

ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta riadenia a informatiky

AUTOREFERÁT
DIZERTAČNEJ PRÁCE

Žilinská univerzita v Žiline
Fakulta riadenia a informatiky

Ing. Andrea Galadíková

Autoreferát dizertačnej práce

Podpora riadenia prevádzky železničných dopravných uzlov v reálnom čase s využitím simulačných modelov

.....
Názov dizertačnej práce

na získanie akademického titulu „**philosophiae doctor**“ (v skratke **PhD.**)
v študijnom programe doktorandského štúdia

aplikovaná informatika

v študijnom odbore:

informatika

Žilina, máj 2024

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia na katedre matematických metód a operačnej analýzy na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline

Predkladateľ: Ing. Andrea Galadíková
Katedra matematických metód a operačnej analýzy
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline

Školiteľ: doc. Ing. Norbert Adamko, PhD.
Simcon, s.r.o.
Žilina

Oponent: prof. Ing. Jozef Gašparík, PhD.
Katedra železničnej dopravy
Fakulta prevádzky a ekonomiky dopravy a spojov
Žilinská univerzita v Žiline

Oponent: doc. Ing. Michael Bažant, PhD.
Katedra softwarových technológií
Fakulta elektrotechniky a informatiky
Univerzita Pardubice

Autoreferát bol rozoslaný dňa: 23. 7. 2024

Obhajoba dizertačnej práce sa koná dňa 22.8.2024 o 9:00 h. pred komisiou pre obhajobu dizertačnej práce schválenou pracovnou skupinou odborovej komisie v študijnom odbore **informatika v študijnom programe aplikovaná informatika**, vymenovanou dekanom Fakulty riadenia a informatiky Žilinskej univerzity v Žiline.

prof. Ing. Ľudmila Jánošíková, PhD.
predsedníčka pracovnej skupiny odborovej komisie
v študijnom odbore **informatika**
v študijnom programe **aplikovaná informatika**

Fakulta riadenia a informatiky
Žilinská univerzita
Univerzitná 8215/1
010 26 Žilina

OBSAH

Úvod.....	4
1 Návrh systému na podporu rozhodovania.....	4
2 Spôsoby získavania aktuálnych informácií o stave železničného dopravného uzla	6
2.1 Statické informácie	7
2.2 Dynamické informácie	7
3 Rozšírenie architektúry ABASim	9
4 Implementácia navrhovaného systému bez sofistikovaných metód	11
5 Integrácia optimalizačných metód do systému na podporu rozhodovania	13
5.1 Integrácia exaktného výpočtu pomocou matematického modelu	14
5.2 Integrácia heuristického prístupu	14
5.3 Integrácia učenia posilňovaním	14
6 Implementácia navrhovaného systému s využitím sofistikovaných metód.....	15
6.1 Priradenie rušnovodičov.....	15
6.1.1 Optimalizačný algoritmus založený na simulácii.....	16
6.1.2 Použitie algoritmov učenia posilňovaním	17
6.2 Určenie poradia úloh.....	19
6.2.1 Optimalizačný algoritmus založený na simulácii – určenie poradia úloh	19
6.2.2 Použitie algoritmov učenia posilňovaním	20
Záver.....	22
Zoznam použitej literatúry	23
Zoznam vlastných publikácií	24

ÚVOD

Železničná doprava má dlhodobu stabilnú miesto v oblasti prepravy osôb a tovarov. Pre jej efektívne fungovanie sa ako jedna z mnohých dôležitých zložiek ukazuje kvalita operatívneho riadenia v železničných dopravných uzloch, preto je nevyhnutné zaoberať sa možnosťami poskytnutia podpory riadenia ich prevádzky. Hoci v súčasnosti existuje mnoho informačných systémov, žiadny z nich nie je dostatočne zameraný na problémy vznikajúce v železničnom dopravnom uzle, akým môže byť napríklad komplikovaná zriaďovacia stanica a tieto systémy sa nezaobierajú takýmito problémami komplexne. Riešením tohto problému by mohlo byť použitie počítačovej simulácie a simulačných modelov železničných dopravných uzlov a to najmä vzhľadom na mnohé dobré vlastnosti počítačovej simulácie. Táto dizertačná práca sa preto zaoberá možnosťami zahrnutia simulačných modelov železničných uzlov do systému na podporu rozhodovania.

Táto práca sa zaoberá návrhom metodiky, v rámci ktorej sú popísané jednotlivé kroky, ktoré je potrebné zabezpečiť, aby bolo možné takýto systém na podporu rozhodovania vytvoriť a zároveň sú v tejto práci navrhnuté spôsoby, ako vytvorenie takéhoto systému zabezpečiť, čím poskytuje komplexný rámec vytvorenia systému na podporu rozhodovania dispečerov v železničnom dopravnom uzle.

1 NÁVRH SYSTÉMU NA PODPORU ROZHODOVANIA

Požiadavky na navrhovaný systém vyplývajú z potrieb samotných dispečerov. Je potrebné, aby takýto systém dokázal nielen prezentovať stav systému v reálnom čase (podobne ako štandardný informačný systém), ale aby zároveň ponúkol aj pohľad do budúcnosti, či minulosti, v používateľsky vhodnej forme. Pod používateľsky vhodnou formou možno rozumieť prívetivé a intuitívne grafické používateľské rozhranie, zahŕňajúce

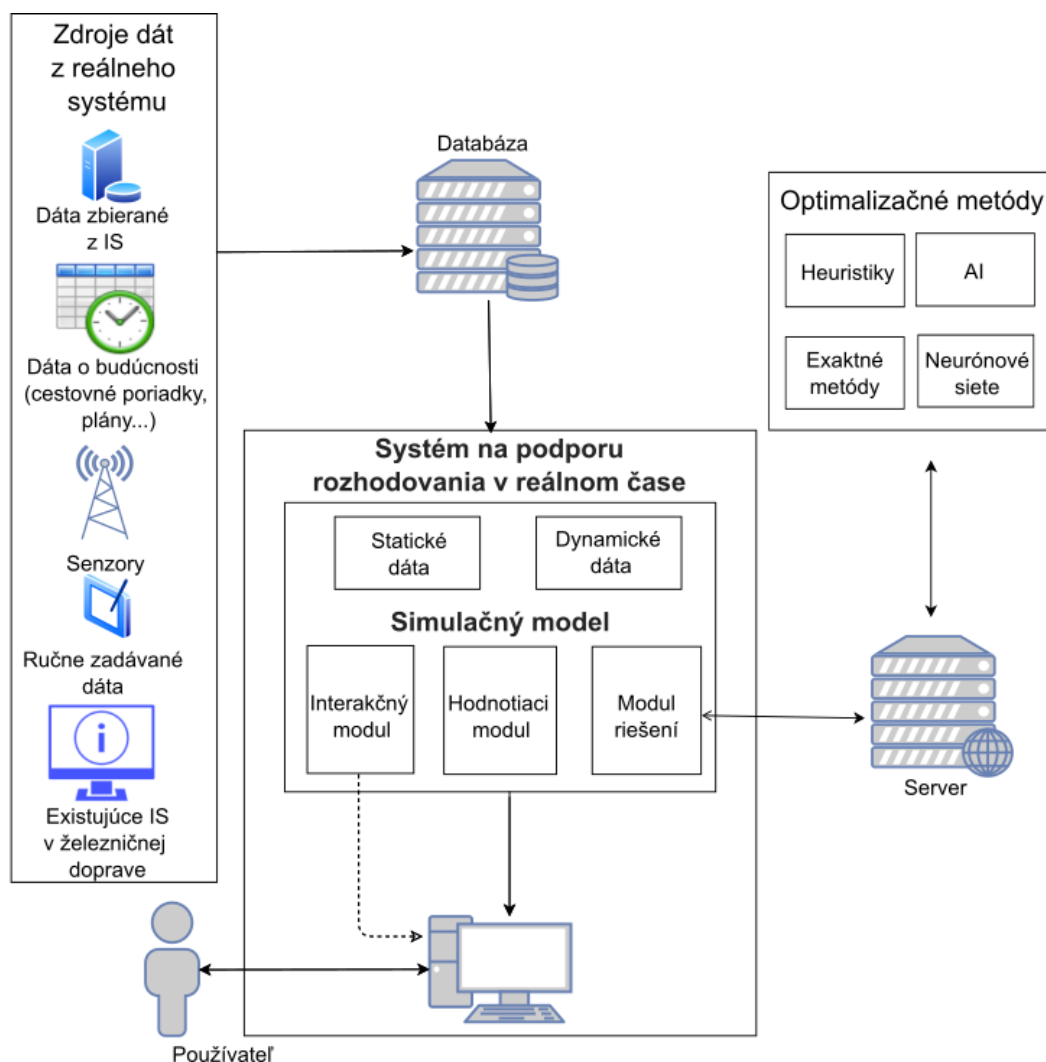
animáciu stavu v železničnom uzle, ako aj výstupy v podobe rôznych grafov a tabuliek. Pohľad do budúcnosti by mal umožniť používateľovi realizáciu viacerých konfigurácií budúceho vývoja v podobe rôznych experimentov. Počas týchto experimentov môže vstupovať do vývoja stavu modelovaného systému a priamo vykonávať niektoré rozhodnutia. Výsledky týchto experimentov (zahŕňajúce rôzne štatistické ukazovatele) by malo byť možné prezentovať v podobe dobre čitateľnej pre používateľa. Možnosť poznať stav systému v budúcnosti je vhodným základom pre aplikáciu sofistikovaných algoritmov poskytujúcich dispečerovi návrh riešení v prípade vzniku nepravidielností v prevádzke železničného dopravného uzla. Na základe vyššie popísaného boli stanovené nasledujúce funkčné požiadavky na navrhovaný systém:

- zadávanie vstupov vyvolané užívateľom
- zadávanie vstupov vyvolané systémom
- spúšťanie simulačných behov
- výber sofistikovaných algoritmov
- zasahovanie o simulačných behov
- získavanie výstupov.

Vzhľadom na definované požiadavky bol navrhnutý systém, v ktorého jadre stojí simulačný model. (obr. 1) Architektúru navrhovaného systému možno rozdeliť do troch základných častí: zdroja dát z reálneho prostredia, jadra systému na podporu rozhodovania v reálnom čase a optimalizačných metód. Zdrojmi dát z reálneho systému železnice sú napríklad dispozičné systémy. Medzi týmito zdrojmi a jadrom systému je vytvorené spojenie pomocou databázového servera, kde sú ukladané dáta z existujúcich systémov[3][8][9] a pre potreby sledovania aktuálnej situácie pomocou simulačných modelov, sú tieto informácie posielané

do jadra navrhovaného systému. Systém zahŕňajúci všetky výhody simulačných modelov, ktorý čerpá aktuálne informácie z reálneho systému za pomoci existujúcich dispozičných systémov [17][18], poskytuje možnosť sledovať budúci vývoj pomocou rôznych vstupov od užívateľa sám o sebe poskytuje dispečerom efektívne možnosti podpory rozhodovania. Napriek tomu sa ukazuje ako vhodné zahrnutie ďalšej, nemenej dôležitej funkcionality. Treťou zložkou navrhovanej architektúry sú preto rôzne optimalizačné metódy, keďže tieto sú potrebné pre kvalitnejšie návrhy rozhodnutí pre dispečera. Takéto algoritmy potrebujú dodávať informácie do jadra navrhovaného systému na základe požiadaviek, vznikajúci v priebehu realizácie simulačných experimentov, preto je pripojenie k týmto metódam zabezpečované pomocou servera.

Jadro systému je okrem kvalitných statických a dynamických informácií tvorené tromi modulmi. V rámci interakčného modulu je zabezpečená interakcia systému s užívateľom. Toto zahŕňa nastavenie počiatočnej konfigurácie experimentov, ako aj ich spúšťanie. Pre tým ako užívateľ nastaví konfiguráciu experimentov je nutné zabezpečiť aktuálnosť simulačného modelu vzhľadom na realitu. Keďže užívateľ môže vykonať viacero experimentov vychádzajúcich z rovnakého počiatočného stavu, je potrebné zabezpečiť uloženie aktuálneho stavu vzhľadom na realitu. Hodnotiaci modul zabezpečuje prezentáciu relevantných informácií pre rozhodovanie dispečera. Nejedná sa len o jeden izolovaný návrh rozhodnutia, ale simulačný model by mal ponúknuť súbor relevantných štatistických ukazovateľov a vyhodnotení, ktoré predstavujú ukazovatele výkonnosti železničného dopravného uzla. Tieto ukazovatele môžu byť zobrazené v grafickej podobe či alfanumerickej podobe vo forme grafov, plánov či tabuliek. Výsledky simulačného modelu môžu užívateľovi ukázať ako sa jednotlivými preverenými variantami zlepšili sledované ukazovatele (napr. súhrnné meškanie vlakov). Pre podporu rozhodovania je potrebné, aby systém na podporu rozhodovania poskytoval údaje o reálnom systéme pred vznikom konfliktnej situácie, aby sa tejto dalo v budúcnosti predchádzať. Keďže sa jedná o podporu operatívneho riadenia, je navyše potrebné, aby výsledky boli dostupné v takmer reálnom čase. Modul riešení nadväzuje spojenie s optimalizačnými metódami pomocou servera.



Obr. 1: Systém na podporu rozhodovania založený na simulačnom modeli

Je zrejmé, že súčasné fungovanie železničnej dopravy nie je schopné poskytnúť všetky informácie, ktoré by simulačný systém potreboval. V prípade, že v možných zdrojoch informácií v reálnom systéme železničnej dopravy (informačné systémy, hardvérové zabezpečenie, senzory a iné) budú existovať všetky potrebné informácie, je ďalej potrebné sa zaoberať možnosťami získavania takýchto informácií v požadovanej forme. Z tohto vyplýva, že je potrebné zaoberať sa možnosťami implementácie komunikačných kanálov smerom k simulačnému modelu. Zabezpečenie kvalitného simulačného modelu, ktorý spĺňa všetky naň kladené požiadavky je nemenej dôležité. Existujúce simulačné modely nedisponujú všetkými funkcionalitami potrebnými pre aktualizáciu stavu vzhľadom na realitu. Je potrebné, aby simulačný model použitý v navrhovanom systéme na podporu rozhodovania umožňoval realizáciu mnohých experimentov v aktuálnom stave. To predpokladá možnosť uchovávať aktuálny stav simulačného modelu. V dátových štruktúrach musí byť možné meniť dáta na základe správ prichádzajúcich z vonkajšieho prostredia. Okrem zmeny dynamických dát je potrebné pamätať aj na možnosť zmeny dát v rámci plánovaných činností.

2 SPÔSOBY ZÍSKAVANIA AKTUÁLNYCH INFORMÁCIÍ O STAVE ŽELEZNIČNÉHO DOPRAVNÉHO UZLA

Informácie, ktoré je potrebné získať z reálneho dopravného uzla je možné rozdeliť na statické a dynamické.

2.1 STATICKÉ INFORMÁCIE

Statické informácie sú napríklad informácie o cestovných poriadkoch, pracovných rozvrhoch a infraštruktúre a tieto informácie je možné pripraviť vopred. Významným zdrojom informácií o plánovaných cestovných poriadkoch je najmä Prevádzkový informačný systém (PIS) [21]. PIS poskytuje možnosti získavania dát pomocou webových služieb (zvyčajne pomocou SOAP protokolu), pričom komunikačné rozhranie je definované presne pre potreby konkrétneho systému, na základe charakteru dát, ktoré tento systém potrebuje. V prípade grafikonu je teda možné vyžiadať si konkrétne informácie pomocou existujúcich webových služieb. Dobrým zdrojom informácií o pracovných rozvrhoch je informačný systém VICOS[19], hoci tento systém nie je používaný na Slovensku, kde sú pracovné zmeny plánované v interných systémoch jednotlivých uzlov, napr. v systéme SAP. Systém SAP má definované komunikačné rozhranie využívajúce webové služby používajúce formát JavaScript Object Notation (JSON) (ale aj iné formáty). Informácie o infraštruktúre sa získavajú transformáciou mapových podkladov v papierovej alebo elektronickej podobe napríklad programom AutoCAD. Veľkú časť informácií o infraštruktúre je možné získať z podrobnej online mapy OpenRailwayMap, ktorá obsahuje informácie o celosvetovej železničnej infraštruktúre. OpenRailwayMap poskytuje verejne dostupné rozhranie, postavené na Representational state transfer (REST) komunikácii a toto rozhranie poskytuje dáta vo formáte JSON. Logickú úroveň a jednotlivé detaily infraštruktúry je však potrebné definovať v spolupráci s prevádzkovateľom, ktorý pozná systém najlepšie. Napriek tomu, že statické informácie sa nezvyknú meniť, v prípade cestovných poriadkov, ale aj pracovných zmien môže v určitých prípadoch prísť k zmene. Túto zmenu je takisto možné získať z vyššie spomenutých systémov, avšak pri návrhu komunikácie je potrebné pamätať aj na takúto situáciu.

2.2 DYNAMICKÉ INFORMÁCIE

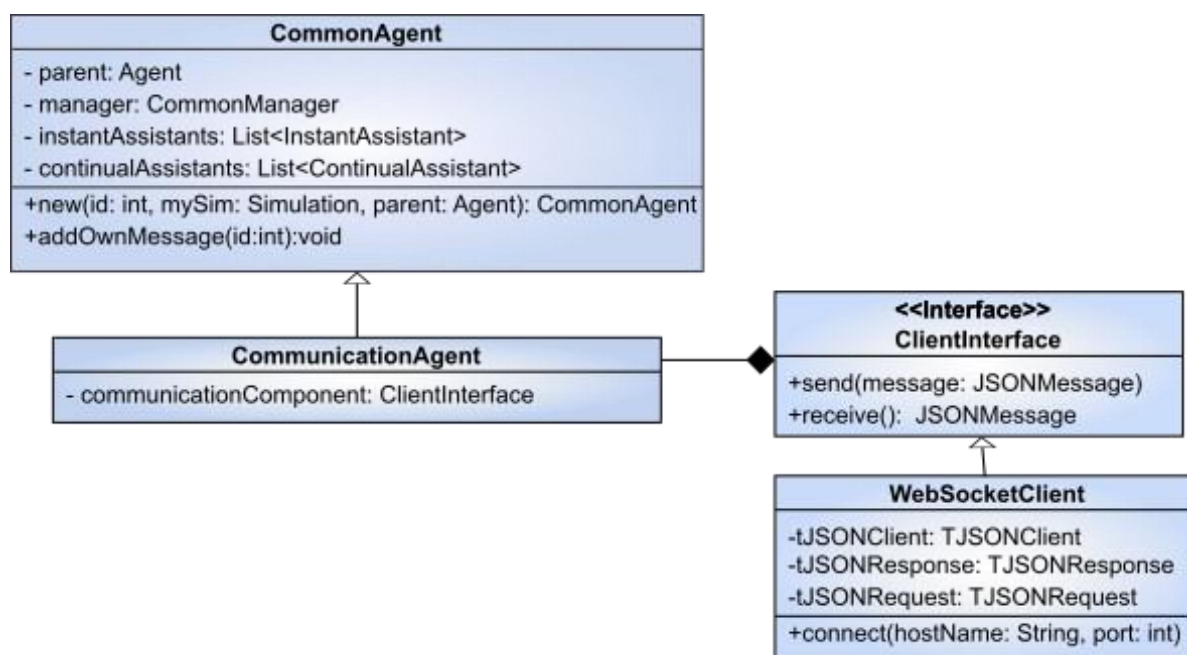
Dynamické informácie je potrebné získavať v reálnom čase a niektoré z nich sa nachádzajú v systémoch PIS a GTN[20]. Z týchto systémov je možné získať najmä informácie o polohe vlaku na trati, v prípade PIS aj o konkrétnej polohe v rámci uzla. Do týchto systémov prichádzajú informácie aj priamo z trate, z rôznych senzorov, GPS lokátorov, RFID lokátorov, prípadne rádiovkej komunikácie a je teda možné informácie zbierať aj priamo z týchto zdrojov. Takéto zariadenia zvyčajne odosielať informácie na server (v prípade systému PIS ide o Tomcat server), kde sa tieto dáta o polohe vlakov agregujú a tento server je schopný ich následne rozposielať klientom v požadovanom formáte. Pokiaľ hovoríme o dátach o vlakoch, systém PIS ich poskytuje štandardne v XML formáte a obsahujú údaje ako typ vozňa, číslo vozňa, hmotnosť a dĺžka vozňa, režim brzdenia, špeciálne vlastnosti (nebezpečný tovar, nesmie ísť cez zväžny pahorok a podobne). Kontrola integrity dát poskytovaných PIS je definovaná na strane systémov požadujúcich informácie, čo znamená, že takáto kontrola musí byť zahrnutá aj v navrhovanom systéme na podporu rozhodovania. Viac problematickým sa ukazuje zber údajov o polohe a stave personálu, nakoľko tieto nie sú takmer vôbec zaznamenávané do informačných systémov, ktoré sú používané v železničných uzloch. Hoci sú v rámci železničných dopravných uzlov používané systémy pre riadenie personálu, tieto systémy sa používajú na evidencie jednotlivých pracovných zmien, prítomnosti zamestnanca v práci a podobne (napríklad systém EDOS). Poloha zamestnanca, prípadne konkrétna práca vykonávaná v aktuálnom čase sa v týchto systémoch nevyskytuje. Poloha personálu je však dôležitým ukazovateľom v rozhodovacom procese dispečera. Ak dispečer vo svojom rozhodovacom procese potrebuje informácie o polohe personálu, zvyčajne ju zisťuje pomocou vysielacky. Toto však môže byť veľmi neefektívne. Pre efektívnu podporu rozhodovania s využitím simulačných modelov sa preto odporúča zaviesť automatické zisťovanie polohy personálu, prípadne môžu byť tieto

informácie zadávané ručne dispečerom (čo však nie je veľmi efektívne). Informácie o polohe personálu je však možné zbierať aj pomocou rôznych technológií ako sú GPS lokátory, bluetooth zariadenia, vysielacky či s využitím moderných zariadení, akými sú smart hodinky. GPS lokátory sú prenosné zariadenia, ktoré umožňujú monitorovanie osôb alebo vozidiel v železničnej doprave. Tieto zariadenie prijímajú GPS signál a na základe prijatých informácií vypočítajú súradnice určujúce ich polohu. Je preto potrebné, aby bolo používané v oblastiach pokrytých GPS signálom. Výhodou GPS lokátorov, ktoré môžu byť zabudované do smart hodínok alebo do vysielacky je, že pre svoje správne fungovanie nepotrebujú žiadne ďalšie hardvérové zariadenia a pomocou softvérovej aplikácie zabudovanej do smart hodínok sú informácie zasielané na server, odkiaľ ich môže získavať akýkoľvek ďalší systém. Nevýhodou je ich nižšia presnosť a to najmä v interiéri, keďže pre správne fungovanie GPS lokátora je potrebné zabezpečiť kontakt so satelitom. Táto nepresnosť sa v interiéri môže pohybovať až okolo 100 metrov. Znížená presnosť je možné zaznamenať aj v exteriéri a to najmä za nepriaznivého počasia. Okrem toho sa na železničnej trati často nachádzajú oblasti (najmä tunely), kde použitie GPS lokátora vôbec nie je možné. V neposlednom rade je potrebné spomenúť možnosť kybernetických útokov, ktoré môžu rušiť alebo manipulovať signály GPS, čo vedie k nepresným alebo falošným údajom o polohe. Niektoré smart hodinky využívajú namiesto GPS lokátora mobilnú sieť, avšak určovanie polohy pomocou mobilnej siete je rovnako (prípadne menej) presné ako GPS. Na rozdiel od GPS, bluetooth zariadenia môžu byť najmä v interiéri lepším riešením. Funkčnosť takéhoto riešenia však požaduje zabudovanie bluetooth majákov do priestoru železničného dopravného uzla. Tieto majáky totiž vysielajú signál, ktorý bluetooth zariadenie prijíma a na základe neho určuje svoju polohu. Pre správne fungovanie zisťovania polohy pomocou bluetooth zariadenia musí byť sieť týchto majákov dostatočne hustá (každých 15-50 metrov). Radio Frequency Identification (RFID) sa v súčasnosti používa na identifikáciu a sledovanie v mnohých odvetviach. Zisťovanie polohy pomocou RFID tagov je vhodným riešením v oblasti vlkovej dopravy a využíva sa pri určovaní polohy vozňov. Pre určovanie polohy jednotlivých vozňov sú RFID tagy pri-pevnené k trati v určitých intervaloch. Keď vlak prechádza cez značky, palubná čítačka zhromažďuje informácie o polohe vozňov a prenáša ich do centrálného riadiaceho systému. Tento systém potom použije informácie na určenie polohy a rýchlosti vlaku a môže podľa toho upraviť návestidlá a výhybky. Technológia RFID je vhodná aj pre určenie polohy personálu, aj keď predpokladá existenciu RFID brán. V prípade, že zamestnanec prejde takouto bránou, systém to zaznamená a teda je možné získať údaj, že zamestnanec sa v danom čase nachádzal pri tejto bráne. Výhodou RFID technológie je jej vysoká presnosť aj za nepriaznivých podmienok okolia. Jej nevýhodou je však nevyhnutnosť inštalácie jednotlivých RFID tagov a RFID brán. Ďalšie možnosti pri určovaní polohy jednotlivých zamestnancov môžu vychádzať z biometrických systémov na rozpoznávanie hlasu. Takéto systémy sú zabudované do vysieláčiek, ktoré personál na trati používa. Keďže moderné vysielacky zväčša disponujú GPS lokátorom, je možné spojiť konkrétnu osobu (pomocou rozpoznania hlasu) s polohou vysielacky. V prípade biometrických systémov však treba upozorniť, že ich presnosť je značne závislá na rôznych fyzikálnych javoch (v prípade rozpoznávania hlasu to môže byť prítomnosť šumu). Hoci sú všetky tieto technológie známe a ich použitie sa javí ako vhodné, v súčasnosti väčšina železničných dopravných uzlov takými zariadeniami nedisponuje. Podobné problémy, aké nastávajú pri zisťovaní polohy personálu súvisia aj s informáciami o činnosti prebiehajúcej na vlaku. V niektorých prípadoch sú takéto informácie hlásené do PIS, avšak nie všetky a najmä nie dostatočne detailne. Získavanie týchto informácií je možné najmä prostredníctvom komunikácie s personálom na trati, ako sú strojvodcovia alebo pracovníci údržby. Takáto forma komunikácie však nie je vždy efektívna, najmä ak ide o rýchle rozhodovanie v reálnom čase. Riešením by mohlo byť monitorovanie železničného dopravného uzla pomocou

senzorov, ktoré by zachytili pohyb vlaku v rámci uzla. Iným prístupom by mohlo byť získavanie údajov z kamerových záznamov, toto však predpokladá zabezpečenie takéhoto záznamu a samozrejme implementáciu algoritmov, ktoré by dokázali vhodne spracovať takéto obrazové dáta. Hoci tento postup nie je ideálny a získavanie informácií automatizovaným spôsobom by bolo efektívnejšie, v rámci navrhovaného systému na podporu rozhodovania navrhujeme možnosť upravovať dynamické dáta používateľom-dispečerom, v prípade, že on takýmito informáciami disponuje.

3 ROZŠÍRENIE ARCHITEKTÚRY ABASIM

Keďže v jadre navrhovaného systému stojí simulačný model, boli analyzované viaceré softvéry slúžiace na tvorbu simulačných modelov v železničnej doprave. Takýmito softvérmi sú OpenTrack, RailSys, Rail Traffic Controller, Villon a iné. Na základe vlastností jednotlivých softvérov sa ako vhodný ukázal simulačný softvér Villon a to najmä vzhľadom k tomu, že sa špecializovaný na tvorbu modelov železničných dopravných uzlov. Keďže tento simulačný softvér nedisponoval všetkými potrebnými funkcionalitami, bolo potrebné ho vhodne upraviť. Tento softvér je postavený na agentovo-orientovanej architektúre ABASim [1]. V tejto architektúre boli navrhnuté rozšírenia a to najmä možnosť získavať informácie z externého prostredia. V rámci architektúry ABASim bolo potrebné rozšíriť funkcionalitu jednotlivých agentov ich rozšírením o využitie služieb určitého typu webového klienta. Pre tieto potreby bol definovaný nový špeciálny typ komunikačného agenta - CommunicationAgent. UML diagram potrebných tried je zobrazený na obr. 2.



Obr. 2: UML diagram tried, potrebných pre implementáciu komunikačného agenta

Ak komunikačný agent prijme správu zo siete, pomocou komunikačného rozhrania ju prevedie na požadovaný formát (správu definovanú v architektúre ABASim) a túto správu odošle ostatným agentom.

Aby bolo možné zabezpečiť aktuálnosť simulačného modelu vzhľadom na skutočnosť a najmä umožniť užívateľovi realizovať viaceré experimenty z jedného východiskového stavu, je potrebné zabezpečiť uloženie a znovu načítanie simulačného modelu. Simulačné modely postavené na architektúre ABASim (ale aj ostatné simulačné modely) sú veľmi komplexné. Ich dátové štruktúry sú rôzne previazané a pri ukladaní aktuálneho stavu

nemôžeme opomenúť ani štruktúry týkajúce sa simulačného jadra. V prípade agentovo orientovanej simulácie je to najmä uloženie stavu poslaných správ v rámci jednotlivých agentov.

Aby bol zabezpečený aktuálny stav simulačného modelu vzhľadom na realitu, je potrebné sa zaoberať možnosťami, ako takúto synchronizáciu realizovať. Jednou z možností je neustále aktualizovanie simulačného modelu. To znamená, že každá informácia, ktorá sa dostane na databázový server bude okamžite poslaná simulačnému modelu. Tento prístup sa môže javiť ako vhodný, vzhľadom na neustálu aktuálnosť, avšak je náročný najmä vzhľadom na zabezpečenie komunikačného mechanizmu. Hoci sa môže zdať, že týmto spôsobom je možné odbúrať použitie databázy, toto nemožno tvorcom systému odporúčať, najmä s ohľadom na možnú stratu dát. Zasahovať do štruktúr bežiaceho simulačného modelu vždy, keď v reálnom systéme nastane zmena tiež nemusí byť vhodné, vzhľadom na rozsiahle väzby v dátových štruktúrach modelu. Druhou možnosťou je aktualizácia po určitom čase (napr. 5 minút), prípadne v čase, kde sa používateľ rozhodne systém používať. V tomto prípade sú teda všetky relevantné dáta uložené v databáze, odkiaľ sú posielané do simulačného modelu. Synchronizácia prebieha spôsobom, že simulácia sa pustí z posledného uloženého bodu, následne sa prebiehajúci simulačný beh koriguje pomocou dát posielaných z databázy. Napríklad sa menia časy aktivít, nastavujú sa cieľové koľaje podľa skutočnosti, pridávajú sa skutočné zdroje, atď.

V simulačných modeloch je potrebné zabezpečiť spracovanie správy prichádzajúce z externého prostredia. Toto môže byť problematické, keďže jednotlivé simulačné modely a softvér slúžiaci na ich tvorbu nevznikli s ohľadom na možnosti komunikácie s externými zdrojmi. Navrhovaný systém na podporu rozhodovania musí umožňovať aktualizáciu stavu simulácie na základe správ zasielaných z externého prostredia cez definované sieťové rozhranie. Týmto krokom distribúcia informácie nekončí, keďže zmenu stavu simulácie zabezpečujú jednotliví agenti, ktorým je potrebné definovať možnosti zmeny stavu simulácie na základe podnetov z externého prostredia. Túto skutočnosť je potrebné mať na zreteli aj v prípade, že by jadro systému na podporu rozhodovania netvorili simulačné modely postavené na architektúre ABASim, ale akékoľvek iné simulačné modely. V rámci architektúry ABASim je potrebné zabezpečiť posielanie jednotlivých správ od komunikačného agenta k jednotlivým agentom, ktorí zabezpečujú správu zdrojov, ktorých sa tieto správy týkajú. Správy prichádzajúce z externého prostredia môžu informovať o zmene stavu vlaku, personálu alebo technológie. Jednotliví agenti, ktorí tieto zdroje spravujú musia teda očakávať správu (s príslušnými parametrami), ktorú im pošle komunikačný agent a túto správu musia vedieť vhodne spracovať, napríklad zmenou polohy konkrétneho vlaku. Takýto zásah z vonkajšieho prostredia do fungovania simulačného modelu nie je jednoduché realizovať, nakoľko jedna informácia o zmene môže mať (a vo väčšine prípadov skutočne má) vplyv na mnohé iné atribúty. Simulačný model, ktorý stojí v jadre systému má totiž určitú počiatočnú konfiguráciu, na základe, ktorej sa predpokladá, že železničný uzol funguje. Počas behu simulačného modelu v reálnom čase však samozrejme dochádza k neočakávaným zmenám v reálnom systéme. Problematické je najmä dodržiavanie jednotlivých krokov technológie vlaku, nakoľko tieto nie sú dostatočne podrobne monitorované v rámci skutočného železničného uzla. Situácie nastávajúce v rámci zabezpečenia udržania aktuálnosti simulačného modelu možno rozdeliť do dvoch skupín. V prvom prípade sa jedná o nesúlad medzi realitou a simulačným modelom, ktorý však nemení charakter definovanej technológie vlaku. V rámci takýchto situácií je potrebné pri spracovaní danej aktivity v simulačnom modeli upraviť vstupujúce dáta, akými sú plánovaná koľaj, personál alebo čas. Plánovaná koľaj a personál sú parametrami správy, ktorá je posielaná príslušnému agentovi. Je preto potrebné zabezpečiť, aby pred týmto poslaním parametre správy zmenil komunikačný agent (jeho vnútorné komponenty), nakoľko tento disponuje informáciami z vonkajšieho prostredia.

V prípade, že sa líši čas trvania aktivity, a teda nastane situácia, že z externého prostredia príde informácia, že aktivita (napríklad presun vlaku z miesta A do miesta B) začala v čase 8:00, avšak podľa plánu mala začať o 7:50. Znamená to, že aktivita v rámci simulačného behu je v inom bode, ako v realite (prípadne nemusí vôbec prebiehať). Je teda potrebné zabezpečiť zmenu dĺžky trvania jednotlivých aktivít. Druhá skupina situácií je o čosi komplikovanejšia. Dochádza k nej vtedy, keď v realite dôjde k významnej zmene v technológii vlaku. V simulačnom modeli predpokladáme, že celý dej sa už odohral, v skutočnosti sa môžeme na časovej osi nachádzať, v akomkoľvek momente. To znamená, že je potrebné zaoberať sa synchronizačným mechanizmom, ktorý zabezpečí zmenu technológie vlaku. Toto predpokladá zásah do pôvodne statických dát počas simulačného behu.

V pôvodnom simulačnom behu bolo definované, ktorý zdroj má byť priradený. Ak príde z reálneho systému informácia, že bol priradený iný zdroj, je potrebné zachytiť správu, ktorá sa posiela agentovi, zabezpečujúcemu priradenie zdroja a zmeniť parameter tejto správy na základe informácie z reálneho systému. Priradenie správneho zdroja už prebehne štandardným spôsobom. Vo fungovaní simulačného modelu sa nič významné nemení, nakoľko agenti vopred nevedia, aký zdroj majú priradiť. Problém by mohol nastať jedine vtedy, ak by zdroj, ktorý bol v realite pridelený, bol už v simulačnom modeli pridelený inému vlaku v danom čase. V prípade, že budú zabezpečené kvalitné zdroje dát z reality by však takáto situácia nemala nastať. Každý vlak, ktorý sa nachádza v simulačnom modeli má v danom čase definovanú svoju polohu. Táto poloha úzko súvisí s aktuálne vykonávaným technologickým krokom, ako aj všetkými technologickými krokmi, ktoré už boli vykonané. Pri synchronizácii stavu v realite a v simulačnom modeli však môže nastať situácia, keď sa vlak nachádza v realite na inom mieste, ako by sa mal podľa preddefinovanej situácie, ktorú zobrazuje simulačným model. Týmto je potrebné sa dôkladne zaoberať a môžu nastať dve situácie. V prvej situácii vlak v realite postupoval podľa určených technologických krokov, avšak cieľová koľaj v technologickom kroku zabezpečujúcim presun bola zmenená. V tomto prípade, stačí vlaku v simulačnom modeli zmeniť polohu. Komplikovanejšia situácia nastáva, ak sa vlak na danú polohu dostal tak, že sa zmenili jeho technologické kroky. Vtedy je potrebné zabezpečiť synchronizáciu tak, že sa jednotlivé technologické kroky odoberú alebo pridajú.

Ak je v rámci synchronizácie simulačného modelu s realitou potrebné odobrať nejaký technologický krok, o ktorom vieme, že sa v realite neuskutoční, najvhodnejšie je nastaviť jeho trvanie na nulový čas.

V prípade, že v realite nastane situácia, že na vlaku je vykonaný technologický krok, ktorý nebol naplánovaný, je potrebné zabezpečiť synchronizačné mechanizmy, ktoré umožnia pridanie takéhoto kroku.

Dĺžka trvania aktivity je zabezpečovaná pomocou posielania správy hold. Túto posiela kontinuálny asistent prislúchajúceho agenta. Aby bolo možné zmeniť dĺžku trvania na základe informácií z reality, je potrebné upraviť metódu hold. Keď sa používateľ rozhodne so systémom realizovať experimenty, je potrebné, aby v určitom čase bola simulácia v stave, v ktorom je možné pokračovať v simulačnom behu v zrýchlennom čase, s použitím plánovaných činností (v realite tieto činnosti ešte neboli vykonané). Vtedy je potrebné, aby všetky bežiacie technologické kroky boli v stave, v ktorom ich koniec vieme odhadnúť pomocou plánu.

4 IMPLEMENTÁCIA NAVRHOVANÉHO SYSTÉMU BEZ SOFISTIKOVANÝCH METÓD

Navrhovaný systém na podporu rozhodovania má poskytovať dispečerovi pomocou simulačného modelu možnosť sledovať aktuálnu situáciu v železničnom dopravnom uzle a využívať výhody simulačných modelov, akými je najmä experimentovanie s rôznymi

variantami budúceho vývoja. Pri implementácii navrhovaného systému sme postupovali podľa navrhutej metodiky popísanej v dizertačnej práci. V prvom kroku boli teda definované požiadavky na daný systém. V nami implementovanom systéme bude realizovaná podpora rozhodovania v rámci údržbového depa. Dispečerovi bude umožnené sledovať aktuálnu situáciu tak, ako to umožňuje vybraný simulačný model. Ako výstupné ukazovatele boli vybrané časy odchodu jednotlivých vlakov zo systému a časy začiatkov a koncov trvania jednotlivých obslužných aktivít. Aby bolo možné implementovať navrhovaný systém, bolo potrebné definovať dáta, ktoré je potrebné čerpať z reálneho systému. Pripojenie k skutočným systémom nebolo možné, najmä z byrokratických dôvodov, ale aj z dôvodu absencie niektorých zdrojov informácií, ktorých použitie navrhujeme v tejto práci. Vzhľadom k tomu bol ako náhrada reality použitý simulačný model. Predpokladáme, že podobné dáta by bolo možné získať aj z reálnych systémov, avšak získavanie týchto dát by bolo oveľa náročnejšie, nakoľko nemusia byť v dostatočnej kvalite, na rozdiel od dát získaných zo simulačného modelu. Aby bolo možné overiť navrhovanú metodiku, bolo určené, že z reality je potrebné získať dáta o čase príchodu vlakov do systému, okrem toho časy začiatkov a koncov jednotlivých technologických krokov, ako aj zdrojov (personálnych), ktoré tieto kroky realizujú. Následne bolo potrebné definovať komunikačné rozhranie. Na základe navrhutej architektúry bola vytvorená serverová aplikácia, ktorej súčasťou je databáza. Táto serverová aplikácia je schopná zbierať údaje z reálneho systému a ukladať ich do danej databázy. Databáza je tvorená tromi tabuľkami a to tabuľkou s informáciami o vlakoch, tabuľkou o jednotlivých zdrojoch (najmä personálnych) a tabuľkou o aktivitách. Informácie z reality sú serverovej aplikácii posielané vo formáte JSON pomocou web socket technológie. Táto následne umožňuje rozposielať dáta ďalším pripojeným stranám, v našom prípade systému na podporu rozhodovania.

Navrhovaných systém na podporu rozhodovania má vo svojom jadre simulačný model vytvorený v simulačnom nástroji Villon. Pri skutočnej implementácii by musel byť vytvorený simulačný model predmetného železničného uzla so všetkými údajmi, ktoré je možné získať z plánov prevádzky daného železničného uzla. Pre demonštráciu riešenia bol použitý simulačný model, ktorý bol použitý aj na nahradenie reality, však upravený tak, aby bolo možné overiť funkčnosť navrhovaného riešenia (boli zmenené plánované technológie, časy príchodov vlakov, personálu a podobne). V rámci implementácie navrhovaného systému bolo upravené simulačné jadro ABASim a bol upravený simulačný softvér Villon tak, aby bolo možné zabezpečiť komunikáciu s vonkajším prostredím. Pre overenie funkčnosti celého systému boli realizované tri testovacie scenáre. Všetky testovacie scenáre vychádzajú zo základnej množiny údajov, ktorá obsahuje 15 vlakov, pričom na každom je potrebné vykonať určité obslužné aktivity. Týmito aktivitami môže byť veľká údržba, malá údržba, čistenie interiéru, čistenie exteriéru, čistenie grafitov a tankovanie.

Testovací scenár 1 zahŕňa meškanie jednotlivých vlakov a čas tohto meškania je modelovaný pomocou spojitého rovnomerného rozdelenia pravdepodobnosti na intervale (0,60) minút. Časy trvania jednotlivých obslužných aktivít sú pevne dané. Bolo realizovaných 30 simulačných behov pre generovanie dát, ktoré nahrádzajú realitu. Pre jednotlivé ukazovatele boli zostrojené 90% intervaly spoľahlivosti a výsledky porovnávajúce stav v systéme, ktorý nahrádza realitu a v navrhovanom systéme na podporu rozhodovania zobrazuje tabuľka 1.

Tabuľka 1: Porovnanie ukazovateľov - testovací scenár 1

<i>Ukazovateľ</i>	<i>Sim. model nahrádzajúci realitu (hod.)</i>	<i>Systém na podporu rozhodovania(hod.)</i>
<i>Celkové meškanie vypravených vlakov</i>	<42.82;48.69>	<42.73;48.55>
<i>Trvanie aktivity Veľká údržba</i>	<38.98;40.52>	<37.99;40.48>

Trvanie aktivity Malá údržba	<16.17;16.17>	<16.17;16.17>
Trvanie aktivity Tankovanie	<5.34;5.70>	<5.25;5.65>
Trvanie aktivity Čistenie grafitov	<14.81; 15.15>	<14.92;15.12>
Trvanie aktivity Čistenie interiéru	<17.5;17.5>	<17.5;17.5>
Trvanie aktivity Čistenie exteriéru	<16.49;16.98>	<16.42;16.85>

Testovací scenár 2 zahŕňa meškanie jednotlivých vlakov a čas tohto meškania je modelovaný pomocou spojitého rovnomerného rozdelenia pravdepodobnosti na intervale (0,60) minút a časy trvania niektorých aktivít sú taktiež náhodne generované.

Tabuľka 2: Porovnanie ukazovateľov - testovací scenár 2

Ukazovateľ	Sim. model nahrádzajúci realitu (hod.)	Systém na podporu rozhodovania(hod.)
Celkové meškanie vypravených vlakov	<63,54;70,54>	<63.35;70.78>
Trvanie aktivity Veľká údržba	<15,36;15,38>	<15.25;15.48>
Trvanie aktivity Malá údržba	<46,23;46,24>	<46.12;46.85>
Trvanie aktivity Tankovanie	<14,52;14,58>	<14.54;14.60>
Trvanie aktivity Čistenie grafitov	<24,96;25,22>	<24.42;24.90>
Trvanie aktivity Čistenie interiéru	<19,41;19,44>	<19.35;19.42>
Trvanie aktivity Čistenie exteriéru	<5,86;5,90>	<5.44;5.95>

Testovací scenár 3 má rovnaké charakteristiky ako scenár 1 a okrem toho zahŕňa zrušenie príchodu 2 vlakov.

Tabuľka 3: Porovnanie ukazovateľov - testovací scenár 3

Ukazovateľ	Sim. model nahrádzajúci realitu (hod.)	Systém na podporu rozhodovania(hod.)
Celkové meškanie vypravených vlakov	<34.78;37.76>	<34.62;36.99>
Trvanie aktivity Veľká údržba	<34.79;36.16>	<34.68;36.14>
Trvanie aktivity Malá údržba	<13.60;14.06>	<13.55;14.12>
Trvanie aktivity Tankovanie	<5.34;5.70>	<5.25;5.65>
Trvanie aktivity Čistenie grafitov	<4.72;5.02>	<4.72;5.02>
Trvanie aktivity Čistenie interiéru	<14.36;15.23>	<14.31;15.21>
Trvanie aktivity Čistenie exteriéru	<13.73;14.51>	<13.73;14.45>

V rámci overovania funkčnosti navrhovaného postupu bol teda vytvorený systém, v ktorého jadre stojí simulačný model. Tento simulačný model bol vytvorený v simulačnom softvéri Villon. Keďže simulačný model tvorí jadro systému na podporu rozhodovania, je schopný prijímať dáta z externého prostredia. Toto je realizované pomocou komunikácie s databázovým serverom, kde sú ukladané dáta popisujúce stav reálneho systému. Načítané údaje sú zobrazované v simulačnom modeli v užívateľsky prívetivej forme. Bola realizovaná séria experimentov, ktoré potvrdili, že simulačný model zobrazuje aktuálnu skutočnú situáciu.

5 INTEGRÁCIA OPTIMALIZAČNÝCH METÓD DO SYSTÉMU NA PODPORU ROZHODOVANIA

Na základe realizovaných výskumov v oblasti použitia optimalizačných metód pri riešení úloh v železničnej doprave sme definovali tri základné typy optimalizačných metód, ktoré by mohli byť vhodné pre integráciu do navrhovaného systému. Týmito metódami sú exaktný výpočet na základe matematického modelu, heuristický prístup v podobe rôznych primárnych heuristik (výmenná heuristika, vkladacia heuristika a iné) alebo metaheuristik (genetický algoritmus, simulované žíhanie, tabu search a iné) a strojové učenie (v prípade úloh riešením v železničných dopravných uzloch najmä algoritmy učenia posilňovaním). Pre tieto tri základné typy boli definované všeobecné postupy, akým by

simulačný model mal definovať požiadavky, tieto požiadavky odoslať a následne prijať odpoveď a túto odpoveď spracovať.

5.1 INTEGRÁCIA EXAKTNÉHO VÝPOČTU POMOCOU MATEMATICKÉHO MODELU

Exaktný prístup predpokladá existenciu matematického modelu, ktorý na základe vstupných informácií vytvorí riešenie. Vstup do matematického modelu je tvorený súborom viacerých typov premenných, zvyčajne sa jedná o binárne alebo celočíselné premenné. Tvorba navrhovaného systému musí poznať súbor týchto premenných. Keďže tieto premenné môžu obsahovať informácie o mnohých častiach systému (napríklad sú potrebné údaje o príchode jednotlivých vlakov a zároveň údaje o pracovných zmenách rušňovodičov), je potrebné tieto údaje zozbierať na jednom mieste. Keďže sa pri tomto type optimalizačného algoritmu (najmä pre jeho časovú náročnosť) predpokladá, že na základe jednej množiny vstupných údajov bude realizovaných viacero rozhodnutí (napríklad pri pridelení rušňovodičov možno na základe jedného algoritmu pridelit' všetkých rušňovodičov) je vhodné realizovať simulačný beh dva krát. Pri prvej realizácii sú postupne odosielané potrebné údaje o aktuálnom simulačnom behu smerom k optimalizačnému algoritmu. Tento optimalizačný algoritmus vytvorí na základe prijímaných informácií potrebnú dátovú štruktúru. Následne realizuje výpočet a čaká na požiadavku o informáciu o pridelení konkrétneho zdroja od simulačného modelu. Keď takúto požiadavku dostane, odošle správu s danou informáciou. Simulačný model musí vedieť takúto požiadavku spracovať. V rámci simulačného modelu je teda potrebné:

- poznať dáta, ktoré vyžaduje optimalizačný algoritmus
- realizovať inicializačný simulačný beh
- počas neho postupne odosielať požadované dáta
- realizovať ďalší simulačný beh
- odosielať požiadavku na pridelenie konkrétneho zdroja
- očakávať odpoveď obsahujúcu informácie o konkrétnom zdroji
- spracovať túto odpoveď a pridelit' konkrétny zdroj

5.2 INTEGRÁCIA HEURISTICKÉHO PRÍSTUPU

Heuristický prístup sa v mnohom podobá na exaktný. Na strane optimalizačného algoritmu totiž stojí metóda, ktorá na základe vstupných údajov takisto prináša určitý výstup (v našom prípade rozhodnutie o pridelení zdroja). Výhodou heuristického prístupu je však väčšia rýchlosť výpočtu (aj keď za cenu straty optimálneho riešenia), preto je v niektorých prípadoch vhodnejšie realizovať výpočet pri každej požiadavke o pridelenie zdroja. Tento prístup umožňuje lepšie zachytiť náhodné deje, ktoré prebiehajú v simulačnom modeli (rovnako ako v reálnom systéme). V prípade, že by mal byť v rámci navrhovaného systému na podporu rozhodovania použitý heuristický prístup, možno ho realizovať rovnako ako exaktný prístup, alebo zvoliť postup, kde je síce realizovaný inicializačný beh na získanie predpokladaných údajov, avšak počas druhého simulačného behu sú spolu s požiadavkou o pridelenie konkrétneho zdroja posielané aj informácie o aktuálnom stave jednotlivých parametrov systému, ktoré heuristický algoritmus potrebuje k svojmu výpočtu. Tento výpočet je potom realizovaný pre každú požiadavku zvlášť.

5.3 INTEGRÁCIA UČENIA POSILŇOVANÍM

Keďže učenie posilňovaním predpokladá existenciu simulačného prostredia, v ktorom môže byť rozhodovací algoritmus (agent) trénovaný k tomu, aby rozhodoval požadovaným spôsobom [5], javí sa tento prístup ako vhodná optimalizačná metóda v oblasti optimalizácie obslužných systémov, akými sú aj železničné dopravné uzly. V prípade učenia posilňovaním, ktoré pre svoje fungovanie vhodne využíva neurónovú sieť je potrebné rozdeliť celý proces použitia tejto metódy do dvoch častí a to trénovanie rozhodovacieho algoritmu (agenta) a používanie natrénovaného agenta.

Pre použitie algoritmov učenia posilňovaním je potrebné definovať:

- jednotlivé stavy prostredia vo forme vstupného stavového vektora. V prípade železničného uzla môže stav definovať poloha jednotlivých vlakov, poloha personálu, vzdialenosti medzi nimi, zoznam prebiehajúcich obslužných činností a podobne. To, ktoré konkrétne ukazovatele prostredia sú zvolené vychádza z charakteru úlohy, ktorá je riešená. Napríklad, v prípade, že rozhodujeme o pridelení vhodnej koľaje je dôležité definovať stav obsadenosti jednotlivých koľají, ale menej dôležité (prípadne úplne bezpredmetné) je aktuálny pracovný stav personálu. Jednotlivé hodnoty tohto vektora je vhodné upraviť na hodnoty z intervalu 0 až 1.

- jednotlivé akcie, ktoré povedú k zmene stavu prostredia. V prípade železničného uzla môže byť takouto činnosťou ukončenie obslužnej činnosti a prechod na inú, pridelenie vhodnej vlakovej trate, pridelenie personálu k vlaku, pridelenie obslužnej činnosti vlaku a podobne.

- odmeny/pokuty, ktoré sa budú udeľovať za každé uskutočnenie akcie. Tieto odmeny môže definovať účelová funkcia, v ktorej je snahou minimalizovať/maximalizovať určitú súhrnnú hodnotu, ako napríklad celkové meškanie vlakov, celkové pracovné vyťaženie personálu, počet využitých/nevyužitých zamestnancov, počet obsluhovaných vlakov a podobne.

V procese tréningu musí byť teda simulačný model schopný odosielať všetky definované informácie o stave prostredia vždy keď je potrebné uskutočniť rozhodnutie. Následne čaká na odpoveď, ktorá obsahuje informáciu, ktorý zdroj má byť pridelený. Po pridelení tohto zdroja stanoviť odmenu podľa definovaných pravidiel a informáciu o tej odmene odoslať optimalizačnému algoritmu.

Je samozrejmé, že proces tréningu predpokladá veľké množstvo realizácií simulačného behu a teda zaberie značný čas. V rámci použitia systému na podporu rozhodovania však užívateľ pracuje s natrénovaným algoritmom učenia posilňovaním. V procese používania tohto typu optimalizačného algoritmu teda simulačný model odosiela celkový stav celého prostredia a dostáva odpoveď obsahujúcu informáciu, ktorý zdroj má byť pridelený.

6 IMPLEMENTÁCIA NAVRHOVANÉHO SYSTÉMU S VYUŽITÍM SOFISTIKOVANÝCH METÓD

Vzhľadom na rôznorodosť a komplexnosť problémov, ktoré musí dispečer riešiť v rámci operatívneho riadenia sme sa rozhodli overiť použiteľnosť navrhovaného riešenia na vybraných izolovaných úlohách. Pre overenie použiteľnosti bolo vybrané údržbové depo, najmä pre jeho jednoduchšie technologické postupy v porovnaní s inými uzlami, akými sú napríklad zriaďovacie stanice. V rámci práce sme sa zaoberali dvomi izolovanými úlohami v dvoch rôznych údržbových depách. Týmito úlohami bolo prideľovanie personálu a stanovenie poradia vykonávania servisných činností.

6.1 PRIRADENIE RUŠŇOVODIČOV

V prevádzke údržbového depa je potrebné priradiť rušňovodičov jednotlivým vlakom, ktoré je potrebné presúvať v rámci depa. Príchod vlakov do depa je v rámci tejto úlohy zanedbaný a úloha sa zameriava výhradne na presuny v rámci depa medzi jednotlivými miestami, na ktorých je potrebné vykonať údržbu. Vlaky sa presúvajú relatívne krátke vzdialenosti, avšak vzhľadom k tomu, že v depe nemôžu prekročiť obmedzenú rýchlosť, tieto presuny v mnohých prípadoch trvajú významne dlhý časový úsek. Tieto časové úseky však nie je možné skrátiť a preto sa ako priestor na optimalizáciu javia vzdialenosti, ktoré musia jednotliví rušňovodiči prejsť, keď sa presúvajú od jedného vlaku k druhému. Pri definovaní tejto úlohy, tak ako k nej budeme pristupovať v rámci riešenia rôznych optimalizačných metód si uvedomujeme značnú mieru abstrakcie, ako aj zanedbania viacerých dôležitých ukazovateľov, ktoré musia byť v skutočnom železničnom dopravnom uzle dodržané (mnohé

zákonne obmedzenia, prípadne obmedzenia vyplývajúce zo zákonníka práce). K tejto abstrakcii sme pristúpili najmä preto, že cieľom práce je poukázať na možnosti spolupráce simulačného modelu a optimalizačných algoritmov v rámci jedného systému na podporu rozhodovania. Spomínané zanedbané obmedzenia vyplývajú najmä z konkrétneho železničného uzla a v prípade nasadenia systému do prevádzky považujeme za jednoduché ich do systému zapracovať.

6.1.1 Optimalizačný algoritmus založený na simulácii

Pre overenia navrhnutého postupu sme sa rozhodli použiť optimalizačný algoritmus založený na simulácii. Výskumy [2][4][7][11][13][14][15] ukázali, že najmä v počiatočných fázach riešenia optimalizačného problému je vhodné zostaviť matematický model a pokúsiť sa nájsť optimálne riešenie. Pre potreby použitia optimalizačného algoritmu bol preto vytvorený matematický model, popisujúci problém priradenia rušňovodičov v údržbovom depe. Tento matematický model bol inšpirovaný modelom, ktorý rieši úlohu okružných jazd.

Riešenie celého problému pomocou matematického modelu je časovo náročné, a preto nevhodné na podporu rozhodovania pri operatívnom rozhodovaní v reálnom čase. Vzhľadom na túto skutočnosť bol navrhnutý optimalizačný algoritmus založený na simulácii. Tento prístup je založený na možnosti redukovať stavový priestor (a tým aj výpočtový čas) obmedzením počtu budúcich úloh, ktoré sa berú do úvahy stanovením časového limitu (nazývaného horizont). Na vykonanie navrhovaného algoritmu je potrebné vykonať inicializačný simulačný beh. Tento beh je dokončený pred začatím optimalizácie a využíva určitý typ jednoduchého algoritmu na alokáciu každého zdroja, ktorý je štandardný v simulačných modeloch tohto typu a zvyčajne je založený na implementácii prioritného zoznamu. Počas tohto simulačného behu sa vytvorí inicializačný dataset, ktorý obsahuje jednotlivé obslužné aktivity, ktoré sa vykonávajú v simulačnom modeli, ako aj pracovné časy jednotlivých vodičov. Následne sa vykoná simulačný beh s použitím optimalizačnej metódami (v našom prípade s použitím exaktného výpočtu pomocou matematického modelu). Počas tohto behu sa každá požiadavka na priradenie zdroj (rušňovodič) k vlaku spracuje tak, že sa vytvorí čiastkový dataset a tento sa použije pri výpočte. Čiastkový dataset sa vytvorí z inicializačného datasetu na základe veľkosti zvoleného časového horizontu. Napríklad ak je časový horizont 3 hodiny, čiastkový dataset pozostáva zo všetkých činností, ktoré sa vyskytnú v priebehu troch hodín od príslušnej požiadavky. Tým sa, samozrejme, výrazne zmenší veľkosť údajov a zníži sa výpočtový čas algoritmov použitých algoritmov.

Na získanie údajov zo simulačného modelu bol vykonaný inicializačný simulačný beh, v ktorom boli jednotliví rušňovodiči priradení k vlakom pomocou funkcie „First Fit“ (aj keď simulačný softvér Villon poskytuje niekoľko algoritmov na personálu, v tomto prípade sa použila funkcia „First Fit“, pretože algoritmus použitý v skutočnom depe nebol pri vývoji modelu nebol známy). Táto funkcia vždy vyberie prvého rušňovodiča, ktorý je v dostupnom stave. Ak v tomto stave nie je žiadny rušňovodič, vlak čaká, kým jeden z rušňovodičov bude k dispozícii. Súbor údajov vytvorený v inicializačnom simulačnom behu pozostáva z jednotlivých posuvných činností a rušňovodičov. Jednotlivé presuny vlakov sa vzťahujú na konkrétny vlak. Časové zmeny jednotlivých rušňovodičov sú modelované ako noví rušňovodiči, hoci v skutočnosti pracuje jeden rušňovodič vo viacerých zmenách. Pre 7-dňový simulačný beh (7-dňová prevádzka v údržbovom depe) bol vytvorený základný súbor údajov s 207 činnosťami a 84 rušňovodičmi, čo však neznamená 84 rušňovodičov, ale 84 jednotlivých pracovných zmien. Algoritmus pre realizáciu exaktného výpočtu bol v tomto prípade implementovaný ako externá aplikácia a to najmä z dôvodu, že bol použitý štandardný riešiteľ, preto bolo v tomto kroku navrhnuté komunikačné rozhranie, ktoré umožňuje komunikáciu po sieti priamym spojením simulačného modelu a externej aplikácie za pomoci socketov. Správy sa posielajú vo formáte JSON. Výpočty pomocou matematického

modelu s použitím tohto súboru údajov boli vykonané pomocou štandardného riešiteľa IP Gurobi [10]. Celá metóda bola implementovaná v jazyku java. Cieľom použitia navrhovaného algoritmu je skrátiť čas, ktorý trvá rušňovodičovi presunúť sa medzi vlakmi. V čase, keď sa rušňovodiči presúvajú, sa v simulačnom modeli nachádzajú v stave „presúvajúci sa“ a „vracajúci sa“. Keďže daná metóda musí byť časovo efektívna, bol určený hraničný čas výpočtu 1 hodina. Experimenty sa uskutočnili s rôznymi dĺžkami časového horizontu (1 hodina, 1,5 hodiny, 2 hodiny, 2,5 hodiny, 3 hodiny a 3,5 hodín) pre 7-dňový simulačný beh.

Tabuľka 4: Porovnanie výsledkov experimentov s rôznym časovým výhľadom

Časový horizont	Presúvajúci sa (hod.)	Vracajúci sa (hod.)
First Fit	30,12	21,52
1 hod	28,22	20,58
1.5 hod	28,76	20,42
2 hod	28,78	19,70
2.5 hod	28,17	19,88
3 hod	29,74	19,83
3.5 hod	28,97	19,33

Keďže simulačný model použitý na vykonanie experimentov je deterministický, experimenty boli vykonané v rámci jedného simulačného behu. Výsledky získané pomocou modelu používajúceho First Fit (pôvodne použitá funkcia) a výsledky získané pomocou navrhovaného prístupu s daným časovým horizontom boli porovnané určením času, o koľko sa trvanie každého stavu znížilo (zvýšilo) v percentách. Na základe výpočtu Pearsonovho korelačného koeficientu sa skúmala korelácia medzi trvaním stavu „vracajúci sa“ a veľkosťou časových horizontov, ale takáto korelácia sa nezistila pre stav „presúvajúci sa“. Je to pravdepodobne preto, že časové rozdiely medzi časovými horizontmi nie sú dostatočne veľké. Experimenty však s väčšími časovými horizontmi nebolo možné vykonať, pretože výpočtový čas prekročil stanovenú hraničnú hodnotu. Tento problém by sa dal v budúcnosti odstrániť použitím heuristických metód.

6.1.2 Použitie algoritmov učenia posilňovaním

Iným prístupom, ktorý bol použitý pre overenie navrhovaného postupu je učenie posilňovaním. Tento prístup sme sa rozhodli použiť na základe predpokladu, že tento prístup je určený na riešenie komplexných problémov[6] a mohol by dokázať vyriešiť problémy danej veľkosti. Bol vybraný Proximal Policy Optimization (PPO) algoritmus. Vychádzajúc z postupu, ktorý bol definovaný pre integráciu algoritmov učenia posilňovaním boli definované tieto parametre:

- prostredie - zoznam časov začiatku plánovaných aktivít (v prípade, že aktivita už prebehla, plánovaný čas začiatku je -1), poloha všetkých vlakov v systéme (počet týchto vlakov sa môže meniť, preto je vektor doplnený na potrebnú veľkosť pomocou núl), poloha rušňovodičov od vlaku, ktorému je potrebné priradiť rušňovodiča;
- akcie - priradený rušňovodič;
- odmeny/pokuty - sú priradené na konci simulačného behu na základe počtu obslužených vlakov, čím viac sa tento počet blíži k maximálnemu počtu obslužených vlakov, tým je odmena vyššia. Táto odmena je normalizovaná v rozmedzí 0 a 1.

Vzhľadom na časovú náročnosť tréningu agenta učenia posilňovaním bol tréning realizovaný na 3 dňovom simulačnom behu, v rámci ktorého bolo potrebné pridelovať jednotlivých rušňovodičov jednotlivým aktivitám presunu. V rámci tohto simulačného behu bolo plánovaných 92 aktivít presunu, maximálny počet vlakov ktoré sa nachádzajú v jednom momente v systéme je 36, v simulačnom modeli sú traja rušňovodiči. Jednotlivé aktivity presunu, súradnice vlakov a vzdialenosť jednotlivých rušňovodičov od konkrétneho vlaku

tvoria vektor popisujúci prostredie. Odmena bola nastavená na základe dokončených technológií jednotlivých vlakov - vlak 92 odišiel obslužený zo systému. Maximálny počet takýchto obslužených vlakov je 65. Odmena bola nastavená nasledovne:

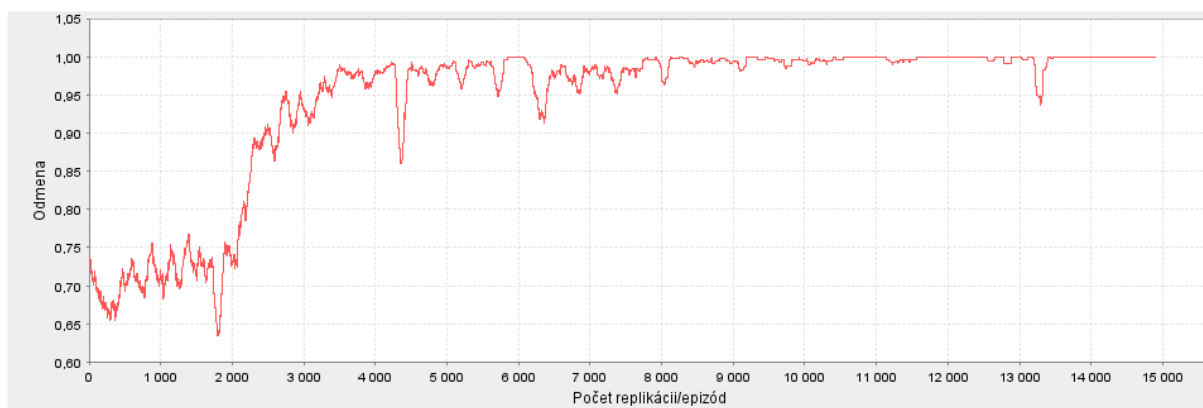
- ak bolo obslužených menej ako 10 vlakov, odmena je 0 (menej ako 10 obslužených vlakov sa počas realizácie experimentov nikdy nevyskytlo)
- ak bolo obslužených viac ako 10 a menej ako 65 vlakov, odmenu vypočítame vzťahom: $(x-10)/(65-10)$, kde x je počet obslužených vlakov;
- ak bolo obslužených 65 vlakov, odmena je 1.

Aby bolo možné použiť algoritmus učenia posilňovaním, musí simulačný model spĺňať požiadavky na použitie tohto algoritmu, to znamená, že musí byť schopný komunikovať s algoritmom učenia posilňovaním. V tejto dizertačnej práci boli navrhnuté spôsoby komunikácie s externými modulmi a preto bolo možné využiť simulačný model železničného dopravného uzla tak, ako to požaduje algoritmus učenia posilňovaním. Algoritmus učenia posilňovaním má slúžiť ako podporný prvok pre rozhodovací proces dispečera, a teda sa predpokladá, že po natrénovaní dokáže takýto agent poskytnúť návrh rozhodnutia pri pridelení zdroja, konkrétne rušňovodiča v údržbovom depe. Pre overenie navrhovaného riešenia bol vybraný PPO agent, teda jeden z algoritmov učenia posilňovaním. Tento algoritmus bol implementovaný v jazyku python, za použitia knižnice PyTorch. Parametre PPO agenta boli nastavené nasledovne:

- počet epoch - 4
- faktor zrážania gamma - 0,999
- veľkosť kroku učenia alfa - 0,003
- veľkosť dávky - 500

Neurónová sieť použitá ako aktor je tvorená 170 neurónmi na vstupnej vrstve, následne dvoma vrstvami s 256 neurónmi a 3 neurónmi na výstupnej vrstve. Neurónová sieť použitá ako kritik je tvorená 170 neurónmi na vstupnej vrstve, dvoma vrstvami s 256 neurónmi a jedným neurónom na výstupnej vrstve. Počas tréningu PPO algoritmu bolo realizovaných 15 000 epizód. Keďže tréningová fáza prebieha pred nasadením systému a teda nemusí byť dosiahnutý nízky čas tréningu, tréning bol realizovaný na bežnom počítači a čas jednej epizódy bol priemerne 20 sekúnd. Tento tréning je zobrazený na grafe na obr. 3.

Z tohto grafu vyplýva, že algoritmus PPO je možné použiť pri realizácii rozhodnutí vo vyššie popísanej úlohe, nakoľko natrénovaný algoritmus dosahuje maximálnu odmenu a teda je zabezpečené obsluženie maximálneho počtu vlakov. Je dôležité upozorniť, že realizovaný experiment bol vykonaný na inej konfigurácii simulačného modelu ako experiment popísaný v kapitole 6.1.1 a preto nie je vhodné porovnávať dosiahnuté výsledky.



Obr. 3: Tréningový proces algoritmu učenia posilňovaním

Na rovnakej konfigurácii ako bol trénovaný PPO algoritmus bol použitý algoritmus First Fit. Tento však nedokázal dosiahnuť počet obslužených vlakov 65, dosiahol len 40 obslužených vlakov. Táto situácia nastala kvôli vzniku deadlocku (dva vlaky idú oproti

sebe po jednej koľají). Na daný problém bol použitý aj presný výpočet založený na matematickom modeli. Tento výpočet (najmä kvôli abstrakcii mnohých vlastností uzla siete) tiež viedol k deadlockom. Je teda možné povedať, že PPO algoritmus dosahuje lepšie výsledky ako First Fit algoritmus. Je dôležité povedať, že vykonané experimenty sa uskutočnili v deterministickom prostredí. V stochastickom prostredí nedosahuje natrénovaný algoritmus dobré výsledky, a preto je v budúcnosti potrebné rozšíriť proces tréningu na stochastické prostredie.

6.2 URČENIE PORADIA ÚLOH

Rôzne vlaky si vyžadujú rôzne typy údržby. Typy údržby závisia od vlastností vlaku, napr. dieselové lokomotívy potrebujú doplniť palivo a elektrické lokomotívy nie. Vlaky s vysokým stupňom degradácie potrebujú dôkladnejšie čistenie, výmenu niektorých častí a pod. Okrem toho by sa typ údržby mohol meniť vzhľadom na to, kedy bola daná činnosť údržby vykonaná naposledy, pretože kontroly sa musia vykonávať v pravidelných intervaloch, napr. raz za mesiac/rok. Určenie poradia, v akom sa majú činnosti údržby vykonávať, môže byť problematické najmä preto, že niektoré typy činností sa môžu vykonávať len na určitých špecifických miestach v rámci depa (napríklad nad údržbovou jamou). Vlaky prichádzajúce do depa môžu mať rôzne priority, najmä pokiaľ ide o ich plánovaný čas odchodu. Preto je niekedy výhodnejšie odložiť údržbu vlaku 1, ktorý prišiel o niekoľko minút skôr, a počkať na vlak 2, ktorý má plánovaný čas odchodu oveľa skôr ako vlak 1. Ak by sme začali údržbu vlaku 1 na určitom mieste, toto miesto bude zablokované a údržba vlaku 2 sa môže vykonať až po dokončení údržby vlaku 1. Toto rozhodnutie môže spôsobiť nežiaduce meškanie vlaku 2. Okrem toho sa môže stať, že údržba vlaku trvá veľmi dlho a počas tohto času by sa na danom mieste mohla vykonávať údržba niekoľko ďalších vlakov. V takomto prípade je potrebné zvážiť, či je vhodnejšie odložiť jeden alebo viacero vlakov. Vo vybranom depe je potrebné vykonať rôzne typy údržbových prác na prichádzajúcich vlakoch, a to ťažkú údržbu, ľahkú údržbu, čistenie interiéru, čistenie exteriéru, čistenie grafitov a tankovanie. Každý vlak si vyžaduje vykonanie určitého počtu týchto údržbových úloh. Úlohy sa môžu vykonávať na konkrétnych miestach vo vnútri údržbového depa - ťažká údržba sa môže vykonávať na štyroch špeciálnych koľajach; čistenie interiéru, čistenie exteriéru a čistenie grafitov sa môže vykonávať len na jednom mieste pozostávajúcom z dvoch koľají; tankovanie sa môže vykonávať len na jednej koľají; a ľahká údržba sa môže vykonávať všade vrátane odstavných koľají. Pre hľadanie vhodného riešenia tejto úlohy sme sa podobne ako pri úlohe priradovania rušnovodičov rozhodli použiť optimalizačný algoritmus založený na simulácii popísaný v kapitole 6.1.1

6.2.1 Optimalizačný algoritmus založený na simulácii – určenie poradia úloh

V rámci inicializačného behu boli zo simulačného modelu pomocou funkcie "First Fit" získané potrebné údaje. Celkový dataset zahŕňa 15 vlakov a 63 údržbových aktivít, ktoré je potrebné vykonať. Pre realizáciu danej metódy boli zozbierané všetky údaje, ktoré sú potrebné pre riešenie matematického modelu, to znamená počet vlakov, počet stanovišť, reálne príchody vlakov, plánované odchody vlakov, zoznam úloh na vlaku, zoznam úloh, ktoré je možné vykonať na danom stanovišti. Definovanie komunikačného rozhrania Ako už bolo spomenuté, táto optimalizačná metóda implementovaná ako externá aplikácia a to najmä z dôvodu, že bol použitý štandardný riešiteľ, preto bolo v tomto kroku navrhnuté komunikačné rozhranie, ktoré umožňuje komunikáciu po sieti priamym spojením simulačného modelu a externej aplikácie za pomoci socketov. Správy sa posielajú vo formáte JSON. Výpočty pomocou matematického modelu s použitím tohto súboru údajov boli vykonané pomocou štandardného riešiteľa IP Gurobi [10]. Celá metóda bola implementovaná v jazyku java. Overenie Pre overenie použiteľnosti navrhovaného riešenia boli realizované

experimenty s rôznym časovým výhľadom. Boli teda použité čiastkové datasety, ktoré boli vytvorené pomocou rôznych časových horizontov od 1 hodiny do 3,5 hodiny. Čiastkové datasety obsahovali v priemere 7 vlakov a 30 úloh. Je dôležité poznamenať, že presný výpočet pracuje s čiastkovým súborom údajov obmedzeným konkrétnym časovým horizontom, a preto nie je potrebné zahrnúť všetky servisné úlohy. Výsledky experimentov sú uvedené v nasledujúcej tabuľke. Celkový čas meškania zahŕňa aj meškania vlakov, ktoré v príslušný deň neodišli. Oneskorenie nedokončeného vlaku sa vypočíta ako rozdiel medzi plánovaným časom odchodu a časom ukončenia simulácie.

Tabuľka 5: Celkové meškание pri použití rôznych časových horizontov

Časový horizont	Meškание v minútach	Počet nevypravených vlakov
First Fit	5234	4
1 hod	4755	4
1.5 hod	4533	4
2 hod	4350	4
2.5 hod	4176	3
3 hod	3120	3
3.5 hod	3445	2

Ako je možné vidieť, navrhovaný algoritmus priniesol zlepšenie oproti algoritmu "First Fit". Z výsledkov vidieť, že medzi zvyšujúcim sa časovým výhľadom a kvalitou riešenia je závislosť, ktorá bola potvrdená pomocou určenia Pearsonovho koeficientu.

6.2.2 Použitie algoritmov učenia posilňovaním

Podobne ako pri riešení úlohy priradenia personálu, aj v prípade úlohy určenia poradia obslužných úloh sme sa rozhodli navrhnúť použitie algoritmov učenia posilňovaním. Reálny systém nie je deterministický, a preto sme (na rozdiel od úlohy priradenia personálu) použili PPO algoritmu v stochastickom prostredí. Pre použitie algoritmu učenia posilňovaním bolo potrebné pre vybranú úlohu definovať nasledujúce parametre:

- prostredie – index aktivity, o ktorej sa aktuálne rozhoduje; plánovaný odchod vlaku; trvanie jednotlivých aktivít vlaku, o ktorom rozhodujeme; najskorší možný čas začiatku jednotlivých aktivít vlaku, o ktorom rozhodujeme; časy vyjadrujúce, koľko budú obsadené jednotlivé koľaje; trvanie jednotlivých aktivít jednotlivých vlakov, ktoré sa nachádzajú v modeli; najskorší možný čas začiatku jednotlivých aktivít jednotlivých vlakov, ktoré sa nachádzajú v modeli; aktuálny simulačný čas,

- akcie - bude úloha, o ktorej rozhodujem realizovaná,
- odmena = $(1 - (\text{súhrnné meškание}) / 3) * 0,5 + (5 - \text{nevypravené vlaky}) * 0,1$.

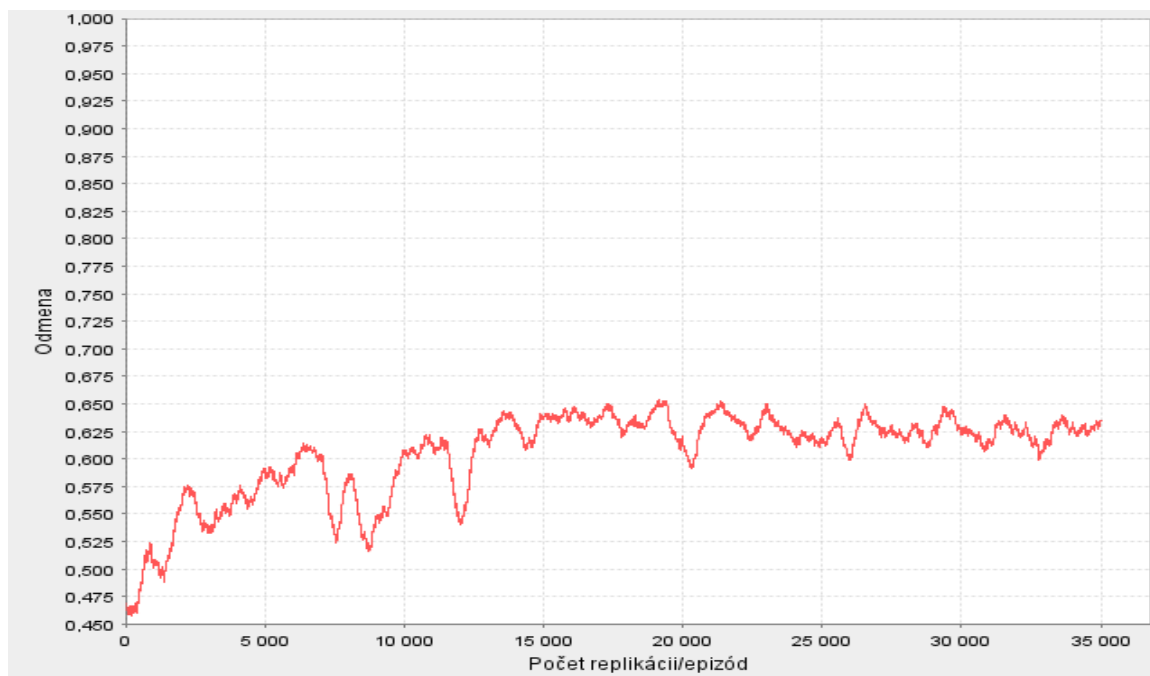
Pre použitie navrhovaného prístupu bol použitý rovnaký komunikačný mechanizmus, ako pri riešení úlohy priradenia personálu. PPO algoritmus bol implementovaný v jazyku python, za použitia knižnice PyTorch. Parametre PPO agenta boli nastavené nasledovne:

- počet epoch - 4
- faktor zrážania gamma - 0,99
- veľkosť kroku učenia alfa - 0,003
- veľkosť dávky - 900

Neurónová sieť použitá ako aktor je tvorená 170 neurónmi na vstupnej vrstve, následne dvoma vrstvami s 256 neurónmi a 3 neurónmi na výstupnej vrstve. Neurónová sieť použitá ako kritik je tvorená 170 neurónmi na vstupnej vrstve, dvoma vrstvami s 256 neurónmi a jedným neurónom na výstupnej vrstve. Pri rozmere neurónovej siete, ako aj pri nastavení vstupných parametrov sme vychádzali zo skúsenosti s riešením deterministickej úlohy priradenia personálu, pričom veľkosť dávky musí byť väčšia ako počet rozhodnutí v jednom simulačnom behu. Obdobne, ako pri úlohe priradenia personálu bol algoritmus učenia posilňovaním spolu so simulačným jadrom nasadený na štandardnom počítači, pričom

na tomto počítači prebiehala aj prípravná fáza. Pre natréňovanie neurónovej siete bolo realizovaných 32 500 simulačných behov (replikácií), nakoľko sme predpokladali, že počet replikácií pri stochastickej úlohe musí byť väčší, ako pri deterministickej úlohe, pri ktorej bolo realizovaných 15 000 replikácií a pri určení presného počtu replikácií sme vychádzali z toho, v akej fáze tréningu je pozorovateľné ustáľovanie dosahovaných výsledkov. Po realizácii tréningu sa však nepodarilo dosiahnuť výsledky, nakoľko algoritmus učenia posilňovaním nevykazoval známky natréňovania (dosahovanie lepších výsledkov, ako v prvotných fázach tréningu). Vzhľadom k tomu bolo potrebné vrátiť sa k nastaveniu jednotlivých parametrov. Ako prvý krok sme rozšírili informácie o stave prostredia. Zahrnuli sme obsadenosť jednotlivých koľají, ako aj plánovaný príchod jednotlivých vlakov. Bohužiaľ, ani toto sa neukázalo ako postačujúce. Preto sme pristúpili k rozšíreniu veľkosti siete a to na sieť pre aktora s 202 neurónov na vstupnej vrstve, potom tri vrstvy s 512 neurónmi a výstupná vrstva s 2 neurónmi a sieť pre kritika s 202 neurónov na vstupnej vrstve, potom tri vrstvy s 512 neurónmi a výstupná vrstva s 1 neurónom. Ani pri takejto zmene však stále nedochádzalo k trénovaniu algoritmu. Následne sme teda rôzne menili parametre alfa, gamma a počet epoch. Výsledky všetkých týchto experimentov neuvádzame v tejto práci vzhľadom k tomu, že nedochádzalo k žiadnemu zlepšeniu a teda výsledky boli podobné, ako by sme použili netréňovanú sieť. Cieľom všetkých týchto experimentov však bolo dosiahnutie takých nastavení, že algoritmus učenia posilňovaním dosahuje lepšie výsledky ako pevne stanovené poradie úloh. Jednotlivé experimenty boli časovo náročné, a teda sme sa pri jednotlivých zmenách nevracali k východiskovému nastaveniu, ale snažili sme sa k zmenám jednotlivých parametrov pristupovať s vedomím, že jednotlivé parametre ovplyvňujú dosahovaný výsledok spoločne a nie izolovane.. Vzhľadom k tomu, je problematické určiť, ktorá zmena priniesla zlepšenie. Tieto parametre sú:

- počet epoch - 12
- faktor zrážania gamma - 0,998
- veľkosť kroku učenia alfa - 0,001
- veľkosť dávky - 900



Obr. 4: Tréningový proces

Obr. 4 zobrazuje dosiahnutý vývoj tréningu algoritmu PPO. Ako môžeme vidieť, dosahované výsledky začínajú byť stabilnejšie od 15 000 replikácií. Bolo vykonaných viac ako 30 000 replikácií, pričom väčšiu stabilitu tréningu sa nepodarilo dosiahnuť.

Takto natrénovaný algoritmus je možné nasadiť ako sofistikovaný algoritmus v systéme na podporu rozhodovania. Nasledujúce tabuľky zobrazuje dosiahnuté výsledky pre 30 replikácii v porovnaní s pôvodne používaným algoritmom s pevne stanoveným počtom úloh, ako aj v porovnaní s netrénovaným algoritmom PPO (náhodný výber). Pre dosiahnuté celkové meškania bol určený 90 percentný interval spoľahlivosti. Tabuľka 11.3 zobrazuje dosiahnuté celkové meškania. V prípade nevypravených vlakov je zahrnuté meškanie do 6 rána nasledujúceho dňa (dĺžka simulačného behu).

Tabuľka 6: Porovnanie výsledkov experimentov s použitím algoritmu PPO pri určení poradia úloh

	<i>Pevne určené poradie</i>	<i>Náhodný výber</i>	<i>Natrénovaný algoritmus</i>
<i>Celkové meškание (hodiny)</i>	<87,9;171,1>	<130,1;201,2>	<35,7;55,3>

Na základe výsledkov môžeme povedať, že algoritmus PPO s nami nastavenými parametrami je vhodný na riešenie danej úlohy v stochastickom prostredí. Z vykonaných experimentov je zrejmé, že úloha je citlivá na nastavenie jednotlivých parametrov. V budúcnosti by sme sa chceli v tejto oblasti venovať ďalšiemu výskumu a to najmä použitiu grafových neurónových sietí [16]. Viacerí autori tento prístup využili pri realizácii výskumu v oblasti železničnej dopravy [22] [12] a preto predpokladáme, že by mohol byť vhodný aj na riešenie typov úloh, akými sa zaoberáme v tejto práci.

ZÁVER

Riadenie železničného dopravného uzla je náročné na všetkých úrovniach, predovšetkým však na operatívnej úrovni. Dispečeri musia prinášať rozhodnutia, tak aby bola zabezpečená efektívna prevádzka celého železničného dopravného uzla. Uvedomujúc si náročný rozhodovací proces dispečerov je potrebné zaoberať možnosťami poskytovania podpory rozhodovania a to najmä v reálnom čase. Dizertačná práca ponúka analýzu niektorých existujúcich informačných, a najmä dispozičných systémov v oblasti železničnej dopravy. Takýmito systémami sú napríklad Prevádzkový informačný systém, systémy spoločnosti Siemens (dôležitým je najmä systém VICOS), Graficko-technologická nadstavba a mnohé ďalšie. Keďže tieto systémy nie sú v súčasnosti dostačujúce a hlavne sa zväčša zaoberajú čiastkovými problémami a nie železničným dopravným uzlom ako celkom, bol navrhnutý nový systém na podporu rozhodovania. Vzhľadom na skúsenosti s vytváraním simulačných modelov navrhujeme systém, ktorý plne využíva výhody, ktoré ponúkajú simulačné modely a to tak, že simulačný model stojí v jadre navrhovaného systému. V dizertačnej práci bola zostavená metodika popisujúca postup tvorby systému na podporu rozhodovania v reálnom čase postaveného na simulačných modeloch železničných dopravných uzlov, ktorá bude nápomocná pri tvorbe a nasadení takéhoto systému. V rámci tejto metodiky sú detailne popísané jednotlivé kroky, ktoré musí tvorca systému nevyhnutne vykonať, ako aj návrhy riešení viacerých problémov, ktoré môžu pri tvorbe takéhoto systému vzniknúť. Realizácia takéhoto systému môže napomôcť dispečerom v ich náročnej práci, zefektívniť operatívne riadenie a tým celé fungovanie železničnej dopravy.

Za teoretický prínos dizertačnej práce považujeme návrh architektúry systému na podporu rozhodovania s využitím simulačných modelov v jadre tohto systému. Architektúra tohto systému je navrhnutá tak, aby sa jednotlivé časti dali implementovať nezávisle. V práci sú definované informácie o reálnom systéme, ktoré sú potrebné pre fungovanie systému na podporu rozhodovania, ako aj možnosti zberu takýchto informácií. Ďalej je dôležitým teoretickým prínosom zostavenie metodiky, v rámci ktorej je popísaný postup, akým systém na podporu rozhodovania vytvoriť a nasadiť v železničnom uzle.

Za praktický prínos považujeme rozšírenie agentovo-orientovanej architektúry ABASim o možnosť komunikácie s externými systémami (rôznymi mikroslužbami, modulmi aj komplexnými softvérovými aplikáciami) pomocou navrhnutého komunikačného agenta využívajúceho služby websocket komunikácie, nakoľko takéto rozšírenie môže byť v budúcnosti využité rôznymi spôsobmi.

Navrhovaný systém na podporu rozhodovania umožňuje čiastočne nahradiť ľudský faktor pri tvorbe návrhov rozhodnutí, tým že umožňuje využívať sofistikované algoritmy. V práci boli vybrané metódy použité pre overenie navrhovaného postupu na vybraných úlohách pridelovania personálu a určovania poradia servisných úloh v reálnych železničných dopravných uzloch. Ďalším z praktických prínosov dizertačnej práce je teda aplikácia výsledkov takýchto metód, ktoré okrem využitia v systéme na podporu rozhodovania, môžu pomôcť zlepšiť samotné simulačné modely, pretože zvyšujú presnosť modelovania vzhľadom na reálne rozhodnutia v praxi. V rámci použitia sofistikovaných metód sme sa zaoberali najmä použitím algoritmov učenia posilňovaním. Nakoľko sa jedná o inovatívny prístup, ktorý zatiaľ nie je v širokej miere v rámci riadenia železničných dopravných uzlov používaný, považujeme za prínosné, že v rámci tejto práce bola vypracovaná metodika nasadenia takýchto algoritmov v rámci systémov na podporu rozhodovania. Veríme, že dizertačná práca bude prínosom v oblasti poskytovania podpory rozhodovania dispečerov v reálnom čase a môže prispieť k budúcemu zefektívneniu operatívneho riadenia v železničnom dopravnom uzle.

ZOZNAM POUŽITEJ LITERATÚRY

- [1] N. Adamko. „Agentovo orientovaná simulácia zložitých obslužných systémov“. In: *Habilitačná práca* (2013), s. 114.
- [2] M. Bažant et al. „Simulation-based rail traffic optimisations applying multicriterial evaluations of variants“. In: *Mendel* 25.1 (2019), s. 139–146.
- [3] J. Čamaj a J. Gašparík. *Informačné a komunikačné technológie v železničnej doprave*. prvé vydanie. Žilinská univerzita v Žiline: EDIS, 2011, s. 179. isbn: 978-80-554-0307-6.
- [4] A. D'Ariano et al. „A branch and bound algorithm for scheduling trains in a railway network“. In: *European Journal of Operational Research* 183.2 (2007), s. 643–657.
- [5] F.L. Da Silva a A.H. Reali Costa. „A survey on transfer learning for multiagent reinforcement learning systems“. In: *Journal of Artificial Intelligence Research* 64 (2019), s. 645–703.
- [6] D. Drungilas et al. „Deep reinforcement learning based optimization of automated guided vehicle time and energy consumption in a container terminal“. In: *zv.* 67. 2023, s. 397–407.
- [7] M. Fischetti a M. Monaci. „Using a general-purpose Mixed-Integer Linear Programming solver for the practical solution of real-time train rescheduling“. In: *European Journal of Operational Research* 263.1 (2017), s. 258–264.
- [8] J. Gašparík a J. Kolář. *Železniční doprava - technologie, řízení, grafiky a dalších 100 zajímavostí*. 1. vydanie. U Pruhonu 22, Praha 7: Grada Publishing, a.s., 2017, s. 432. isbn: 978-80-271-0058-3.
- [9] J. Gašparík, D. Lichner a P. Blaho. „Železničná dopravná prevádzka - základy dopravnej prevádzky“. In: *ISBN 978-80-554-0996-2* 1 (2015), s. 407.
- [10] Gurobi. *Gurobi Optimizer*. [cit.2022-05-06]. Dostupné na internete: www.gurobi.com.
- [11] P. Hanczar a A. Zandi. „A novel model and solution algorithm to improve crew scheduling in railway transportation: A real world case study“. In: *Computers and Industrial Engineering* 154 (2021).
- [12] J. S. W. Heglund et al. „Railway Delay Prediction with Spatial-Temporal Graph Convolutional Networks“. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, s. 1–6.

- [13] J. Janáček et al. „Optimization of periodic crew schedules with application of column generation method“. In: *Transportation Research Part C: Emerging Technologies* 83 (2017), s. 165–178.
- [14] L. Jánošíková, A. Kavička a M. Bažant. „Optimal operation scheduling and platform track assignment in a passenger railway station“. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 228.3 (2014), s. 271–284.
- [15] Ľ. Jánošíková a M. Kreml. „Train platforming problem“. In: *Lecture Notes in Geoinformation and Cartography* (2015), s. 149.
- [16] T. N. Kipf a M. Welling. „Semi-supervised classification with graph convolutional networks.“ In: *In International Conference on Learning Representations (ICLR)* (2017).
- [17] P. Kučera a P. Drdla. „Geografická metóda pro Automatické stavění vlakových cest“. In: *Vědeckotechnický sborník Správy železnic* 3/2020 (2020), s. 3–18.
- [18] J. Majerčák et al. *ŽELEZNIČNÁ DOPRAVNÁ PREVÁDZKA. Technológia železničných staníc*. 1. vydanie. Žilinská univerzita v Žiline: EDIS, 2008, s. 230. isbn: 978-80-8070-887-0.
- [19] Siemens AG Mobility Division. *Efficient rail transport with the Vicos OC operations control system family*. [cit.2021-08-02]. Dostupné na internete: <https://www.mobility.siemens.com/global/en.html>.
- [20] V. Polach. „Plán vlakové dopravy v GTN zcela nově“. In: *Technologie* 4 (2021), s.52–55.
- [21] J. Tkáč. „Predpis PREVÁDZKOVÝ INFORMAČNÝ SYSTÉM“. In: *Železnice Slovenskej republiky* (2018).
- [22] R. Wang et al. „The Method about the Prediction and Analysis of Causes for Railway Accidents based on Graph Convolutional Neural Network“. In: *2023 7th International Conference on Transportation Information and Safety (ICTIS)*. 2023, s. 496–501.

ZOZNAM VLASTNÝCH PUBLIKÁCIÍ

- [1] A. Galadíková a N. Adamko. "Simulation-based methods to support the real-time management of railways nodes". In: *Transportation Research Procedia* 55 (2021), s. 1345-1352
- [2] A. Galadíková. "Využitie algoritmov učenia posilňovaním pri optimalizácii železničných uzlov". In: *Využitie kvantitatívnych metód vo vedeckovýskumnej činnosti a v praxi. Zborník zo seminára. - 1. vyd. - Bratislava: Vydavateľstvo EKONÓM* (2021), ISBN 978-80-225-4831-1., s. 25-31
- [3] A. Galadíková a N. Adamko. "Simulation-based optimization of personnel assignment in railway nodes". In: *2022 IEEE 16th International Scientific Conference on Informatics, Informatics 2022 - Proceedings* (2022), s. 100–105
- [4] A. Galadíková. "Communication architecture for real-time decision support systems in railway nodes". In: *TRANSCOM 2023: 15th International scientific conference of young scientists on sustainable, modern and safe transport. - 1 vyd. - Amsterdam: Elsevier, 2023. - s. 1203-1210*
- [5] A. Galadíková a N. Adamko. "Usage of proximal policy optimization algorithm for personnel assignment in railway nodes". In: *Proceedings of The International Conference on Information and Digital Technologies (IDT)*, 2023. - 1 vyd. - Danvers: Institute of Electrical and Electronics Engineers, 2023. - ISBN 979-8-3503-0586-9. - s. 37-43