÁCE	doc. Ing. Jan Fábry, Ph.D.		
KATEDRA	KRVLK - Katedra řízení výroby, logistiky a kvality	ROK ODEVZDÁNÍ	2023
POČET STRAN	74		
POČET OBRÁZKŮ	15		
POČET TABULEK	14		
POČET PŘÍLOH	2		
STRUČNÝ POPIS	<p>Cílem této diplomové práce je analýza a aplikace modelů s prioritními frontami se zaměřením na systém dobíjení elektromobilů.</p> <p>V teoretické části je vysvětleno téma hromadné obsluhy, včetně popisu všech částí. Tato část dále obsahuje popis vybraných řešení úloh hromadné obsluhy a zejména úloh s paralelními procesory.</p> <p>V praktické části je představen model hromadné obsluhy, kde je řešen systém dobíjení elektromobilů. Model je analyzován s použitím heuristické metody a v modifikované formě modelu je využit software MPL for Windows. Dále se model věnuje přiřazením priorit vybraným prvkům</p>		
KLÍČOVÁ SLOVA	Systém hromadné obsluhy, modely front, režimy front, prioritní fronty, heuristický algoritmus, MPL for Windows, elektromobilita,		

ANNOTATION

AUTHOR	Bc. Michal Vild		
FIELD	Specialization International Supply Chain Management		
THESIS TITLE	The Use of Mass Service Models with Priority		
SUPERVISOR	doc. Ing. Jan Fábry, Ph.D.		
DEPARTMENT	KRVLK - Department of Production, Logistics and Quality Management	YEAR	2023
NUMBER OF PAGES			
	74		
NUMBER OF PICTURES			
	15		
NUMBER OF TABLES			
	14		
NUMBER OF APPENDICES			
	2		
SUMMARY	<p>The aim of this thesis is the analysis and application of priority queuing models with a focus on the EV charging system.</p> <p>In the theoretical part, the topic of bulk service is explained, including a description of all parts. This part also contains a description of selected solutions to the bulk servicing problems and in particular to the parallel processor problems.</p> <p>In the practical part, a model of the bulk service is presented, where the EV charging system is solved. The model is analysed using a heuristic method and the MPL for Windows software is used in a modified form of the model. Furthermore, the model deals with the assignment of priorities to selected elements</p>		
KEY WORDS	Systems of Mass service, Queue Models, Queue Modes, Priority Queues, Heuristic Algorithm, MPL for Windows, Electromobility		