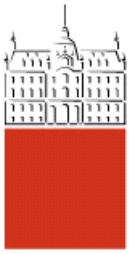
Univerza
v Ljubljani
Fakulteta
za gradbeništvo
in geodezijo



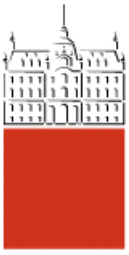
Jamova cesta 2
1000 Ljubljana, Slovenija
http://www3.fgg.uni-lj.si/

DRUGG – Digitalni repozitorij UL FGG
http://drugg.fgg.uni-lj.si/

V zbirki je izvirna različica doktorske disertacije.

Prosimo, da se pri navajanju sklicujete na bibliografske podatke, kot je navedeno:

University
of Ljubljana
Faculty of
Civil and Geodetic
Engineering



Jamova cesta 2
SI – 1000 Ljubljana, Slovenia
http://www3.fgg.uni-lj.si/en/

DRUGG – The Digital Repository
http://drugg.fgg.uni-lj.si/

This is an original PDF file of doctoral thesis.

When citing, please refer as follows:

Šemrov, D. 2016. Časovno načrtovanje železniškega prometa z uporabo metode spodbujevanega učenja. = Railway traffic scheduling with use of reinforcement learning. Doctoral dissertation. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo. (Mentor Žura, M., somentor Todorovski, L.)

http://drugg.fgg.uni-lj.si/5445/

Datum arhiviranja / Archiving Date: 29-02-2016

Univerza
v Ljubljani
Fakulteta za
Gradbeništvo in geodezijo

DOKTORSKI ŠTUDIJSKI PROGRAM III. STOPNJE
GRAJENO OKOLJE

Kandidatka:

DARJA ŠEMROV

Časovno načrtovanje železniškega prometa z uporabo metode spodbujevanega učenja

Doktorska disertacija št.: 32/GO

Railway traffic scheduling with use of reinforcement learning

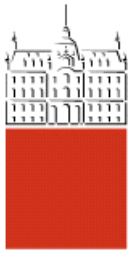
Doctoral thesis No.: 32/GO

Komisija za doktorski študij je na 17. seji, 11. maja 2011, po pooblastilu 30. seje Senata Univerze v Ljubljani z dne 20. januarja 2009, dala soglasje k temi doktorske disertacije.

Za mentorja je bil imenovan doc. dr. Marijan Žura,
za somentorja je bil imenovan izr. prof. dr. Ljupčo Todorovski

Ljubljana, 11. februar 2016

Univerza
v Ljubljani
Fakulteta za
gradbeništvo in
geodezijo



Komisijo za oceno ustreznosti teme doktorske disertacije v sestavi:

- doc. dr. Marijan Žura, UL FGG
- prof. dr. Bogdan Zgonc, UL FGG
- izr. prof. dr. Ljupčo Todorovski, UL FU
- doc. dr. Tomaž Maher, UL FGG

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 18. redni seji, 2. marca 2011.

Poročevalce za oceno doktorske disertacije v sestavi:

- prof. dr. Bogdan Zgonc, UL FGG
- doc. dr. Tomaž Maher, UL FGG
- izr. prof. dr. Drago Sever, UM FGPA

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 23. redni seji, 4. novembra 2015.

Komisijo za zagovor doktorske disertacije v sestavi:

- prof. dr. Matjaž Mikoš, dekan UL FGG, predsednik
- izr. prof. dr. Marijan Žura, UL FGG, mentor
- prof. dr. Ljupčo Todorovski, UL FU, somentor
- prof. dr. Bogdan Zgonc, UL FGG
- doc. dr. Tomaž Maher, UL FGG
- izr. prof. dr. Sever Drago, UM FGPA

je imenoval Senat Fakultete za gradbeništvo in geodezijo na 25. redni seji, 27. januarja 2016.

POPRAVKI

Stran z napako	Vrstica z napako	Namesto	Naj bo

»Ta stran je namenoma prazna.«

IZJAVA O AVTORSTVU

Podpisana **Darja Šemrov** izjavljam, da sem avtorica doktorske disertacije z naslovom: **»Časovno načrtovanje železniškega prometa z uporabo metode spodbujevanega učenja«**.

Izjavljam, da je elektronska različica v vsem enaka tiskani različici.

Izjavljam, da dovoljujem objavo elektronske različice v digitalnem repozitoriju.

Ljubljana, 11. februar 2016

……

(podpis kandidatke)

»Ta stran je namenoma prazna.«

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK: **656.2: 004.451.26:519.8:(043)**

Avtor: **Darja Šemrov, univ. dipl. inž. grad.**

Mentor: **izr. prof. dr. Marijan Žura, univ. dipl. inž. grad.**

Somentor: **prof. dr. Ljupčo Todorovski**

Naslov: **Časovno načrtovanje železniškega prometa z uporabo metode spodbujevanega učenja**

Tip dokumenta **doktorska disertacija**

Obseg in oprema: **110 str., 10 pregl., 51 sl., 10 pril.**

Ključne besede: **vozni red, časovno replaniranje vlakov, učenje Q**

Izvleček

Zanesljivost železniškega prometa najpogosteje povezujemo s točnostjo vlakov, torej primerjamo odstopanje dejanskih prihodov/odhodov vlakov s prihodi/odhodi, objavljenimi v voznem redu. Manjšo zamudo vlaka omilimo ali celo izničimo s časovnimi dodatki v voznem redu, večja zamuda pa povzroči tako imenovane sekundarne zamude ostalih vlakov na omrežju. Odseki prog, na katerih je visoka izkoriščenost kapacitete, so še posebej podvrženi nastanku zamud, saj večje število vlakov pomeni večje število možnih konfliktov in višjo stopnjo interakcije med vlaki, posledično pa je težje omejiti sekundarne zamude. Osebji upravljavca in prevoznika sta zadolženi, da železniški promet poteka varno, nemoteno in v skladu z voznim redom. Pa vendar lahko zaradi nepredvidenih dogodkov nastanejo zamude; v tem primeru je treba vlakom določiti nove čase prihodov in odhodov. Časovno načrtovanje voženj vlakov je kompleksen optimizacijski problem, ki ga dispečerji trenutno rešujejo na osnovi izkušenj, vendar z večanjem števila vlakov kompleksnost problema narašča, zato dispečerji vedno bolj potrebujejo sistem za pomoč pri odločanju, ki bi predlagal optimalno vodenje vlakov glede na zadani cilj, npr. minimalne zamude vseh vlakov. Časovno načrtovanje voženj vlakov sodi v skupino NP-polnih problemov, kjer odpovedo klasične matematično-računalniške metode optimiranja, nakazuje pa se uporabnost pristopov umetne inteligence. V okviru doktorske disertacije smo razvili algoritem časovnega načrtovanja voženj vlakov, ki temelji na metodi spodbujevanega učenja, natančneje učenja Q. Agent, ki se uči iz nagrad in kazni, ki jih pridobi iz okolja, išče optimalno strategijo vodenja vlakov glede na izbrano kriterijsko funkcijo.

BIBLIOGRAPHIC – DOCUMENTALISTIC INFORMATION AND ABSTRACT

UDC: **656.2: 004.451.26:519.8:(043)**

Author: **Darja Šemrov, B. Sc.**

Supervisor: **assoc. prof. Marijan Žura, Ph. D.**

Co-supervisor: **prof. Ljupčo Todorovski, Ph. D.**

Title: **Railway traffic scheduling with use of reinforcement learning**

Document type **PhD Thesis**

Scope and tools: **110 p., 10 tab., 51 figs., 10 ann.**

Keywords: **timetable, train rescheduling, Q learning**

Abstract

The reliability of railway traffic is commonly evaluated with train punctuality, where the deviations of actual train arrivals/departures and train arrivals/departures published in the timetable are compared. Minor train delays can be mitigated or even eliminated with running time supplements, while major delays can lead to so-called secondary delays of other trains on the network. Railway lines with high capacity utilization are more likely subject to delays, since a greater number of trains means a larger number of potential conflicts and more interactions between trains. Consequently, the secondary delays are harder to limit. Railway manager and carrier personnel are responsible for safe, undisturbed and punctual railway traffic. But unforeseen events can lead to delays, which calls for train rescheduling, where new train arrivals and departures are calculated. Train rescheduling is a complex optimization problem, currently solved based on dispatcher's expert knowledge. With the increasing number of trains the complexity of the problem grows, the need for a decision support system increases. Train rescheduling is considered an NP-complete problem, where conventional mathematical and computer optimization methods fail to find the optimal solution, but artificial intelligence approaches have some measure of success. In this dissertation an algorithm for train rescheduling based on reinforcement learning, more precisely Q-learning, was developed. The Q-learning agent learns from rewards and punishments received from the environment, and looks for the optimal train dispatching strategy depending on the objective function.

ZAHVALA

Hvala mentorju izr. prof. dr. Marijanu Žuri in somentorju prof. dr. Ljupču Todorovskemu za vso pomoč, znanje in čas, ki sta ga namenila za širjenje mojega znanja s področja spodbujevanega učenja. Hvala za vse spodbudne besede, pomoč pri pisanju članka, predvsem pa za potrpežljivost. Vem, dolgo je trajalo, vendar je bilo vredno. Presegli smo na začetku zastavljeni cilj.

Jure. Hvala za pomoč, neštete ure, pogovore o temi doktorske disertacije in o življenju nasploh ... Veš, da se strinjam s tabo, zato še toliko bolj cenim tvoj čas, ki si mi ga namenil v času nastajanja doktorske disertacije.

Rok, vesela sem, da sem imela sotrpina, s katerim sva skupaj stopala na za naju neznano področje umetne inteligence. Hvala za vse pogovore, ideje, razmišljanja …

Hvala Mateji in Andreju za skrben in natančen pregled slovenskega in angleškega besedila.

Kar nekaj je še takšnih, ki so prispevali svoj košček k nastajanju doktorske disertacije. Nehvaležno, in hitro se mi lahko zgodi, da koga nehote pozabim, se je zahvaljevati posameznikom, zato gre vsem vam skupna zahvala. Hvala vsem, ki ste me v času pisanja doktorske disertaciji spraševali: »KAKO?«, »ZAKAJ?«, »ČEMU?«. Vaša vprašanja so me spodbudila k razmišljanju in zato je doktorska disertacija boljša. Hvala vsem, ki ste me spraševali: »JE ŽE?«, »KDAJ BO?«, »KAJ ČAKAŠ?«. Ta vprašanja so me vedno spodbudila, da čim prej dokončam.

Ne nazadnje hvala staršema. Hvala za zaupanje – nikoli nista niti za trenutek podvomila v uspeh, hvala, ker sta se vedno veselila dobrih rezultatov in vedno z razumevanjem sprejela dni, ko so bili vlakci in članek prepovedana tema.

»Ta stran je namenoma prazna.«

KAZALO VSEBINE

KAZALO PREGLEDNIC XI

KAZALO SLIK XII

LIST OF TABLES XV

LIST OF FIGURES XVI

SEZNAM PRILOG XIX

OKRAJŠAVE IN SIMBOLI XXI

SLOVAR MANJ ZNANIH BESED IN TUJK XXIII

1 UVOD 1

1.1 Namen in cilj doktorske disertacije 4

1.2 Hipoteze 5

1.3 Vsebina doktorske disertacije 6

2 ČASOVNO NAČRTOVANJE ŽELEZNIŠKEGA PROMETA 7

2.1 Vozni red 10

2.2 Operativno izvajanje in korekcija voznega reda v realnem času 11

2.3 Kompleksnost replaniranja voženj vlakov 15

2.4 Zmanjševanje kompleksnosti časovnega načrtovanja voženj vlakov 21

2.5 Pregled pristopov k časovnemu načrtovanju voženj vlakov 23

3 SISTEM ZA POMOČ PRI ČASOVNEM NAČRTOVANJU ŽELEZNIŠKEGA PROMETA 33

3.1 Spodbujevano učenje 33

3.2 Učenje Q 35

3.3 Uporaba metode učenja Q za časovno načrtovanje voženj vlakov 39

3.3.1 Agent 41

3.3.2 Okolje 41
3.3.3 Stanja okolja 45
3.3.4 Akcije 46
3.3.5 Spodbuda 50
3.3.6 Ocena max$Q(s_{t+1}, a_{t+1})$ v končnem stanju okolja 51
3.3.7 Simulator 51
3.3.8 Učenje Q 55
3.3.9 Učenje Q z zakasnjeno nagrado 58
3.3.10 Učenje Q z zakasnjeno nagrado in sledmi 75
3.3.11 Parametrična študija 85

4 UPORABA ALGORITMA UČENJA Q NA REALNEM PRIMERU ŽELEZNIŠKE INFRASTRUKTURE 90

5 UGOTOVITVE RAZISKOVANJA 99
5.1 Preverjanje postavljenih hipotez 99
5.2 Izvirni znanstveni prispevek 101
5.3 Smernice za nadaljnje raziskovanje 102

6 RAZPRAVA IN ZAKLJUČEK 104
6.1 Razprava o uporabi »ukrojenega« simulacijskega orodja 104
6.2 Razprava o (ne)upoštevanju predznanja 104
6.3 Razprava o kriterijski funkciji 105
6.4 Zaključek 106

7 POVZETEK 109

LITERATURA IN VIRI 113

KAZALO PREGLEDNIC

Preglednica 1: Skupne zamude vlakov za različne pristope replaniranja..... 20

Preglednica 2: Eksperiment a – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ in $\boldsymbol{\varepsilon}$. 65

Preglednica 3: Eksperiment b – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ in $\boldsymbol{\varepsilon}$. 69

Preglednica 4: Eksperiment c – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ in $\boldsymbol{\varepsilon}$. 72

Preglednica 5: Eksperiment a – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ in $\boldsymbol{\varepsilon}$. 79

Preglednica 6: Eksperiment b – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ in $\boldsymbol{\varepsilon}$. 81

Preglednica 7: Eksperiment c – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ in $\boldsymbol{\varepsilon}$. 83

Preglednica 8: 20 scenarijev zamud, kjer so z S3_i označeni scenariji s tremi zamujenimi vlaki in z S5_i označeni scenariji s petimi zamujenimi vlaki..... 91

Preglednica 9: Rezultati, izračunani s strategijo FIFO in učenjem Q za 20 scenarijev zamud, podanih v Preglednici 8 – kriterij uspešnosti so minimalne skupne zamude 93

Preglednica 10: Rezultati, izračunani s strategijo FIFO in učenjem Q za 20 scenarijev zamud, podanih v Preglednici 8 – kriterij uspešnosti so minimalni stroški zamud..... 97

KAZALO SLIK

Slika 1: Princip spodbujevanega učenja ... 3
Slika 2: Definicija medpostajnega odseka ... 9
Slika 3: Definicija prostorskega odseka ... 9
Slika 4: Primer grafikona voznega reda (vir: SŽ) ... 11
Slika 5: Grafikon voznega reda ... 15
Slika 6: Konflikt, ki ga povzroči zamuda Vlaka 1002 ... 15
Slika 7: Konflikta, ki ju povzroči sekundarna zamuda Vlaka 101 ... 16
Slika 8: Replanirani vozni red – primer 1 ... 17
Slika 9: Replanirani vozni red – primer 2 ... 18
Slika 10: Replanirani vozni red – primer 3 ... 18
Slika 11: Replanirani vozni red – primer 4 ... 19
Slika 12: Replanirani vozni red – primer 5 ... 19
Slika 13: Primer krivulje učenja, kjer se znanje agenta poslabšuje ... 39
Slika 14: Poenostavitev modela na območju postaje ... 41
Slika 15: Brezizhodna situacija – primer 1 ... 43
Slika 16: Brezizhodna situacija – primer 2 ... 43
Slika 17: Primer voznega reda dveh zaporednih vlakov ... 47
Slika 18: Del drevesa možnih stanj okolja in akcij ... 48
Slika 19: Trajanje simulacije s programskim orodjem Vissim ... 52
Slika 20: Princip učenja Q ... 56
Slika 21: Učenje Q – definicija nagrade ... 57
Slika 22: Princip učenja Q z zakasnjeno nagrado ... 59
Slika 23: Konvergenca vrednosti $\mathbf{Q}(\mathbf{st}, \mathbf{at})$ pri $\boldsymbol{\alpha} = \mathbf{0,1}$, $\boldsymbol{\gamma} = \mathbf{0,9}$, $\boldsymbol{rT} = \mathbf{15}$ ... 60
Slika 24: Eksperiment a – železniška infrastruktura ... 61
Slika 25: Eksperiment a – vozni red ... 62
Slika 26: Eksperiment a – krivulje učenja za različna semena (α = 0,3; $\boldsymbol{\gamma}$ = 0,3; ε = 0,3) ... 64

Slika 27: Eksperiment a – replanirani vozni red (α = 0,3; γ = 0,3; ε = 0,3, Scenarij 2)..... 66

Slika 28: Eksperiment 3a – replanirani vozni red (α = 0,3; γ = 0,7; ε = 0,3, Scenarij 2)..... 66

Slika 29: Eksperiment b – vozni red 67

Slika 30: Eksperiment b – replanirani vozni red (α = 0,1; γ = 0,3; ε = 0,1, Scenarij 1)..... 70

Slika 31: Eksperiment b – replanirani vozni red (α = 0,7; γ = 0,9; ε = 0,1, Scenarij 1)..... 70

Slika 32: Eksperiment c – železniška infrastruktura 71

Slika 33: Eksperiment c – vozni red 71

Slika 34: Eksperiment c – replanirani vozni red (α = 0,3; γ = 0,3; ε = 0,3, Scenarij 1)..... 73

Slika 35: Eksperiment c – replanirani vozni red (α = 0,7; γ = 0,3; ε = 0,3, Scenarij 1)..... 74

Slika 36: Princip učenja Q z zakasnjeno nagrado in sledmi..... 77

Slika 37: Konvergenca vrednosti $\mathbf{Q}(\mathbf{st}, \mathbf{at})$ pri $\boldsymbol{\alpha} = \mathbf{0,1}$, $\boldsymbol{\gamma} = \mathbf{0,9}$, $\boldsymbol{rT} = \mathbf{15}$ 78

Slika 38: Eksperiment a – krivulje učenja za različna semena (α = 0,3; $\boldsymbol{\gamma}$ = 0,3; ε = 0,3) 79

Slika 39: Eksperiment a – replanirani vozni red (α = 0,3; γ = 0,7; ε = 0,3, Scenarij 2)..... 80

Slika 40: Eksperiment b – replanirani vozni red (α = 0,7; γ = 0,9; ε = 0,1, Scenarij 1)..... 82

Slika 41: Eksperiment c – replanirani vozni red (α = 0,7; γ = 0,3; ε = 0,3, Scenarij 1)..... 84

Slika 42: Različne $\boldsymbol{\varepsilon}$-požrešne funkcije, upoštevane v raziskavi 86

Slika 43: Skupne zamude vlakov po $\boldsymbol{\alpha}$ 87

Slika 44: Skupne zamude vlakov po $\boldsymbol{\gamma}$ pri pogoju $\boldsymbol{\alpha} = \mathbf{0,9}$ 87

Slika 45: Krivulja učenja in krivulja standardne deviacije pri $\boldsymbol{\alpha} = \mathbf{0,9}$, $\boldsymbol{\gamma} = \mathbf{0,1}$ ter a) $\boldsymbol{\varepsilon 1} = \boldsymbol{x} - \mathbf{1}$, b) $\boldsymbol{\varepsilon 2} = \mathbf{0,51} + \mathbf{e}(\mathbf{10} * (\mathbf{x} - \mathbf{0.4} * \mathbf{25})/\mathbf{25}))$, c) $\boldsymbol{\varepsilon 3} = \mathbf{0,51} + \mathbf{e}(\mathbf{10} * (\mathbf{x} - \mathbf{0.4} * \mathbf{35})/\mathbf{35}))$, d) $\boldsymbol{\varepsilon 4} = \mathbf{0,9}$, e) $\boldsymbol{\varepsilon 5} = \mathbf{0,7}$, f) $\boldsymbol{\varepsilon 6} = \mathbf{0,5}$, g) $\boldsymbol{\varepsilon 7} = \mathbf{0,3}$, h) $\boldsymbol{\varepsilon 8} = \mathbf{0,1}$ 88

Slika 46: Predlagana $\boldsymbol{\varepsilon}$-požrešna funkcija za različno število ponovitev učenj 89

Slika 47: Shema enotirne proge Ljubljana–Jesenice 90

Slika 48: Začetni vozni red za progo Ljubljana–Jesenice 90

Slika 49: Začetni in s strategijo FIFO replanirani vozni red..... 94

Slika 50: Začetni in z algoritmom učenja Q replanirani vozni red 95

Slika 51: S strategijo FIFO in z algoritmom učenja Q replanirani vozni red 95

»Ta stran je namenoma prazna.«

LIST OF TABLES

Table 1: Total train delays obtained with different rescheduling aproaches 20

Table 2: Experiment a – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account. 65

Table 3: Experiment b – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account. 69

Table 4: Experimnet c – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account. 72

Table 5: Experiment a – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account. 79

Table 6: Experiment b – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account. 81

Table 7: Experiment c – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account. 83

Table 8: The twenty delay scenarios used in the experiments: the S3_i are scenarios with 3 delayed trains and S5_i are scenarios with 5 delayed trains. 91

Table 9: Results obtained with the FIFO strategy and with the Q-learning algorithm on the 20 delay scenarios from Table 8 – objective of minimizing the total delays 93

Table 10: Results obtained with the FIFO strategy and with the Q-learning algorithm on the 20 delay scenarios from Table 8 – objective of minimizing the delay costs 97

LIST OF FIGURES

Figure 1: Reinforcement learning setting 3
Figure 2: Definition of open section 9
Figure 3: Definition of block section 9
Figure 4: Example of train timetable (SŽ) 11
Figure 5: Train timetable 15
Figure 6: Conflict caused by delayed train 1002 15
Figure 7: Conflicts caused by delayed train 101 16
Figure 8: Rescheduled timetable – example 1 17
Figure 9: Rescheduled timetable – example 2 18
Figure 10: Rescheduled timetable – example 3 18
Figure 11: Rescheduled timetable – example 4 19
Figure 12: Rescheduled timetable – example 5 19
Figure 13: Example of learning curve where agent knowledge worsens 39
Figure 14: Simplification of railway station layout 41
Figure 15: Dead lock – example 1 43
Figure 16: Dead lock – example 2 43
Figure 17: Example of timetable for two successive train 47
Figure 18: Part of possible environment states and actions 48
Figure 19: Computation time obtained with VISSIM software for different number of runs 52
Figure 20: How Q learning works 56
Figure 21: Q learning – definition of reward 57
Figure 22: How Q learning with delayed reward works 59
Figure 23: Convergence of $\mathbf{Q}(\mathbf{st}, \mathbf{at})$ values with $\boldsymbol{\alpha} = \mathbf{0.1}$, $\boldsymbol{\gamma} = \mathbf{0.9}$, $\boldsymbol{rT} = \mathbf{15}$ 60
Figure 24: Experiment a – railway layout 61
Figure 25: Experiment a – Timetable 62
Figure 26: Experiment a – Learning curves for different seeds (α = 0.3; $\boldsymbol{\gamma}$ = 0.3; ε = 0.3) 64

Figure 27: Experiment a – Rescheduled timetable (α = 0.3; γ = 0.3; ε = 0.3, Scenario 2)..... 66

Figure 28: Experiment 3a – Rescheduled timetable (α = 0.3; γ = 0.7; ε = 0.3, Scenario 2)... 66

Figure 29: Experiment b – Timetable 67

Figure 30: Experiment b – Rescheduled timetable (α = 0.1; γ = 0.3; ε = 0.1, Scenario 1)..... 70

Figure 31: Experiment b – Rescheduled timetable (α = 0.7; γ = 0.9; ε = 0.1, Scenario 1)..... 70

Figure 32: Experiment c – railway layout..... 71

Figure 33: Experiment c – Timetable 71

Figure 34: Experiment c – Rescheduled timetable (α = 0.3; γ = 0.3; ε = 0.3, Scenario 1)..... 73

Figure 35: Experiment c – Rescheduled timetable (α = 0.7; γ = 0.3; ε = 0.3, Scenario 1)..... 74

Figure 36: How Q learning with delayed reward and eligibility traces works..... 77

Figure 37: Convergence of $\mathbf{Q}(\mathbf{st}, \mathbf{at})$ values with $\boldsymbol{\alpha} = \mathbf{0.1}$, $\boldsymbol{\gamma} = \mathbf{0.9}$, $\boldsymbol{rT} = \mathbf{15}$..... 78

Figure 38: Experiment a – Learning curves for different seeds (α = 0.3; $\boldsymbol{\gamma}$ = 0.3; ε = 0.3)..... 79

Figure 39: Experiment a – Rescheduled timtable (α = 0.3; γ = 0.7; ε = 0.3, Scenario 2)..... 80

Figure 40: Experiment b – Rescheduled timtable (α = 0.7; γ = 0.9; ε = 0.1, Scenario 1)..... 82

Figure 41: Experiment c – Rescheduled timtable (α = 0.7; γ = 0.3; ε = 0.3, Scenario 1)..... 84

Figure 42: Different $\boldsymbol{\varepsilon}$–greedy functions studied in the research..... 86

Figure 43: Total train delay by $\boldsymbol{\alpha}$ 87

Figure 44: Total train delay by $\boldsymbol{\gamma}$, where $\boldsymbol{\alpha} = \mathbf{0.9}$..... 87

Figure 45: Learning and standard deviation curves for $\boldsymbol{\alpha} = \mathbf{0.9}$, $\boldsymbol{\gamma} = \mathbf{0.1}$, $\boldsymbol{\varepsilon}1$, and a) $\boldsymbol{\varepsilon 1} = \boldsymbol{x} - \mathbf{1}$, b) $\boldsymbol{\varepsilon 2} = \mathbf{0,51} + \mathbf{e}(\mathbf{10} * (\mathbf{x} - \mathbf{0.4} * \mathbf{25})/\mathbf{25}))$, c) $\boldsymbol{\varepsilon 3} = \mathbf{0,51} + \mathbf{e}(\mathbf{10} * (\mathbf{x} - \mathbf{0.4} * \mathbf{35})/\mathbf{35}))$, d) $\boldsymbol{\varepsilon 4} = \mathbf{0.9}$, e) $\boldsymbol{\varepsilon 5} = \mathbf{0.7}$, f) $\boldsymbol{\varepsilon 6} = \mathbf{0.5}$, g) $\boldsymbol{\varepsilon 7} = \mathbf{0.3}$, h) $\boldsymbol{\varepsilon 8} = \mathbf{0.1}$ 88

Figure 46: Proposed $\boldsymbol{\varepsilon}$-greedy function for different number of iterations 89

Figure 47: The layout of the single-track railway between Ljubljana and Jesenice 90

Figure 48: Initial timetable for railway line between Ljubljana and Jesenice 90

Figure 49: Initial and with FIFO strategy rescheduled timetable..... 94

Figure 50: Initial and with Q learning algorithm rescheduled timetable..... 95

Figure 51: With FIFO strategy and with Q learning algorithm rescheduled timetable 95

»Ta stran je namenoma prazna.«

SEZNAM PRILOG

Priloga A: Učenje Q z zakasnjeno nagrado, Eksperiment a – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Priloga B: Učenje Q z zakasnjeno nagrado, Eksperiment b – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Priloga C: Učenje Q z zakasnjeno nagrado, Eksperiment c – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Priloga D: Učenje Q z zakasnjeno nagrado in sledmi, Eksperiment a – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Priloga E: Učenje Q z zakasnjeno nagrado in sledmi, Eksperiment b – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Priloga F: Učenje Q z zakasnjeno nagrado in sledmi, Eksperiment c – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Priloga G: Parametrična študija – Krivulje učenja za Scenarij zamud 1

Priloga H: Parametrična študija – Krivulje učenja za Scenarij zamud 2

Priloga I: Parametrična študija – Krivulje učenja za Scenarij zamud 3

Priloga J: Parametrična študija – Krivulje učenja za Scenarij zamud 4

»Ta stran je namenoma prazna.«

OKRAJŠAVE IN SIMBOLI

a_{acc}	pospešek vlaka
a_{dec}	pojemek vlaka
a_t	akcija agenta v času t
c	strošek zamud
c_f	strošek minute zamude tovornega vlaka
c_p	strošek minute zamude potniškega vlaka
$d_{i,t}$	(trenutna) zamuda vlaka i v času t
$d_{i,T}$	zamuda vlaka i v končnem stanju T
$d_{f,T}$	zamuda tovornega vlaka v končnem stanju T
$d_{p,T}$	zamuda potniškega vlaka v končnem stanju T
d_{skupna}	skupna zamuda vseh vlakov
L	lokacija vlaka
N	število vlakov
P	prosta pot
r_{t+1}	nagrada, ki jo agent prejme, ko izvede akcijo a_t v stanju s_t
s_t	stanje okolja v času t
$t_{oj,i,VR}$	čas odhoda vlaka j na postajo i po voznem redu
$t_{oj,i,VR,R}$	čas odhoda vlaka j na postajo i po replaniranem voznem redu
v_{max}	maksimalna hitrost vlaka
$\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$	max. vrednost Q v stanju s_{t+1} po izbiri akcije a_t v stanju s_t
$Q(s_t, a_t)$	vrednost Q v stanju s_t
α	stopnja učenja
γ	faktor diskontiranja nagrade
ε	razmerje med raziskovanjem in izkoriščanjem

»Ta stran je namenoma prazna.«

SLOVAR MANJ ZNANIH BESED IN TUJK

dispečer (ang. *dispatcher*) – v Pravilniku o delovnih mestih izvršilnih železniških delavcev (Uradni list RS, št. 126/2007: 18754–18761) so navedena tri delovna mesta vodenja prometa, in sicer: prometnik, ki je odgovoren za vodenje vlakov na postaji ali daljinsko vodeni postaji, progovni prometnik, ki je odgovoren za vodenje vlakov na progovnem odseku TKo proge ali na TKo progi, ter vlakovni dispečer, ki je odgovoren za urejanje vlakovnega prometa na določenem odseku proge ali na določeni progi. V doktorski disertaciji bomo za osebje upravljavca, ki je zadolženo za vodenje vlakov, uporabili izraz dispečer, neodvisno od delovnega mesta.

nov vozni red – v Uredbi o izdelavi voznega reda omrežja javne železniške infrastrukture (Uradni list RS, št. 73/2012: 7372–7380) besedna zveza »nov vozni red« označuje vozni red, ki je pripravljen in usklajen s prosilci ter bo uveljavljen v naslednjem voznorednem obdobju. V doktorski disertaciji besedno zvezo nov vozni red uporabljamo za operativen načrt prihodov in odhodov vlakov, ki ga pripravi dispečer po tem, ko se pojavi zamuda in vlaki ne vozijo več po voznem redu omrežja. Torej je v doktorski disertaciji nov vozni red rezultat časovnega replaniranja voženj vlakov.

brezizhodno stanje (ang. *deadlock*) – na enotirni progi, kjer je tir namenjen vožnji v obe smeri, lahko pride do situacije, ko vlaka, ki vozita v različnih smereh, ne moreta nadaljevati vožnje, saj se njuni vozni poti prekrivata. V primeru, da bi nadaljevala vožnjo, bi prišlo do čelnega trka.

sestajanje vlakov – skupni naziv za križanja, prehitenja, srečanja in dohitenja vlakov (Prometni pravilnik, Uradni list RS, št. 50/2011: 6824–6931), kjer je **križanje** sestajanje dveh vlakov iz nasprotnih smeri na postajah enotirnih prog; **srečanje** sestajanje dveh vlakov iz nasprotnih smeri na postajah ali na odprti progi dvotirne proge, ko vozita vsak po svojem tiru; **prehitenje** sestajanje dveh ali več vlakov iz iste smeri na postajah ali na odprti progi dvotirne proge v primeru, ko zaporedni vlak nadaljuje vožnjo pred sprednjim vlakom in se zamenja njun vrstni red; **dohitenje** sestajanje dveh ali več vlakov iste smeri na postajah enotirne ali dvotirne proge, s katere najprej odpelje sprednji vlak pred zaporednim vlakom; vrstni red vožnje ostane nespremenjen.

»Ta stran je namenoma prazna.«

1 UVOD

Hiter razvoj gospodarstva, vedno večje povpraševanje po kvalitetnih in cenovno ugodnih izdelkih in storitvah ter vedno večja želja po mobilnosti vplivajo na rast in pomen prometa. In obratno, prometni sistem vpliva na razvoj gospodarstva in družbe, zato je cilj vsake družbe učinkovito prometno omrežje. V zadnjih desetletjih se je največja pozornost namenjala razvoju cestnega prometa, v zadnjem času pa se zaradi doseženih kapacitet cestnega omrežja in (pre)velikega vpliva cestnih transportnih sredstev na okolje vedno bolj razmišlja o preusmeritvi prometnih tokov s cest na železnice.

Pri izbiri prometnega sredstva sta uporabnikom najpomembnejša ekonomičnost in kvaliteta prometnega sredstva. Ekonomičnost izrazimo s ceno prevozne storitve, kvaliteta oz. učinkovitost prometnega sredstva pa se najpogosteje nanaša na pogostost storitev, potovalne čase in zanesljivost (König, 2002; Tornquist, 2006). Zanesljivost železniškega prometa, za katero sicer obstajajo številne definicije, se najpogosteje vrednoti na osnovi točnosti vlakov oz. velikosti in pogostosti zamud (Landex, 2008). Najbolj učinkoviti ukrep za zagotavljanje kvalitetnih storitev v železniškem prometu je izgradnja dodatne infrastrukture. Ta ukrep se zaradi ekonomskih in prostorskih razlogov pogosto izkaže za neizvedljivega, zato upravljavci železniške infrastrukture iščejo rešitve za zagotavljanje večje izkoriščenosti in hkrati zadostno zanesljivost storitev v okviru obstoječe infrastrukture. Luethi in sodelavci (2007; 2009) so raziskovali vpliv spremljanja vlakov v realnem času in dokazali, da se z uvedbo evropskega sistema za nadzor in upravljanje vlakov, nivo 2 (ang. *European Train Control System, Level 2*), kvaliteta storitev izboljša, vendar je tudi ta ukrep drag, njegova implementacija pa počasna. Cenejši in takoj izvedljiv ukrep je kvalitetno načrtovanje voznega reda ter učinkovito izvajanje in vodenje prometa z ustreznim časovnim replaniranjem voženj vlakov ob nastanku zamud.

Odstopanja od voznega reda oz. zamude v železniškem prometu delimo na (D'Ariano, 2008):

- zamude, na katere lahko vplivajo strojevodje (pospeševanje, zaviranje, hitrost vožnje) in potniki (trajanje postanka zaradi izstopanja in vstopanja potnikov);
- zamude, ki lahko nastanejo zaradi odstopanj na infrastrukturi (pokvarjeni signali, motnje v prometu, izredni dogodki, vremenske razmere …).

Po vzroku nastanka zamude delimo zamude na (Kecman, 2012):

- primarne zamude, ki nastanejo zaradi zunanjega dogodka;
- sekundarne zamude, ki jih povzročijo primarne zamude in so posledica medsebojnega vpliva med vlaki (domino efekta).

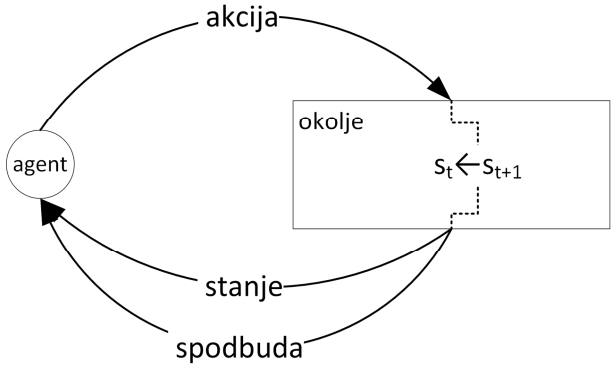
Primarne zamude se pojavijo naključno in jih je težko predvideti in omejiti. Na zmanjšanje verjetnosti nastanka sekundarnih zamud pa lahko vplivamo z zmanjšanjem števila vlakov. Večja, kot je namreč gostota vlakov, večji in dolgotrajnejši je tudi vpliv na ostale vlake (Huisman in Boucherie, 2001; Luethi, Laube in Medeossi, 2007; Wegele in Schnieder, 2004). Abril in sodelavci (2007) ugotavljajo, da sekundarne zamude vlakov naraščajo eksponentno s povečevanjem izkoriščenosti kapacitete.

Ne glede na vrsto in vzrok nastanka zamud le-te zahtevajo spremembe v časih odhodov vlakov s postaj, spremembe dolžin postankov na postajah, spremembo v vrstnem redu vlakov ali celo izbiro obvozne poti, da se lahko zagotovi varnostne zahteve odvijanja železniškega prometa in prepreči trke vlakov. Korekcije voznega reda kot odgovor na zamude z namenom, da bi bili stroški zamud vseh vlakov minimalni oz. da bi bile posledice zamud čim manjše za vse vlake na omrežju (Tornquist, 2006), imenujemo časovno replaniranje voženj vlakov. Glavno vlogo pri izvajanju, vodenju in urejanju vlakovnega prometa imajo dispečerji, ki morajo za učinkovito zajezitev domino efekta širjenja zamud čim hitreje zaznati zamudo in najti odgovor na nastalo situacijo (Tornquist, 2006). Dispečerji morajo pri izbiri korekcijskih akcij upoštevati množico vhodnih podatkov, kot so trenutne lokacije vlakov, smeri potovanj vlakov, hitrosti vlakov, dolžine postankov, rangi vlakov, kapacitete infrastrukture, in hkrati zagotavljati visoko stopnjo varnosti pri vodenju in urejanju vlakovnega prometa, na voljo pa imajo le nekaj minut. Dispečerji zato korekcijske akcije izbirajo na osnovi izkušenj (D'Ariano et al., 2008; Hara et al., 2006). S povečevanjem števila vlakov narašča kompleksnost problema in vedno bolj se kaže potreba po učinkovitem sistemu za pomoč pri odločanju.

V teoriji kompleksnosti razdelimo probleme glede na zahtevnost reševanja v dva razreda, in sicer v razred problemov, rešljivih v polinomskem času (P-problem), in v razred, v katerem problem ni nujno rešljiv v polinomskem času (NP-problem, ang. *nondeterministic polynomial time*). V drugi skupini so problemi, katerih izračun optimalne rešitve je časovno zelo zahteven in običajno iskanje rešitve traja nesprejemljivo dolgo. Avtorji problem replaniranja voženj vlakov obravnavajo podobno kot splošne probleme razvrščanja, za katere je znano, da so NP-polni problemi. Replaniranje voženj vlakov je specifičen problem časovnega razvrščanja, za katerega ne obstaja dokaz, v kateri razred NP-problemov ga uvrščamo. Tako avtorji problem uvrščajo v razred NP-težkih problemov (Gély et al., 2006; Ping et al., 2001;

Sajedinejad et al., 2011; Strotmann, 2007; Tazoniero, Gonçalves in Gomide, 2007), drugi pa so mnenja, da je problem bolj enostaven, in ga uvrščajo v razred NP-polnih problemov (D'Ariano, 2008; Ge, 2009; Kroon, Romeijn in Zwaneveld, 1997; Wegele in Schnieder, 2004), vsi pa so si enotni, da je načrtovanje voženj vlakov kompleksen in težko rešljiv problem, saj prostor možnih rešitev narašča eksponentno z velikostjo problema. Seveda se ob trditvi, da replaniranje voženj vlakov sodi v skupino NP-problemov, postavi vprašanje, ali obstaja način reševanja problema, pri katerem dobimo dobro rešitev v času, ki je primeren za replaniranje v realnem času. Možnost se ponuja z uporabo hevrističnih algoritmov, ki najdejo rešitev v krajšem času, vendar ni nujno, da je rešitev optimalna, saj hevristične metode ne preiščejo celotnega prostora rešitev.

V doktorski disertaciji za reševanje problema replaniranja voženj vlakov predlagamo uporabo algoritmov umetne inteligence. Zaradi analogije učenja agenta z učenjem dispečerja smo se izmed množice algoritmov umetne inteligence odločili za spodbujevano učenje. Osnovni princip spodbujevanega učenja je učenje agenta, ki z raziskovanjem okolja išče optimalno strategijo. Agent opazuje trenutno stanje okolja in izbira ter izvaja akcije. Po izvedeni akciji se stanje okolja spremeni in okolje agentu vrne spodbudo (nagrado). Velikost nagrade odraža uspešnost izvedene akcije; akcija, ki izboljša stanje okolja, vrne visoko nagrado, druge pa nizko ali pa negativno nagrado. Cilj agenta je najti strategijo, ki maksimizira vsoto spodbud, ki jih prejme iz okolja (Russell in Norvig, 2003). V spodbujevanem učenju sta okolje in agent ločena, kar omogoča uporabo algoritma učenja za različne modele okolja. Okolje je pasivno, saj se o akcijah odloča agent. Okolje mora biti definirano tako, da se stanje okolja po izvedbi akcije spremeni. V primeru uporabe spodbujevanega učenja za replaniranje voženj vlakov predlagamo, da je agent dispečer, ki je zadolžen za vodenje vlakov, okolje železniška infrastruktura skupaj z vlaki in voznim redom, nagrada pa je signal, ki agentu sporoči, v kolikšni meri replanirani vozni red odstopa od začetnega voznega reda. Princip spodbujevanega učenja je podan na prikazu Slika 1.



Slika 1: Princip spodbujevanega učenja
Figure 1: Reinforcement learning setting

Spodbujevano učenje je bilo uspešno uporabljeno že pri reševanju različnih problemov, tudi pri upravljanju in vodenju cestnega in železniškega prometa. Abdulhai in Kattan (2003) sta nakazala potencialno uporabo spodbujevanega učenja v transportu, istega leta so Abdulhai in sodelavci (2003) dokazali, da z uporabo učenja Q zagotovijo boljši nivo uslug v cestnem prometu. Avtorji so poročali o uspešni aplikaciji učenja Q za krmiljenje posameznih križišč (Gregoire et al., 2007; Shoufeng et al., 2008), kasneje tudi za krmiljenje večjega števila križišč (Arel et al., 2010; Marsetič et al., 2014; Prashanth et al., 2011). Na področju železniškega prometa je bilo učenja Q uspešno uporabljeno na treh področjih, in sicer so Zou in sodelavci (2006)metodo učenja Q predlagali za reševanje problema upravljanja voženj vlakov lahke železnice, Wong in sodelavci (2008; 2010) za pogajanja med upravljavci in prevozniki o ceni uporabnine, Hirashima (2011) pa za učinkovito razporejanje vagonov (s čim manj premiki) na ranžirnih postajah; metode Q niso predlagali za optimizacijo vodenja vlakov po nastanku zamude, kar ponuja možnost širjenja področja uporabe učenja Q in s tem doprinos k znanosti.

1.1 Namen in cilj doktorske disertacije

Dispečerji so pri izvajanju in urejanju železniškega prometa postavljeni pred vedno večje izzive, saj se z naraščanjem števila vlakov in izkoriščenosti železniške infrastrukture povečuje verjetnost nastanka zamud, večji vpliv pa ima primarna zamuda tudi na ostale vlake. Povečuje se tudi število omejitev, ki jih je treba upoštevati, zato še dodatno narašča zahtevnost iskanja optimalne rešitve in vedno bolj se kaže potreba po računalniško podprti pomoči pri replaniranju voženj vlakov. V doktorski disertaciji je prikazana kompleksnost problema časovnega načrtovanja voženj vlakov, ki je po eni strani povezana z analiziranjem potencialnih kombinacij akcij, ki vodijo k optimalni rešitvi, po drugi strani pa z zahtevo, da je treba na zamudo odgovoriti kar se da hitro. Namen doktorske disertacije je predstaviti vse omejitve in pogoje, ki jih je treba upoštevati tako pri konstruiranju kot pri replaniranju voznega reda, ter izdelati algoritem za pomoč pri odločanju, ki bi v realnem času predlagal (skoraj) optimalno rešitev vodenja vlakov po nastanku zamude.

Zahtevnost problema in zahteva po rešitvi v izredno kratkem času onemogočata uporabo eksaktnih deterministčnih algoritmov, zato je bil cilj razviti algoritem, ki bo omogočal prilagajanje voznega reda spreminjajočim se razmeram v železniškem prometu v realnem času ob zagotavljanju visoke kvalitete storitev in stabilnosti voznega reda. Ob poznanem voznem redu in zamudi želimo z algoritmom poiskati takšno korekcijo voznega reda, da bo strošek zamud vseh vlakov čim manjši.

1.2 Hipoteze

Predmet doktorske disertacije je preveritev uporabnosti metode spodbujevanega učenja za časovno replaniranje voženj vlakov v realnem času z upoštevanjem vseh omejitev, ki veljajo za proces izvajanja in vodenja železniškega prometa, ter vseh varnostnih zahtev, ki morajo biti ob tem izpolnjene. Ob tem se postavijo naslednje hipoteze, ki jih bomo skozi disertacijo potrdili (ali ovrgli):

1. Spodbujevano učenje je primerno za časovno replaniranje voženj vlakov

Učni proces v spodbujevanem učenju, torej učenje agenta, ki sprejema odločitve, je podoben procesu učenja dispečerja. Dispečer sprejema odločitve na osnovi izkušenj; prav tako tudi agent nadgrajuje znanje z izkušnjami, ki jih pridobiva z raziskovanjem okolja in interpretacijo informacij, ki mu jih sporoča okolje. Na osnovi prepoznavanja problemov, upoštevanja omejitev in zahtev pri vodenju železniškega prometa ter samoučenja agenta bi se metoda lahko uporabljala za replaniranje voženj vlakov v realnem času.

1.1 Metoda spodbujevanega učenja je uporabna za upravljanje zaporednih voženj in voženj v različni smeri po istem tiru

Na dvotirni progi je vsak tir namenjen vožnji v eno smer in obstaja nevarnost naleta (hitrejši vlak dohiti počasnejšega), na enotirni progi pa je poleg nevarnosti naleta tudi nevarnost čelnega trka (trčenja vlakov iz nasprotnih smeri) ali nastanka brezizhodne situacije, kjer si vlaka iz nasprotnih smeri onemogočata nadaljevanje vožnje. Metoda spodbujevanega učenja omogoča, da agent z omejitvami ali spodbudami prepozna možnost nastanka naleta, trčenja in brezizhodne situacije ter predlaga brezkonflikten vozni red.

1.2 Metoda spodbujevanega učenja je primerna za različne vidike uspešnosti replaniranja

Uspešnost replaniranja ocenimo s kvaliteto rešitve, kar pomeni, koliko se približamo zastavljenemu cilju. Uporaba metode spodbujevanega učenja omogoča enostavno definiranje različnih ciljev ali kombinacijo le-teh. S poskusi bomo poskušali dokazati, da se agent nauči različnih strategij, če mu definiramo različne cilje (npr. minimizacijo skupnih zamud ter minimizacijo stroškov zamud, pri čemer upoštevamo različne prioritete vlakov).

2. Izbira vrednosti parametrov algoritma učenja Q vpliva na uspešnost replaniranja

Na hitrost konvergiranja k optimalni rešitvi ter kvaliteto rešitve vplivajo naslednji parametri: velikost nagrade, parameter učenja α, parameter diskontiranja prihodnje nagrade β ter razmerje med raziskovanjem in izkoriščanjem znanja ε. Vrednosti parametrov oz. njihova kombinacija, ki vodi k najboljši rešitvi, se določijo za vsak primer uporabe učenja Q posebej.

3. Način nagrajevanja

Pri spodbujevanem učenju agent običajno prejme nagrado iz okolja po vsaki izvedeni akciji, vendar je možna tudi implementacija algoritma, pri kateri agent prejme nagrado v končnem stanju, t. i. učenje Q z zakasnjeno nagrado. Učenje Q z zakasnjeno nagrado je uporabno pri reševanju problemov, kjer iz vmesnih stanj ne moremo določiti, kako blizu cilja smo. Pri replaniranju voženj vlakov je uspešnost algoritma, če upoštevamo kriterij skupnih zamud na končnih postajah, znana šele v končnem stanju. S poskusi bomo preverili uspešnost sprotnega nagrajevanja in nagrajevanja v končnem stanju.

1.3 Vsebina doktorske disertacije

V prvem poglavju sta opredeljena problem in predmet raziskovanja, predstavljene so teze, razložena namen in cilj, podana ocena dosedanjih raziskav ter opisana struktura doktorske disertacije.

V drugem poglavju so podane osnovne informacije o vodenju železniškega prometa, teoretične osnove časovnega načrtovanja voženj vlakov, omejitve pri vodenju vlakov zaradi infrastrukture, omejitve z vidika zagotavljanja varnosti železniškega prometa ter omejitve, ki izhajajo iz nacionalnih predpisov in zakonov, ter pregled obstoječih raziskav na obravnavanem področju. Opravljena je analiza pristopov k časovnemu načrtovanju voženj vlakov na železniškem omrežju, podane so glavne značilnosti ter prednosti in slabosti posameznih pristopov.

V tretjem poglavju so podrobneje predstavljeni osnovni principi umetne inteligence in spodbujevanega učenja, podrobneje učenja Q, ter uporaba te metode za časovno načrtovanje voženj vlakov. Različne implementacije spodbujevanega učenja ter parametrična študija so izdelane na enostavnem primeru železniškega omrežja.

V četrtem poglavju je uspešnost predlaganega algoritma učenja Q prikazana na realnem železniškem omrežju – na progi Ljubljana–Jesenice. Uspešnost algoritma je ovrednotena s primerjavo rezultata replaniranja po pravilu »*First-In-First-Out*«, ki je pogosto uporabljeno pravilo pri vodenju prometa v realnih situacijah.

V petem in šestem poglavju so predstavljeni rezultati raziskovanja, potrjene (ali ovržene) postavljene hipoteze in podane usmeritve za nadaljnje raziskovanje.

2 ČASOVNO NAČRTOVANJE ŽELEZNIŠKEGA PROMETA

Zaradi lažjega in boljšega razumevanja postopka časovnega načrtovanja voženj vlakov ter razumevanja vzrokov in posledic omejitev in pogojev, ki so potrebni za varno odvijanje železniškega prometa, so v tem poglavju na kratko opisani procesi vodenja vlakov in razloženi nekateri osnovni pojmi.

Bistveni lastnosti, ki razlikujeta cestni in železniški promet, sta utirjenost železniškega vozila in dolga zavorna razdalja vlaka. Vozna pot utirjenega vozila je natančno določena; vozilo ne more obiti ovire ali preprečiti trka, kot je to mogoče v cestnem prometu. Poleg tega je koeficient trenja železniškega vozila veliko manjši od cestnega vozila, kar pomeni, da je zavorna sila veliko manjša. Zaradi manjše zavorne sile in velike teže vlaka je zavorna razdalja vlaka veliko daljša od zavorne razdalje cestnih vozil. Skladno s Signalnim pravilnikom (Uradni list RS, št. 123/2007: 18085–18185) upoštevamo, da je zavorna razdalja na regionalnih progah 700 m, na glavnih progah pa 1000 m. Zaradi dolge zavorne razdalje vlaka in topologije terena strojevodja ne more voziti po načelu preglednosti, kot je to mogoče v cestnem prometu, ampak vozi v skladu z navodili, ki mu jih preko signalov sporoča osebje upravljavca železniške infrastrukture, zadolženo za vodenje prometa.

Prvi signali so bili likovni signali, ki so signalne znake dajali z lego in barvo signala, kasneje so razvili svetlobne signale, ki dajejo signal z mirnimi in/ali utripajočimi lučmi. Ne glede na izvedbo signala pa je osnovna naloga ostala nespremenjena, to je enostavno, nedvoumno in zanesljivo obveščanje strojevodje. Strojevodja vozi po navodilih, ki mu jih preko signalov sporoča prometno osebje, saj le osebje, zadolženo za vodenje vlakov, pozna lokacije vseh vlakov in podatke o razpoložljivi infrastrukturi na območju, ki ga upravlja. Zato lahko samo prometno osebje dovoli oz. prepove vožnjo vlaka ter svoje odločitve prek signalnih naprav sporoča strojevodji. Strojevodja se za uvoz na postajo ali v naslednji prostorski odsek ravna po glavnih signalih. V grobem lahko signalne znake delimo na takšne, ki prepovedujejo vožnjo, in takšne, ki jo dovoljujejo. Če signal kaže signalni znak »stoj« ali je nerazsvetljen, potem strojevodja ne sme nadaljevati z vožnjo; le-to lahko nadaljuje, ko s signalnim znakom glavnega signala dobi dovoljenje za vožnjo. Z različnimi signalnimi znaki prometno osebje sporoča različne pogoje za nadaljevanje vožnje, npr. uvoz na postajo z omejeno hitrostjo. Natančnejši opis pomenov signalnih znakov in njihov pomen pri vodenju vlakov je podan pri Theeg ter Vlasenko (2009) ter v Signalnem pravilniku (2007).

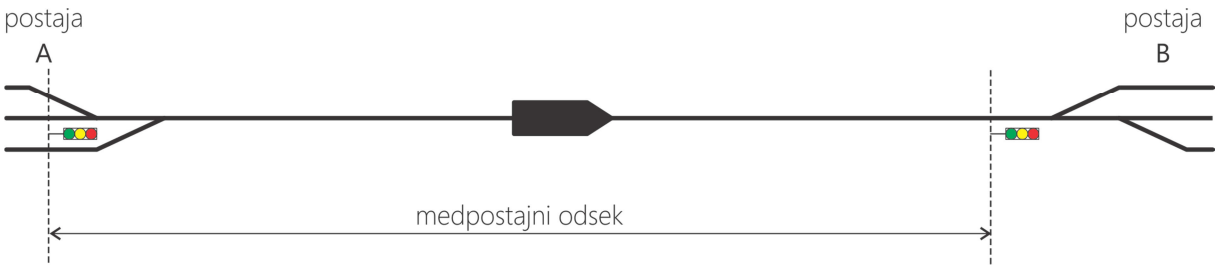
Temeljno načelo izvajanja in vodenja prometa je varen in urejen vlakovni promet. Varnost v železniškem prometu je ogrožena pri vožnji vlakov, ki se gibljeta drug proti drugemu in imata isto vozno pot (čelno ogrožanje), v primeru, ko hitrejši vlak dohiti počasnejšega na isti vozni

poti (ogrožanje pri sledenju), ter na območju kretnic, pri čemer imata lahko vlaka enako ali različno smer vožnje (bočno ogrožanje), zato mora biti vsaka vlakovna pot s signalno-varnostnimi napravami zavarovana za nalet, čelno in bočno trčenje. Varnost prometa se zagotavlja z zagotovitvijo proste vlakovne poti, postavitvijo kretnic na vozni poti v pravilno in natančno lego, postavitvijo ostalih kretnic in raztirnikov v položaj, ki zagotavlja bočno zaščito vožnje, z zavarovanjem prometa na cestnih prehodih in ustavitvijo premika (Theeg in Vlasenko, 2009).

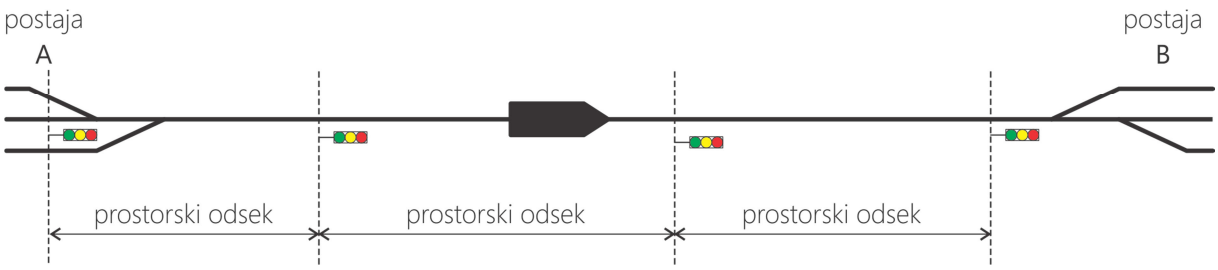
Zavarovanje voženj vlakov v isti smeri se zagotavlja s principom voženj v prostorskem razmaku med zaporednima vlakoma. To lahko dosežemo na dva načina (Hansen in Pachl, 2008). Prvi način je vožnja v fiksnem prostorskem razmiku, kjer progo razdelimo na prostorske odseke, zavarovane s signalno-varnostnimi napravami. Pri tem načinu se varnost železniškega prometa zagotavlja s pogojem, da je na enem prostorskem odseku lahko samo en vlak. Dolžina odseka je vsota zavorne poti pred prvim glavnim signalom, dolžine prostorskega odseka (razdalja med zaporednima signaloma) in dolžine prepeljavne poti (rezerve), ki znaša 50–200 metrov. Drugi način je vožnja v gibljivem prostorskem razmiku, kjer vodenje vlakov poteka s pomočjo v tir vgrajenega vodnika, ki informacije o prostosti odseka kontinuirano prenaša na lokomotivo. Ob progi ni klasičnih signalno-varnostnih naprav. Minimalna razdalja med zaporednima vlakoma je enaka dolžini gibljivega odseka, to je vsoti zavorne in prepeljavne poti. Nikjer na omrežju slovenskih železnic drugi sistem še ni implementiran, zato sistem za pomoč pri vodenju vlakov, razvit v okviru doktorske disertacije, omogoča replaniranje prometa v načinu fiksnega prostorskega razmika.

Na enotirnih progah, kjer je tir namenjen vožnji v obe smeri, je treba zagotoviti tudi zavarovanje voženj vlakov v nasprotni smeri. Pred odpravo vlaka v prostorski odsek med sosednjima prometnima mestoma morajo biti na postaji, ki bo sprejela vlak, vsi izvozni signali v smeri obravnavanega prostorskega odseka v legi za prepovedano vožnjo. S tem se zavaruje vlak pred nasproti vozečimi vlaki. Vlak v nasprotni smeri lahko nadaljuje z vožnjo, ko je prvi vlak prispel na postajo ter so izpolnjeni vsi pogoji za zavarovanje vozne poti (Prometni pravilnik, Uradni list RS, št. 50/2011: 6824–6931). Princip zavarovanja voženj vlakov v isti smeri je enak za eno-, dvo- in večtirne proge. Pri enotirnih progah je treba zagotoviti tudi zavarovanje voženj vlakov iz nasprotne smeri, zato v doktorski disertaciji uporabnost in učinkovitost predlaganega algoritma za časovno replaniranje vlakov testiramo na enotirni progi. Dvotirna proga, kjer je vsak tir namenjen vožnji v eno smer, je poenostavitev problema. In če je algoritem uspešen pri reševanju problema časovnega načrtovanja voženj vlakov na enotirni progi, lahko sklepamo, da je uspešen tudi pri reševanju zamud na dvo- in večtirnih progah.

Odprta proga, območje med sosednjima postajama, je lahko en prostorski odsek (t. i. medpostajni prostorski odsek), lahko pa je s signalno-varnostnimi napravami razdeljen na dva ali več odsekov (t. i. blokovni prostorski odseki). V primeru, da je med izvoznim signalom ene in uvoznim signalom druge postaje samo en odsek, je kapaciteta železniške infrastrukture majhna, saj je med postajama lahko največ en vlak. Kar pomeni, da je malo možnosti za odpravo zamud, optimizacija vodenja prometa pa je zelo lokalna. Z razdelitvijo tira med postajama na več ustrezno zavarovanih prostorskih odsekov se poveča zmogljivost železniške infrastrukture. Vsak odsek mora biti zavarovan z glavnim prostornim signalom, ki se v trenutku, ko prva os vlaka uvozi na prostorski odsek, postavi v položaj »stoj«; ko zadnja os vlaka zapusti blokovni odsek, pa v položaj »prosto«. Večje število blokovnih odsekov med postajama omogoča vožnjo več zaporednih vlakov (teoretično je lahko toliko zaporednih vlakov, kolikor je blokovnih odsekov), kar omogoča več možnosti optimizacije vodenja prometa, večji medsebojni vpliv med vlaki in hkrati večjo kompleksnost problema. Sodoben način vodenja prometa je centralno vodenje prometa na progi, kjer so odseki med postajama razdeljeni na več blokovnih odsekov; takšen način vodenja vlakov je implementiran v predlagani algoritem za časovno replaniranje voženj vlakov. Območji medpostajnega prostorskega odseka ter blokovnega prostorskega odseka sta ponazorjena na prikazih Slika 2 in Slika 3.



Slika 2: Definicija medpostajnega odseka
Figure 2: Definition of open section



Slika 3: Definicija prostorskega odseka
Figure 3: Definition of block section

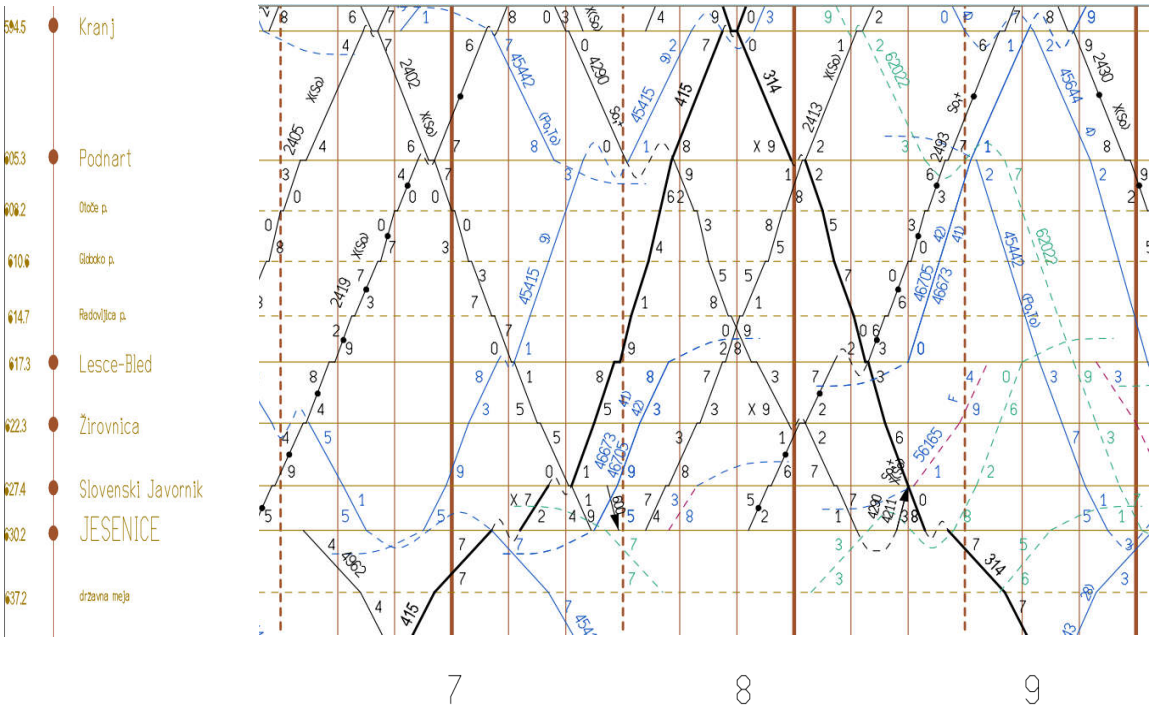
2.1 Vozni red

Vozni red je temeljni tehnološki dokument v železniškem prometu, ki ga izdela in uveljavi upravljavec železniške infrastrukture. Vozni red hkrati opredeljuje povpraševanje uporabnikov železniških storitev (potniški in tovorni promet) ter razpoložljive materialne, kadrovske in tehnične zmogljivosti, ki so potrebne za njegovo dosledno izvajanje (Zgonc, 2012). Iterativni proces usklajevanja vlakovnih poti se prične 18 mesecev pred objavo novega voznega reda. V tem času je naloga konstruktorjev, da izdelajo brezkonflikten in izvedljiv vozni red, v katerem upoštevajo in uskladijo vse vlakovne poti, za katere so zaprosili prevozniki, v katerem ima vsak vlak optimalno traso z minimalnimi postanki zaradi prometnih razlogov in so upoštevani vsi parametri zmogljivosti železniške infrastrukture (npr. prepustna zmogljivost prog, zmogljivost ranžirnih in potniških postaj, zmogljivost električnih napajalnih postaj in voznega omrežja ter omejitve osne obremenitve ali hitrosti na posameznih odsekih), podatki o voznih časih, postajni intervali in intervali sledenja vlakov ter tehnični in varnostni pogoji. V kolikor so v voznem redu predvidene vlakovne poti, ki materialno ali kadrovsko niso izvedljive (npr. ni zadostnega števila lokomotiv ali strojevodij), lahko nastanejo velike zamude. Z voznim redom se natančno določijo trase vlakov. Kvalitetno izdelan vozni red je nujen za zagotavljanje neoviranega in varnega odvijanja železniškega prometa, racionalne uporabe virov, nižanja stroškov poslovanja in povečanja izkoriščenosti železniške infrastrukture. Vozni red torej vsebuje vse potrebne informacije o časovnem poteku voženj, ki jih potrebujejo udeleženci v železniškem prometu. Z voznim redom določenim odhodom, prihodom in postankom na prometnih mestih morata v čim večji meri slediti osebji upravljavca in prevoznikov, saj neupoštevanje povzroča zamude in s tem povezane stroške (Zgonc, 2012). Rezultat konstruiranja voznega reda je grafikon prometa vlakov.

Grafikon prometa vlakov je dokument voznega reda upravljavca, ki vsebuje grafični prikaz tras vseh vlakov ter osnovne podatke o pripadajočem progovnem odseku ali progi (Prometni pravilnik, 2011). Trase so v grafikonu prikazane z linearnimi grafi funkcije poti in časa, kjer naklon grafa ponazarja hitrost vlaka, preseki tras s časovnimi linijami pa predstavljajo čas prihoda in odhoda s prometnega mesta. Trase so vrisane v koordinatni sistem, v katerem so praviloma na abscisni osi časovni podatki in na ordinatni osi oznake za prometna mesta. Črte, ki označujejo časovne podatke, so v minutnem ali desetminutnem intervalu. Vsaka črta, ki označuje uro, mora biti opisana z oznako časa in odebeljena. Črte, ki označujejo prometna mesta, se rišejo v razmerju dolžin odseka proge med prometnima mestoma. V podaljšku teh črt mora biti navedeno ime postajnega mesta. Za zagotavljanje preglednosti grafikona in večje zanesljivosti pri uporabi se trase prikazujejo s črtami različnih debelin, barv in z različnimi simboli. Čas prihoda vlaka se vpiše na levi strani trase, čas odhoda oz. čas

prevoza vlaka pa se vpiše(-ta) na desni strani trase vlaka. Številke se vpišejo v ostre kote, ki jih tvorijo grafi tras in črte, ki označujejo prometna mesta.

Na prikazu Slika 4 je primer grafikona voznega reda, v katerem so trase potniških vlakov prikazane s črno, trase tovornih pa z modro barvo. Rang vlaka je ponazorjen z debelino črte; vlak višjega ranga je prikazan z debelejšo črto. S polno črto so označene redne trase, s prekinjeno črto so prikazane izredne trase, s tanko zeleno prekinjeno črto pa so označene sistemske trase. Ob vsaki trasi je napisana številka vlaka, iz katere se lahko razbere vrsta vlaka (potniški/tovorni, notranji/mednarodni promet), rang vlaka, smer vožnje ter mesta sestajanj vlakov.



Slika 4: Primer grafikona voznega reda (vir: SŽ)
Figure 4: Example of train timetable (SŽ)

2.2 Operativno izvajanje in korekcija voznega reda v realnem času

Načrtovanje voznega reda je prvi korak v vodenju železniškega prometa, sledi mu izvajanje prometa v skladu z voznim redom. Za vodenje in urejanje vlakovnega prometa v skladu z voznim redom je zadolžen dispečer, katerega naloga je neprestano spremljanje odvijanja železniškega prometa, analiziranje relevantnih podatkov o lokacijah vlakov in stanju infrastrukture ter ocenjevanje, ali dejansko stanje sovpada z voznim redom, prepoznavanje in napovedovanje zamud ter izbira in implementacija korekcijskih ukrepov za zajezitev vpliva zamud (Rodriguez, 2007). Hiter in pravilen odziv dispečerja na nastalo zamudo vpliva na zagotavljanje zanesljivosti in točnosti storitev. Pri izbiri korekcijski akcij mora dispečer upoštevati, da akcije ne smejo voditi v nastanek konfliktov vodenja prometa (npr. da bi dva

vlaka hkrati zasedala odsek) in konfliktov zagotavljanja virov (osebja, vlečnih sredstev ...). Hkrati mora zagotavljati čim višjo kakovost storitev – npr. z upoštevanjem in zagotavljanjem prestopanj potnikov in neprekinjenosti logističnih verig v blagovnem prometu (Fay, 2000). Dinamično urejanje vlakovnega prometa je nujno, da se ohrani točnost vlakov in da se minimizirajo posledice zamud.

Če je/bo nastala zamuda, mora dispečer pričeti z izvajanjem ukrepov za odpravljanje zamude, torej z replaniranjem voznega reda ob upoštevanju vseh pravil za zagotavljaje varnosti in urejenosti železniškega prometa. Do zahtev po replaniranju voznega reda običajno privedejo motnje, kot so sprememba hitrosti na posameznih odsekih (npr. uvedba počasne vožnje, to je vožnje s hitrostjo, ki je manjša od dovoljene progovne hitrosti in manjša od hitrosti, predpostavljene v voznem redu), zamuda enega ali več vlakov, zahteva za traso, ki ni predvidena v voznem redu (zahtevek za traso, vložen znotraj 24 ur pred nameravano vožnjo vlaka), zunanji vplivi, ki vplivajo na odvijanje železniškega prometa (npr. vozilo na nivojskem prehodu), ter okvare in napake na železniški infrastrukturi in voznem parku (npr. nedelujoč signal, nepravilna lega kretnice, okvara lokomotive) (Dorfman in Medanic, 2004).

V železniškem prometu so motnje, ki povzročajo manjša odstopanja od teoretičnih voznih časov, pričakovane in se jim ni mogoče ogniti, zato se pri konstruiranju voznega reda teoretičnemu voznemu času in/ali času med zaporednima vlakoma in/ali času križanja vlakov doda časovni dodatek, t. i. tamponski čas (Kecman et al., 2012). Časovna dodatka (med zaporednima vlakoma in pri križanju vlakov) zagotavljata stabilnost voznega reda in dispečerjem omogočata čas za izbiro in implementacijo korekcijskih akcij, časovni dodatki voznemu času pa zmanjšujejo vpliv različnih voznih dinamik zaradi različnih vremenskih pogojev ali različnih voznih lastnosti lokomotiv. Tamponski čas sestoji iz dveh komponent, in sicer iz varnostne, ki preprečuje trk vlakov, in komponente zanesljivosti sistema, ki zmanjšuje domino efekt širjenja zamud. Tamponski čas torej omogoča kompenziranje manjših odstopanj in s tem povečuje stabilnost sistema (Luethi et al., 2007). Velja, da se z večanjem časovnih dodatkov manjša verjetnost nastanka sekundarnih zamud, posledično pa se zmanjša potreba po replaniranju. Dodajanje tamponskih časov zmanjšuje uporabno kapaciteto železniške infrastrukture, zato se upravljavci zaradi zagotavljanja večjega števila vlakov pogosto odločijo za zmanjšanje časovnih dodatkov na vrednost, ki je čim bliže varnostni komponenti.

Motnje, ki jih ni mogoče kompenzirati s tamponskim časom, povzročijo zamude in zahtevajo korekcijo (replaniranje) voznega reda, lahko pa tudi replaniranje virov (voznega parka in osebja). V nadaljevanju podajamo kratek pregled ukrepov oz. prilagoditev, ki jih avtorji predlagajo v procesu replaniranja vlakov, in sicer:

- spremembe hitrosti vlakov na odprti progi;
- spremembe trajanja postankov;
- spremembe vrstnega reda vlakov;
- ukinitve zamujenih vlakov;
- spremembe vlakovnih poti;
- izbire obvoznih poti;
- dodajanje vlakov;
- spremembe v vzorcu postankov.

Po nastanku zamude je ena izmed možnosti, ki jih ima na voljo dispečer pri reševanju konfliktov, sprememba vlakovnih poti na odprti progi s prilagajanjem hitrosti vlakov. Za učinkovito replaniranje vlakov po tej metodi mora dispečer natančno poznati trenutne lokacije in hitrosti vlakov ter imeti možnost, da strojevodji sporoči spremembo hitrosti vlaka v vsakem trenutku. To omogoča evropski sistem za nadzor in upravljanje vlakov, nivo 2, ki pa, kot smo že omenili, ni pogosto implementirani sistem na evropskem železniškem omrežju.

Pri načinu vodenja vlakov, ki je trenutno najpogosteje uporabljen na železniških progah v Evropski uniji, dispečer ve, kateri odseki so zasedeni ter kateri vlak je na posameznem odseku, vendar ne pozna natančne lokacije vlaka znotraj odseka; ve, kakšno hitrost naj bi imel vlak glede na vozni red, vendar ne pozna dejanske hitrosti vlaka, zato so aktivnosti replaniranja namesto k spremembam hitrosti vlakov najpogosteje usmerjene k spremembam postankov vlakov na postajah. Postajni tiri namreč omogočajo sestajanje vlakov in prilagajanje postankov. Dispečerji določijo nove čase prihodov in odhodov s postaj, pri čemer morajo upoštevati, da postanek vlaka ne sme biti krajši od časa, ki ga zahtevajo od prometno-tehnične operacije, upoštevati pa morajo tudi kapaciteto železniške infrastrukture, da preprečijo brezizhodno situacijo (glej prikaza Slika 15 in Slika 16 na str. 43).

V primeru zamud z zelo velikim vplivom avtorji predlagajo povečanje kapacitete za preostale vlake z ukinitvijo zamujenega vlaka ali izbiro obvozne poti (Jespersen-Groth et al., 2006). Odpoved odhoda vlaka povzroči več novih konfliktov, kajti odpoved vlaka z začetne postaje hkrati pomeni odpoved odhoda tega vlaka v obratni smeri; hkrati lahko ta vlak povzroči konflikt na postaji, ker zaseda tir. Alternativa odpovedi vlaka na celotni trasi je skrajšanje vlakovne poti; vlak ne pripelje do končne postaje, ampak obrne, še preden doseže z voznim redom predvideno končno postajo.

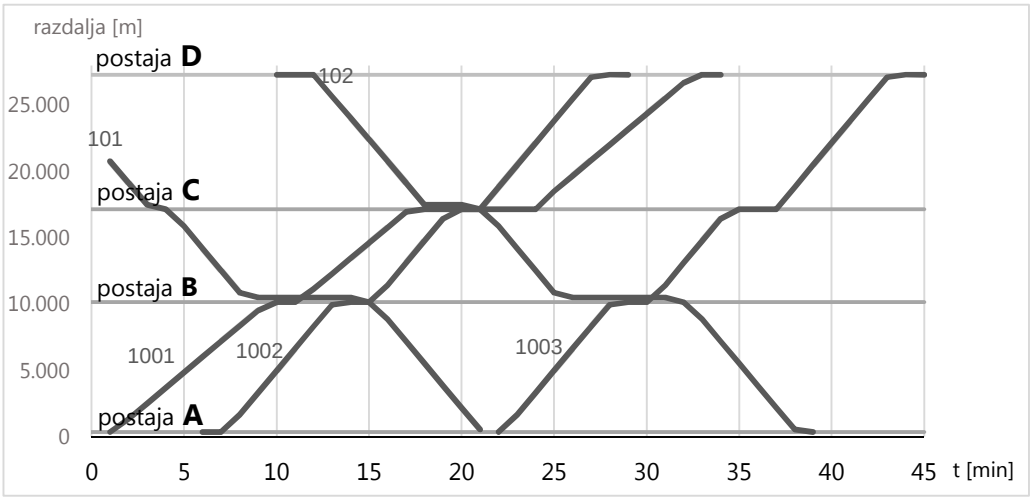
V tujini uporabljajo tudi metodo dodajanja vlaka na vmesni postaji po voznem redu (Jespersen-Groth et al., 2006). To pomeni, da v primeru zamude vlaka na vmesni postaji dodajo vlak, ki odpelje po voznem redu. Ko zamujeni vlak pripelje do te postaje, ga izključijo iz prometa. S takšnim pristopom se zmanjšajo zamude potnikov, domino efekt širjenja zamude se hitro zajezi, vendar se pogosto zgodi, da viri (kadrovski in materialni) niso razpoložljivi na določeni postaji in ob času, ko nastane zamuda.

Dispečerji se za zmanjšanje zamud in omejitve širjena zamud lahko odločijo za ukrep ukinitve postankov zamujenega in/ali drugih vlakov (Nagasaki, Eguchi in Koseki, 2003). Na katerih postajah v določenih urah postanek ni potreben, se določi na osnovi raziskave potovalnih navad. V primeru izbire in uvedbe tega ukrepa je pomembno, da so potniki pravočasno obveščeni o novem vzorcu postankov, da lahko pravočasno izberejo alternativo.

Ne glede na izbrano strategijo korekcije voznega reda mora biti dispečerjev niz akcij izvedljiv in učinkovit. Ker je železniški promet dinamičen proces, mora dispečer na zamudo odgovoriti v nekaj minutah in zato ne more preiskati vseh izvedljivih rešitev, da bi določil optimalno. Odločitve o korekcijskih akcijah zato temeljijo na izkušnjah in intuiciji dispečerjev (Hara et al., 2006; D'Ariano et al., 2008). Zaradi časovne omejitve se dispečerji pogosto odločijo le za manjše popravke voznega reda (Fay, 2000). Njihovi ukrepi so sicer izvedljivi in ne pride do trka vlakov, vendar niso optimalni, zato bi pri delu potrebovali sistem za pomoč pri odločanju, ki bi hitro in nepristransko preveril uspešnost različnih akcij. Algoritem, ki smo ga razvili v sklopu doktorske disertacije, je zasnovan tako, da s spremembami dolžin postankov (s tem tudi odhodov vlakov) predlaga replanirani vozni red, v katerem so upoštevani vsi vlaki in vsi predvideni postanki. Algoritem predlaga tudi postanke vlakov na postajah, kjer sicer po voznem redu postanek ni predviden, če je takšen postanek potreben za optimalno rešitev. Takšen pristop je tudi najpogosteje obravnavan v literaturi. Odpravljanje zamud z odpovedjo zamujenega vlaka na celotni trasi ali na delu le-te, s spremembo vzorcev postankov vlakov ali z dodajanjem točnega vlaka v algoritmu ni predvideno, saj takšni ukrepi niso običajni pri replaniranju vlakov na slovenskem železniškem omrežju.

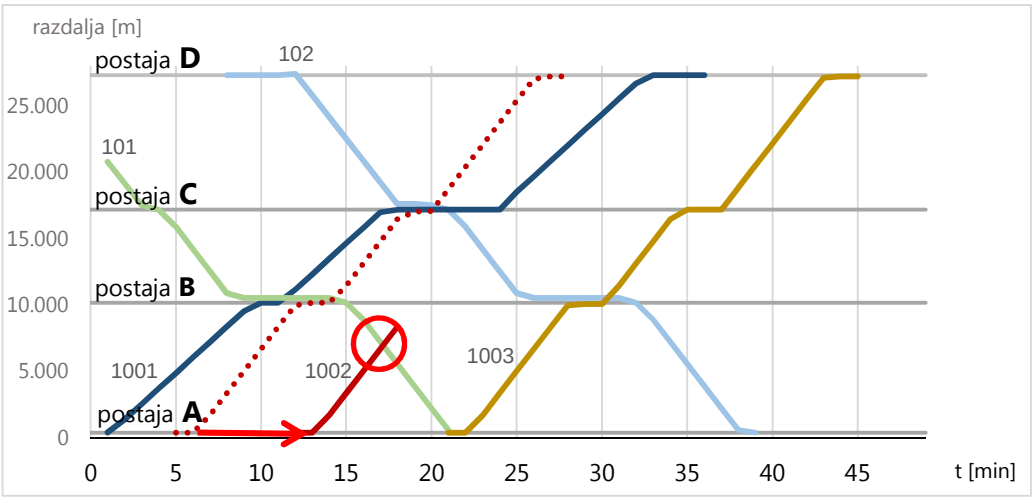
2.3 Kompleksnost replaniranja voženj vlakov

Za ilustracijo občutljivosti železniškega prometa na zamude in kompleksnosti časovnega replaniranja vlakov podajamo krajši primer, kjer je zaradi zamude enega vlaka treba korigirati vozni red. Grafikon voznega reda za pet vlakov, ki vozijo na enotirni progi med postajama A in D, z vmesnima postajama B in C, je prikazan na prikazu v nadaljevanju.



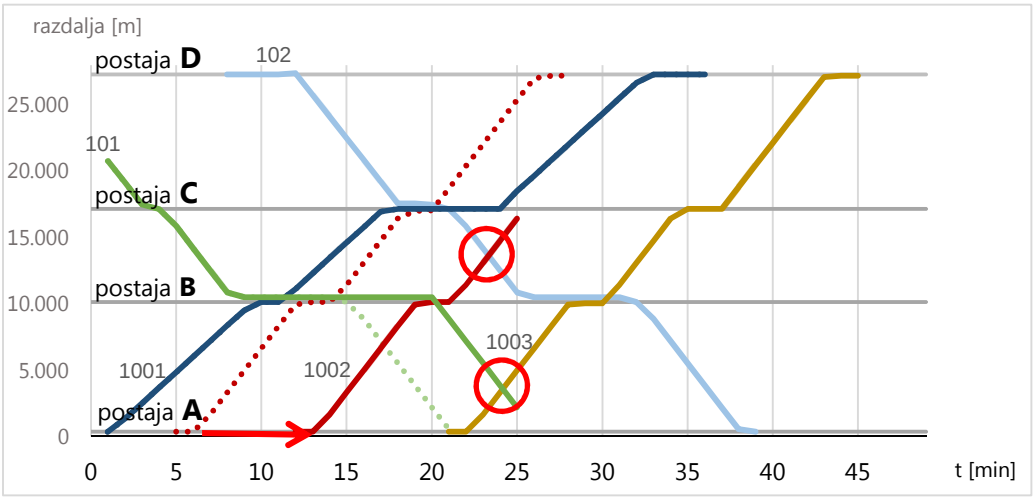
Slika 5: Grafikon voznega reda
Figure 5: Train timetable

V nadaljevanju obravnavamo primer, ko Vlaku 1002 pripišemo zamudo 5 min. S prikaza Slika 6 je razvidno, da brez prilagajanja vlakovne poti (podaljšanja postanka ali spremembe hitrosti) Vlaka 1002 v 17. minuti nastane konflikt med vlakoma 1002 in 101. Velja namreč, da se vlakovne poti vlakov na enotirni progi lahko križajo samo na postajah ali izogibališčih, kjer dodatni tiri to omogočajo. Križanje vlakovnih poti na odseku med postajama bi v naravi pomenilo trk vlakov.



Slika 6: Konflikt, ki ga povzroči zamuda Vlaka 1002
Figure 6: Conflict caused by delayed train 1002

Za odpravo konflikta v 17. minuti lahko spremenimo čas odhoda Vlaka 1002 s postaje 2 ali podaljšamo postanek Vlaka 101 na postaji B. Izberemo drugo možnost in upoštevamo, da ima Vlak 1002 na postaji B postanek tako dolg, kot je predviden v voznem redu, ter da vozi s hitrostjo, predpisano z voznim redom. Vlakom 101, 102 in 1003 ne spreminjamo vlakovnih poti.

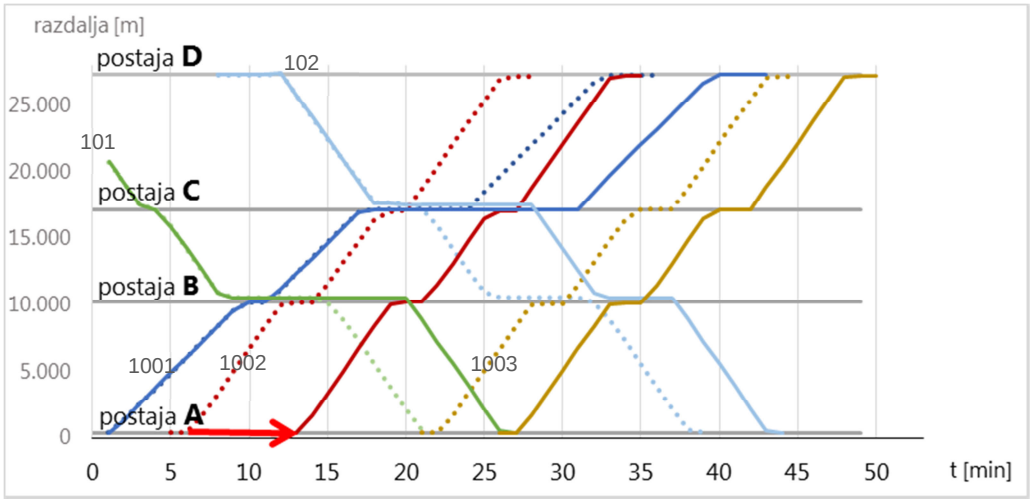


Slika 7: Konflikta, ki ju povzroči sekundarna zamuda Vlaka 101
Figure 7: Conflicts caused by delayed train 101

S prikaza Slika 7 je razvidno, da smo s podaljšanjem postanka Vlaka 101 na postaji B odpravili konflikt v 17. minuti, vendar sta nastala nova konflikta v 24. minuti, ki ju je treba v naslednjih korakih replaniranja odpraviti. Vozni red iterativno korigiramo, dokler ni rezultat brezkonflikten vozni red. Torej, rezultat replaniranja je vozni red, v katerem so vsi takšni konflikti odpravljeni ter so upoštevane vse prometno-tehnične in varnostne omejitve, ki so zahtevane pri izvajanju in vodenju železniškega prometa.

V nadaljevanju podajamo nekaj možnih pristopov in rešitev problema časovnega replaniranja voženj vlakov zaradi zamude Vlaka 1002. Več pristopov, ki se uporabljajo za replaniranje, je opisanih v poglavju 2.2.

Ena izmed možnosti, ki se intuitivno ponuja pri replaniranju, je premik vseh še neizvedenih aktivnosti v desno. Replanirani vozni red z upoštevanjem tega pravila je prikazan na prikazu v nadaljevanju (Slika 8), iz katerega je razvidno, da so vlaki 1001, 101 in 102 prvi del poti (do prve naslednje postaje) prevozili skladno z voznim redom, nato so se odhodi vseh vlakov prilagodili zamudi Vlaka 1002. Z uporabo pravila zamika še neizvedenih aktivnosti v desno za urejanje vlakovnega prometa se ohranijo hitrosti vlakov, dolžine postankov, mesta križanj, vrstni red vlakov, ne nastajajo pa novi konflikti. Vseeno pa takšnega pristopa ne moremo opredeliti kot optimizacijo problema, temveč le kot reševanje posameznega konflikta. Pristop ni učinkovit na progah z visoko izkoriščenimi kapacitetami, ker ni vrzeli med vlaki, ki bi kompenzirala zamudo, zato se lahko zamuda prenaša na druge vlake več ur.

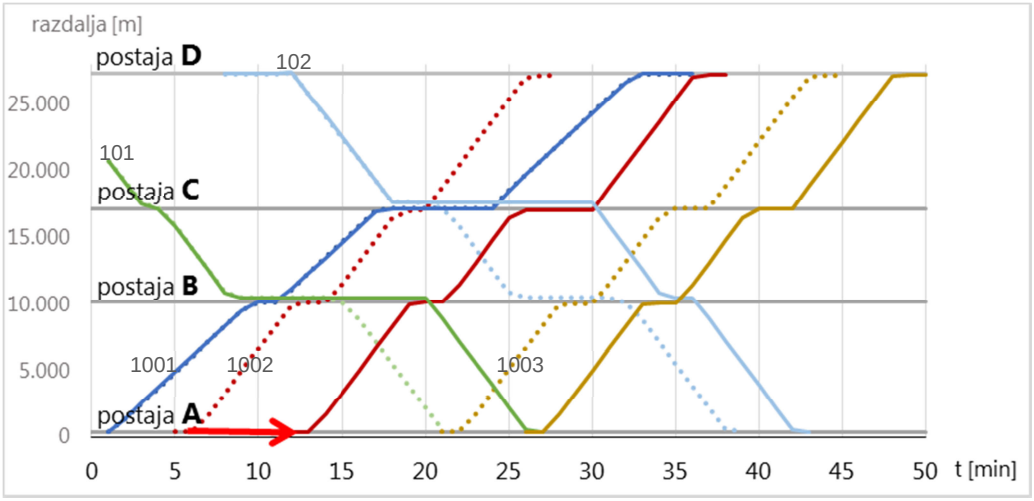


Slika 8: Replanirani vozni red – primer 1
Figure 8: Rescheduled timetable – example 1

Drugačen pristop k reševanju konflikta zaradi zamude vlaka je sprememba dolžin postankov na postajah, pri čemer je treba upoštevati, da mora imeti vlak vsaj minimalen čas postanka, ki je potreben zaradi prometno-tehničnih razlogov (npr. čas, potreben za vstop in izstop potnikov). Pri tem načinu replaniranja se lahko spremenita vrstni red in mesta križanj vlakov. V nadaljevanju sta podana dva primera korekcij voznega reda, kjer sledimo različnima ciljema. V prvem primeru želimo, da zamujeni vlak čim prej nadaljuje z vožnjo, torej ko so izpolnjeni vsi varnostni in prometno-tehnološki pogoji, v drugem primeru pa želimo, da se ohranja točnost vlakov, torej imajo točni vlaki najvišjo, zamujeni pa najnižjo prioriteto.

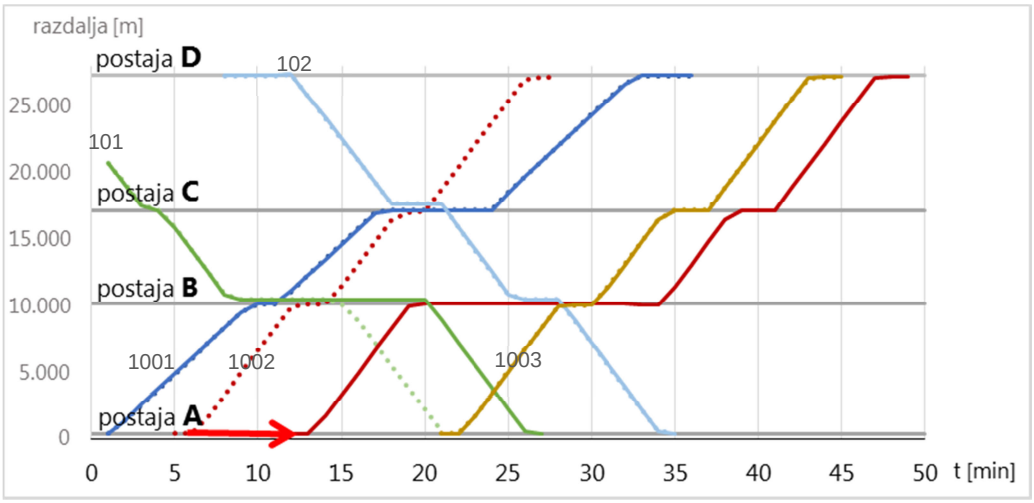
V prvem primeru smo torej upoštevali, da dobi Vlak 1002 dovoljenje za vožnjo takoj, torej v 14. minuti. V tem primeru se vrstni red vlakov 1001 in 1002 na postaji C ne spremeni. Da hitrejši vlak (Vlak 1002) ne ostane ujet za počasnejšim vlakom (Vlakom 1001), je treba predvideti zamenjavo vrstnega reda na kasnejših postajah. Na prikazu Slika 9 je prikazan

rezultat replaniranja voženj vlakov s spreminjanjem postankov, pri čemer smo upoštevali, da gre zamujeni vlak čim prej na pot. Mesta križanj vlakov so se v tem primeru ohranila.



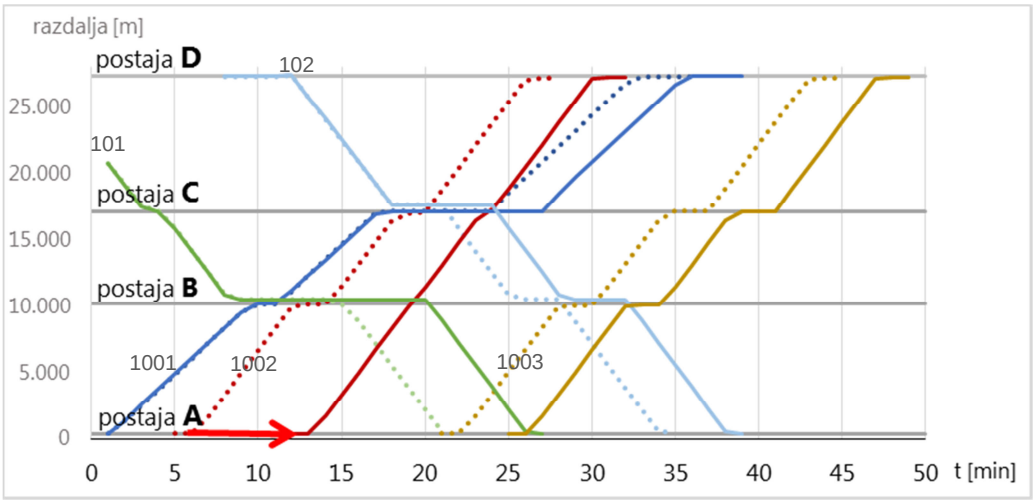
Slika 9: Replanirani vozni red – primer 2
Figure 9: Rescheduled timetable – example 2

V drugem primeru je bil cilj ohraniti točnost čim več vlakov, zato se za traso zamujenega vlaka izkoristijo časovne vrzeli. Rezultat replaniranja voženj vlakov s prilagajanjem dolžin postankov je prikazan na prikazu Slika 10.



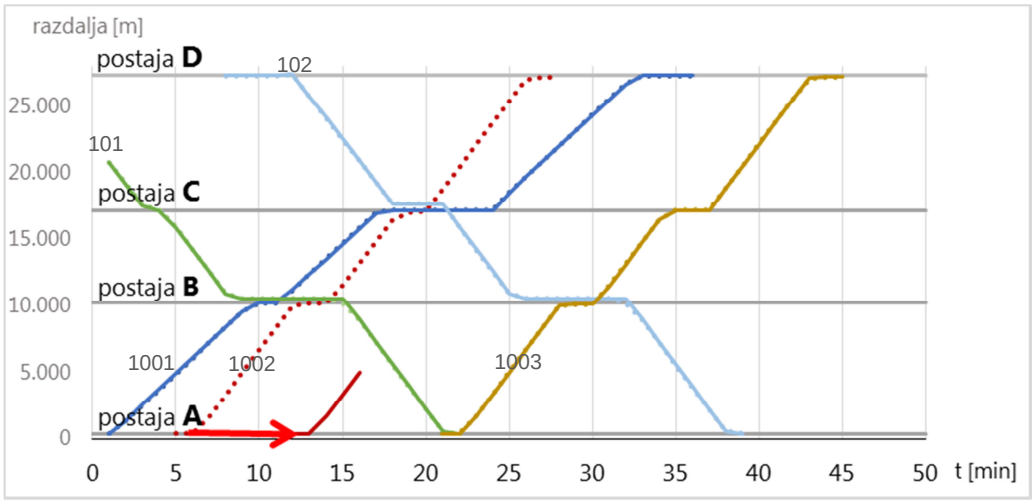
Slika 10: Replanirani vozni red – primer 3
Figure 10: Rescheduled timetable – example 3

Tretji pristop k reševanju konflikta in odprave posledic zamude je odpoved postankov vlakov. Pri takšnem načinu replaniranja je treba oceniti, kako takšen ukrep vpliva na potnike, in potnike o spremembi postankov pravočasno obvestiti. Na prikazu Slika 11 je prikazan primer, kjer postanke odpovemo samo za zamujeni vlak. Lahko bi odpovedali postanke tudi drugim vlakom in tako skrajšali čas, potreben za odpravo zamud, vendar bi povzročili dodatne negativne vplive na potnike.



Slika 11: Replanirani vozni red – primer 4
Figure 11: Rescheduled timetable – example 4

Četrta možnost za odpravo konflikta je ukinitev zamujenega vlaka. Ta ukrep je sicer neugoden za potnike, ki so želeli na zamujeni vlak, vendar ima pozitiven učinek na potnike, ki so na vlakih, na katere bi vplivala zamuda. Tudi pri tem pristopu replaniranja ne gre za optimizacijo problema, temveč za odpravo konflikta zaradi zamude.



Slika 12: Replanirani vozni red – primer 5
Figure 12: Rescheduled timetable – example 5

Iz velike množice možnih rešitev smo prikazali samo pet primerov replaniranih voznih redov. Tako pri konstrukciji voznega reda kot tudi pri replaniranju voženj vlakov je treba zagotoviti varnostne zahteve, to pomeni, da se vlaki lahko sestajajo samo na območju postaj oz. tam, kjer dodatni tiri to omogočajo. Iz diagramov replaniranih voznih redov je razvidno, da je ta pogoj upoštevan v vseh petih primerih. Pri replaniranju pa je pomembna tudi učinkovitost izvedenih korekcijskih ukrepov. Učinkovitost replaniranega voznega reda se najpogosteje meri z velikostjo zamud, zato so v nadaljevanju povzete zamude posameznih vlakov na končni postaji obravnavanega območja in vsota zamud vseh vlakov.

Preglednica 1: Skupne zamude vlakov za različne pristope replaniranja.
Table 1: Total train delays obtained with different rescheduling aproaches.

Primer/Vlak	**Zamuda [min]**					
	101	102	1001	1002	1003	∑
Primer 1: »Premik aktivnosti v desno«	6	6	8	7	6	33
Primer 2: »Zamujeni vlak čim prej«	6	5	0	10	6	27
Primer 3: »Čim več točnih vlakov«	6	0	0	21	0	27
Primer 4: »Ukinitev postankov zamujenega vlaka«	6	5	4	4	5	24
Primer 5: »Ukinitev zamujenega vlaka«	0	0	0	/	0	?

Iz preglednice je razvidno, da je pristop »Premik še neizvedenih aktivnosti v desno« najmanj uspešen, saj imajo vsi vlaki zamudo; njihova skupna zamuda pa je največja. Skupne zamude v primerih 2 in 3 so enake, vendar je učinkovitost pristopov različna za posamezne vlake. O izbiri med primeroma 2 in 3 bi lahko odločal dodatni kriterij, npr. ali želimo čim večje število točnih vlakov ali pa zamudo razporedimo med več vlakov. V primeru 4 so skupne zamude sicer najmanjše, vendar je treba upoštevati, da takšen pristop ni uspešen za potnike, ki so želeli vstopiti/izstopiti na postajah, na katerih je bil predvideni postanek ukinjen. Zadnji primer bi lahko obravnavali kot učinkovit pristop k replaniranju, če bi upoštevali samo skupne zamude vlakov na končnih postajah, vendar ima ukinjeni vlak druge negativne učinke, ki jih je treba upoštevati pri vrednotenju uspešnosti replaniranja. V takšnem primeru je treba upoštevati stroške potnikov, ki morajo spremeniti načrt potovanja (izbrati drugo prevozno sredstvo ali kasnejši vlak).

V obravnavanem primeru smo uspešnost replaniranja voženj vlakov ocenjevali glede na skupne zamude vlakov na končni postaji in ob predpostavki, da so stroški zamud za vse vlake enaki. To je samo en vidik uspešnosti replaniranja. Pri vrednotenju uspešnosti replaniranja bi lahko upoštevali, da so stroški zamud bolj zasedenih vlakov višji (upoštevamo zamude glede na število potnikov), ali pa upoštevamo višji strošek zamude vlakov, ki v jutranji konici vozijo v smeri večjih mest (upoštevamo zamude dijakov, študentov in zaposlenih, ki se z vlakom vozijo v Ljubljano, Celje ali Maribor). Uspešnost replaniranja lahko vrednotimo tudi z vidika upravljavca (upoštevamo stroške porabe energije) ter z vidika potnikov (upoštevamo stroške potnikov zaradi zamujenega prestopanja), lahko pa upoštevamo kombinacije različnih kriterijev.

Iz predstavljenega primera je razvidno, da reševanje enega konflikta lahko vodi v več novih konfliktov ter da je uspešnost replaniranja odvisna tudi od ciljev, ki si jih zastavimo. Za vodenje in urejanje vlakovnega prometa so zadolženi dispečerji, ki imajo za izbiro korekcijskih akcij in implementacijo le-teh na voljo le kratek čas. Običajno se ne preveri, ali je bila njihova odločitev optimalna glede na zastavljeni cilj (Gély, Dessagne in Lérin, 2006). Kompleksnost replaniranja voženj vlakov se s povečevanjem števila vlakov in postaj ter dejstva, da se vlaki lahko sestajajo samo na postajah (Pachl, 2011), še povečuje, zato je jasno, da dispečerji potrebujejo sistem za pomoč pri odločanju, ki jim bo pomagal najti kvalitetno rešitev v razumno kratkem času.

2.4 Zmanjševanje kompleksnosti časovnega načrtovanja voženj vlakov

Časovno načrtovanje voženj vlakov, razen nekaterih enostavnih primerov, spada v razred NP-problemov, kar pomeni, da je množica rešitev zelo velika, kar rezultira v (pre)veliko časovno zahtevnost algoritma. Število novih brezkonfliktnih voznih redov narašča eksponentno z večanjem števila vlakov, števila odsekov ter števila postaj in postajnih tirov. Avtorji zato za zmanjšanje kompleksnosti problema predlagajo decentralizirani pristop k vodenju železniškega prometa (Corman et al., 2011; Geske, 2006; Ghosh, 2001; Kuster et al., 2008; Letia et al., 2008). Poglavitna lastnost decentraliziranega pristopa je, da je železniško omrežje razdeljeno na krajše odseke, na katerih dispečerji avtonomno zagotavljajo varen, urejen in nemoten promet. Krajši odseki pomenijo manjše število odsekov in postaj, manjše število vlakov in krajše časovno okno, kar močno zmanjša kompleksnost problema. V primeru decentraliziranega pristopa dispečerji poznajo razmere (položaje in hitrosti vlakov ter kapacitete železniške infrastrukture) samo na odsekih, za katere so zadolženi, in na osnovi teh informacij vodijo promet na območju, za katerega so zadolženi. Dispečer zaradi manjše kompleksnosti uspe uspešno voditi promet na odseku, za katerega

je zadolžen, med tem ko je globalna učinkovitost vodenja vlakov vprašljiva, saj je zelo odvisna od komunikacije in usklajenosti med dispečerji.

Z določitvijo časovnega okna se izločijo vlaki, ki so izven obravnavanega časovnega okvira, posledično se zmanjša kompleksnost problema, saj manj vlakov pomeni manj možnih konfliktov in tudi manj možnih rešitev. Kuster in sodelavci (2008) predlagajo, da se velikost časovnega okvira prilagodi resnosti zamude. Wegele in Schnieder (2004) kompleksnost problema zmanjšata z določitvijo neodvisnih verig. V neodvisni verigi so zajeti vlaki, katerih zamuda se prenaša na druge vlake, medtem ko se zamude med dvema verigama ne prenašajo. Takšen pristop omogoča razdelitev problema na več manjših problemov, ki jih rešujemo ločeno (lahko tudi na različnih računalnikih). Geske (2006) pa predlaga razdelitev obravnavanega območja na več manjših območij s približno enakim številom blokovnih odsekov, pri čemer je treba uskladiti čas in hitrost vlaka, ki zapušča eno in vstopa v drugo območje. Corman in sodelavci (2010; 2011) so izdelali algoritem, s katerim povežejo več manjših območij in poiščejo globalni optimum na železniškem omrežju.

Slabost upoštevanja krajšega časovnega okna ali manjšega območja je kratkovidnost rešitve, saj spremembe voznega reda lahko povzročijo konflikte z vlaki, ki niso zajeti v obravnavanem časovnem okviru, ali z vlaki, ki so izven obravnavanega območja. Mascis in sodelavci (2004) utemeljujejo, da konflikti izven časovnega okvira niso tako pomembni kot trenutni konflikti, saj lahko nepredvideni dogodki še vplivajo na bodoče konflikte.

Na hitrost reševanja problema vplivamo tudi z natančnostjo modelov, s katerimi ponazorimo realno stanje, in sicer ločimo mikroskopske in makroskopske modele. Mikroskopski modeli zelo natančno upoštevajo železniško infrastrukturo ter potreben čas med zaporednima vlakoma in natančno računajo čas zasedenosti odsekov in so zato časovno zelo zahtevni. Za razliko od mikroskopskih modelov makroskopski modeli ne upoštevajo vseh lastnosti in kapacitet železniškega omrežja in postajo obravnavajo kot točko, odprto progo pa kot povezavo. Zaradi poenostavitev in opustitve računanja dejanskih časov zasedenosti odsekov je časovna zahtevnost makroskopskih modelov majhna. Zato avtorji običajno za replaniranje vlakov na večjem območju uporabljajo makroskopski model omrežja (Acuna-Agost et al., 2011; Chiu et al., 2002; Dollevoet et al., 2012; Gély et al., 2006; Kumazawa et al., 2010; Min et al., 2011; Nagasaki et al., 2003; Törnquist Krasemann, 2012).

2.5 Pregled pristopov k časovnemu načrtovanju voženj vlakov

V tem poglavju je podan pregled relevantnih del, ki obravnavajo časovno načrtovanje voženj vlakov, predvsem z vidika zmanjšanja vpliva odstopanj od voznega reda zaradi zamude enega ali več vlakov. V grobem razlikujemo dve vrsti časovnega načrtovanja železniškega prometa, in sicer za potrebe voznega reda ter časovno načrtovanje v realnem času (replaniranje voženj vlakov po nastanku zamude). Vozni red se izdeluje in usklajuje do 18 mesecev, kar pomeni, da imajo konstruktorji voznega reda na voljo dovolj časa za usklajevanje in optimiziranje naročenih voznih poti, medtem ko mora osebje, odgovorno za vodenje vlakov v primeru nastanka zamude, korigirati vožnje vlakov v realnem času. Ravno zaradi časovne komponente je področje replaniranja vlakov za raziskovalce bolj zanimivo. V literaturi tako zasledimo objave, ki dokazujejo uporabnost različnih tehnik in pristopov za reševanje tega kompleksnega optimizacijskega problema. V nadaljevanju navajamo samo pomembnejše pristope replaniranja, ki jih je mogoče zaslediti v strokovni in znanstveni literaturi. Obširen pregled in primerjavo pristopov konstruiranja in replaniranja voznih redov so objavili Assad (1980), Cordeau in sodelavci (1998) ter Cacchiani in sodelavci (2014).

V literaturi za časovno replaniranje vlakov avtorji uporabljajo različne termine, kot so replaniranje (ang. *rescheduling*) (Dotoli et al., 2013; Kecman et al., 2012), upravljanje zamud (ang. *delay management*) (Kuster, Jannach in Friedrich, 2008; Dollevoet, 2012) in upravljanje manjših motenj (ang. *disturbance management*) (Luethi, 2009; Törnquist, 2007). V vseh primerih je bistvo korekcija voznega reda zaradi zamude enega ali več vlakov, zato v pregledu znanstvenega področja ne razlikujemo med temi poimenovanji.

Prvi hevristični pristop je razvil Fay (2000), ki ugotavlja, da je analitični pristop neučinkovit pri reševanju problema časovnega replaniranja vlakov, zato predlaga kombinacijo analitičnih in hevurističnih pristopov. Fay je prvi razvil in opisal hevristični pristop, v katerem je na osnovi intervjujev z dispečerji definiral približno 100 pravil za odločanje v primeru zamud. Za uporabo posameznih pravil pri odločanju je nastavil od 2 do 8 kriterijev, ki jih morajo izpolnjevati okoliščine. Ker kriteriji niso vedno povsem doseženi, je samo izbiro pravil v modelu izvajal z uporabo teorije mehke logike. Izbrana pravila je potem uporabil pri simulaciji scenarijev z uporabo PETRI-mrež. Rezultat teh simulacij je čas odhodov vlakov, pri čemer upošteva dva vlaka, in sicer Vlak 1, ki je na postaji in bi moral po voznem redu iti že na pot, in Vlak 2, ki z zamudo prihaja na postajo in katerega potniki želijo prestopiti na Vlak 1. Njegov algoritem omogoča določitev novih časov in vrstnega reda odhodov za Vlak 1 in Vlak 2, pri čemer v pravilih upošteva velikost zamude, število potnikov, ki prestopajo, čas med zaporednima vlakoma po zamudi, dolžino potovanja do končne postaje ter kriterijsko funkcijo. V kriterijski funkciji je upošteval skupno zamudo vlakov, uteženo z rangom vlaka,

skupno zamudo potnikov ter morebitne dodatne stroške zaradi potreb po dodatnem osebju in/ali vlakih.

Ping in sodelavci (2001) so predstavili optimizacijo časovnega replaniranja vlakov na dvotirni progi z uporabo genetskih algoritmov (ang. *genetic algorithm*). Glavna parametra sta dolžina postanka posameznega vlaka na posamezni postaji in vrstni red vlakov. Kot kriterij za izbor optimalne rešitve je uporabljena minimalna skupna zamuda vseh vlakov.

Ho in Yeung (2001a) sta na diskretnem odločitvenem modelu železniškega vozlišča primerjala uporabnost hevrističnih metod, in sicer sta metode genetski algoritem, tabu iskanje (ang. *tabu search*) in simulirano ohlajanje (ang. *simulated annealing*) primerjala z metodo dinamičnega programiranja. Avtorja sta primerjala kvaliteto rešitve in čas, potreben za določitev vrstnega reda vlakov na vozlišču v primeru zamude enega vlaka. Kot kriterij optimalnosti je bila uporabljena vrednost utežene skupne zamude vlakov. V zaključkih podata ugotovitev, da so za praktično uporabo testirane hevristične metode v primeru večjega števila vlakov uspešnejše od metode dinamičnega programiranja.

Medanic in Dorfman (2002a; 2002b) sta predstavila strategijo za replaniranje voženj vlakov na enotirni progi z uporabo požrešnega algoritma, ki v vsakem koraku išče lokalni optimum v upanju, da bo našel tudi globalnega. Kompleksnost problema sta zmanjšala z uporabo dogodkovnega modela (ang. *discrete-event model*). Prioritete vlakov so določene na osnovi njihove lokacije in hitrosti vlakov v bližini. Uspešnost replaniranja sta ocenjevala glede na zamude in porabljeno energijo. Zamude sta zmanjševala s prilagajanjem vrstnega reda vlakov in trajanjem postankov vlakov, čim nižjo porabo energije pa s prilagajanjem hitrosti vlakov na odprti progi. V nadaljevanju svojih raziskav sta pristop, predstavljen leta 2002, nadgradila iz modela enotirne proge v model za optimizacijo replaniranja na širši mreži (Dorfman in Medanic, 2004).

Nagasaki in sodelavci (2003) so za replaniranje voženj vlakov na dvotirni progi predlagali PERT-metodo (stohastična metoda planiranja projektov, ki temelji na tehniki mrežnega planiranja). V PERT-diagramih so vnaprej opredelili vse dogodke, dejavnosti ter njihove medsebojne odvisnosti, ki nastanejo pri vodenju vlakov. Za čas dejavnosti pa so uporabili potovalne čase in čase postankov. Replaniranje voznega reda so izvajali na dva načina: v prvem so spreminjali prestopne postaje, v drugem pa so spreminjali vzorec postankov. V kriterijski funkciji so z vidika potnika finančno ovrednotili zamude, število prestopanj in občutek gneče. Slabost uporabe tehnike PERT predstavlja potreba po vsakokratnem prilagajanju mrežnega diagrama, saj je za uporabo predlaganega pristopa na drugačni konfiguraciji železniške infrastrukture treba določiti nove odvisnosti med dejavnostmi.

Ghoseiri in sodelavci (2004) so razvili večkriterijski optimizacijski model replaniranja vlakov. Za izbor optimalne rešitve so upoštevali kriterija minimalne porabe energije in minimalno porabljenega časa potnikov. V predlaganem algoritmu predlagajo najprej generiranje Pareto omejitev obeh kriterijev in v drugem koraku izvedbo večkriterijalne analize.

Wegele in Schnieder (2004) sta prikazala optimizacijo replaniranja voženj vlakov z uporabo genetskih algoritmov. Pri generiranju možnih rešitev sta uporabila sledeče strategije: spreminjanje časa postanka, prilagajanje hitrosti vlakov in spreminjane vrstnega reda vlakov. Izhodiščne rešitve sta generirala z metodo razveji in omeji (ang. *branch and bound*), le-te pa sta v nadaljevanju iteracijsko optimizirala z uporabo genetskih algoritmov. Uporabnost predlaganega pristopa sta prikazala na relativno veliki mreži (enotirne in dvotirne proge), ki sta jo modelirala z uporabo PETRI-mrež. S predlaganim algoritmom sta reševala problem določitve nove trase za dodani vlak ter problem optimizacije časovnega replaniranja vlakov ob nastanku zamud. V kriterijski funkciji sta upoštevala vsoto zamud vseh vlakov na vseh postajah in kazen zaradi spremembe postajnega tira in zamujenega prestopanja. Optimizirani scenarij sta primerjala s scenarijem, pri katerem bi vsak vlak imel svoj tir in med vlaki ne bi prišlo do medsebojnega vpliva.

Ekspertni (ang. *knowledge-based*) sistem za upravljanje in nadzor tovornega vlakovnega prometa, ki so ga razvili Tazoniero in sodelavci (2005; 2007), sestoji iz treh modulov. V diskretnem simulacijskem modulu so vgrajene operativne omejitve problema (npr. samo en vlak na odseku) in praktične omejitve problema (npr. polnjenje vlakov z gorivom na samo določenih postajah). V modulu za optimizacijo trajektorij se z metodo mešanega linearnega programiranja v kombinaciji z mehkim genetskim algoritmom preračunavajo hitrosti vlakov s ciljem, da je dejanska trajektorija čim bliže trajektoriji, predvideni v voznem redu. V kolikor optimalne rešitve zaradi sprememb stanja na omrežju naletijo na ovire, se dodatni ukrepi definirajo v modulu strateškega odločanja, kjer so z uporabo mehke logike vgrajena hevristična dispečerska pravila (spremembe prioritet in hitrosti vlakov). Optimum predstavljajo minimalne zamude vseh vlakov.

Norio in sodelavci (2005) za reševanje problema replaniranja vlakov na hitri progi predlagajo uporabo algoritma, ki združuje PERT-metodo mrežnega planiranja in optimizacijsko metodo simuliranega ohlajanja (ang. *Simulated Annealing*). V algoritmu upoštevajo različne korekcijske akcije, ki jih ima dispečer na voljo (ukinitev vlaka, sprememba tirov, sprememba časa odhodov s postaj …). Uspešnost replaniranja so ocenjevali z vidika potnikov.

Hara in sodelavci (2006) so prav tako uporabili PERT-tehniko planiranja. Za merjenje učinkovitosti metode predlagajo izračun stroškov potnikov, v katerih so zajeti potovalni čas,

prestopanje in gneča. Avtorji so predstavili tri različne pristope k reševanju replaniranja voženj vlakov. V prvem primeru so preverili učinkovitost možnosti spremembe tira, po katerem vozi vlak. V drugem pristopu so predlagali preveritev možnosti spremembe postankov; predlagajo preveritev postaj, kjer je možnost, da se vlak ustavi (ob tiru mora biti peron in postajni tir mora biti ustrezne dolžine), in na podlagi tega določijo nove (dodatne) postanke vlakov. In tretjič, predlagajo upoštevanje dejstva, da se zaradi zamude vlaka poveča število potnikov na postaji, ki čakajo vlak. Povečano število potnikov pa se odraža v daljšem času, potrebnem za vstop potnikov, zato predlagajo uporabo metode, pri kateri vlakom namerno podaljšajo postanek, da ohranijo interval med vlaki. Iz rezultatov raziskave je razvidno, da se zaradi zamude enega vlaka stroški potnikov povečajo in da se z replaniranjem, pri katerem ima dispečer na voljo vse tri možnosti (spremembo voznega tira, dodatni postanek in ohranjanje intervala med vlaki), ti stroški lahko zmanjšajo. V kasnejši raziskavi (Kumazawa, Hara in Koseki, 2010) so izboljšali natančnost kriterijske finkcije in dodali čas, ko potniki čakajo na peronu. Predlagajo dvostopenjski pristop, in sicer najprej replaniranje voženj vlakov in s tem določitev prihodov in odhodov vlakov s postaj, nato pa vrednotenje voznega reda glede na ocene in simulacije vedenja potnikov.

Geske (2006) je za optimizacijo replaniranja vlakov uporabil kombinacijo deklarativnega programiranja z omejitvami (ang. *constraint programming*) ter genetske algoritme za iskanje lokalnega optimuma – simulirano ohlajanje, optimizacijo s kolonijami mravelj (ang. *ant algorithms*) in *flood algorithm*. Kombinacijo je uporabil zaradi velikosti obravnavanega omrežja. Kompleksnost reševanja problema optimizacije replaniranja je zmanjšal z delitvijo obravnavanega omrežja na manjša, približno enako velika območja. S programiranjem z omejitvami je omejil prostor možnih rešitev, genetske algoritme pa je vključil v reševanje lokalnih konfliktov vlakov na postajah in minimiziranje skupnih zamud vlakov.

Gély in sodelavci (2006) so v modelu za replaniranje vlakov z numeričnimi spremenljivkami opisali prihode in odhode vseh vlakov na postajah (poimenovali so jih *vozlišča*) in z binarnimi spremenljivkami vrstni red vlakov skozi vozlišča (ali gre vlak pred naslednjim ali ne), izbiro tirov (ali vlak uporabi določen tir ali ne) ter dodatne postanke (ali se vlak ustavi na postaji, kjer po voznem redu ni imel postanka, ali ne). Cilj njihovega modela so minimalne skupne zamude vseh vlakov na vseh postajah. Za optimizacijo so uporabili hibridni algoritem, sestavljen iz uporabe genetskega algoritma za pridobitev suboptimalne izvedljive rešitve, le-to pa so v nadaljevanju optimizirali z uporabo optimizacijskega okolja IBM ILOG CPLEX.

Rodriguez (2007) se je v članku omejil na replaniranje voženj vlakov v območju vozlišč, saj le-ta pogosto predstavljajo ozka grla, kjer primarna zamuda vlaka v velikosti nekaj sekund lahko vodi v sekundarne zamude v velikosti nekaj minut. Ker ima dispečer od trenutka, ko

zazna zamudo vlaka, do trenutka, ko vlak vstopi na območje križišča, le malo časa, avtor postavi pogoj, da mora program omogočati izračun (skoraj) optimalne rešitve v treh minutah. Za zaznavanje konfliktov med vlaki uporabi model sestave posamične obdelave (ang. *job-shop model*) z dodatnimi omejitvami virov. Optimizacijo replaniranja z upoštevanjem zaznanih konfliktov izvede z uporabo metod razveji in omeji ter tehniko programiranja z omejitvami.

Avtorici Mladenović in Čangalović (2007) obravnavata problem optimizacije replaniranja prometa na enotirni progi kot problem sestave posamične obdelave (ang. *job-shop problem*). Za optimizacijski algoritem uporabita programiranje z omejitvami. Uspešnost modela sta preverili za sedem različnih kriterijskih funkcij, in sicer minimalne vrednosti maksimalnih zamud, utežene maksimalne zamude, skupne zamude, utežene skupne zamude, maksimalno odstopanje od predvidenih postankov na postajah, trajanje potovanj med postajama ter število zamujenih vlakov. Za zmanjšanje prostora možnih rešitev in s tem pospešeno iskanje optimuma sta uporabili omejevanje z uporabo hevrističnih pravil. Optimizacije replaniranja so bile izvedene z uporabo komercialnega orodja ILOG Solver CP.

Luethi in sodelavci (2007) so predstavili koncept dinamičnega posodabljanja voznih redov vlakov. Vozni red posodabljajo glede na lokacije in hitrosti vlakov vsako sekundo. Testni primer je bil vzpostavljen v mikroskopsko simulacijskem okolju programa OpenTrack. Obravnavano omrežje so razdelili na območja zgostitve in območja kompenzacij, kjer je kapaciteta večja oz. je izkoriščenost kapacitete nižja. Cilj optimizacije je maksimizacija prometnega toka skozi območja zgostitve, preko katerih morajo vlaki voziti z optimalno hitrostjo, brez nepotrebnega ustavljanja pred signali in z minimalnim časom med zaporednima vlakoma.

D'Ariano in sodelavci (2007) vlakovno vožnjo obravnavajo kot obdelavo izdelka (ang. *job*) v modelu sestave posamične obdelave (ang. *job-shop model*), kjer odseki proge predstavljajo stroje, vlaki pa izdelke. Model problema optimizacije so izdelali z uporabo alternativnih grafov. Z uporabo algoritma razveji in omeji so omejili prostor spremenljivk in tako pospešili izračun optimalne odločitve. Kriterij optimalnosti so minimalne skupne zamude. Kasneje so model nadgradili z upoštevanjem podatkov o dejanskih lokacijah in hitrosti vlakov na začetku vsakega odseka in možnostjo spreminjanja njihove hitrosti, kar pomeni, da so v modelu poleg zelenega (»prosta pot«) in rdečega (»stoj«) upoštevali tudi rumeni signal, ki pomeni dovoljeno vožnjo z zmanjšano hitrostjo (D'Ariano, Pranzo in Hansen, 2007). Tako so zmanjšali časovni razmik med zaporednima vlakoma in posledično dodatno zmanjšali skupne zamude. V zaključku so poudarili pomembnost natančnega in detajlnega modeliranja infrastrukture (dolžine odsekov, omejitve hitrosti, vzponov …) in vlakov (hitrosti, pospeška in

pojemka, lokacije …). Poleg možnosti časovnega replaniranja vlakov v primeru nastanka zamud so kasneje v model optimizacije vpeljali še možnost izbire obvozne poti (D'Ariano et al., 2007). Z modelom so preverili smiselnost uvedbe fleksibilnega voznega reda. Za razliko od običajnega (togega) voznega reda, kjer so podani točni termini prihodov in odhodov vlakov, so v njem podani maksimalni časi prihodov in minimalni časi odhodov. Rezultati so pokazali, da so v primeru uporabe fleksibilnega voznega reda skupne zamude manjše (D'Ariano et al., 2007; D'Ariano, Pacciarelli in Pranzo, 2008). Z izboljšavami v algoritmu so dosegli krajši čas računanja (Corman et al., 2010a). V nadaljnjih raziskavah so primerjali uspešnost centraliziranega in decentraliziranega pristopa replaniranja. Centralizirani pristop je uspešnejši pri upoštevanju krajšega časovnega okna in manjših zamud, v primeru upoštevanja večjega časovnega okna in večjih zamud pa priporočajo uporabo decentraliziranega pristopa (Corman et al., 2011).

Kuster in sodelavci (2008) so namesto takojšnje optimizacije na večjem omrežju iskali možnosti za uspešno reševanje problema na lokalni ravni. Predlagajo, da se pri reševanju problema replaniranja voženj vlakov najprej upošteva relativno majhno časovno okno, ki se ga v naslednjih iteracijah povečuje in se tako zagotovi globalna optimalna rešitev. Velikost začetnega okna se v začetni fazi prilagodi velikosti zamude, nato pa se velikost časovnega okna povečuje v odvisnosti od časa, ki ga ima dispečer na voljo – prej zazna zamudo, več časa ima za optimiranje. Z raziskavo so pokazali, da je njihov pristop uspešen in zaradi majhne časovne zahtevnosti nakazuje možnost uporabe v realnih situacijah.

V prvem v nizu člankov avtorice Törnquist (Törnquist in Davidsson, 2002) je predstavljen simulacijski model za napoved ocenjenega časa prihoda vlakov. V model so poleg kapacitetnih omejitev vključeni tudi medsebojni vplivi odločitev upravljavca omrežja in različnih prevoznikov. Za slednje je uporabljen večagentni simulator nosilcev odločanja in njihovih medsebojnih pogajanj. Simulacijski model je bil uporabljen v eksperimentalni študiji homogenega prometa na enotirni progi (Törnquist in Persson, 2005). Avtorja za replaniranje voženj vlakov predlagata iteracijski dvonivojski postopek. Zgornji nivo upravlja razpored prehitevanj in križanj, spodnji nivo pa določa čas zasedenosti posameznega odseka s posameznim vlakom glede na zaporedje vlakov, določeno na zgornjem nivoju. Na spodnjem nivoju predlagane rešitve se nadalje optimizirajo na zgornjem nivoju. Optimizacija temelji na hevristiki, in sicer predlagata uporabo metod tabu iskanje (ang. *Tabu Search*) in simulirano ohlajanje (ang. *Simulated Annealing*). Kriterij optimalnosti so minimalni skupni stroški zamud z vidika uporabnikov, kjer so upoštevani naslednji parametri: velikost zamude vlakov na vsaki postaji, število potnikov, ki jih na posamezni postaji zamuda prizadene, in stroški zamujenih povezav. V naslednji študiji (Törnquist in Persson, 2007) sta avtorja obravnavala bolj

kompleksen problem – heterogeni promet, večje omrežje in večje število vlakov kot v prvi študiji. Problem sta formulirala kot MILP (ang. *Mixed Integer Linear Program*) in predlagala uvedbo strategij, ki zmanjšajo število binarnih spremenljivk, ki na račun kvalitete rešitve (rešitev je običajno suboptimalna) pohitrijo iskanje rešitve. Predlagala in preverila sta učinkovitost štirih različnih strategij za zmanjšanje prostora možnih rešitev. V strategijah sta omejila spremembo vrstnega reda vlakov in/ali spremembo tira, pri čemer v prvi strategiji ne dovolita sprememb v vrstnem redu vlakov, dovolita pa spremembo tira; v četrti pa dovolita vse izvedljive spremembe vrstnega reda vlakov in spremembe tira. Za reševanje problema uporabita algoritem razveji in omeji. Avtorja ugotavljata, da je rešitev, pridobljena z zadnjo strategijo, sicer optimalna, vendar je časovno zahtevna, zato predlagata uporabo strategije, v kateri je število sprememb vrstnega reda vlakov odvisno od karakteristik zamud, saj so rezultati pokazali, da obstaja povezava med karakteristikami zamud in hitrostjo konvergence k optimalni rešitvi. Kasneje je Törnquist (2007) raziskovala ključne spremembe, ki vodijo k optimalni rešitvi za različne velikosti zamud, zmanjšanju števila spremenljivk in časovne zahtevnosti. Z raziskavo je dokazala, da v predlagani formulaciji problema izbira kriterijske funkcije ne vpliva na čas izračuna ter da kljub omejitvi časovnega okna lahko dobimo rešitev, ki je ustrezna dolgoročno (vpliv na vlake izven časovnega okna je majhen). V naslednji raziskavi (Törnquist Krasemann, 2012) je dokazala, da se z uporabo požrešnega algoritma namesto algoritma razveji in omeji čas, potreben za iskanje rešitve v prej obravnavanih konfiguracijah železniškega prometa in zamud ter prej obravnavani formulaciji problema, skrajša.

Acuna-Agost in sodelavci (2011) so problem replaniranja voženj vlakov modelirali kot MIP. V primerjavi z drugimi avtorji, ki problem formulirajo kot MIL (npr. Törnquist, 2007), avtorji natančneje upoštevajo gibanje vlaka (upoštevajo spremembe potovalnih časov zaradi zaviranja/pospeševanja vlakov) in omogočajo modeliranje železniške infrastrukture s prostorskimi odseki (upoštevajo, da je lahko na medpostajnem odseku, razdeljenem na več prostorskih odsekov, več vlakov). V kriteriju uspešnosti upoštevajo stroške zamud in stroške zaradi spremembe tira/perona na postaji.

Min in sodelavci (2011) so problem replaniranja voženj vlakov prav tako modelirali kot MIP. V algoritmu upoštevajo hevristiko, ki je kombinacija dveh pristopov. Predlagajo, da se najprej oštevilči nekaj zaporedij vlakov, nato se s funkcijo, ki predvidi, kako se bodo razrešili konflikti med vlaki, izbere najbolj ustrezno zaporedje. Z eksperimenti so dokazali, da je njihov pristop uspešnejši od Törnquistinega in Perssonovega pristopa (Törnquist in Persson, 2007).

Tudi Dotoli in sodelavci (2013) so problem replaniranja voženj vlakov modelirali kot MIP, vendar so se omejili na enotirne proge. Predlagajo uporabo hevrističnega algoritma, s katerim iterativno odpravljajo konflikte, ki nastanejo znotraj izbranega časovnega okna.

Narayanaswami in Rangarajin (2013) predlagata, da se pri modeliranju problema replaniranja voženj vlakov kot MIL-zamude in omejitve replaniranja vključijo že v sam model. Prednost takšnega pristopa je, da se replanirajo samo zamujeni vlaki, ostali pa sledijo začetnemu voznemu redu.

Pellegrini in sodelavci (2014) so predlagali modeliranje problema kot MILP za vodenje vlakov tako v prostorskem kot tudi v gibljivem prostorskem razmiku. Na primerih vozlišč so pokazali, da z upoštevanjem gibljivega prostorskega razmika hitreje omejijo domino efekt širjenja zamud. V kriteriju optimalnosti upoštevajo skupne zamude ali največjo zamudo vlaka.

Louwerse in Huisman (2014) obravnavata primere motenj, ki povzročajo zamude vlakov, večje od ene ure. Z uporabo celoštevilčnega programiranja optimizirajo nivo storitev z vidika potnikov, in sicer tako, da se določijo vlaki, ki se izločijo iz voznega reda; za vlake, ki ostanejo, pa se določi nov vozni red z upoštevanjem omejitev kapacitete železniške infrastrukture.

Zhan in sodelavci (2015) so problem prav tako modelirali kot MILP in za primer popolne zapore hitre proge optimizirali utežene skupne zamude in število ukinjenih vlakov, in sicer z določitvijo postaj, na katerih se ustavijo vlaki, določitvijo vrstnega reda odhodov vlakov, da se vpliv zamude čim prej izniči, ter določitvijo vlakov, ki se ukinejo. S testi na realnem primeru hitre proge Peking–Šanghaj so dokazali, da je njihov pristop uspešnejši od strategije »kdor prvi pride, prvi gre«.

Sajedinejad in sodelavci (2011) problem replaniranja voženj vlakov obravnavajo s simuliranjem na osnovi diskretnih dogodkov. Del njihovega simulacijskega orodja predstavlja optimizacijski modul, ki predlaga replanirane vozne rede na osnovi genetskih algoritmov.

Xu in sodelavci (2015) so s simuliranjem na osnovi diskretnih dogodkov preverili uspešnost različnih strategij replaniranja voženj vlakov na dvotirnih progah, in sicer strategijo, da vlaki vozijo samo po pravem tiru, ter strategijo, da hitrejši vlak, ujet za počasnejšim, le-tega prehiti po nepravem tiru. Z eksperimenti so dokazali, da predlagana implementacija pravila prehitevanja po nepravem tiru za skoraj polovico zmanjša zamude v primerjavi s pristopom, kjer se prilagajajo le odhodi vlakov na postajah.

Caimi in sodelavci (2012) so razvili binarni linearni optimizacijski model za pomoč pri odločanju dispečerjev. Avtorji v prispevku ugotavljajo, da dispečerji pri svojem delu

upoštevajo samo prihode vlakov v naslednjih 20 minutah, saj dlje, kot je predviden prihod vlaka, večja je verjetnost zamude, in večje, kot je časovno okno, večja je kompleksnost problema. Avtorji problem formulirajo kot kombinatorični problem z omejitvami, saj z upoštevanjem stopničastega grafikona zasedenosti prostorskih odsekov v diskretnih časovnih korakih določajo različne vlakovne poti. Vlakovno pot izberejo z upoštevanjem kriterija zadovoljstva potnikov, ki ga merijo s točnostjo in zanesljivostjo storitev. Predlagani pristop so ovrednotili na primeru postaje Berne v Švici.

Fan in sodelavci (2012) so za štiri različne scenarije zamud preverili uspešnost osmih algoritmov, in sicer so to: surova sila (ang. *brute force*), kdor prvi pride, je prvi postrežen (ang. *First-Come-First-Served – FIFO*), tabu iskanje (ang. *tabu search*), simulirano ohlajanje (ang. *simulated annealing*), genetski algoritmi (ang. *genetic algorithms*), optimizacija s kolonijami mravelj (ang. *ant colony optimization*), dinamično programiranje (ang. *dynamic programming*) in izločanje z odločitvenimi drevesi (ang. *decision tree based elimination*). Z eksperimenti na območju vozlišča so dokazali, da motnje zaradi zamude enega vlaka uspešno rešujejo z enostavnimi pristopi, kot je FIFO. V primeru kompleksnejših scenarijev pa z uporabo algoritma optimizacije s kolonijami mravelj in genetskimi algoritmi dosežejo za približno 30 % boljše rezultate kot z algoritmom FIFO.

Jesperson in sodelavci (2009) so predstavili problem dinamičnega replaniranja voznega reda potniških vlakov, voznih sredstev in posadke na makroskopskem nivoju v primeru večjih motenj. V modelu strukturirajo pravila odločanja in povežejo vloge različnih udeležencev pri organizaciji, vodenju in upravljanju potniškega železniškega prometa v proces upravljanja odstopanj. To jim omogoča okvirno replaniranje vseh treh področij na nacionalni ravni, vendar pa njihov pristop ni primeren za natančno replaniranje voženj vlakov, saj ne upoštevajo npr. kapacitete postaj in odprtih prog.

Vsem objavam je skupno, da obravnavajo reševanje konfliktov, ki nastanejo zaradi zamude vlaka, in ocenjujejo učinkovitost možnih rešitev, vendar pa se razlikujejo v obsegu oz. kompleksnosti (tako časovni kot prostorski), načinu modeliranja in matematični formulaciji problema, predvidenih oz. dovoljenih korekcijskih akcijah ter v uporabljenih kriterijih optimalnosti.

Kompleksnost optimizacije replaniranja vlakov narašča z naraščanjem števila vlakov pa tudi s povečevanjem obravnavanega območja, zato pristope delimo v dve skupini, in sicer na tiste, ki obravnavajo večje število postaj in vlakov na omrežju (Acuna-Agost et al., 2011; Dollevoet et al., 2012; Kecman et al., 2012), in na tiste, pri katerih se replanira manjše število

vlakov na zelo lokalnem območju (Caimi et al., 2012; Fay, 2000; Fan et al., 2012; Ke-Ping, 2010).

Naslednja delitev je delitev glede na nivo obravnave vožnje vlaka. Avtorji upoštevajo konstantno hitrost vlaka, običajno V_{max} (Ke-Ping, 2010) ali pa upoštevajo tudi pospeševanje in zaviranje vlaka (Acuna-Agost et al., 2011; Medanic in Dorfman, 2002b).

Avtorji v svojih raziskavah uporabljajo namenska programska orodja, namenjena modeliranju železniškega omrežja in simulaciji prometa, kot so npr. OpenTrack (Luethi et al., 2007), splošna simulacijska orodja, npr. na osnovi PETRI-mrež (Wegele in Schnieder, 2004), PERT-metode (Hara et al., 2006; Norio et al., 2005), ali pa uporabljajo modeliranje v okviru programskih rutin (Caimi, 2012).

Naslednja delitev pristopov replaniranja voženj vlakov je glede na matematično formulacijo problema. Avtorji najpogosteje problem modelirajo kot sestav posamične obdelave (ang. *job-shop problem*) z različnimi omejitvami (Corman et al., 2011; D'Ariano, 2008; Olivera, 2001). Poleg najbolj poznanega in najpogosteje uporabljenega linearnega programiranja (ang. *linear mixed programming*) raziskovalci predlagajo uporabo hevrističnih pristopov, kot je razveji in omeji (D'Ariano et al., 2007; Wegele in Schnieder, 2004), uporabo genetskih algoritmov (Ping et al., 2001), tabu iskanja (Corman et al., 2010a; Ho in Yeung, 2001b), *tree search* (Acuna-Agost et al., 2011; Chiu et al., 2002), simuliranega ohlajanja (Ho in Yeung, 2001b; Norio et al., 2005; Törnquist in Persson, 2005) in optimizacijo s kolonijami mravelj (Geske, 2006), s hevristikami (Fay, 2000; Sajedinejad et al., 2011; Tazoniero et al., 2007), genetskih algoritmov (Ping et al., 2001), programiranje z omejitvami (Olivera, 2001; Geske, 2006; Rodriguez, 2007).

Pristope lahko delimo tudi glede na kriterij optimalnosti, saj avtorji upoštevajo skupne zamude vseh vlakov (Corman, 2010b; Gély et al., 2006), utežene zamude vlakov (Fan, Roberts in Weston, 2012) ali število zamujenih vlakov (Törnquist, 2007). V literaturi lahko zasledimo primere, kjer avtorji optimizirajo problem z vidika potnikov, zato pri kriteriju optimalnosti poleg zamud upoštevajo tudi stroške zaradi spremembe perona, spremembe vzorcev postankov, zamujene povezave in odpovedi vlakov (Acuna-Agost et al., 2011; Caimi et al., 2012; Hara et al., 2006; Nagasaki et al., 2003; Norio et al., 2005) ali pa z vidika potnikov in upravljavca, kjer v kriteriju uspešnosti upoštevajo velikost zamud in porabo energije (Medanic in Dorfman, 2002a; Ghoseiri et al., 2004).

V dostopni literaturi ni bibliografske enote, ki bi problem časovnega replaniranja voženj vlakov obravnavala z uporabo spodbujevanega učenja.

3 SISTEM ZA POMOČ PRI ČASOVNEM NAČRTOVANJU ŽELEZNIŠKEGA PROMETA

V predhodnih poglavjih smo nakazali kompleksnost načrtovanja in replaniranja voženj vlakov in s tem potrebo po učinkovitem sistemu za pomoč dispečerjem. V tem poglavju sta predstavljena spodbujevano učenje, predvsem njegove značilnosti, ki nakazujejo uporabnost pri replaniranju voženj vlakov, ter formulacija problema in implementacijskih detajlov algoritma, ki temelji na principu učenja Q.

Razvoj agentnih modelov (ang. *agent-based models*) omogoča razvoj novih pristopov, ki so učinkoviti pri reševanju kompleksnih nalog, torej verjetno tudi za reševanje optimizacijskega problema replaniranja vlakov na večjem železniškem omrežju. V agentnih modelih je agent entiteta, ki se samostojno odloča o akcijah, potrebnih za doseganje cilja. Osnovna ideja agentnih modelov je opisati sistem z njegovimi komponentami, kar omogoča modeliranje poljubno velikih sistemov, pri katerih agent znanje nadgrajuje in prilagaja izbiro akcij glede na stanje okolja. Ravno dejstvo, da ima agent sposobnost učenja in s tem prilagajanja novim situacijam, je razlog, da smo v doktorski disertaciji preverili uporabnost agentnega modela, natančneje spodbujevanega učenja, za reševanje problema časovnega replaniranja voženj vlakov.

3.1 Spodbujevano učenje

Princip umetne inteligence zaradi splošnosti ponuja širok spekter uporabe in ponuja priložnost za reševanje problemov, ki so do sedaj veljali za nerešljive, ter raziskovanje novih področij in s tem prispevek k znanosti. Uporaba umetne inteligence je že na začetku podala mnogo pomembnih in spodbudnih rezultatov pri reševanju kompleksnih problemov, tudi na področju transportnih sistemov.

Izziv na področju uporabe umetne inteligence je, kako izboljšati sposobnosti agentove inteligence. Eden izmed mehanizmov je, podobno kot pri ljudeh, učenje. Učenje ne pomeni pomnjenja niza znakov (učenje na pamet), temveč pridobivanje znanj iz izkušenj ali z iskanjem pravil v podatkih. Učenje je lahko nadzorovano (ang. *supervised learning*), nenadzorovano (ang. *unsupervised learning*) ter spodbujevano (ang. *reinforcement learning*). Nadzorovano učenje je princip, kjer agent na osnovi učne množice generalizira znanje (s podanih primerov izlušči zakonitosti, jih posploši na poljuben vzorec in odgovori na vprašanje, ki ga še ni videl). Ta vrsta učenja se uporablja predvsem za razvrščanje vzorcev v kategorije ter za iskanje relacij med spremenljivkami. Nenadzorovano učenje je princip, kjer agent podatkom išče skrite strukture in pravila ali pa jih razvršča v gruče. Nenadzorovano

učenje agenta ne pozna principa dodeljevanja nagrad, zato tudi ne informacije o uspešnosti. Z učenjem agenta optimalnih akcij v danem okolju se ukvarja spodbujevano učenje. Nasprotno od nadzorovanega učenja, kjer učenje poteka na osnovi baze znanja, ki jo pripravi učitelj, spodbujevano učenje zahteva, da se agent uči neposredno v interakciji z okoljem in reagira (izbere akcijo) glede na spodbudo, ki jo je prejel iz okolja.

Spodbujevano učenje združuje dve disciplini, dinamično programiranje in nadzorovano učenje, in omogoča reševanje problemov, ki jih disciplini ločeno ne uspeta rešiti. Uporabnost dinamičnega programiranja omejuje velikost in kompleksnost problema, uporabnost nadzorovanega učenja pa priprava ustreznih učnih primerov. Replaniranje vlakov sodi v razred NP-problemov, kar nakazuje, da dinamično programiranje brez poenostavitev ni primeren pristop. Zaradi kompleksnosti problema časovnega replaniranja vlakov optimalen odgovor v dani situaciji pogosto ni poznan, hkrati pa je težko predvideti vse situacije, ki se lahko zgodijo. Tudi najboljši dispečerji so lahko slabi učitelji, saj ni nujno, da vedno poznajo najboljše akcije, hkrati pa je prostor možnih rešitev tako velik, da je težko zagotoviti dovolj velik učni vzorec (Zou, Xu in Zhu, 2006). Torej, tudi nadzorovano učenje ni primerno, saj učenje, ki temelji na slabih učnih vzorcih, ni uspešno.

Spodbujevano učenje opredeljuje načelo dodeljevanja spodbud, ki jih agent prejema v procesu raziskovanja okolja, agent pa mora biti sposoben informacije, ki jih prejme iz okolja, interpretirati. Okolje v spodbujevanem učenju mora biti dinamično, to pomeni, da mora akcija, ki jo agent izvede, voditi v spremembo stanja okolja. Glede na novo stanje okolja agent prejme spodbudo (nagrado) za izbrano akcijo. Spodbuda, ki jo agent prejme iz okolja, agentu pove, kako uspešna je bila njegova akcija pri doseganju zastavljenega cilja. Agent v procesu učenja spreminja akcije (obnašanje) s ciljem, da se nauči optimalne strategije – torej, po kateri poti naj gre iz začetnega do končnega stanja, da bo prejel čim višjo nagrado (Sutton in Barto, 1998).

Pri spodbujevanem učenju je pomembno poudariti, da se agent odloča na osnovi trenutnega stanja, ki povzema vse pomembne informacije o sekvenci predhodnih stanj. Pri tem ni nujno, da so v informaciji o trenutnem stanju ohranjene vse informacije o predhodnih stanjih, temveč samo tiste informacije, ki so pomembne za prihodnja stanja. Torej, okolje mora imeti lastnost Markova, da agent lahko predvidi bodoče stanje glede na trenutno stanje in izbrano akcijo v tem stanju.

3.2 Učenje Q

Watkins (1989) je prvi predstavil algoritem učenja Q (ang. *Q learning algorithm*), ki je trenutno najpogosteje uporabljeni algoritem spodbujevanega učenja. Bistvo algoritma je iterativno spreminjanje matrike $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ glede na izbrano akcijo a_t v trenutnem stanju $\mathrm{s_t}$. Osnovni elementi spodbujevanega učenja so: agent, akcija, stanje okolja ter nagrada; definirajo se za vsak problem posebej. V nadaljevanju so opisane bistvene lastnosti osnovnih elementov, ki so za lažje razumevanje predstavljene na primeru igre šaha.

Agent je entiteta, odgovorna za izbor akcije a_t v času t, za interpretacijo spodbud, ki jih dobi iz okolja, ter za učenje optimalne strategije. Agent v vsakem stanju iz množice možnih akcij izbere akcijo a_t, ki vodi v spremembo stanja okolja iz s_t v s_{t+1}. Agent za izvedeno akcijo iz okolja dobi spodbudo (nagrado) $r(s_{t+1}, a_{t+1})$, ki opisuje uspešnost izvedene akcije glede na spremembo stanja okolja po izvedeni akciji. Cilj agenta je določiti optimalno pot od začetnega do končnega stanja z opazovanjem spodbud, ki jih prejme iz okolja v nizu izvedenih akcij. V procesu učenja agent posodablja matriko $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ z raziskovanjem različnih akcij, ki jih izvede v vsakem stanju. Agent nima spomina, da bi v vsakem trenutku vedel, kateri niz akcij je bil najbolj uspešen. Z raziskovanjem novih akcij si lahko znanje tudi poslabša. V igri šaha je agent naš soigralec oz. nasprotnik, ki je odgovoren za premikanje svojih figur, o naslednji potezi pa se odloča glede na stanje (postavitev) figur na šahovnici.

Akcije $a_t \in A(s_t)$, med katerimi izbira agent, so vnaprej določene, in sicer tako, da odražajo dovoljene akcije v problemu, ki ga rešujemo. V posameznih stanjih je množica možnih akcij lahko podmnožica vseh možnih akcij, torej so akcije odvisne od trenutnega stanja okolja. V vsakem stanju okolja ima agent običajno možni vsaj dve akciji, med katerima izbira; izbira določene akcije je odvisna od strategije izbire akcije. V igri šaha so akcije dovoljeni premiki figur, torej agent pri izbiri akcij upošteva pravila premikanja za šest različno gibajočih se figur (npr. kmet se premika za eno polje naprej, razen v prvi potezi, ko se lahko za dve polji; skakač se premika v obliki črke L v katerokoli smer in lahko preskakuje figure) in ne bo izbral nedovoljene akcije (npr. ne bo premikal kmeta v obliki črke L). Agent se lahko tudi priuči, katere akcije so nedovoljene, in sicer tako, da mu okolje vrne negativno nagrado. Praviloma se agentu poda množica dovoljenih akcij, med katerimi izbira, saj ni cilj učenja, da se agent nauči, katere akcije niso dovoljene, temveč da se nauči strategije zaporedja akcij, ki vodijo h globalnem optimumu. Agent (igralec šaha) se o naslednji potezi odloča glede na postavitev figur in izbere akcijo (premik figure), za katero že ve, da bo uspešna, ali pa naključno izbere akcijo in tako pridobiva nove izkušnje.

Stanje okolja je skupina spremenljivk, ki skupaj opišejo karakteristike (stanje) okolja, ki so pomembne za posamezni problem. V vsakem diskretnem časovnem koraku $t \in \{0, 1, 2 \ldots\}$ je

agent v novem stanju $s_t \in S$, torej mora biti okolje dinamično, kar pomeni, da se mora stanje okolja po vsaki izvedeni akciji spremeniti ter agentu v različnih stanjih vračati različne vrednosti spodbud. V igri šaha je stanje okolja opisano s položajem vseh figur. Po izvedeni potezi, ko eden izmed igralcev premakne figuro, se stanje okolja spremeni.

Nagrada (spodbuda) $r_{t+1} \in \mathbb{R}$, ki jo agent prejme pri prehodu v novo stanje, definira uspešnost izvedene akcije. Nagrada poleg uspešnosti akcije (ali je bila akcija uspešna ali ne) agentu vrne tudi informacijo, v kolikšni meri je bila akcija uspešna. Nagrada mora biti definirana tako, da agent ne more vplivati nanjo, mora pa mu služiti kot orodje za spremembo akcije. Znotraj posameznega problema lahko definiramo različne funkcije nagrad, zato je izbira funkcije nagrade ena izmed ključnih nalog pri reševanju problema in mora biti prilagojena problemu, ki ga rešujemo z uporabo spodbujevanega učenja. Funkcija nagrade definira cilj agenta, pri čemer moramo poudariti, da ni cilj agenta, da ob vsakem prehodu v novo stanje prejme najvišjo nagrado, temveč da se nauči strategije izbire akcij, ki imajo največjo vsoto nagrad, ki jo agent prejme na poti od začetnega do končnega stanja. Tako se agent v procesu učenja nauči strategije za doseganje optimalne rešitve. V osnovnem principu učenja Q velja, da agent prejme nagrado po vsaki izvedeni akciji, vendar ni nenavadno, da agent dobi nagrado v končnem stanju. Tudi pri igri šaha šteje samo cilj, to je prisiliti nasprotnikovega kralja v položaj, ko le-ta nima možnosti za premik, tega pa tudi ne more preprečiti (šah-mat). Vmesni koraki niso pomembni, saj vmesne akcije in stanja okolja ne povedo, koliko bližje smo cilju oz. zmagi, zato je nagrada (zmaga) znana šele v končnem stanju (v šah-mat poziciji).

Učenje agenta oz. posodabljanje vrednosti $Q(s_t, a_t)$ opišemo z enačbo (1):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) - Q(s_{t,} a_t) \right], \qquad (1)$$

kjer je:

α … stopnja učenja;

γ … faktor diskontiranja nagrade;

r_{t+1} … nagrada, ki jo agent prejme, ko izvede akcijo a_t v stanju s_t;

$\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$ … max. vrednost Q v stanju s_{t+1} po izbiri akcije a_t v stanju s_t;

$Q(s_t, a_t)$ … vrednost Q v stanju s_t.

Parameter α vpliva na stopnjo učenja, velja $0 < \alpha \leq 1$. Vrednost parametra α je lahko konstantna ali pa se med učenjem spreminja. Parameter α ne sme biti enak 0, saj se v tem primeru matrika Q s časom ne spreminja (agent se ničesar ne nauči):

$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + 0 * \left[r_{t+1} + \gamma \max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) - Q(s_{t,} a_t) \right]$, sledi

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t).$$

V primeru $\alpha = 1$ se vrednost $Q(s_t, a_t)$ posodobi le v odvisnosti od nagrade r_{t+1} in (diskontirane) vrednosti $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + 1 * \left[r_{t+1} + \gamma \max_{a(t+1)} Q(s_{t+1}, a_{t+1}) - Q(s_{t,} a_t)\right] \text{ oz.}$$

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \left[r_{t+1} + \gamma \max_{a(t+1)} Q(s_{t+1}, a_{t+1}) - Q(s_{t,} a_t)\right], \text{ sledi}$$

$$Q(s_t, a_t) \leftarrow \left[r_{t+1} + \gamma \max_{a(t+1)} Q(s_{t+1}, a_{t+1})\right].$$

Parameter γ je faktor diskontiranja prihodnjih nagrad, velja $0 \leq \gamma < 1$. Vrednosti, ki so bližje 0, pomenijo, da na novo vrednost $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ močneje vplivajo trenutne nagrade. Bolj kot so vrednosti bližje 1, bolj na novo vrednost $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ vpliva tudi vrednost matrike Q v času $t + 1$. Iz enačbe (1) sledi, da mora biti v primeru neskončnega procesa učenja vrednost parametra γ manjša od 1, da zagotovimo konvergenco k optimalni rešitvi, sicer bi vrednosti Q naraščale proti neskončnosti.

Na začetku učenja se lahko vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ inicializirajo na enako prednastavljeno vrednost za vse pare stanj-akcij (običajno se matrika inicializira na vrednost 0) ali pa se posameznim parom stanj-akcij pripišejo vrednosti, ki favorizirajo posamezne akcije v določenih stanjih. Favoriziranje posameznih akcij ob inicializaciji kriterijske funkcije imenujemo predznanje, ki običajno skrajša proces učenja.

Standardni postopek algoritma učenja Q lahko povzamemo, kot sledi:

1. inicializacija matrike $Q(s_t, a_t)$;
2. stanje okolja $s_t \in S$;
3. izbira in izvedba akcije $a_t \in A(s_t)$; odvisno od izbrane strategije izbire akcije (najpogosteje ε-greedy metoda);
4. stanje okolja se spremeni iz s_t v s_{t+1}, okolje vrne nagrado $r_{t+1} \in \mathbb{R}$;
5. posodobitev matrike $Q(s_t, a_t)$;
6. stanje $s_{t+1} = s_t$;
7. pojdi na 3, dokler ni s_t končno stanje;
8. ponovi korake 2–7 za določeno število epizod.

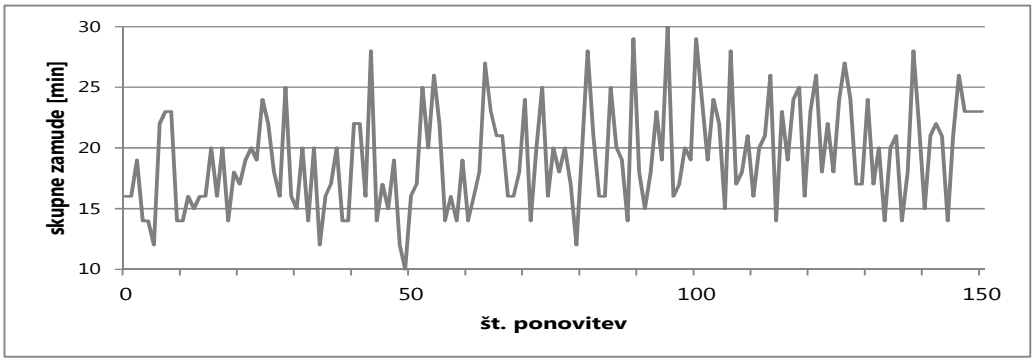
Zaporedje korakov 3–7 predstavlja eno iteracijo*. Iteracija* je vsaka sprememba stanja okolja po izvedeni akciji (pri igri šaha je vsaka poteza, ki jo izvede eden od igralcev, ena iteracija). Enkrat izvedeno zaporedje korakov 2–8 imenujemo epizoda. *Epizoda* je množica vseh iteracij od začetnega do končnega staja (pri igri šaha je ena igra, torej od začetka do zmage enega igralca, ena epizoda). Večkratne ponovitve epizod z namenom, da agent preveri učinkovitost različnih akcij, imenujemo *ponovitve učenja* (pri igri šaha to pomeni, da igro večkrat odigramo).

Omenili smo že, da agent izbere akcijo ter da so v enem stanju možne različne akcije. Torej se postavi vprašanje, katero akcijo agent izbere. V procesu učenja je pomembno, da agent tako raziskuje kot tudi izkorišča že pridobljeno znanje. Agent med procesom učenja izkorišča že pridobljeno znanje, torej izbira takšne akcije v danem stanju, da prejme čim boljšo nagrado in hkrati raziskuje okolje, da pridobi nova znanja in preveri uspešnost nove strategije. S tem agent povečuje možnost, da najde globalni optimum. Učenje Q sodi v skupino t. i. *off-policy* algoritmov, saj agent pridobiva koristne izkušnje, tudi kadar raziskuje in izbira akcije, ki se kasneje izkažejo za neoptimalne. Pri uporabi spodbujevanega učenja je izziv uravnotežiti razmerje med raziskovanjem in izkoriščanjem, saj le-to razmerje vpliva na trajanje učenja in kvaliteto naučene strategije. Po eni strani preveč raziskovanja preprečuje maksimizacijo kratkoročne nagrade (kratkovidnost agenta), ker lahko izbrana »raziskovalna« akcija vodi v negativno spodbudo okolja. Po drugi strani pa izkoriščanje znanja lahko prepreči maksimizacijo dolgoročne nagrade, ker izbrana akcija ni nujno globalno optimalna. V literaturi je uporabljenih več pristopov k uravnoteženju razmerja med raziskovanjem in izkoriščanjem. Najpogosteje uporabljeni pristop za določevanje razmerja med raziskovanjem in izkoriščanjem znanja je ***ε-greedy*** metoda (Sutton in Barto, 1998). Razloga za izbiro pristopa ε-greedy sta dva, in sicer da metoda ne zahteva poznavanja podatkov v zvezi z raziskovanjem, in dejstvo, da s tem pristopom agent najde skoraj optimalno rešitev v veliko primerih uporabe spodbujevanega učenja. Metoda ε-greedy vključuje izbiro akcije z verjetnostjo $(1-\varepsilon)$ za akcije z najvišjo vrednostjo $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$, t. i. požrešne akcije, ter z verjetnostjo ε izbiro naključnih akcij, torej neodvisno od vrednosti matrike Q v času $t+1$. Vrednost $\varepsilon = 0{,}1$ pomeni, da bo agent v 10 % primerov izbral naključno akcijo, med tem ko bo v 90 % med vsemi akcijami $a_t \in A(s_t)$ izbral tisto, ki ima najvišjo vrednost $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$; ***ε-greedy*** metodo strategije izbire akcije opišemo z enačo:

$$p(a_{t+1}|s_{t+1}) = \begin{cases} 1-\varepsilon, a_{t+1} = a_{t+1,\ požrešna};\ \ a_{t+1,\ požrešna} = arg\max_{a_{(t+1)}}(Q(s_{t+1}, a_{t+1})) \\ \varepsilon, a_{t+1} \neq a_{t+1,požrešna} \end{cases} \quad (2)$$

V praksi se pogosto uporabljajo izpeljanke ε-greedy metode, kjer se ε s časom spreminja, npr. ***ε-first*** metoda, kjer v začetnih korakih agent samo raziskuje, nato pa samo izkorišča znanje; *decreasing-**ε** method*, kjer je na začetku relativno visoka stopnja raziskovanja, ki se z vsakim korakom zmanjšuje (Tokic, 2010). Cilj v obeh primerih je, da agent postopoma vedno bolj izkorišča znanje in manj raziskuje, torej izbira tiste kombinacije stanj in akcij, za katere se je naučil, da vrnejo dobre rezultate.

Kot smo že omenili pri opisu agenta, se lahko, zaradi raziskovanja novih akcij, znanje agenta poslabša. V nadaljevanju je prikazana krivulja, iz katere je razviden omenjeni pojav.



Slika 13: Primer krivulje učenja, kjer se znanje agenta poslabšuje
Figure 13: Example of learning curve where agent knowledge worsens

S prikaza Slika 13 je razvidno, da je agent v 50. ponovitvi preizkusil strategijo, ki je vodila v najboljši rezultat (najnižje skupne zamude), vendar se zaradi nadaljnjega raziskovanja okolja in s tem posodabljanja matrike Q znanje agenta slabša, saj krivulja učenja konvergira k vedno višjim vrednostim skupnih zamud. Pojav, da agentu s stalnim spodbujanjem k raziskovanju novih akcij poslabšamo znanje, je logičen, saj se agent odloča na osnovi posodabljanja vednosti matrike Q in nima spomina o tem, katera strategija je vrnila najboljši rezultat. Zato je pri formulaciji problema, ki ga rešujemo z učenjem Q, treba posebno pozornost nameniti nastavitvi vrednosti (funkcij) parametrov, ki vplivajo na učenje $(\alpha,\ \gamma \text{ in } \varepsilon)$, ter številu ponovitev učenja.

3.3 Uporaba metode učenja Q za časovno načrtovanje voženj vlakov

V okviru doktorske disertacije smo si zadali izziv razviti sistem za pomoč pri reševanju problema optimizacije vodenja vlakov po nastanku zamude. Zaradi prednosti v primerjavi z drugimi tehnikami in pristopi, uporabljenimi za replaniranje vlakov, predstavljenih v poglavju 2.5, predlagamo uporabo algoritma, ki temelji na učenju Q. Prednosti algoritma učenja Q, pomembne za reševanje optimizacijskega problema časovnega (re)planiranja, so predstavljene v nadaljevanju.

Glavni motiv je bil razviti algoritem za pomoč pri odločanju dispečerjev, ki bi bil brez prilagoditev uporaben za vsa železniška omrežja.

V literaturi predstavljene tehnike za časovno replaniranje voženj vlakov zahtevajo model okolja in so zato uporabne samo na modelu infrastrukture, za katero je bila pripravljena baza znanja. Fay (2000) v raziskavi ugotavlja, da je za reševanje že manjšega problema treba definirati več kot 100 pravil, ki so odvisna od položajev in hitrosti vlakov ter železniške infrastrukture, hkrati je izpostavil problem, ki nastane zaradi kompleksnosti replaniranja vlakov, in sicer da se rešitve med dispečerji razlikujejo. Podoben problem nastane pri

uporabi pristopa nadzorovanega učenja, kjer je uspešnost agenta odvisna od kvalitete učnega vzorca. Pri reševanju problema časovnega načrtovanja voženj vlakov je optimalna rešitev odvisna od velikega števila spremenljivk, zato je težko pripraviti kvalitetno bazo znanja, na kateri bi agent gradil svoje znanje. Pri spodbujevanem učenju agent ne pozna okolja in z dinamičnim raziskovanjem različnih akcij v danem stanju opazuje spremembe v okolju in namesto na učnih vzorcih znanje gradi na svojih izkušnjah; agent se uči razmerij med stanjem okolja, izvedeno akcijo in nagrado z dinamično interakcijo z okoljem. Z uporabo algoritma učenja Q problem replaniranja voženj vlakov rešujemo univerzalno, saj je edino, kar se spremeni pri uporabi algoritma na različnih železniških infrastrukturah, model okolja v simulatorju, med tem ko algoritem učenja ostaja isti. Tako so v primeru npr. dodatnih tirov ali sprememb dolžin odsekov pri uporabi pristopov, ki temeljijo na modelu okolja, potrebne prilagoditve, pri uporabi metode učenja Q pa ne.

Drugi motiv je bilo vprašanje »Kaj je cilj replaniranja?«. Pod pojmom problem optimiranja se na splošno razume problem, pri katerem se k predhodno dani nalogi najde rešitev, ki pri vnaprej določeni kriterijski funkciji doseže maksimalno (oz. minimalno) vrednost. Za problem replaniranja voženj vlakov ne obstaja definicija kriterijske funkcije, ki bi korektno pokrivala vse interese replaniranega voznega reda. V literaturi zasledimo dve vrsti formulacije kriterijske funkcije, in sicer uspešnost predlaganega pristopa ugotavljajo za enega ali za več kriterijev. Kriterijska funkcija se lahko nanaša na potnike, upravljavca ali oba skupaj. Večina avtorjev učinkovitost pristopa ocenjuje glede na zmožnost zmanjšanja skupnih zamud (D'Ariano et al., 2008; Geske, 2006; Tazoniero et al., 2007; Törnquist Krasemann, 2012). Nagasaki in sodelavci (2003) menijo, da je vrednotenje novega voznega reda glede na zamude vlakov neustrezno, saj so v tem primeru vlaki enakovredni ne glede na število potnikov na vlaku, zato predlagajo uporabo kriterijske funkcije, ki upošteva zamude potnikov, število prestopanj in občutek gneče. Tornquist (2007) je v svoji raziskavi v prvem primeru uporabila kriterijsko funkcijo, ki minimizira skupne zamude prometa (vsoto vseh zamud na končni postaji), v drugem primeru pa kriterijsko funkcijo, ki minimizira stroške zamude in upošteva kazen za vlake, ki so pripeljali pred voznim redom. Wegele in Schnieder (2004) sta rešitev problema ocenjevala glede na strošek zamude in spremembo tira (oz. spremembo perona, ob katerem se vlak ustavi po replaniranem voznem redu). Ghoseiri in sodelavci (2004) ter Medanic in Dorfan (2002a; 2002b) v kriterijski funkciji upoštevajo porabo goriva in potovalni čas (zamude). Algoritem učenja Q omogoča definiranje poljubno velike in poljubno kompleksne kriterijske funkcije ter njeno dinamično spreminjanje in dodajanje kriterijev v kriterijski funkciji. To omogoča enostavno preverjanje učinkovitosti algoritma pri različno zastavljenih ciljih. V nadaljevanju poglavja so opisane komponente učenja Q za časovno (re)planiranje vlakov.

3.3.1 Agent

Po analogiji z vodenjem železniškega prometa v realnem svetu, kjer dispečer odloča o prometu vlakov, smo v algoritmu upoštevali, da se agent odloča o vodenju vlakov. Agent (dispečer) se v našem algoritmu odloča vsako minuto in za vse obravnavane vlake na omrežju.

3.3.2 Okolje

Vlakovni promet opredeljujejo štirje glavni elementi, in sicer železniška infrastruktura (dolžina in število blokovnih odsekov ter dolžina in število postajnih tirov), vlaki (njihove vozne karakteristike, kot so maksimalna hitrost ter pospešek pri zaviranju in speljevanju), vozni red (položaj in hitrost vlakov, določena z voznim redom) ter varnostne in prometno-tehnične zahteve, ki jih je treba zagotavljati pri vodenju železniškega prometa. Vse štiri elemente smo definirali v okolju, v katerem se agent uči. V praksi so ti elementi zelo natančno načrtovani in implementirani, v doktorski disertaciji pa smo sprejeli odločitve o treh poenostavitvah, predstavljenih v nadaljevanju, ki pa ne vplivajo na dokaz o uporabnosti algoritma učenja Q na področju replaniranja voženj vlakov.

Prva poenostavitev se nanaša na modeliranje železniške infrastrukture, in sicer na natančnost modeliranja območja kretnic. Odseki med postajama, torej na odprti progi, imajo relativno enostavno strukturo – en ali več prostorskih odsekov, ki jih omejujejo glavni signali. V modelu železniške infrastrukture, ki ga uporabljamo v doktorski disertaciji, se položaj signalno-varnostnih naprav modelira natančno. Na območju postaj in križišč, kjer je modeliranje železniške infrastrukture zaradi kretnic bolj kompleksno, smo model poenostavili. Prikaz (Slika 14) prikazuje model postaje s tremi postajnimi tiri. Leva stran prikazuje bolj natančno modelirano območje kretnic, saj so upoštevane predpisane razdalje med koncem ene in začetkom druge kretnice. V modelu okolja, ki ga uporabljamo v doktorski disertaciji, območje kretnic upoštevamo poenostavljeno, saj ne upoštevamo segmentov AB in CD. Takšna poenostavitev ne vpliva na rezultat, saj so ti odseki običajno krajši od dolžine vlaka (Ghoseiri et al., 2004).

Slika 14: Poenostavitev modela na območju postaje
Figure 14: Simplification of railway station layout

Druga poenostavitev se nanaša na modeliranje vlaka, in sicer na njegovo dolžino in hitrost. Vlak mora skozi odseke v logičnem vrstnem redu, torej mora vstopiti in zapustiti odsek 2,

preden vstopi in zapusti odsek 3. V realni situaciji je zaradi dolžine vlaka zadnji del le-tega še na odseku 2, medtem ko je sprednji del vlaka že na odseku 3. Tako sta kratek čas zasedena oba odseka. V disertaciji smo predpostavili, da je vlak točkovni objekt, in nismo upoštevali prekrivanja zasedenosti odsekov. Vlaki po progi ali delih proge lahko vozijo z največjo dovoljeno progovno hitrostjo, ki se določi glede na tehnično stanje proge, njeno opremljenost in tehnične značilnosti vlaka (Signalni pravilnik, 2007). V algoritmu upoštevamo, da vlak vozi preko vseh odsekov z enako hitrostjo (hitrost se pripiše vlaku in ne odsekom), zato se čas zasedenosti posameznih odsekov v realnem primeru in v modelu razlikuje.

Tretja poenostavitev se nanaša na signale. S signalnimi znaki se izvršilni delavci obveščajo in sporazumevajo o pogojih vožnje, stanju proge, hitrosti, premiku in nevarnostih na progi (Signalni pravilnik, 2007). Praviloma so to svetlobni signalni znaki, ki se dajejo z zeleno, rumeno, rdečo barvo ali pa kombinacijo dveh barv ter z mirnimi in/ali utripajočimi lučmi (glej poglavje 2). Okolje učenja Q smo poenostavili do te mere, da nas zanima samo, ali je odsek prost – torej, ali lahko vlak zapelje na odsek ali ne, zato sta uporabljena samo signalna znaka »prosto« (zeleni signalni znak) ter »stoj« (rdeči signalni znak). Signalni znaki, ki signalizirajo npr. vožnjo z omejeno hitrostjo, previden uvoz na postajo z 10 km/h ali previden izvoz s postaje z 10 km/h, v modelu okolja niso upoštevani.

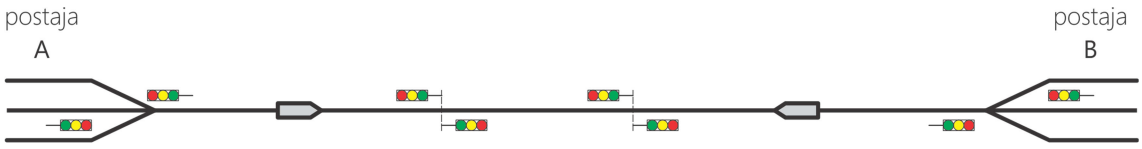
Pri reševanju problemov z učenjem Q lahko v okolju definiramo omejitve, ki so značilne za obravnavani problem. Z omejitvami zmanjšamo število možnih akcij v danem stanju in posledično pohitrimo proces učenja. Cilj algoritma, predstavljenega v doktorski disertaciji, je optimalen in izvedljiv vozni red, torej morajo biti upoštevana pravila, ki izhajajo iz principov vodenja vlakov, opisanih v poglavju 2, določil Prometnega pravilnika (Uradni list RS, št. 50/2011: 6824–6931) in določil Signalnega pravilnika (Uradni list RS, št. 123/2007: 18085–18185). Pravila in omejitve, ki smo jih definirali v okolju, so naslednje:

1. Na enem odseku je lahko samo en vlak

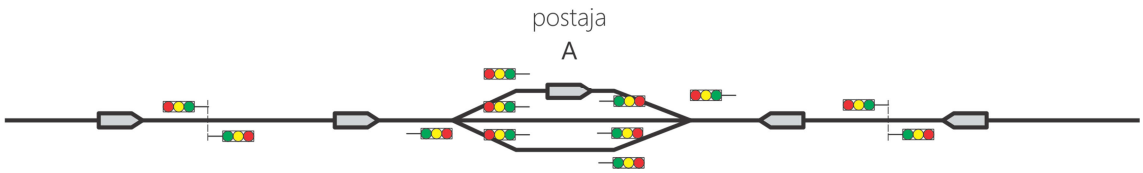
   V poglavju 2 je opisan princip vodenja vlakov, ki velja za večino evropskih železniških omrežij, to je vožnja vlakov v fiksnem prostorskem razmiku, kjer se varnost železniškega prometa zagotavlja s pogojem, da je na odseku lahko samo en vlak. Algoritem učenja Q nam omogoča, da agent prejme negativno nagrado za neželeno akcijo. Tako bi lahko agentu dodelili kazen, če bi dva vlaka hkrati postavil na isti odsek; agent bi se s časom naučil, da ta akcija ne vodi k optimalni strategiji, vendar pa, ker takšne akcije v realnem svetu niso dovoljene, le-te preprečimo z omejitvami v okolju, ki zmanjšujejo množico možnih akcij v nekem stanju.

2. Kontrola brezizhodne situacije

Na prikazu Slika 15 je primer, kjer Vlak 1 potuje s postaje A na postajo B in Vlak 2 v obratni smeri. Ker vlaka potujeta po enotirni progi v različnih smereh, ne smeta hkrati zapeljati s postaje, če vmes ni dodatnih tirov, ki bi omogočili križanje vlakov. Akcija, ki bi dovolila odhod vlakoma 1 in 2, bi vodila v brezizhodno situacijo ali trk vlakov. Na spodnjem prikazu (Slika 16) je prikazan primer, ki zaradi neupoštevanja kapacitete postajnih tirov in pravila, da je na enem postajnem tiru lahko samo en vlak, prav tako vodi v nerešljivo situacijo.



Slika 15: Brezizhodna situacija – primer 1
Figure 15: Dead lock – example 1



Slika 16: Brezizhodna situacija – primer 2
Figure 16: Dead lock – example 2

V okolju učenja Q smo za preprečitev čelnega trka in brezizhodne situacije iz primera 1 (Slika 15) definirali omejitev, da vlak lahko zapusti postajo, če na kateremkoli odseku do naslednje postaje ni vlaka, ki bi vozil v nasprotni smeri. Ta omejitev je skladna z osnovnim principom vodenja vlakov na enotirnih progah, kjer se čelna zaščita zagotavlja z določitvijo smeri privolitve, ki preprečuje, da bi katerikoli glavni signal v nasprotni smeri od privolitve pokazal signalni znak za dovoljeno vožnjo. Privolitev vozne smeri pomeni stanje, ko je naprava tehnično usmerjena v delovanje samo v eni, trenutno izbrani smeri, v drugi smeri pa je privolitev voženj onemogočena.

Nastanek brezizhodne situacije, podane v primeru 2 (Slika 16), je nemogoče matematično opredeliti in preprečiti, zato okolje učenja Q v vsaki iteraciji izvaja kontrolo kapacitete postaj, torej število postajnih tirov primerja s številom vlakov, ki so že na postaji, s številom vlakov, ki se postaji približujejo iz obeh smeri. V primeru, da je kapaciteta postaje presežena, okolje agentu vrne negativno nagrado.

3. Dva ali več vlakov ne sme istočasno zapustiti postaje v isti smeri

   Odločitev o odhodih vlakov s postaj se sprejema vsako minuto za vse vlake hkrati. V kolikor imata vlaka na isti postaji ob isti minuti za cilj isto postajo, bi ob izbiri akcije, ki obema vlakoma hkrati dovoli odhod s postaje, vlaka hkrati zasedla naslednji odsek, kar bi v realnosti pomenilo trk. S pravilom, da lahko v eni smeri samo en vlak zapusti postajo, smo takšne dogodke preprečili. Glede na strategijo izbire akcije se agent odloči o tem, kateri vlak bo prvi zapustil postajo. Če bo agent upošteval že pridobljeno znanje, potem se bo odločil za akcijo z $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$ oz. za poljubno akcijo, če bo raziskoval nove možnosti.

4. Vlak ne sme s postaje pred odhodom, predvidenim v voznem redu

   Dispečerji morajo promet vlakov voditi in izvajati v skladu z voznim redom, torej morajo upoštevati z voznim redom določene odhode vlakov s postaj. Pri urejanju vlakovnega prometa po nastanku zamude se odhod vlaka lahko prestavi (zakasni), nikakor pa odhod vlaka ne sme biti pred odhodom, predvidenim v voznem redu:

$$t_{o\,j,i,VR} \leq t_{o\,j,i,VR_R}, \tag{3}$$

   kjer je:

   $t_{o\,j,i,VR}$ … čas odhoda vlaka j na postaji i po voznem redu;

   $t_{o\,j,i,VR_R}$ … čas odhoda vlaka j na postaji i po replaniranem voznem redu.

5. Vlak se lahko ustavi samo pri glavnih signalih

   Agent se samo na lokacijah glavnih signalov odloča, ali ima vlak dovoljenje za vožnjo ali ne. Za vse vlake, ki so na medpostajnem odseku in niso na lokaciji glavnih prostorskih signalov, je edina možna akcija »nadaljuj z vožnjo«, saj se vlaki lahko ustavljajo samo na lokacijah signalno-varnostnih naprav.

6. Če je glede na vrednost $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$ več akcij enakovrednih in agent izbira akcijo na osnovi že pridobljenega znanja, potem imajo akcije z vsaj enim odhajajočim vlakom prednost pred akcijo, kjer vsi vlaki počakajo.

3.3.3 Stanja okolja

Stanje okolja v spodbujevanem učenju lahko definiramo na različne načine, z vključevanjem različnih parametrov. Množica vseh možnih stanj okolja je običajno definirana kot kartezični produkt vseh parametrov, ki ustrezajo različnim lastnostim opazovanega okolja. Parametri morajo biti izbrani tako, da se stanje okolja dinamično spremeni v vsaki iteraciji, torej po vsaki izvedeni akciji. Za reševanje problema replaniranja voženj vlakov smo pri definiranju možnih stanj okolja upoštevali tri parametre, in sicer: lokacije vlakov (podatek, na katerem odseku je posamezni vlak), proste poti (podatek o tem, ali je naslednji odsek prost ali ne; na enotirni progi še dodatni podatek o tem, ali je kakšen vlak v nasprotni smeri) ter čas (čas v minutah od t_1 do t_T, odvisno od časovnega okna, ki ga upoštevamo). Ti podatki so znani dispečerjem v realnih situacijah, zato je predlagani pristop uporaben za replaniranje vlakov na obstoječih sistemih železniške infrastrukture v Evropski uniji.

Upoštevanje časa v opisu stanja s podaljševanjem časovnega okna povečuje prostor možnih stanj. V primeru končnega števila stanj in akcij je konvergenca matrike Q k optimalni vrednosti zagotovljena (Watkins in Dayan, 1992), zato smo se odločili, da z omejitvijo časovnega okna omejimo prostor možnih stanj in s tem zagotovimo hitrejše konvergiranje znanja k optimalni vrednosti.

V predlaganem pristopu je sistem v končnem stanju ob času T, to je v minuti, ko zadnji vlak prispe na končno postajo. V primeru nastanka brezizhodne situacije, ko vlaki ne morejo prispeti na končno postajo, je to čas, ko agent ugotovi nastanek brezizhodne situacije. Ali pa je to čas maksimalnega časovnega okna, ki ga uporabnik definira za posamezni primer, znotraj katerega se izvaja učenje; s tem se prepreči nerazumno podaljševanje postankov vlakov in omeji število možnih stanj.

Množico vseh možnih stanj, ne glede na to, ali so stanja v času t_i izvedljiva ali ne, npr. v isti minuti ne moreta biti dva vlaka hkrati na istem odseku, zapišemo kot:

$$množica\ stanj\ \{s_t\} = \{\mathrm{L}\}\ \mathrm{x}\ \{\mathrm{P}\}\ \mathrm{x}\ \{\mathrm{čas}\}, \tag{4}$$

kjer pomeni:

L … lokacije vlakov $|\{\mathrm{L}\}| = |\mathrm{X}|^{|N|}$;

P … proste poti vlakov $|\{\mathrm{P}\}| = 2^{|\mathrm{N}|}$;

N … število vlakov;

X … številka odseka.

Množica $\big((2,7,1),(0,1,0),6\big)$ predstavlja stanje v šesti minuti (6), ko je prvi vlak na odseku 2, drugi vlak pa na odseku 7 in tretji vlak na odseku 1 $((2,7,1))$, prvi in tretji vlak ne smeta s postaje, drugi vlak pa prične ali nadaljuje vožnjo $((0,1,0))$.

V primerih, ki jih obravnavamo v nadaljevanju, je število vseh možnih stanj po enačbi (4):

- 2.200 za primer, v katerem obravnavamo dva vlaka na petih odsekih v 22 minutah;
- 106.496 za primer, v katerem obravnavamo tri vlake na osmih odsekih v 26 minutah;
- 3 x 10^{47} za primer, v katerem obravnavamo 26 vlakov na 27 odsekih v 280 minutah.

3.3.4 Akcije

Dispečer se pri vodenju vlakov odloča, kdaj in kateri vlak nadaljuje pot, in svojo odločitev preko signalnih znakov »stoj« (rdeči signalni znak) in »prosto« (zeleni signalni znak) sporoča strojevodjam. Po analogiji z realnim svetom se tudi agent v učenju Q odloča, kateri vlak mora v naslednji minuti ostati pred signalnim znakom in kateri lahko nadaljuje pot. Še enkrat ponovimo, da v realnem svetu dispečer lahko s kombinacijami mirnih in/ali utripajočih luči sporoča poleg »stoj« in »prosto« tudi druge akcije, kot npr. »prosto, pričakuj omejitev hitrosti« ali »previden uvoz na postajo z 10 km/h«. Takšne akcije nimajo večjega vpliva na rezultat replaniranja, zato jih v predlaganem algoritmu ne upoštevamo (glej poglavje 3.3.2). Prav tako ponavljamo, da agent na začetku procesa učenja ne pozna ne modela okolja niti omejitev, ki izhajajo iz problema replaniranja vlakov. Da se agent ne odloča med akcijami, ki so v realnem svetu neizvedljive oz. prepovedane, smo v okolje vnesli varnostne omejitve, značilne za železniški promet (glej poglavje 2 in poglavje 3.3.2), tako se agent odloča samo med akcijami, ki so tudi v realnem svetu izvedljive. Če je vlak na postaji, potem se agent lahko odloča, da vlaku podaljša postanek, kljub temu da ima prosto pot, v kolikor pa je vlak na medpostajnem odseku in ima prosto pot, mora agent izbrati akcijo »nadaljevanje vožnje«, saj se vlak ne sme ustavljati na medpostajnem odseku.

Množico vseh možnih akcij, ne glede na to, ali so v posameznem stanju dovoljene ali ne, zapišemo, kot sledi:

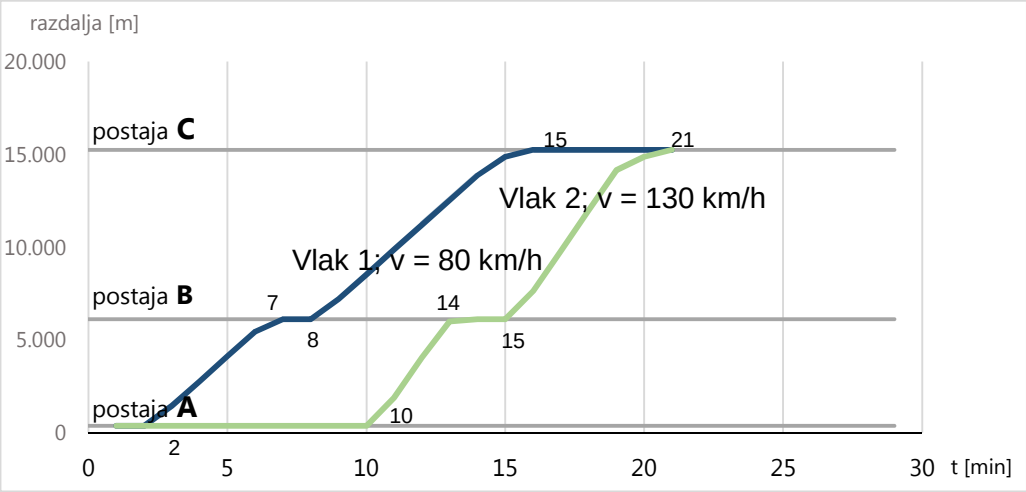
$$množica\ akcij\ \{a_t\} = \{a_{1,t}, a_{2,t}, \dots, a_{n,t}\};\ a_{i,t} = 0\ ali\ 1\ za\ vsak\ vlak\ i\ od\ 1\ do\ n, \tag{5}$$

kjer je:

$|množica\ akcij\ \{a_t\}| = 2^{|N|}$.

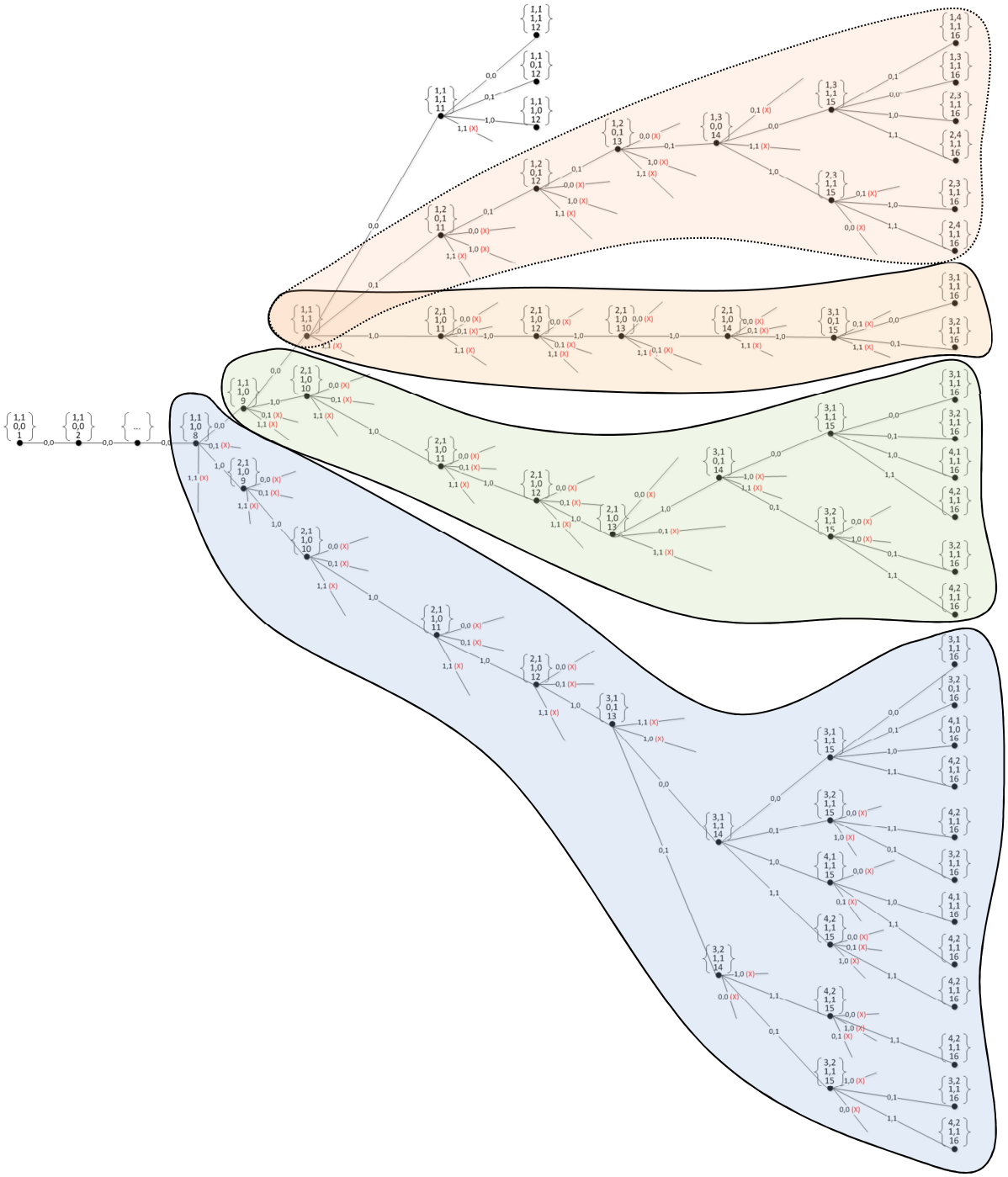
V primeru dveh vlakov so možne štiri akcije: $\{(0,0),\ (0,1),\ (1,0),\ (1,1)\}$, v primeru treh vlakov pa osem: $\{(0,0,0),\ (0,0,1),\ (0,1,0),\ (1,0,0),\ (0,1,1),\ (1,1,0),\ (1,0,1),\ (1,1,1)\}$, kjer je 0 oznaka za nenadaljevanje vožnje in 1 oznaka za nadaljevanje vožnje.

Za lažje razumevanje učenja Q, spreminjanja stanj ter možnih in dovoljenih akcij je v nadaljevanju prikazan del drevesa možnih stanj in akcij od začetka epizode do 16. minute za vozni red, prikazan na spodnjem prikazu Slika 17, in primer, ko Vlak 1 zamuja šest minut.



Slika 17: Primer voznega reda dveh zaporednih vlakov
Figure 17: Example of timetable for two successive train

Na prikazu Slika 18 so na povezavah napisane akcije, med katerimi agent vsako minuto izbira. Na prikazu so na vozliščih podani opisi stanja okolja. Ponovimo, stanje okolja je opisano z lokacijami vlakov, prostimi potmi in časom. Tako zapis $\begin{Bmatrix}4,3\\1,0,\\14\end{Bmatrix}$ ponazarja stanje po voznem redu, ko je prvi vlak na odseku 4 (medpostajni odsek med postajama B in C) in drugi na odseku 3 (na postaji B) – prva vrstica; ko ima prvi vlak prosto pot, drugi pa ne, saj naslednji odsek zaseda Vlak 1 – druga vrstica; to poteka v 14. minuti (zadnja vrstica zapisa).



Slika 18: Del drevesa možnih stanj okolja in akcij
Figure 18: Part of possible environment states and actions

Algoritem je prednastavljen tako, da agent izvaja samo smiselne akcije (npr. ni akcije vzvratne vožnje) in akcije, ki upoštevajo omejitve pri vodenju vlakov, predstavljene v poglavju 2, zato so v posamezni iteraciji lahko dovoljene samo nekatere iz nabora vseh možnih akcij. Oznaka *(x)* pri akcijah pomeni, da akcija ni dovoljena zato, ker vlak še nima odhoda, predvidenega po voznem redu, ker je odsek, na katerega želi vlak, zaseden oz. zaradi omejitve, da dva vlaka ne smeta hkrati zapustiti postaje v isti smeri.

Na prikazu Slika 18 je do osme minute možna akcija samo $(0,0)$, saj obravnavamo primer, ko Vlak 1 zamuja šest minut, torej ima možen odhod v $t = 8\,min$, Vlak 2 pa ima po voznem redu odhod v $t = 10\,min$. Torej se agent šele v $t = 8\,min$ odloča med akcijami, ki dovoljujejo odhod Vlaku 1 (akciji $(1,1)$ ali $(1,0)$). Akciji $(1,1)$ in $(0,1)$ nista dovoljeni, saj Vlak 2 ne sme s postaje pred odhodom, predvidenim v voznem redu.

Z modro barvo je na drevesu možnih stanj in akcij označena veja, ko se agent odloči, da Vlak 1 zapusti začetno postajo takoj, ko ima vlak prosto pot, torej v osmi minuti. Dokler Vlak 1 ne prispe na naslednjo postajo, to je v 13. minuti, je edina možna akcija $(1,0)$, saj se vlak na medpostajnem odseku ne sme ustavljati, Vlak 2 pa ne sme na zasedeni odsek. V času $t = 13\,min$ se agent odloči, ali Vlak 2 zapusti prvo postajo, glede na to, ali se je medpostajni odsek sprostil (akcija $(0,1)$) ali ne (akcija $(0,0)$). Akciji, ki dovoljujeta vožnjo Vlaku 1 (akciji $(1,1)$ ali $(1,0)$), nista dovoljeni, saj ima Vlak 1 na postaji po voznem redu predviden postanek, dolg vsaj eno minuto. V času $t = 14\,min$ se agent glede na prejšnjo akcijo odloči, kateri vlak nadaljuje vožnjo; če je Vlak 2 v $t = 13\,min$ zapustil postajo, potem lahko izbira samo med akcijami, ki dovoljujejo vožnjo temu vlaku, sicer ima na voljo vse štiri možne akcije.

Z zeleno obarvana veja drevesa prikazuje primer, ko agent Vlaku 1 dovoli vožnjo na prvi medpostajni odsek v deveti minuti. Do takšnega primera pride, ko agent poizkuša odziv okolja, če Vlaku 1 podaljša postanek, kljub temu da bi po voznem redu že lahko odšel, in v primeru, ko Vlak 1 zamuja sedem minut in mu agent 1 dovoli vožnjo takoj, ko je mogoče. Zelo je pomembno, da agent vlaku podaljšuje postanek, kljub temu da so izpolnjeni vsi pogoji, da lahko vlak zapusti postajo, saj le tako agent lahko preveri uspešnost strategije z drugačnim vrstnim redom vlakov ali s prilagajanjem postankov za križanje vlakov. Struktura zeleno obarvane veje je podobna modri veji, le da so akcije in stanja zamaknjena za eno minuto.

Oranžno obarvani veji prikazujeta primer, ko agent glede na možen odhod Vlaka 1 v osmi ali deveti minuti le-temu postanek podaljšuje, oz. primer, ko imata oba vlaka odhod s prve postaje, možen v deseti minuti. Ker ima Vlak 2 po voznem redu odhod v $t = 10\,min$ (in nima zamude), se drevo razcepi na dve veji, in sicer ločeno za primer, ko v deseti minuti postajo zapusti Vlak 1, in primer, ko postajo najprej zapusti Vlak 2.

Nadaljevanje drevesa smo samo nakazali, saj bi agent teoretično lahko preizkušal strategijo, kjer obema vlakoma podaljšuje postanek na začetni postaji. Drevo možnih stanj in akcij je prikazano samo do 16. minute, saj smo ponazorili logiko vodenja železniškega prometa na medpostajnem odseku in na postaji. Nadaljevaje drevesa je podobno.

3.3.5 Spodbuda

Funkcija nagrade določa uspešnost izvedene akcije a_t v stanju s_t. Cilj agenta je doseganje maksimalne vrednosti končne nagrade, torej je funkcija nagrade bistvenega pomena za usmerjanje agenta k doseganju cilja. V algoritmu za replaniranje voženj vlakov smo v skladu z osnovnim principom učenja Q najprej preverili učinkovitost sprotnega dodeljevanja nagrad. V tem primeru je agent prejel nagrado v velikosti odstopanja od voznega reda po vsaki izvedeni akciji.

$$r_{t+1} = -d_{i,t}, \tag{6}$$

kjer je:

$d_{i,t}$ … (trenutna) zamuda vlaka i v času t.

Več o principu nagrajevanja in uspešnosti takšnega pristopa je zapisano v poglavju 3.3.8.

V nadaljnjih eksperimentih (poglavji 3.3.9, 3.3.10) smo sprotno dodeljevanje nagrad nadomestili s t. i. zakasnjeno nagrado, ki jo agent prejme v končnem stanju okolja. V tem primeru agent v vseh vmesnih stanjih prejme nagrado $r_{t+1} = 0$, le v končnem stanju nagrado, katere vrednost je premo sorazmerna z nasprotno vrednostjo vsot zamud vseh vlakov, pri čemer je zamuda definirana kot absolutna vrednost razlike med predvidenim prihodom na končno postajo po voznem redu in prihodom na končno postajo po replaniranem voznem redu. Nagrada, ki jo prejme agent za svoje odločitve, je enaka vsoti nagrad vseh vlakov ob koncu eksperimenta:

$$r_{t+1} = -\sum_{i=1}^{n} d_{i,T} = -d_{skupna}. \tag{7}$$

kjer je:

$d_{i,T}$ … zamuda vlaka i v končnem stanju *T*.

Za zagotavljanje konvergence učenja mora biti nagrada omejena, torej mora obstajati takšen $R_0 < \infty$, da velja za vsak $(s,a) \in S \times A, r(s,a) < R_0$ (Watkins in Dayan, 1992). Vlaki, ki ne pridejo na končno postajo, imajo običajno zaradi nastanka brezizhodne situacije, lahko pa tudi zato, ker agent enemu ali več vlakom podaljšuje postanek, neskončno zamudo. Da zagotovimo konvergiranje k optimalni rešitvi, omejimo zamudo posameznega vlaka tako, da vlak, ki ne pride do končne postaje, dobi majhno nagrado v vrednosti r_{min}. Vrednost nagrade, ki jo prejme agent iz okolja, določimo po enačbi:

$$r_{t+1} = \begin{cases} d_{skupna}; & d_{skupna} \geq r_{min} \\ r_{min}; & d_{skupna} < r_{min} \end{cases}. \tag{8}$$

Vrednost r_{min} dispečer oceni glede na izkušnje o velikosti zamud na obravnavanem odseku.

3.3.6 Ocena max$Q(s_{t+1}, a_{t+1})$ v končnem stanju okolja

Vrednosti $Q(s_{t,}a_t)$ se po enačbi (1) posodabljajo v odvisnosti od vrednosti matrike Q v naslednjem stanju (torej vrednosti Q_{t+1}). V primeru reševanja problema replaniranja vlakov smo končno stanje sistema definirali kot čas T, ki ga dosežemo, ko so vsi vlaki na končnih postajah, ko ne morejo nadaljevati vožnje zaradi nastanka brezizhodne situacije oz. ko preteče čas uporabniško določenega časovnega okna. Vsem situacijam je skupno, da ko agent doseže končno stanje, nima več možnosti izbire naslednje akcije, in matrika Q se ne posodablja več. Vrednost matrike Q v končnem stanju se mora zato oceniti. Ocena, ki jo predlagamo, temelji na predpostavki, da je velikost zamude vlakov (torej vrednost nagrade) v času $T-1$ in T enaka in da je $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) = Q(s_{t,}a_t)$. Določimo jo, kot sledi:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma * \max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) - Q(s_{t,}a_t) \right]$$

$$0 = \alpha \left[r_{t+1} + \gamma * \max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) - \max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) \right]$$

$$\alpha * r_{t+1} = \alpha * \max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) * (1-\gamma)$$

$$\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) = \frac{r_{t+1}}{(1-\gamma)}. \quad (9)$$

3.3.7 Simulator

Ker imajo akcije dispečerjev velik vpliv na razvoj dogodkov in na kvaliteto storitev, je cilj, da se vplivi odločitev ovrednotijo pred njihovo implementacijo. V ta namen so zelo koristne simulacije. Simulacijska orodja so uporabna v fazi konstruiranja voznega reda, za oceno vplivov načrtovanih gradbenih in organizacijskih sprememb, za načrtovanje odvijanja železniškega prometa v času izvajanja del, za hitro posredovanje odgovorov na kratkoročne in *ad-hoc* zahteve uporabnikov storitev ter za pomoč pri odpravi posledic zamude. S simulacijami ponazorimo dogajanje v realnosti (brez dodatnih stroškov in nevarnosti) ter ocenimo različne variante voznih redov.

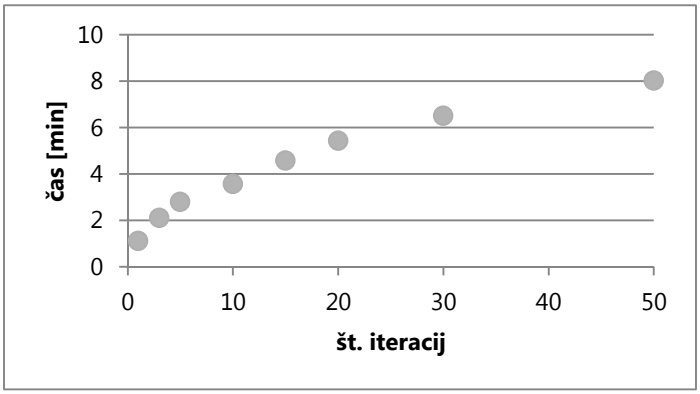
Za natančno simuliranje železniškega prometa se uporabljajo mikroskopski modeli, ki natančno izračunajo vozne čase. Mikroskopski modeli v izračunih voznih časov upoštevajo karakteristike lokomotive (vlečno moč, pospeške pri speljevanju in pojemke pri zaviranju), odpore železniških vozil (zračni upor, odpor zaradi kotaljenja, odpor zaradi trenja v ležajih), odpore proge (odpor zaradi vzpona, odpor zaradi krivine, odpor zaradi predora) ter lastnosti železniške infrastrukture (velikosti in dolžine horizontalnih in vertikalnih geometrijskih elementov prog, omejitve hitrosti, območja kretnic). Rezultat so v vsakem trenutku natančno

izračunane hitrosti vlakov. Takšna simulacijska orodja omogočajo natančno določevanje zasedenosti posameznih odsekov.

Na trgu obstajajo simulacijska orodja, namenjena izključno simulacijam železniškega prometa, npr. OpenTrack. Programsko orodje OpenTrack je namenjeno natančnemu določevanju potovalnih časov, določevanju izkoriščenosti kapacitet železniške infrastrukture in izračunu porabe energije. Modeliranje železniške infrastrukture v OpenTrack je zamudno, poleg tega pa program ne omogoča optimizacije voznega reda, temveč samo simulira vožnje vlakov po uporabniško določenem voznem redu. Naslednja pomanjkljivost simulacijskega orodja OpenTrack je ta, da ga ni mogoče upravljati s pomočjo drugih programskih orodij, kar pomeni, da ni možno vzpostaviti komunikacije med okoljem in agentom, kar je nujno potrebno za uporabo simulacijskega orodja kot okolja za spodbujevano učenje. Zato smo v začetnih eksperimentih vzpostavili okolje v uveljavljenem mikroskopskem programu za simuliranje kopenskega in vodnega prometa, in sicer v Vissim. Vissim preko COM-vmesnika omogoča programiranje in s tem krmiljenje signalov in upravljanje voženj vlakov, torej lahko agent učenja Q okolju sporoča in testira različne akcije. Začetne eksperimente replaniranja voženj vlakov z uporabo učenja Q smo izvajali na enostavnem modelu železniške infrastrukture (odsek proge s tremi postajami) in tremi vlaki.

Prvi rezultati uporabe učenja Q in simulacijskega orodja Vissim so bili spodbudni, saj se je izkazalo, da je rezultat nov brezkonflikten vozni red (torej so upoštevane vse omejitve, ki smo jih definirali v okolju učenja) ter da agent v različnih poskusih učenja predlaga drugačen vozni red, torej se uči.

Vendar morajo pristopi, ki jih želimo uporabiti v realnem času, imeti dve lastnosti, in sicer da najdejo dobro rešitev (ne nujno optimalno) ter da rešitev izračunajo v razumno kratkem času. V nadaljevanju je prikaz (Slika 19), ki prikazuje čas, potreben za različno število ponovitev.



Slika 19: Trajanje simulacije s programskim orodjem Vissim
Figure 19: Computation time obtained with VISSIM software for different number of runs

S prikaza Slika 19 je razvidno, da sistem za enostaven primer železniške infrastrukture in majhno število vlakov potrebuje približno štiri minute za deset ponovitev in osem minut za 50 ponovitev. S povečanjem časovnega okna se poveča čas trajanja simulacije, s povečanjem števila odsekov in števila vlakov se poveča tudi kompleksnost problema (več možnih stanj in več možnih akcij, ki jih mora agent preveriti), zato bi bil čas, potreben za izračun (skoraj) optimalnega voznega reda na bolj kompleksni infrastrukturi in z večjim številom vlakov, veliko daljši – zato pristop ni primeren za uporabo v realnih situacijah.

Kot so avtorji že poročali (Fay, 2000; Hulea et al., 2007; Kecman et al., 2012), se je tudi v našem primeru izkazalo, da so zaradi upoštevanja množice vhodnih podatkov za izračun natančnih hitrosti in lokacij vlakov mikroskopski modeli neuporabni za delo v realnem času. Tako enostaven primer, kot smo ga obravnavali mi, dispečer reši veliko hitreje, zato smo poenostavili kompleksne izračune dejanskih hitrosti vlakov in za preveritev uporabnosti metode učenja Q za časovno replaniranje voženj vlakov razvili novo simulacijsko orodje. »Ukrojeno« simulacijsko orodje smo razvili v programskem okolju MS Access z uporabo programskega jezika Visual Basic for applications – VBA.

V simulacijskem orodju uporabnik definira dva elementa železniškega prometa, in sicer železniško infrastrukturo in vlake. Osnovni elementi, ki definirajo železniško progo, so povezave (linki), ki v naravi predstavljajo prostorske odseke in postajne tire. Tako v naravi kot v simulatorju so odseki zavarovani s signalno-varnostnimi napravami. Simulacijsko orodje omogoča natančno definiranje dolžin linkov (odsekov), pri čemer je lahko dolžina odseka poljubna, njihovo število ni omejeno, kar omogoča izdelavo modela z upoštevanjem poljubne kapacitete obravnavane železniške proge. Simulator omogoča modeliranje eno-, dvo- ali večtirnih prog, zato se odsekom pripiše tudi atribut o smeri vožnje vlakov. Pri tem so linki lahko enosmerni in dovoljujejo samo vožnjo vlakov v eni smeri, kar je običajno pri dvo- ali večtirnih progah, ali dvosmerni, kjer vlaki vozijo v obe smeri, kot je to primer pri enotirnih progah. Odsekom določimo tudi vrsto linka – ali je torej odsek na odprti progi ali na postajnem območju, pri čemer je vsak postajni tir definiran kot posamezni odsek.

Drugi element železniškega prometa so vlaki, ki jih v simulatorju lahko definiramo poljubno veliko. Vlakom pripišemo statične in dinamične parametre. Statični so: vozni red, maksimalna hitrost vlaka, pospešek pri pospeševanju in pojemek pri zaviranju vlaka, smer vlaka ter identifikacijska številka vlaka. Identifikacijska številka vlaka se uporablja za identifikacijo vlaka, kar uporabniku omogoča npr. različno upoštevanje stroškov zamud glede na rang vlaka ali glede na število potnikov. Dinamična parametra sta replanirani vozni red in lokacija vlaka. Lokacija vlaka se nanaša na trenutno lokacijo vlaka, in sicer na prostorski odsek, na katerem je vlak.

Simulator agentu učenja Q omogoča testiranje različnih akcij v procesu replaniranja voženj vlakov in omogoča učenje agenta na način, kot je opisan v nadaljevanju.

V okolju učenja Q uporabnik definira vozni red vlakov, v katerem so za vsak vlak definirani začetni in končni link (ni nujno, da začetni in končni link posameznega vlaka sovpadata z začetnim in končnim linkom obravnavane proge), odhodi vlakov s posameznih postaj ter trajanje postanka vlaka na posamezni postaji. Čas trajanja postanka definira uporabnik, pri čemer je čas trajanja postanka t_P vlaka j na postaji i lahko po voznem redu $t_{Pj,i,VR} = 0$, v tem primeru vlak prevozi postajo brez ustavljanja.

Ob pričetku simulacije so vlaki na svojih začetnih linkih, njihova hitrost je $v = 0\ km/h$, vse signalno-varnostne naprave so postavljene v signalni znak »stoj«. Simulator agentu v vsaki iteraciji vrne informacijo o stanju okolja ter nabor v trenutnem stanju izvedljivih akcij. Iz nabora vseh možnih akcij so le nekatere izvedljive; in sicer glede na vozni red (oz. možen odhod po nastanku zamude), na prostost linkov (na enem odseku je lahko samo en vlak), prostost medpostajnega odseka (zavarovanje voženj v nasprotni smeri na enotirni progi po načelu privolitve smeri vožnje) ter lokacijo vlaka (naj ponovimo: če je vlak na odprti progi, potem ima agent na voljo samo akcijo, da vlak nadaljuje z vožnjo). Iz nabora izvedljivih akcij agent glede na strategijo izbire akcije izbere akcijo in spremeni signalne znake. Simulator glede na izbrano akcijo izračuna (z upoštevanjem trenutne hitrosti in pospeška vlaka) nove lokacije vlakov glede na prevoženo razdaljo v časovni enoti ali lokacijo signala, ki kaže signalni znak »stoj«. Simulator preveri, ali ima vlak v nadaljevanju poti signalni znak »stoj« – v tem primeru vlak prestavi na lokacijo signalno-varnostne naprave in ne na lokacijo, ki bi jo vlak prevozil v časovnem koraku. Simulator nima omejitev glede velikosti časovnega koraka, kar omogoča izračun nove lokacije vlakov npr. vsako sekundo, vsako minuto, vsakih pet minut. V vseh eksperimentih smo upoštevali časovni korak ene minute, saj so s takšno natančnostjo tudi podani prihodi in odhodi vlakov v voznem redu.

Simulator glede na lokacije, število vlakov ter kapacitete postaj preverja možnost nastanka brezizhodne situacije, ki nastane zaradi prevelikega števila vlakov na območju postaje glede na število postajnih tirov.

Zunanji program omogoča uporabniku definiranje zamud poljubnim vlakom, v poljubni velikosti in na poljubni postaji (ne pa tudi na območju odprte proge).

Če na kratko povzamemo: za vodenje vlakov skrbi, podobno kot v realni situaciji, dispečer, agent učenja Q, ki se odloča, kateri vlaki lahko nadaljujejo z vožnjo, in svoje odločitve preko signalov sporoča vlakom. Simulator vsako minuto izračuna nove lokacije vlakov in skrbi, da se vlaki ustavijo na lokaciji signalno-varnostnih naprav, če je signalni znak »stoj«, in ne na

lokaciji, ki bi jo vlak prevozil v časovnem koraku. Simulator pozna lokacije vlakov, pozna varnostni načeli »en vlak na enem odseku« ter »dovoljenje za vožnjo s privolitvijo smeri« ter prepozna nastanek brezizhodne situacije in agentu vrne informacijo, katere akcije so v trenutnem stanju izvedljive.

Čas trajanja simulacije smo preverili na primerljivi železniški infrastrukturi (kot v prvem delu eksperimenta), torej tri postaje, trije vlaki. Z novim simulacijskim orodjem izvedemo 1000 ponovitev učenja v samo dveh sekundah, kar je bistveno hitreje kot pri uporabi simulacijskega orodja Vissim.

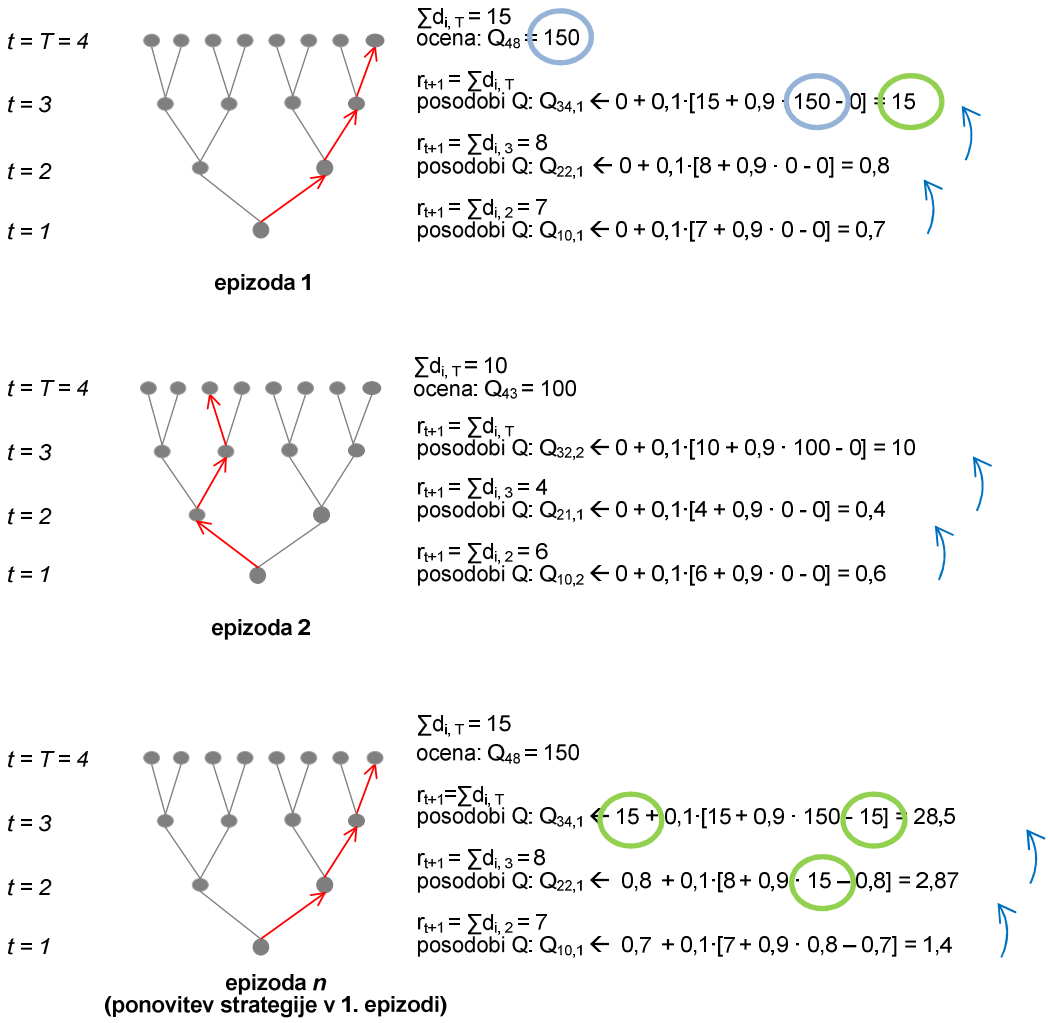
Spoznanja, pridobljena v eksperimentu, lahko povzamemo z naslednjimi besedami:

- mikrosimulacijska orodja so namenjena natančnemu izračunu voznih časov, porabe energije, natančni določitvi izkoriščenosti kapacitet itd., vendar so zaradi časovne zahtevnosti neuporabna pri replaniranju v realnem času;
- »ukrojeni« simulator omogoča simuliranje poljubne železniške infrastrukture (poljubno število in dolžine prostorskih odsekov ter poljubno število in dolžine postajnih tirov) za poljubno število vlakov v poljubno velikem časovnem oknu;
- v simulator vgrajeno predznanje o varnostnih načelih v železniškem prometu se uspešno upošteva – vozni redi, ki jih predlaga agent, so izvedljivi;
- simulator uspešno prepozna možnost nastanka brezizhodne situacije (ang. *dead lock*) pri vožnji vlakov v nasprotnih smereh;
- »ukrojeni« simulator vrne rezultat dovolj hitro, tako bi bil lahko uporaben za delo v realnem času; zato so vsi eksperimenti, opisani v nadaljevanju, izvedeni s tem simulacijskim orodjem.

3.3.8 Učenje Q

Osnovni princip učenja Q (enačba 1) je sprotno prejemanje nagrad (okolje agentu vrne vrednost nagrade v vsaki iteraciji) ter sprotno posodabljanje matrike Q (agent v vsaki iteraciji posodobi vrednost matrike Q). Za lažje razumevanje principa učenja Q podajamo kratek primer, na katerem sta prikazana princip dodeljevanja nagrad in izračun vrednosti Q. V izračunu sta upoštevani v literaturi najpogosteje uporabljeni vrednosti parametrov stopnje učenja in diskontiranja nagrade, in sicer $\alpha = 0{,}1$, $\gamma = 0{,}9$. Predpostavljeno je, da je matrika Q inicializirana na vrednost 0.

Na prikazu Slika 20 je z rdečimi puščicami označena smer gibanja agenta, z modrimi pa smer posodabljanja matrike Q. Z modrima krogoma je označeno, kako se ocenjena vrednost matrike Q v končnem stanju propagira v stanju $T-1$, z zelenimi pa, kako se ob ponovitvi iste strategije upoštevajo prej izračunane vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$.

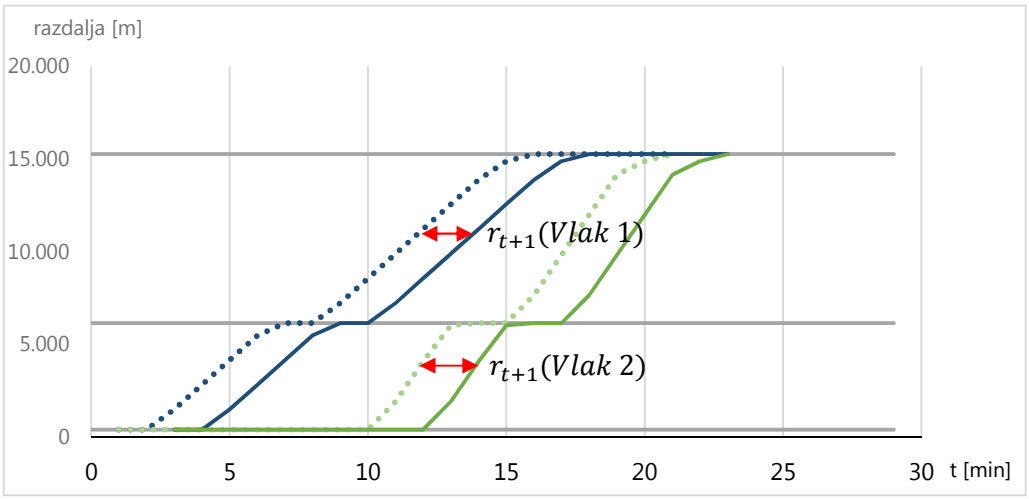


Slika 20: Princip učenja Q
Figure 20: How Q learning works

S prikaza Slika 20 je razvidno, da agent v vsaki iteraciji prejme iz okolja nagrado (informacijo o velikosti trenutnih skupnih zamud) in skladno z enačbo (1) sproti posodablja vrednosti $Q(s_t, a_t)$. Vrednost $Q(s_t, a_t)$ v končnem stanju je ocenjena skladno z enačbo (9). Agent v procesu učenja izbira različne akcije, ki vodijo v različne strategije, in posodablja matriko Q, zato je na prikazu predstavljeno spreminjanje vrednosti Q v različnih strategijah. Prikazano je tudi posodabljanje matrike Q v primeru ponovitve strategije.

V začetnih eksperimentih smo skladno z osnovnim principom učenja Q ter dejstva, da v realnem svetu dispečer ves čas spremlja, ali dejansko stanje sovpada z voznim redom, v algoritmu učenja nagrado intuitivno definirali kot razliko med replaniranim in dejanskim

voznim redom (zamudo), ki jo agent prejme v vsaki iteraciji. Primer velikosti nagrade r_{t+1} v $t = 11\,min$ je prikazan na prikazu spodaj (Slika 21).



Slika 21: Učenje Q – definicija nagrade
Figure 21: Q learning – definition of reward

Pri definiranju nagrade na območju postaje je treba upoštevati z voznim redom predpisano dolžino postanka. Na prikazu Slika 21 se v $t = 15\,min$ vlakovni poti po voznem redu in replaniranju dotikata, vendar ima Vlak 2 kljub temu zamudo, saj mora imeti glede na vozni red vsaj eno minuto dolg postanek.

Velikost nagrade je odvisna od točnosti vlaka, zato se s takšnim načinom nagrajevanja agent odloča za akcije, s katerimi vlaki nadaljujejo z vožnjo čim prej, ko je to mogoče. V naravi, kjer so vlaki različnih prioritet (imajo različne stroške zamud) in vozijo z različnimi hitrostmi, je za globalno optimalno rešitev treba spremeniti vrstni red vlakov (glej primer na prikazu Slika 10 – Vlak 1002 in Vlak 1003) ali podaljšati postanek vlaka zaradi križanja (glej primer na prikazu Slika 10 – Vlak 101 in Vlak 1002). V primeru sprotnega dodeljevanja nagrad in posodabljanja matrike Q bo agent sicer preizkusil strategijo podaljševanja postanka vlaka, kar pomeni, da bo vlak zadržal na postaji, kljub temu da ima prosto pot in možen odhod po voznem redu, vendar bo za takšno akcijo slabo nagrajen. Zato bo v naslednjih poskusih odpravil vlak čim prej, saj takšne akcije zagotavljajo višje nagrade in višje vrednosti Q, ki jih agent upošteva pri učenju.

Testi so pokazali, da agent tudi pri zelo velikem številu iteracij (> 50000) in ne glede na kombinacijo parametrov $\alpha, \gamma \text{ in } \varepsilon$ ne najde optimalne strategije, zato rezultatov serije poskusov ne bomo prikazovali. V teoriji bi agent moral vsako stanje obiskati neskončnokrat, kar pomeni, da bi se agent verjetno naučil strategije zamenjave vrstnega reda, vendar takšno število ponovitev ni primerno za uporabo v realnem času.

Spoznanja, pridobljena v eksperimentu, lahko povzamemo z naslednjimi besedami:

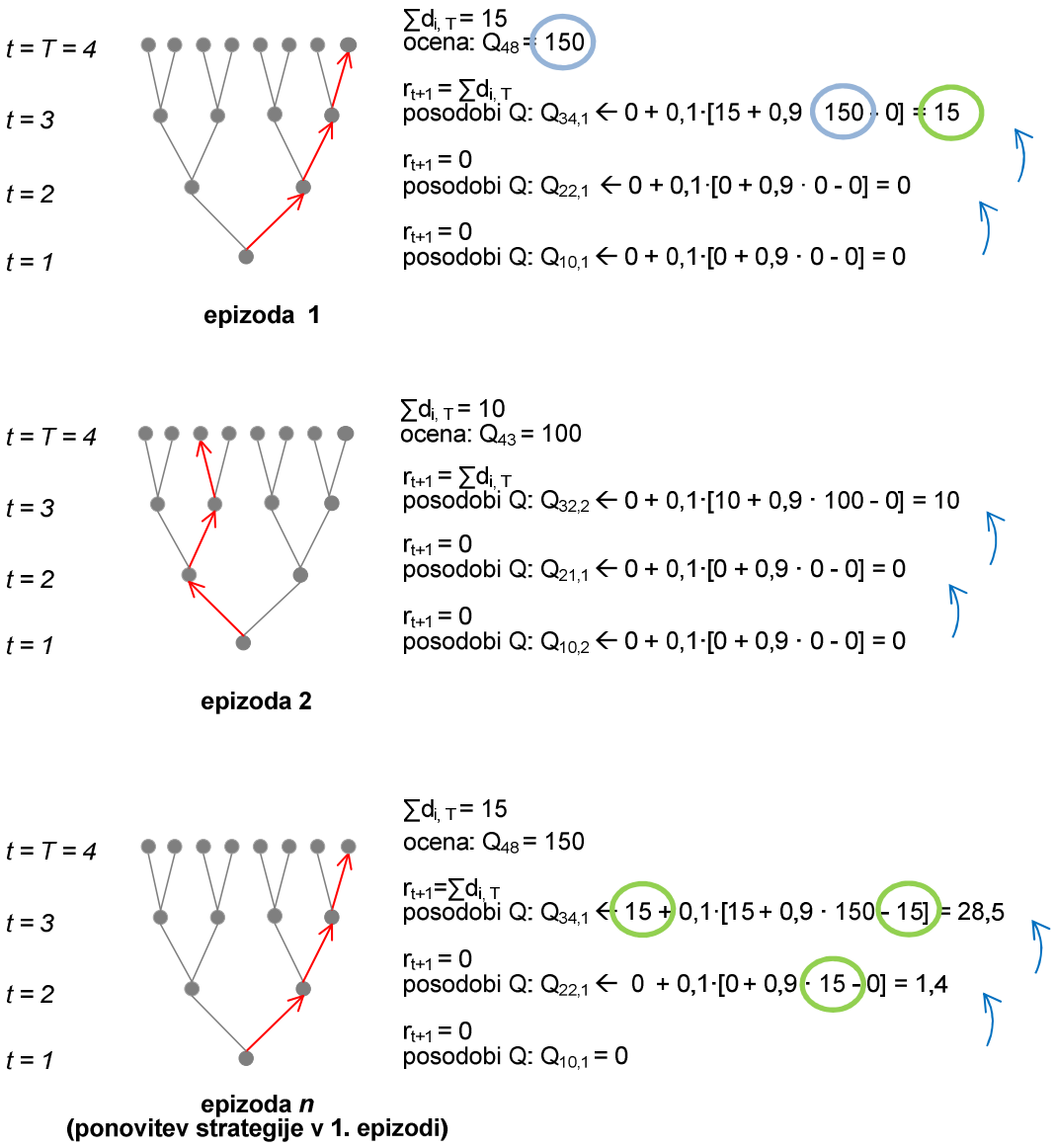
- v primeru sprotnega dodeljevanja nagrad je treba posebno pozornost nameniti definiciji nagrade na območju postaje;
- podaljševanje postankov je pomembno pri zagotavljanju globalnega optimuma;
- učenje s sprotnim dodeljevanjem nagrade je manj uspešno, saj agent za podaljševanje postanka prejema majhne nagrade, kljub temu da je strategija podaljševanja postanka lahko globalno uspešnejša.

3.3.9 Učenje Q z zakasnjeno nagrado

Pri problemu replaniranja voženj vlakov je uspešnost strategije (zaporedja akcij v danih stanjih od začetnega do končnega stanja) poznana šele v končnem stanju, ko torej pridejo vsi vlaki na končno postajo. Vmesne, trenutno manj uspešne akcije se lahko izkažejo kot nujne za doseganje globalnega optimuma. Podobno je pri igri šaha, ko pri posamezni potezi ne moremo oceniti, kako blizu konca igre smo. Zato je poleg drugih implementacijskih detajlov, ki so potrebni za uspešno reševanje problema igre šaha z učenjem Q, pomemben tudi ta, da agent prejme nagrado šele v končnem stanju, to je, ko eden od igralcev zmaga ali se igra konča z remijem (Mannen, 2003).

Formulacijo učenja Q na način, da agent prejme nagrado le v končnem stanju, imenujemo učenje Q z zakasnjeno nagrado. V primeru uporabe učenja Q za reševanje problema replaniranja vlakov učenje z zakasnjeno nagrado pomeni, da agent v vsaki iteraciji prejme nagrado $r_{t+1} = 0$, v končnem stanju pa nagrado $r_{t+1} = -\sum_{i=1}^{n} d_{i,T}$.

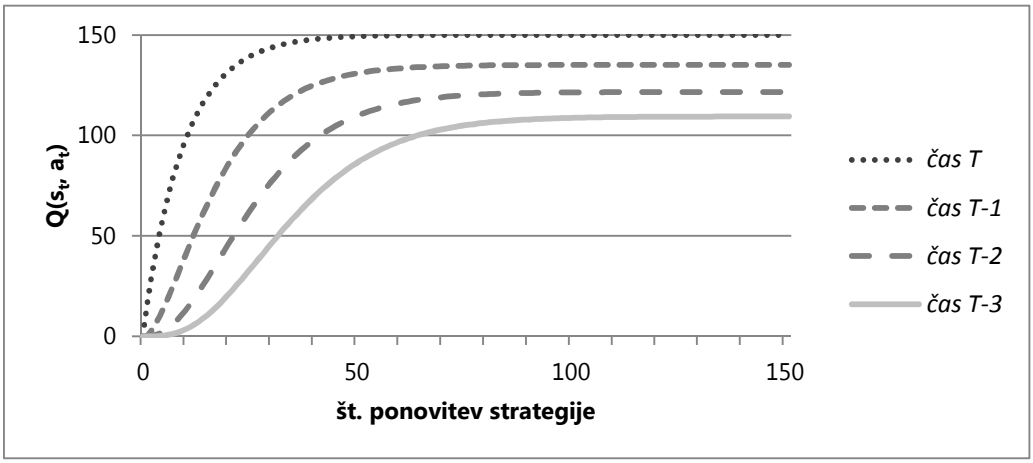
Na prikazu Slika 22 je prikazan proces dodeljevanja nagrad in posodabljanja vrednosti matrike Q pri učenju z zakasnjeno nagrado. Tudi v tem primeru smo upoštevali, da je matrika Q inicializirana na vrednost 0, ter upoštevali vrednosti $\alpha = 0{,}1$ in $\gamma = 0{,}9$. Ponovno je z rdečimi puščicami označena smer gibanja agenta, z modrimi pa smer posodabljanja matrike Q, z modrima krogoma propagiranje ocenjene vrednosti matrike Q v končnem stanju v predhodnem stanju ter z zelenimi propagiranje vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ v naslednjih ponovitvah strategije.



Slika 22: Princip učenja Q z zakasnjeno nagrado
Figure 22: How Q learning with delayed reward works

S prikaza je razvidno, da so po prvi izvedbi epizode vrednosti nagrad in vrednosti Q v vseh, razen v predzadnjem in končnem stanju, enake nič. Na prikazu *n*-te iteracije je ponazorjeno, kako se pri naslednjih ponovitvah strategije vrednost nagrade, ki jo je agent prejel v končnem stanju, propagira v predhodnem stanju. S takšnim načinom nagrajevanja preprečimo, da bi agent izbiral predvsem akcije, s katerimi čim prej izbere odhod vlaka, saj so v začetnih ponovitvah zaradi $r_{t+1} = 0$ vse akcije enakovredne, pri zadostnem številu ponovitev učenja pa se učinkovitost strategije propagira proti začetnim stanjem.

Ker so nagrade omejene (glej poglavje 3.3.5), so tudi vrednosti matrike Q omejene (Humphrys, 1996). Na prikazu Slika 23 je prikazana konvergenca vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ v času T, $T-1$, $T-2$ ter $T-3$ pri uporabi učenja Q z zakasnjenimi nagradami, in sicer za $\alpha = 0{,}1$, $\gamma = 0{,}9$ in $r_T = \sum_{i=1}^{n} d_{i,T} = -15$; iz enačbe (9) sledi $\max_{a_{(t+1)}} Q(s_T, a_T) = 150$.



Slika 23: Konvergenca vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ pri $\alpha = 0{,}1$, $\gamma = 0{,}9$, $r_T = 15$
Figure 23: Convergence of $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ values with $\alpha = 0.1$, $\gamma = 0.9$, $r_T = 15$

S prikaza Slika 23 je razvidno, da vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ v končnem stanju T konvergirajo k ocenjeni vrednosti $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1}) = \frac{r_{T+1}}{(1-\gamma)}$ in v vsaki predhodni iteraciji $(T-n)$ k vrednosti $\max_{a_{(t+1)}} Q(s_{t+1}, a_{t+1})$, diskontirani z γ^n.

V nadaljevanju poglavja je predstavljen primer uporabe učenja Q z zakasnjeno nagrado na zelo enostavnem primeru železniške infrastrukture. V vseh eksperimentih, predstavljenih v nadaljevanju, je naloga agenta, da zaradi zamude vlaka poišče optimalno rešitev, torej ustrezen vrstni red vlakov in odhode s postaj, da je skupna zamuda vseh vlakov na končni postaji minimalna.

Eksperiment smo glede na smer vožnje vlakov razdelili na tri sklope, in sicer:
a) vlaka vozita zaporedno;
b) vlaka vozita v različnih smereh;
c) kombinacija zaporednih voženj in voženj v različnih smereh.

Delitev eksperimenta na tri sklope omogoča, da ločeno preverimo, ali agent prepozna varnostna načela, ki veljajo za železniški promet. V prvem eksperimentu to pomeni, da je dal drugemu vlaku dovoljenje za vožnjo šele, ko je prvi vlak sprostil odsek (načelo, da je na enem odseku lahko samo en vlak), v drugem pa načelo privolitve vožnje iz različnih smeri (načelo, da morajo biti v nasprotni smeri privolitve vožnje vsi signali »stoj«), v tretjem sklopu

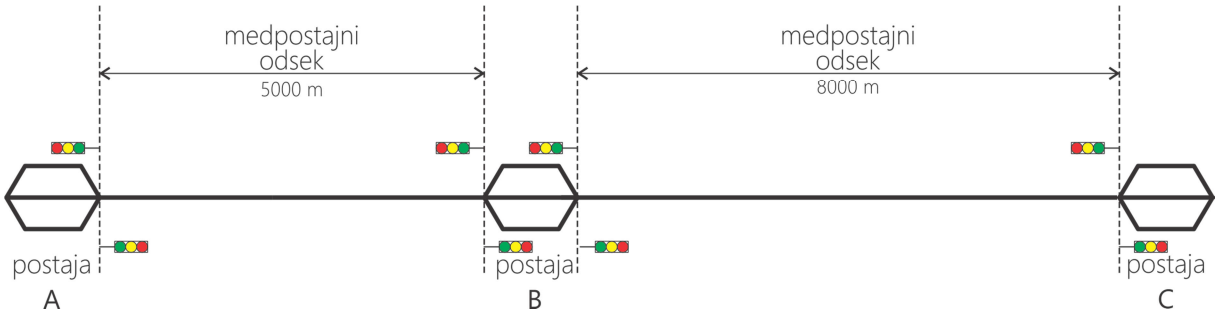
pa smo preverili, ali agent prepozna kombinacijo različnih zahtev ter jih tudi pravilno upošteva.

Pri učenju z uporabo algoritmov spodbujevanega učenja se postavi vprašanje, kako dolgo se mora agent učiti. V teoriji bi moral agent obiskati vsako stanje neskončnokrat, da bi se naučil optimalne strategije (Russell in Norvig, 2003), vendar je pri reševanju problemov v realnem času zelo pomembno, da dobimo kvalitetno (ne nujno optimalno) rešitev v razumno kratkem času, zato smo v eksperimentih agentu omejili število ponovitev učenja.

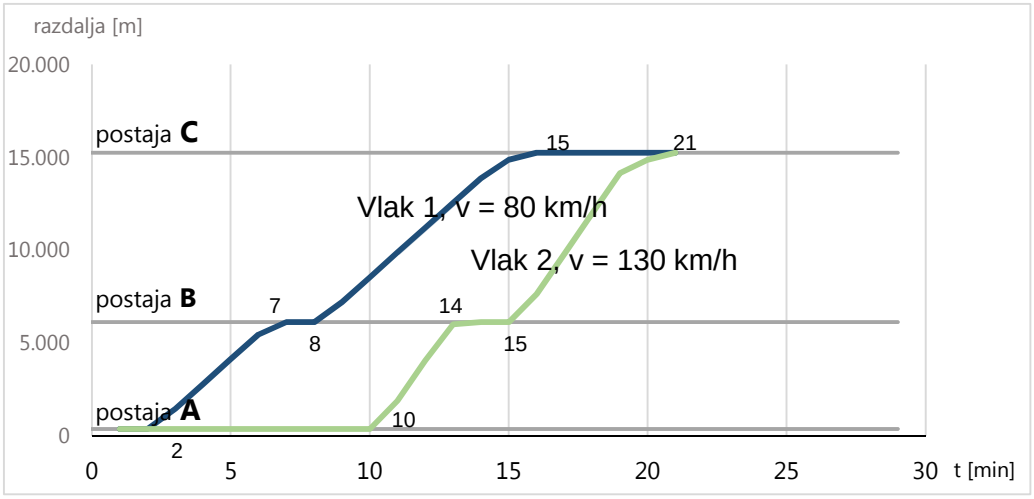
V vseh eksperimentih smo obnašanje agenta razdelili v dve fazi, in sicer v fazo bolj aktivnega učenja in fazo izkoriščanja pridobljenega znanja. Agent se je prvih *n*-ponovitev učil pri $\alpha = konst.,\ \gamma = konst.$ in $\varepsilon = konst.$, v zadnjih treh ponovitvah (t. i. testnih iteracijah) pa sta bila parametra $\alpha = 0$ in $\varepsilon = 0$. S testnimi iteracijami smo dobili informacijo, katera strategija (zaporedje akcij) je za agenta trenutno optimalna, saj se v testnih ponovitvah agent ne uči niti ne preizkuša novih akcij. Poudarjamo, da strategija, ki je agentu trenutno optimalna, ni nujno tudi dejanska optimalna rešitev problema (agent ima pomanjkljivo znanje).

a) Vlaka vozita zaporedno

Železniško infrastrukturo sestavljajo tri postaje (imenovane A, B in C) in dva medpostajna odseka. Postajni odsek med postajama A in B je dolg 5 km, drugi medpostajni odsek, odsek med postajama B in C, je dolg 8 km (glej prikaz Slika 24). V tem eksperimentu sta upoštevana le dva vlaka, ki vozita po enotirni progi. Vlaka imata vnaprej določen vozni red in s tem definirano hitrost ($v_{Vlak\ 1} = 80\ km/h, v_{Vlak\ 2} = 130\ km/h$), odhode s postaj, smer vožnje in vrstni red (glej prikaz Slika 25).



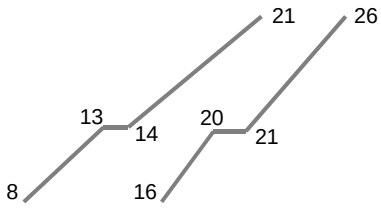
Slika 24: Eksperiment a – železniška infrastruktura
Figure 24: Experiment a – railway layout



Slika 25: Eksperiment a – vozni red
Figure 25: Experiment a – Timetable

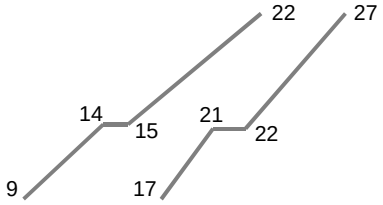
Eksperiment lahko glede na velikost začetne zamude razdelimo na dva primera, in sicer v prvem so poskusi, kjer je začetna zamuda prvega vlaka tako velika, da za optimalno rešitev problema ni treba spremeniti vrstnega reda vlakov. V drugem primeru je začetna zamuda prvega vlaka tako velika, da je zamenjava vrstnega reda vlakov nujna. Za dano železniško infrastrukturo in vozni red smo analitično določili, da pri zamudi prvega vlaka na začetni postaji $d_{1,1} < 7\ min$ za optimalno rešitev ni potrebna sprememba vrstnega reda vlakov, sicer pa je, zato smo izvedli serijo poskusov za tri scenarije zamud: z začetno zamudo prvega vlaka na prvi postaji v velikosti 6, 7 in 8 min. V nadaljevanju so prikazane analitično določene optimalne rešitve problema za vse tri scenarije zamud:

a) Scenarij 1: $d_{1,1} = 6\ min$

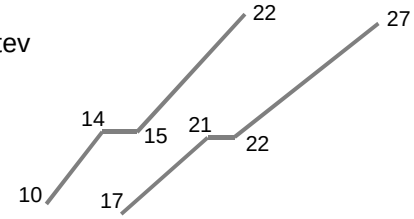


$\sum d_{i,T} = 6\ min + 6\ min = 12\ min$

b) Scenarij 2: $d_{1,1} = 7\ min$



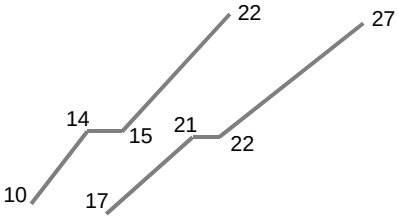
oz. za optimalno rešitev je treba obrniti vrstni red vlakov:



$\sum d_{i,T} = 7\ min + 7\ min = 14\ min$

$\sum d_{i,T} = 0\ min + 12\ min = 12\ min$

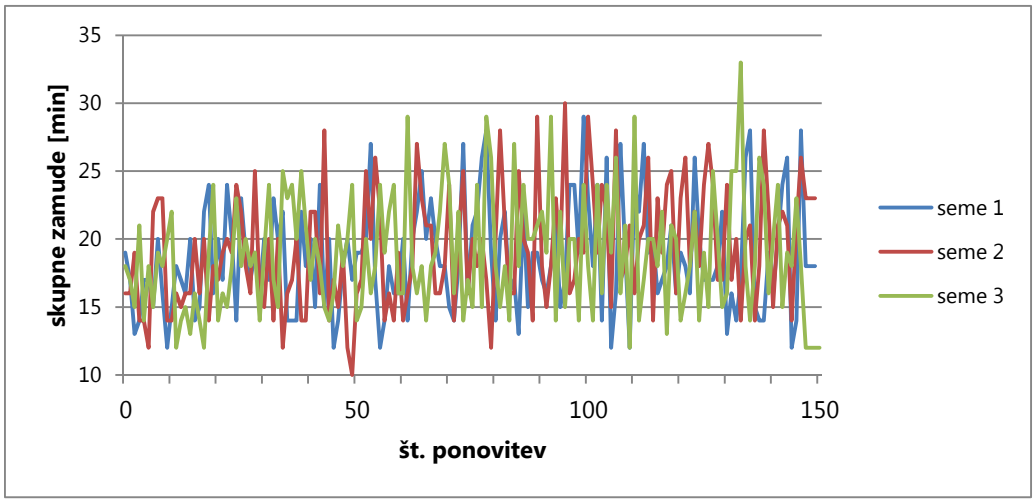
c) Scenarij 3: $d_{1,1} = 8\ min$



$\sum d_{i,T} = 0\ min + 12\ min = 12\ min$

V primeru b) je skupna zamuda vlakov manjša, če zamenjamo vrstni red vlakov, sicer je hitrejši vlak »ujet« za počasnejšim. Enak rezultat bi dobili, če bi počasnejši vlak zapustil prvo postajo takoj, ko se sprosti prvi medpostajni odsek (odhod v 14. minuti), vendar bi moral na postaji B podaljšati postanek, saj postajo lahko zapusti šele v 22. minuti, ko hitrejši vlak zapusti drugi medpostajni odsek. Ker nas zanima vsota zamud vseh vlakov na končni postaji, sta ta dva primera replaniranega voznega reda enakovredna.

Za vse tri scenarije zamud smo izvedli serijo poskusov za različne kombinacije vrednosti parametrov stopnje učenja, faktorja diskontiranja nagrade ter razmerja med raziskovanjem in izkoriščanjem. Za vsak scenarij zamude smo izvedli učenje z različnimi vrednostmi parametrov, in sicer
$\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ in za vse kombinacije parametrov, torej skupaj 125 kombinacij parametrov. Vsako kombinacijo parametrov α, γ in ε smo simulirali z desetimi različnimi semeni za generacijo naključnih vrednosti, saj na rezultat vpliva, v kakšnem zaporedju se zvrstijo akcije, s katerimi agent raziskuje in izkorišča znanje. Na prikazu Slika 26 so prikazane tri krivulje učenja za primer, kjer agent replanira vozni red v primeru, ko ima prvi vlak sedem minut zamude (Scenarij 2). Zaradi preglednosti so prikazane krivulje učenja samo za tri različna semena.



Slika 26: Eksperiment a – krivulje učenja za različna semena (α = 0,3; γ = 0,3; ε = 0,3)
Figure 26: Experiment a – Learning curves for different seeds (α = 0.3; γ = 0.3; ε = 0.3)

Zgornji prikaz (Slika 26) dokazuje, da izbira semena za generacijo naključnih spremenljivk vpliva na uspešnost učenja, zato se učenje izvede z različnimi semeni; kot rezultat učenja upoštevamo minimalno vrednost skupnih zamud v končni iteraciji. Za primer, prikazan na zgornjem prikazu, kot rezultat učenja za posamezno kombinacijo α, γ in ε upoštevamo rešitev, dobljeno s semenom 3.

Rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud so podani v Prilogi A in povzeti v Preglednici 2, v nadaljevanju poglavja pa so podane le ključne ugotovitve.

Preglednica 2: Eksperiment a – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov α, γ in ε.
Table 2: Experiment a – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account.

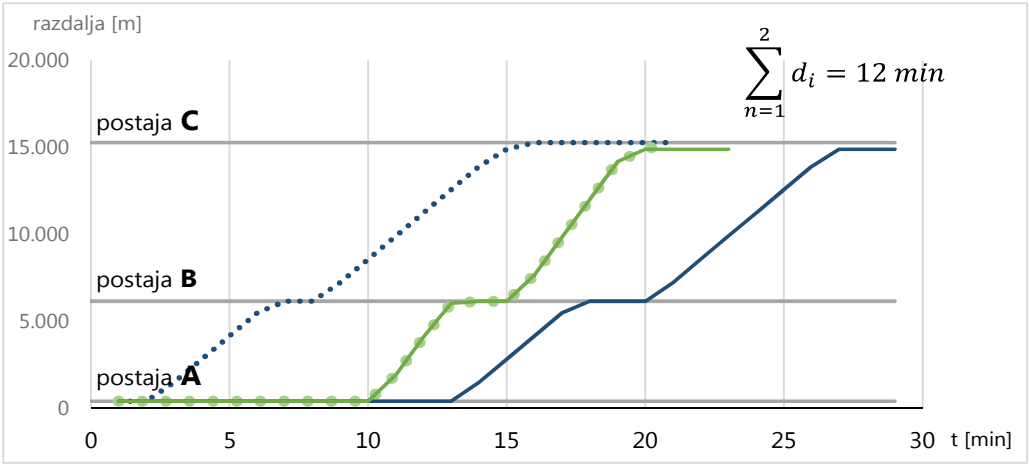
$\sum d_{i,min}$	Scenarij 1			Scenarij 2			Scenarij 3		
	Št. ponovitev			Št. ponovitev			Št. ponovitev		
	50	100	150	50	100	150	50	100	150
12	58 %	43 %	33 %	36 %	13 %	10 %	21 %	6 %	7 %
13	6 %	7 %	4 %	10 %	5 %	2 %	5 %	4 %	2 %
14	16 %	8 %	16 %	22 %	18 %	10 %	13 %	6 %	9 %
15	6 %	6 %	7 %	7 %	2 %	4 %	5 %	5 %	2 %
16	6 %	13 %	11 %	10 %	22 %	24 %	17 %	11 %	13 %
17	3 %	6 %	9 %	1 %	10 %	8 %	11 %	14 %	7 %
18	2 %	4 %	6 %	5 %	15 %	17 %	13 %	18 %	23 %
19	1 %	4 %	2 %	6 %	5 %	3 %	6 %	13 %	9 %
20	1 %	6 %	5 %	2 %	2 %	6 %	6 %	5 %	4 %
21	1 %	3 %	2 %	2 %	2 %	4 %	4 %	2 %	9 %
22	/	/	1 %	/	2 %	6 %	1 %	9 %	6 %
23	/	/	2 %	/	2 %	2 %	/	2 %	4 %
24	/	/	2 %	/	1 %	2 %	/	2 %	2 %
25	/	/	/	/	1 %	/	/	1 %	2 %
26	/	/	/	/	/	2 %	/	/	/
27	/	/	/	/	/	/	/	1 %	1 %

Agent je v vseh primerih vrnil rezultat, ki ustreza prometno-tehničnim in varnostnim zahtevam, torej je predlagal vozni red, ki je izvedljiv, ni pa nujno optimalen. V vseh obravnavanih scenarijih zamude je optimalna rešitev 12 minut, agent pa za Scenarij 1 izračuna zamude v velikosti med 12 in 24 min, za Scenarij 2 med 12 in 26 min in za Scenarij 2 med 12 in 27 min – odvisno od vrednosti parametrov α, γ, ε in števila ponovitev učenja. Iz preglednice je razvidno, da je agent nekoliko bolj uspešen pri reševanju problema po Scenariju 1, kjer optimalno rešitev najde v 58 %, 43 % oz. 33 % (odvisno od števila ponovitev učenja), pri reševanju problema po Scenariju 2 in Scenariju 3, ko je za optimalno rešitev treba podaljšati postanek Vlaku 1, je uspešnost veliko nižja – samo v enem primeru preseže 30 %.

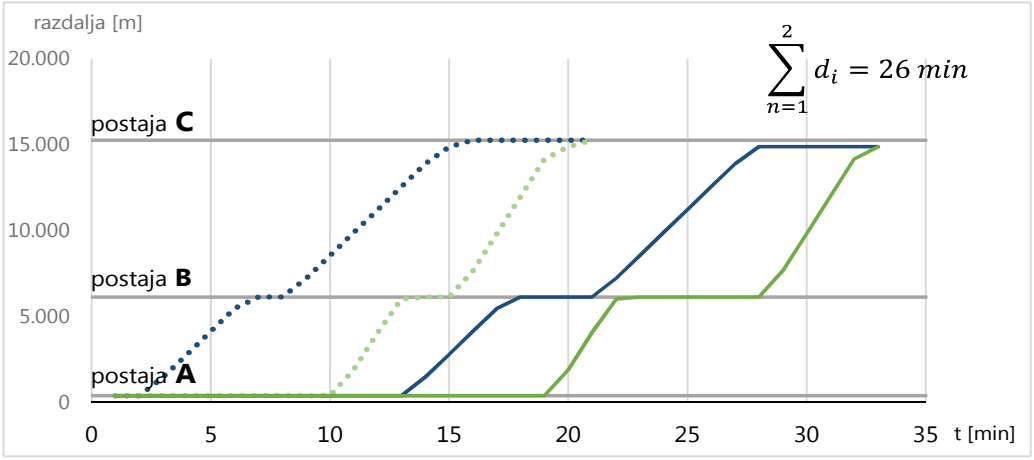
Iz Preglednice 2 je razvidno tudi, da se s povečevanjem števila ponovitev znanje agenta poslabšuje, saj se zmanjšuje delež poskusov, v katerih agent najde optimalno rešitev. S tem smo potrdili tezo, da če agenta silimo v nadaljnje raziskovanje, lahko poslabšamo njegovo znanje. Agent namreč nima spomina o najbolj uspešni strategiji (strategiji, ki je kadarkoli vrnila minimalno vrednost skupnih zamud). Pri večjem številu ponovitev bi se agentovo

znanje izboljšalo, vendar je za uporabo algoritma v realnem času pomembno hitro in učinkovito učenje, zato eksperimentov z večjim številom ponovitev nismo izvajali.

V nadaljevanju sta prikazana dva primera replaniranih voznih redov (optimalen in »slab« vozni red), ki jih predlaga agent za isti scenarij zamude (Scenarij 2), vendar z različno vrednostjo parametra γ.



Slika 27: Eksperiment a – replanirani vozni red (α = 0,3; γ = 0,3; ε = 0,3, Scenarij 2)
Figure 27: Experiment a – Rescheduled timetable (α = 0.3; γ = 0.3; ε = 0.3, Scenario 2)

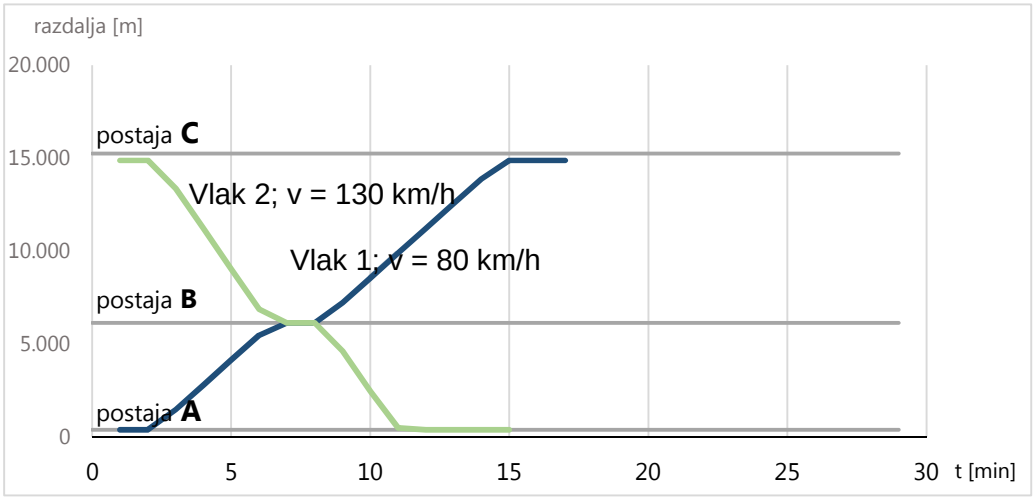


Slika 28: Eksperiment 3a – replanirani vozni red (α = 0,3; γ = 0,7; ε = 0,3, Scenarij 2)
Figure 28: Experiment 3a – Rescheduled timetable (α = 0.3; γ = 0.7; ε = 0.3, Scenario 2)

S prikazov Slika 27 in Slika 28 je razvidno, da replanirana vozna reda ustrezata zahtevi, da je samo en vlak na enem odseku, vendar se rešitvi razlikujeta. V prvem primeru se je agent pravilno naučil (glede na analitično rešitev), da je optimalna strategija tista, pri kateri podaljša postanek Vlaku 1 in zamenja vrstni red vlakov, v drugem primeru pa se je naučil, da je optimalna strategija tista, ko obema vlakoma še dodatno prestavi odhod, kar je seveda nesmiselno. Torej je pri uporabi algoritma učenja Q treba posebno pozornost nameniti izbiri vrednosti parametrov $\alpha, \gamma, \varepsilon$ (več v poglavju 3.3.11).

b) Vlaka vozita v različnih smereh

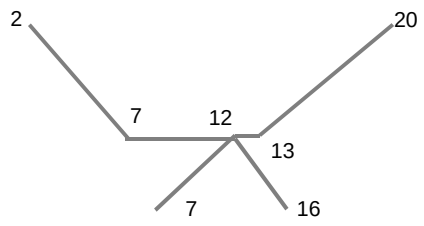
Vožnja zaporednih vlakov je značilna za dvo- ali večtirne proge, kjer je vsak tir namenjen vožnji v eno smer. Na enotirnih progah je treba zagotoviti varnost tudi vlakom, ki si vozijo nasproti. Da bi zagotovili uporabnost algoritma na različnih železniških infrastrukturah, smo preverili delovanje algoritma pri vožnji vlakov v nasprotnih smereh, kjer mora agent upoštevati načelo privolitve smeri vožnje. Izhodiščni vozni red za primer je podan na prikazu v nadaljevanju (Slika 29), upoštevana infrastruktura pa je enaka kot v Eksperimentu 3a (Slika 24), torej tri postaje z medpostajnima odsekoma, dolgima 5 km in 8 km.



Slika 29: Eksperiment b – vozni red
Figure 29: Experiment b – Timetable

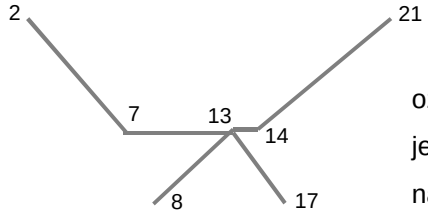
Za obravnavano železniško infrastrukturo in vozni red smo analitično določili, da je pri zamudi prvega vlaka na začetni postaji $d_{1,1} \leq 5\,min$ za optimalno rešitev treba podaljšati postanek Vlaka 2 na postaji B in ohraniti postajo, na kateri se vlaka križata (postaja B). Pri začetni zamudi $d_{1,1} > 5\,min$ je za optimalno rešitev treba spremeniti mesto križanja vlakov, torej je treba Vlaku 1 podaljšati postanek na postaji A. V nadaljevanju so prikazane optimalne rešitve problema za vse tri scenarije zamud:

a) Scenarij 1: $d_{1,1} = 5\,min$

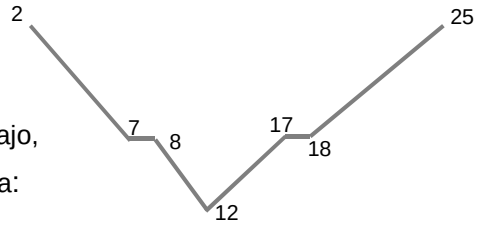


$\sum d_{i,T} = 5\,min + 4\,min = 9\,min$

b) Scenarij 2: $d_{1,1} = 6\,min$



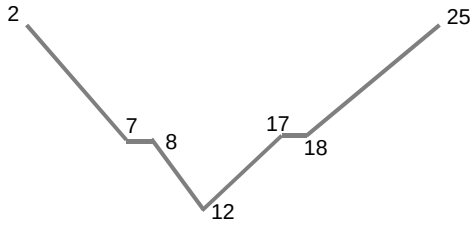
oz. za optimalno rešitev je treba spremeniti postajo, na kateri se vlaka križata:



$\sum d_{i,T} = 6\,min + 5\,min = 11\,min$

$\sum d_{i,T} = 0\,min + 10\,min = 10\,min$

c) Scenarij 3: $d_{1,1} = 7\,min$



$\sum d_{i,T} = 0\,min + 10\,min = 10\,min$

Za vse tri scenarije zamud smo izvedli serijo poskusov za različne kombinacije vrednosti parametrov stopnje učenja, faktorja diskontiranja nagrade ter razmerja med raziskovanjem in izkoriščanjem. Za vsak scenarij zamude smo izvedli učenje z različnimi vrednostmi parametrov, in sicer
$\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ in za vse kombinacije parametrov, torej skupaj 125 kombinacij parametrov. Vsako kombinacijo parametrov α, γ in ε smo simulirali z desetimi različnimi semeni za generacijo naključnih vrednosti. Rezultati za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije so podani v Prilogi B, komentar rezultatov pa je v nadaljevanju.

Preglednica 3: Eksperiment b – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov α, γ in ε.
Table 3: Experiment b – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account.

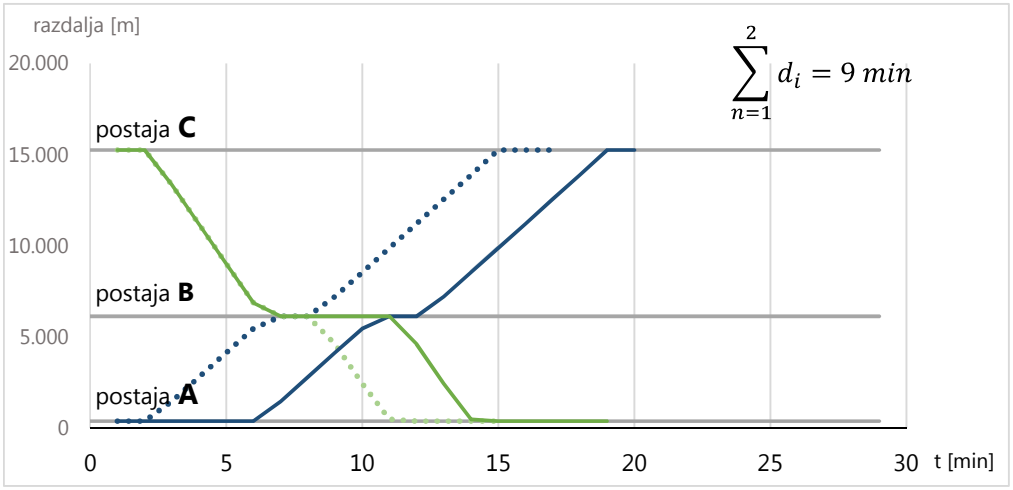
$\sum d_{i,min}$	Scenarij 1			Scenarij 2			Scenarij 3		
	Št. ponovitev			Št. ponovitev			Št. ponovitev		
	50	100	150	50	100	150	50	100	150
9	18 %	5 %	6 %				14 %*	4 %*	5 %*
10	14 %	14 %	10 %	0 %	0 %	0 %	1 %	2 %	/
11	44 %	35 %	38 %	31 %	16 %	17 %	4 %	2 %	1 %
12	10 %	16 %	15 %	20 %	23 %	24 %	2 %	/	/
13	10 %	18 %	15 %	31 %	26 %	21 %	39 %	28 %	23 %
14	2 %	2 %	8 %	10 %	14 %	17 %	12 %	24 %	19 %
15	2 %	3 %	2 %	3 %	13 %	16 %	21 %	14 %	23 %
16	1 %	5 %	2 %	5 %	5 %	1 %	7 %	12 %	12 %
17	/	/	1 %	/	2 %	2 %	1 %	6 %	7 %
18	/	1 %	1 %	/	/	2 %	1 %	5 %	3 %
19	/	1 %	2 %	/	/	/	/	2 %	4 %
20	/	/	/	/	/	1 %	/	1 %	1 %
21	/	/	/	/	1 %	/	/	/	2 %
22	/	/	1 %	/	/	/	/	/	/

* Pri analitičnem reševanju smo upoštevali, da Vlak 2 prispe na končno postajo v 12. minuti, torej gre lahko Vlak 1 s postaje A v $t = 12\ min$. Agent učenja Q upošteva, da vlak lahko zasede prosti odsek takoj, ko je to mogoče. V obravnavanem primeru Vlak 2 zapusti medpostajni odsek med postajama A in B v 11. minuti in se v začetku 12. minute ustavi pred signalnim znakom na postaji A. Sprostitev medpostajnega odseka v $t = 11\ min$ omogoča, da Vlak 1 zapusti začetno postajo že v 11. minuti, zato agent dobi optimalen rezultat zamud v velikosti 9 min, analitično pa smo kot optimalno rešitev določili skupno zamudo v velikosti 10 min.

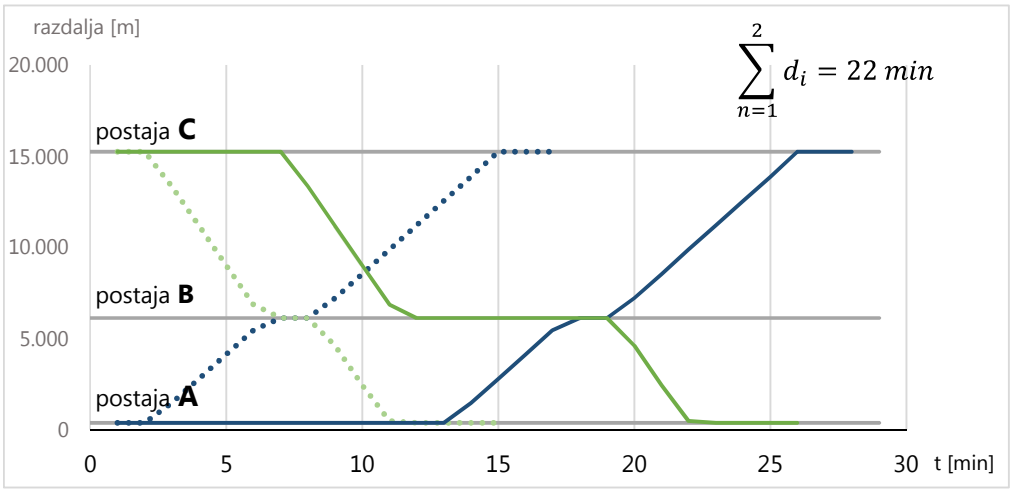
Agent je za Scenarij 1 izračunal zamude v velikosti med 9 in 22 min, optimalna rešitev je devet minut, za Scenarij 2 in Scenarij 3 je za optimalno rešitev treba obrniti vrstni red vlakov; tu je analitično določena optimalna rešitev deset minut, agent pa je v primeru Scenarija 2 izračunal vrednosti med 11 in 21 min, v primeru Scenarija 3 pa vrednosti med 9 in 21 min. V primeru Scenarija 2, v katerem mora v primerjavi s Scenarijem 3 za eno minuto dlje podaljšati postanek Vlaku 1 ter obrniti vrstni red vlakov, agent ne najde optimalne rešitve.

Tudi v primeru voženj vlakov iz nasprotnih smeri se agentovo znanje s povečevanjem števila ponovitev poslabša.

V nadaljevanju sta podana primer uspešnega in primer manj uspešnega replaniranja voženj vlakov v nasprotnih smereh.



Slika 30: Eksperiment b – replanirani vozni red (α = 0,1; γ = 0,3; ε = 0,1, Scenarij 1)
Figure 30: Experiment b – Rescheduled timetable (α = 0.1; γ = 0.3; ε = 0.1, Scenario 1)

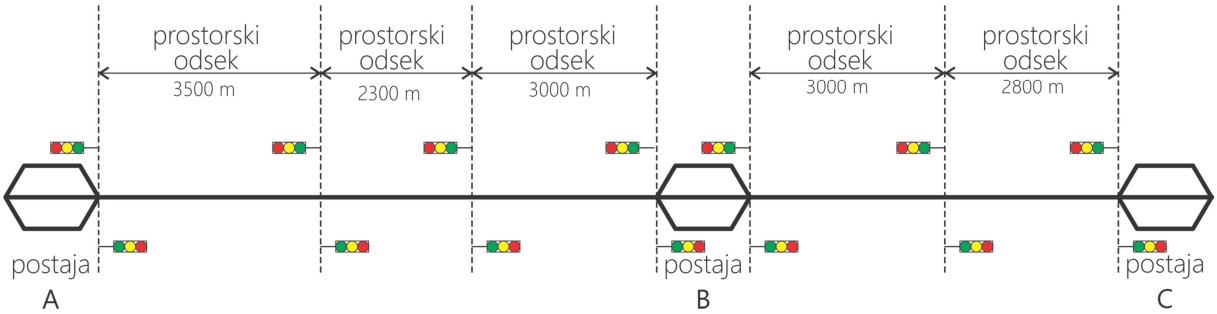


Slika 31: Eksperiment b – replanirani vozni red (α = 0,7; γ = 0,9; ε = 0,1, Scenarij 1)
Figure 31: Experiment b – Rescheduled timetable (α = 0.7; γ = 0.9; ε = 0.1, Scenario 1)

S prikazov Slika 30 in Slika 31 je razvidno, da agent v replaniranem voznem redu v »dobri« in »slabi« rešitvi upošteva načelo privolitve smeri vožnje in vlak zapusti postajo šele, ko medpostajni odsek ni več zaseden. Strategiji, ki se ju agent nauči pri različnih kombinacijah parametrov α, γ in ε, se razlikujeta. Z analitičnim načinom reševanja smo ugotovili, da za optimalno rešitev v Scenariju 1 ni treba zamenjati vrstnega reda vlakov, torej gre Vlak 1 na pot takoj, ko je to mogoče (po preteku trajanja zamude). Takšno rešitev je tudi agent pri kombinaciji parametrov α = 0,1; γ = 0,3; ε = 0,1 prepoznal kot optimalno. Pri kombinaciji α = 0,7; γ = 0,9; ε = 0,1 pa se je agent naučil, da je smiselno še dodatno podaljševati postanek Vlaku 1, kar seveda ni smiselna rešitev.

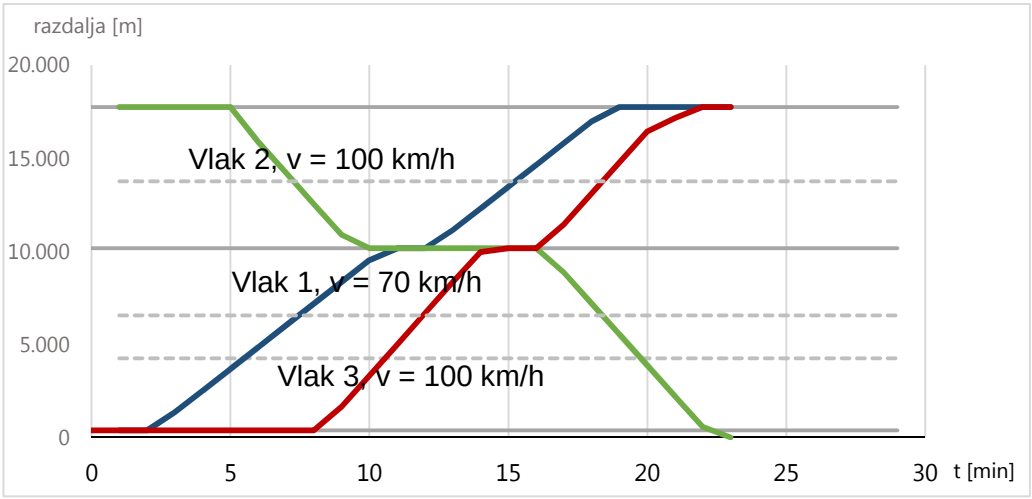
c) Kombinacija zaporednih voženj in voženj v različnih smereh

V tem eksperimentu smo že preverili uspešnost agenta pri reševanju problema replaniranja voženj vlakov za kombinacijo zaporednih voženj in voženj v različnih smereh. Odseki prog med postajama so pogosto zaradi zagotavljanja večje izkoriščenosti železniške infrastrukture razdeljeni na prostorske odseke (glej poglavje 2), zato v tem eksperimentu upoštevamo spremenjeno železniško infrastrukturo. Tudi sicer smo odsek med postajama A in B razdelili na tri prostorske odseke, odsek med postajama B in C pa na dva prostorska odseka. Dolžine odsekov so razvidne s prikaza v nadaljevanju (Slika 32). S tem smo preverili, ali agent upošteva, da je na vsakem prostorskem odseku lahko samo en vlak in hkrati da je na odseku med postajama lahko več zaporednih vlakov (teoretično toliko, kolikor je prostorskih odsekov med postajama).



Slika 32: Eksperiment c – železniška infrastruktura
Figure 32: Experiment c – railway layout

Začetni vozni red, za katerega smo preverili uspešnost učenja z zakasnjeno nagrado, je podan na prikazu Slika 33, kjer črtkane vodoravne črte ponazarjajo meje prostorskih odsekov.



Slika 33: Eksperiment c – vozni red
Figure 33: Experiment c – Timetable

Izvedli smo serijo poskusov za tri scenarije zamud: s primarno zamudo prvega vlaka na začetni postaji v velikosti 5 min, s primarno zamudo drugega vlaka na začetni postaji v velikosti 5 min ter s primarno zamudo tretjega vlaka na začetni postaji v velikosti 5 min. Velikost primarne zamude je v vseh treh scenarijih izbrana naključno.

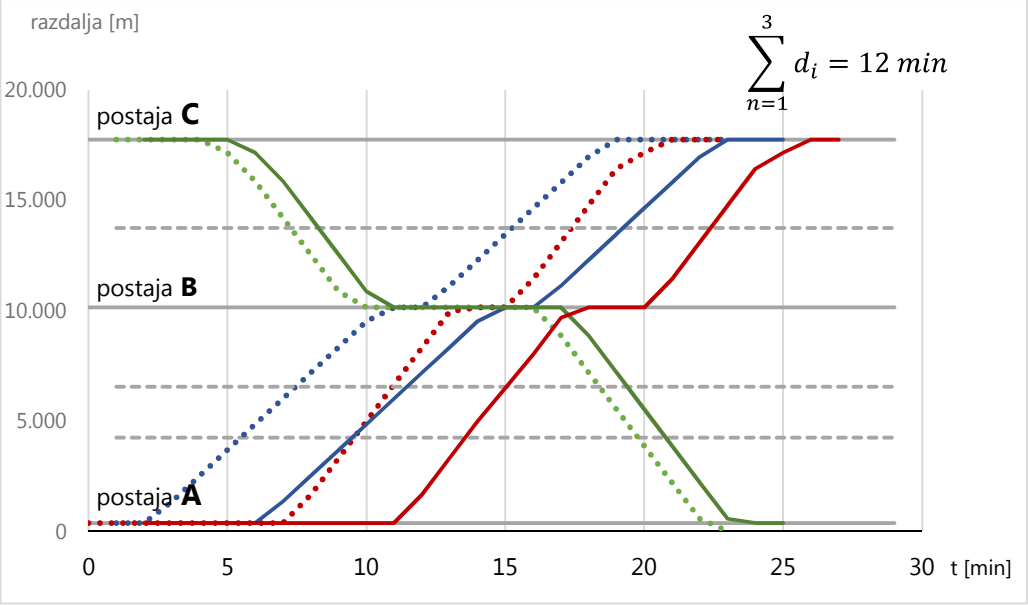
Tudi v tem eksperimentu smo za vse tri scenarije zamud izvedli serijo poskusov za različne kombinacije vrednosti parametrov stopnje učenja, faktorja diskontiranja nagrade ter razmerja med raziskovanjem in izkoriščanjem. Za vsak scenarij zamude smo izvedli učenje z različnimi vrednostmi parametrov, in sicer
$\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ in za vse kombinacije parametrov, torej skupaj 125 kombinacij parametrov. Vsako kombinacijo parametrov α, γ in ε smo simulirali z desetimi različnimi semeni za generacijo naključnih vrednosti. Rezultati za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije so podani v Prilogi C, komentar rezultatov pa je v nadaljevanju.

Preglednica 4: Eksperiment c – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov α, γ in ε.
Table 4: Experimnet c – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account.

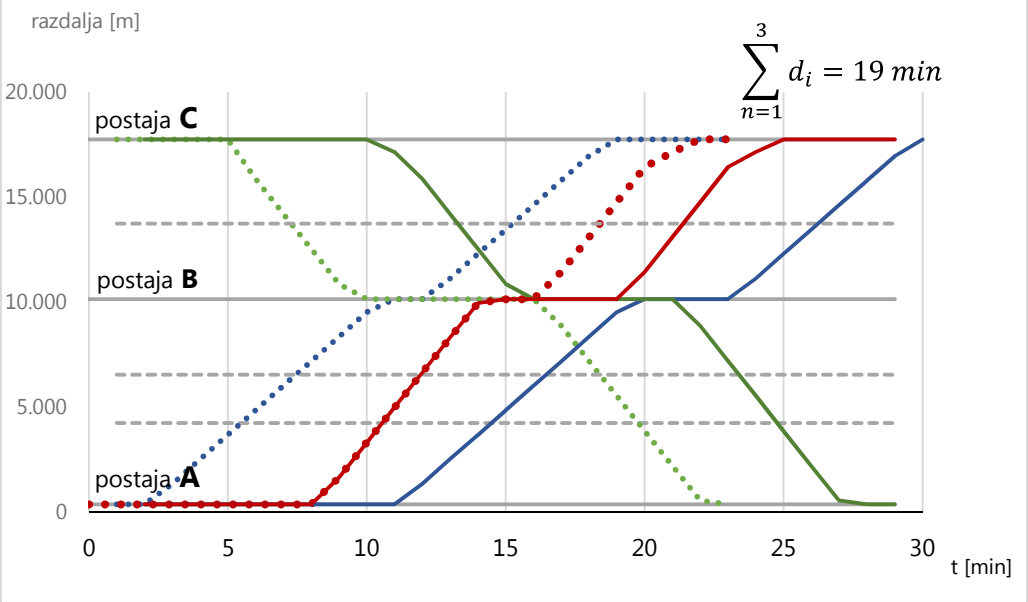
	Scenarij 1			**Scenarij 2**			**Scenarij 3**		
$\sum d_{i,min}$	**Št. ponovitev**			**Št. ponovitev**			**Št. ponovitev**		
	50	**100**	**150**	**50**	**100**	**150**	**50**	**100**	**150**
6				29 %	7 %	2 %			
7				13 %	10 %	5 %			
8				8 %	12 %	16 %	1 %	/	1 %
9				10 %	16 %	6 %	14 %	2 %	/
10				4 %	9 %	8 %	16 %	8 %	3 %
11				11 %	2 %	15 %	15 %	6 %	2 %
12	10 %	3 %	1 %	8 %	5 %	4 %	21 %	22 %	24 %
13	42 %	24 %	28 %	11 %	15 %	22 %	10 %	17 %	16 %
14	22 %	20 %	11 %	2 %	10 %	4 %	6 %	10 %	10 %
15	16 %	24 %	17 %	3 %	9 %	4 %	10 %	11 %	8 %
16	6 %	13 %	10 %	/	2 %	4 %	3 %	6 %	7 %
17	2 %	6 %	9 %	/	2 %	2 %	2 %	5 %	7 %
18	1 %	4 %	9 %	/	2 %	3 %	2 %	3 %	5 %
19	2 %	1 %	8 %	/	/	3 %	/	5 %	2 %
20	/	2 %	5 %	/	/	1 %	/	2 %	6 %
21	/	3 %	1 %	/	/	/	/	3 %	5 %
22	/	1 %	2 %	/	/	1 %	/	/	1 %
23	/	/	/	/	/	/	/	/	2 %
24	/	/	/	/	/	/	/	/	1 %

Agent je najnižjo vrednost za skupne zamude v končnem stanju v velikosti 12 min izračunal v 10 %, če se je učil 50 ponovitev. Po Scenariju 2 agent v 29 % oz. manj, odvisno od števila ponovitev, izračuna najnižjo vrednost zamud (šest minut). Po Scenariju 3 agent najnižjo vrednost zamud izračuna v zanemarljivo majhnem deležu poskusov. Kot smo pričakovali, glede na rezultate eksperimentov 3a in 3b, se tudi v tem eksperimentu agentovo znanje s povečanjem števila ponovitev poslabšuje, rezultati pa so v velikem razponu.

V nadaljevanju podajamo primera agentovih rešitev za Scenarij 1. Na prikazih so s črtkano vodoravno črto označene lokacije prostorskih signalov, ki odsek med postajama delijo na prostorske odseke. V vsakem takšnem odseku je v danem trenutku lahko le en vlak.



Slika 34: Eksperiment c – replanirani vozni red (α = 0,3; γ = 0,3; ε = 0,3, Scenarij 1)
Figure 34: Experiment c – Rescheduled timetable (α = 0.3; γ = 0.3; ε = 0.3, Scenario 1)



Slika 35: Eksperiment c – replanirani vozni red (α = 0,7; γ = 0,3; ε = 0,3, Scenarij 1)
Figure 35: Experiment c – Rescheduled timetable (α = 0.7; γ = 0.3; ε = 0.3, Scenario 1)

Prikaza Slika 34 in Slika 35 sta dokaz, da agent upošteva načeli, da je na vsakem odseku samo en vlak ter da je na odseku med postajama lahko več vlakov. Kombinaciji parametrov, za katere prikazujemo rezultata učenja, sta izbrani tako, da je razvidno, da izbira vrednosti parametra α vpliva na uspešnost učenja. Na prikazu Slika 34 agent predlaga, da vlak, ki zamuja (Vlak 1), zapusti postajo takoj, ko je to mogoče; vrstnega reda vlakov ne spreminja. Velikost skupne zamude v končnem stanju je devet minut. Obratno pa na prikazu Slika 35 agent predlaga zamenjavo vrstnega reda vlakov. Predlagani replanirani vozni red je sicer korekten (brez konfliktov), vendar je v tem primeru skupna zamuda za štiri minute večja, kot je v primeru, ko agent ohrani vrstni red.

Spoznanja, pridobljena v eksperimentu, lahko povzamemo z naslednjimi besedami:

- različne kombinacije parametrov so različno uspešne pri reševanju optimizacije problema replaniranja vlakov (več v poglavju 3.3.11);
- v primeru scenarijev, ko je treba podaljšati postanek vlaku, je agent manj uspešen (velja tako za zaporedne vožnje kot tudi vožnje v različnih smereh);
- uspešnost učenja Q z zakasnjeno nagrado ni zadovoljiva;
- s povečevanjem števila ponovitev se agentovo znanje poslabšuje.

3.3.10 Učenje Q z zakasnjeno nagrado in sledmi

V nadaljevanju raziskave smo iskali možnosti za izboljšanje uspešnosti algoritma, predvsem pa smo poskušali najti način, da bi se agentovo znanje že pri majhnem številu ponovitev izboljšalo in konvergiralo proti optimalni vrednosti.

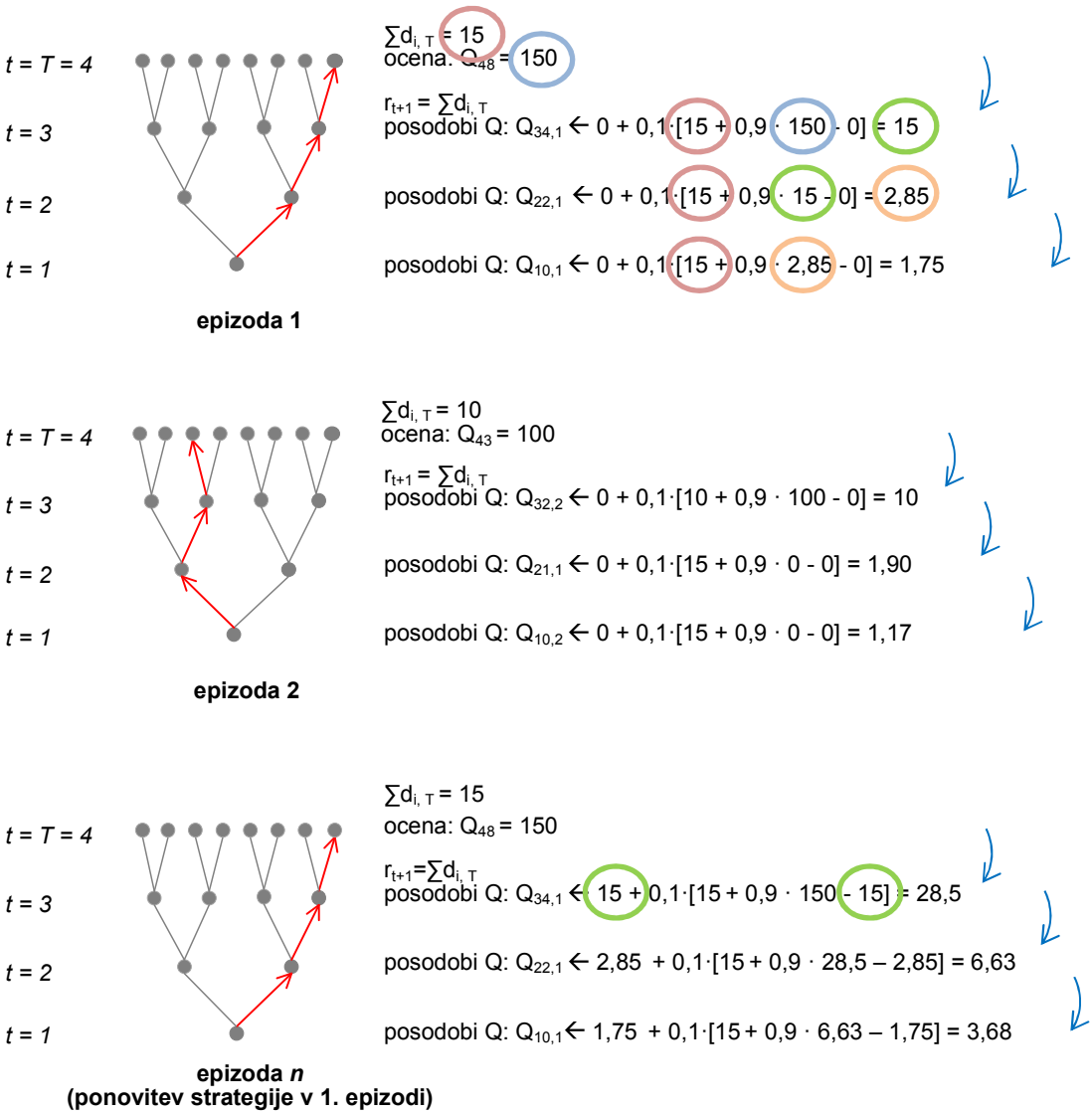
Primer replaniranja voženj vlakov je specifičen, saj informacijo o uspešnosti strategije dobimo šele v končnem stanju. Pri uporabi učenja Q z zakasnjenimi nagradami se je izkazalo, da agent zna reševati tovrstne probleme, vendar tudi pri enostavnih primerih algoritem ni zadovoljivo učinkovit. V Eksperimentu 3 smo pokazali, da se informacija o uspešnosti strategije (torej o velikosti nagrade) od končnega stanja proti začetnemu prenaša postopoma. Pri replaniranju vlakov je pomembno, da se ta informacija čim prej prenese do začetnega stanja, saj je pomembno, kdaj imajo vlaki odhod z začetne postaje ter v kakšnem vrstnem redu. Torej je treba poiskati način, kako to informacijo o uspešnosti strategije učinkovito posredovati proti začetnemu stanju. Eden izmed načinov za povečanje učinkovitosti metode učenja Q je uporaba sledi (ang. *eligibility traces*), to pomeni, da se za nazaj posodobijo vrednosti Q v nizu že izvedenih akcij. V literaturi sta najpogosteje obravnavani dve metodi, in sicer Watkinsova in Pengova (Sutton in Barto, 1998).

Po Watkinsovi metodi sledi uporabimo samo za akcije, izvedene do prve raziskovalne akcije in prve naslednje požrešne akcije, nato pa informacijo pripišemo vsem že obiskanim parom stanj in akcij. Slabost metode je, da se s prekinitvijo sledi vsakič, ko agent raziskuje (ne izbere požrešne akcije), izgubijo prednosti uporabe sledi. Slabost je izrazita predvsem v začetnih fazah učenja, ko agent običajno veliko raziskuje in se vrednosti matrike Q posodobijo samo za zadnji ali zadnja dva koraka (Sutton in Barto, 1998). Metoda Peng ne razlikuje med raziskovanjem agenta in požrešnimi akcijami, torej se uporaba sledi pri posodabljanju vrednosti ne prekine. Slabost metode Peng je zelo kompleksna implementacija, poleg tega ni nujno, da z uporabo metode Peng rešitev konvergira k optimalni rešitvi (Sutton in Barto, 1998).

Za namen reševanja replaniranja voženj vlakov predlagamo drugačen pristop uporabe sledi. Iz izkušenj v Eksperimentu 3 je jasno, da je uporaba učenja Q z zakasnjeno nagrado bolj uspešna od učenja Q, zato smo ohranili princip nagrajevanja. Torej agent v vseh stanjih, razen v končnem, prejme nagrado $r_{t+1} = 0$; v končnem stanju pa prejme nagrado $r_{t+1} = -\sum_{i=1}^{n} d_{i,T}$. Slabost takšne implementacije učenja Q je po našem mnenju v trajanju potovanja informacije o uspešnosti strategije od končnega proti začetnemu stanju, zato smo v tem eksperimentu preverili učinkovitost posodabljanja matrike Q od končnega proti začetnemu stanju po postopku, opisanem v nadaljevanju.

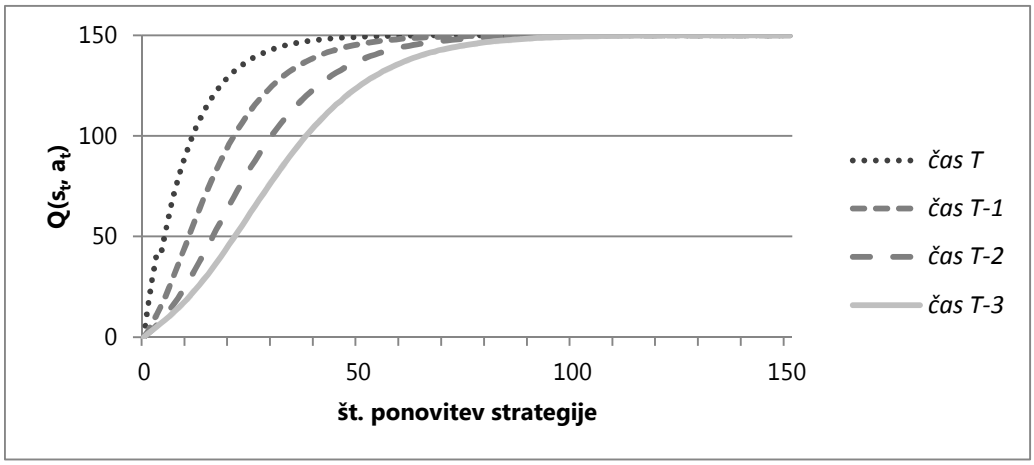
Ker uporabljamo učenje Q za končno število stanj, tudi v tem eksperimentu vrednost Q v končnem stanju ocenimo po enačbi (9), vrednosti matrike Q v ostalih obiskanih stanjih posodobimo po enačbi (1), kjer v vseh korakih upoštevamo vrednost končne nagrade $r_{t+1} = -\sum_{i=1}^{n} d_{i,T}$ (glej prikaz Slika 36), vrednosti matrike Q pa se po končani epizodi posodabljajo od končnega stanja proti začetnemu. Podobno kot Watkins tudi mi predlagamo prekinitev verige posodabljanja vrednosti Q. Vendar za razliko od njegove metode verige ne prekinemo v odvisnosti od izbrane akcije, temveč v odvisnosti od trenutne in nove vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$. Predlagamo, da se posodabljanje vrednosti Q po sledeh strategije izvaja ne glede na to, ali je bila akcija v posameznem stanju požrešna ali ne, in da se posodabljanje prekine v stanju, kjer bi vrednost Q posodobili na nižjo vrednost glede na trenutno.

Predlagani princip nagrajevanja in posodabljanja matrike Q je prikazan v nadaljevanju, temelji pa na naslednjih predpostavkah: matrika Q je inicializirana na vrednost 0, $\alpha = 0{,}1$, $\gamma = 0{,}9$, $r_T = 15$. Z rdečimi puščicami je označena smer gibanja agenta, z modrimi smer posodabljanja matrike Q. Z rdečimi krogi je poudarjeno upoštevanje nagrade (agent nagrado prejme v končnem stanju in njeno vrednost upošteva pri vseh posodobitvah), z modro, zeleno oz. oranžno barvo pa je ponazorjeno upoštevanje vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ v sledečih izračunih.



Slika 36: Princip učenja Q z zakasnjeno nagrado in sledmi
Figure 36: How Q learning with delayed reward and eligibility traces works

Na prikazu v nadaljevanju (Slika 37) je prikazana konvergenca vrednosti $Q(s_t, a_t)$ v času T, $T-1$, $T-2$ ter $T-3$ pri uporabi učenja Q z zakasnjenimi nagradami in v doktorski disertaciji predlaganim načinom upoštevanja posodabljanja matrike Q po sledeh strategije. Vhodni podatki so: $\alpha = 0{,}1$, $\gamma = 0{,}9$ in $r_T = \sum_{i=1}^{n} d_{i,T} = 15$; iz enačbe (9) sledi $\max_{a_{(t+1)}} Q(s_T, a_T) = 150$.



Slika 37: Konvergenca vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ pri $\alpha = 0{,}1$, $\gamma = 0{,}9$, $r_T = 15$
Figure 37: Convergence of $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ values with $\alpha = 0.1$, $\gamma = 0.9$, $r_T = 15$

S prikaza Slika 37 je razvidno, da na posamezni veji vse vrednosti $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ konvergirajo k ocenjeni vrednosti $\max_{a_{(t+1)}} Q(s_T, a_T)$. Poudarjamo, da posamezna veja ne poteka nujno od začetnega do končnega stanja, saj verigo parov stanj in akcij posodabljamo tako dolgo, dokler ni nova vrednost $\mathrm{Q}(\mathrm{s_t}, \mathrm{a_t})$ nižja od trenutne.

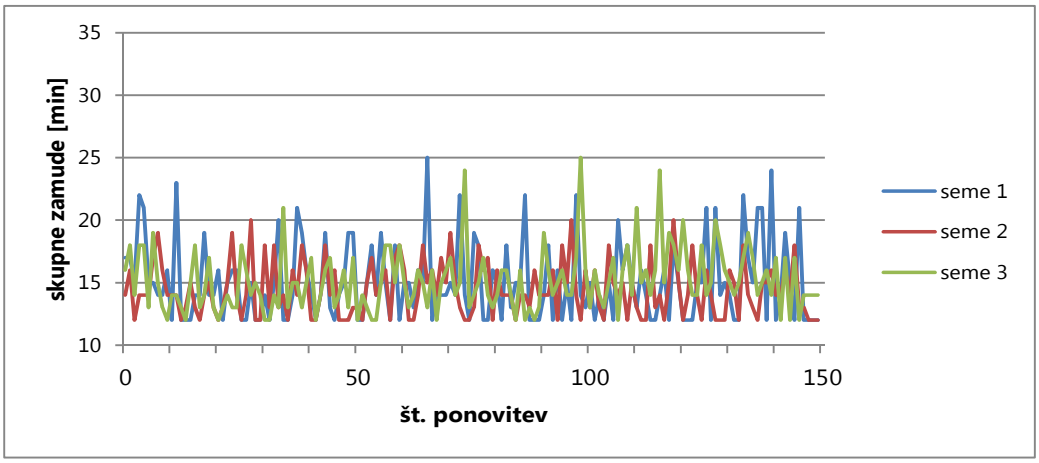
V nadaljevanju zaradi zagotavljanja primerljivosti implementacije učenja Q z zakasnjenimi nagradami in implementacije učenja Q z zakasnjeno nagrado in sledmi obravnavamo enake konfiguracije železniške infrastrukture, enake vozne rede in enake scenarije zamud kot v eksperimentu, kjer smo upoštevali učenje Q z zakasnjeno nagrado.

a) Vlaka vozita zaporedno

Zaradi primerljivosti pristopov učenja Q z zakasnjeno nagrado in pristopom, predlaganim v tem poglavju, ponovno obravnavamo primer železniške infrastrukture, prikazan na prikazu Slika 24 (območje treh postaj, ki so oddaljene 5 km in 8 km), voznega reda na prikazu Slika 25 (dva vlaka, ki vozita v isti smeri) in tri scenarije zamud ($\mathrm{d_{1,1}} = 6\ \mathrm{min}$, $\mathrm{d_{1,1}} = 7\ \mathrm{min}$, $\mathrm{d_{1,1}} = 8\ \mathrm{min}$). Spomnimo, v prvem scenariju je začetna zamuda prvega vlaka takšna, da za optimalno rešitev problema ni treba spremeniti vrstnega reda vlakov, v drugih dveh primerih pa je zamenjava vrstnega reda vlakov nujna.

V vseh primerih je optimalna rešitev $\sum \mathrm{d_{i,T}} = 12\ \mathrm{min}$.

Učenje agenta po metodi, kjer agent prejme nagrado v končnem stanju, vrednosti matrike Q pa se posodobijo od končnega proti začetnemu stanju, je prikazan v nadaljevanju. Učenje agenta poteka z upoštevanjem enakih vhodnih podatkov, kot so upoštevani pri učenju z zakasnjeno nagrado (prikaz Slika 26).



Slika 38: Eksperiment a – krivulje učenja za različna semena (α = 0,3; γ = 0,3; ε = 0,3)
Figure 38: Experiment a – Learning curves for different seeds (α = 0.3; γ = 0.3; ε = 0.3)

S prikaza Slika 38 je razvidno, da agentovo znanje konvergira proti nižjim vrednostim skupnih zamud v končnem stanju, razpršenost rezultatov v posameznih ponovitvah je veliko nižja kot v primeru uporabe učenja Q z zakasnjeno nagrado. Tudi v predlagani implementaciji učenja Q je proces učenja odvisen od izbire semena za generacijo naključnih spremenljivk, zato se tudi pri tej metodi učenje izvede z različnimi semeni; kot rezultat učenja pa upošteva minimalna vrednost (primer na zgornjem prikazu, kjer kot rezultat upoštevamo rešitev, dobljeno s semenom 1 ali 2).

Rezultati učenja za vseh 125 kombinacij parametrov ($\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$), za različno število ponovitev učenja in za tri scenarije so podani v Prilogi D in povzeti v Preglednici 5, v nadaljevanju poglavja pa so podane le ključne ugotovitve.

Preglednica 5: Eksperiment a – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov α, γ in ε.
Table 5: Experiment a – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account.

$\sum d_{i,min}$	Scenarij 1			Scenarij 2			Scenarij 3		
	Št. ponovitev			Št. ponovitev			Št. ponovitev		
	50	100	150	50	100	150	50	100	150
12	100 %	100 %	98 %	59 %	65 %	64 %	94 %	94 %	92 %
13	/	/	2 %	9 %	3 %	4 %	6 %	6 %	8 %
14	/	/	/	32 %	32 %	32 %	/	/	/

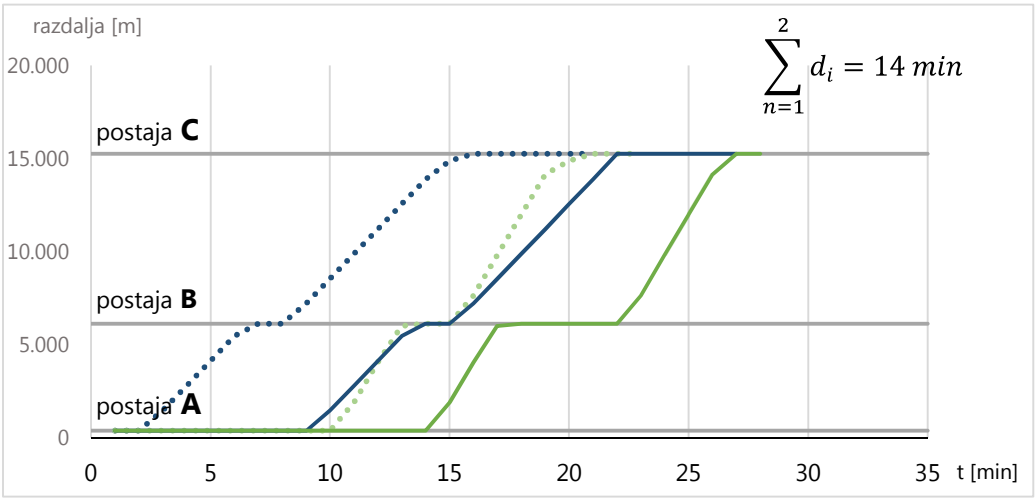
Iz Preglednice 5 je razvidno, da se je agent v primeru Scenarija 1 uspešno naučil replanirati, saj je pri 50 in 100 ponovitvah našel optimalno rešitev v vseh kombinacijah parametrov α, γ in ε, v primeru s 150 ponovitvami pa je uspešen v 98 %. Rezultati manj uspešnega učenja se

od optimalne vrednosti razlikujejo le za eno minuto, torej je učenje precej bolj učinkovito kot učenje z upoštevanjem zakasnjene nagrade.

V primeru Scenarija 2 je agent nekoliko manj uspešen, saj najde optimalno rešitev pri 50 ponovitvah v 59 %, pri 100 ponovitvah v 65 % in pri 150 ponovitvah v 64 %. Ker so vsi rezultati znotraj treh minut in ker se s številom ponovitev znanje agenta izboljšuje, smatramo učenje za učinkovito.

V primeru Scenarija 3, pri katerem mora v primerjavi s Scenarijem 2 agent v manjši meri podaljšati postanek Vlaku 1 za dosego optimalne rešitve, je agent pričakovano dosegel boljše znanje – optimalno rešitev je našel v 94 % (oz. 92 % pri 150 ponovitvah), v ostalih poskusih pa je bila zamuda le za minuto minuto večja.

V nadaljevanju podajamo primer replaniranega voznega reda, ki ga predlaga agent. Primer je izračunan z enakimi vhodnimi podatki kot primer, ki se je pri učenju Q z zakasnjeno nagrado izkazal kot neuspešen (glej prikaz Slika 28).



Slika 39: Eksperiment a – replanirani vozni red (α = 0,3; γ = 0,7; ε = 0,3, Scenarij 2)
Figure 39: Experiment a – Rescheduled timtable (α = 0.3; γ = 0.7; ε = 0.3, Scenario 2)

Prikaz dokazuje, da smo ohranili agentovo upoštevanje varnostnega načela, da je na enem odseku samo en vlak (vozni red, ki ga predlaga agent, je brez konfliktov). Analitično smo določili, da je za optimalno rešitev treba spremeniti vrstni red vlakov. Tudi v primeru uporabe učenja Q z zakasnjeno nagrado in sledmi se agent pri kombinaciji parametrov α = 0,3; γ = 0,7; ε = 0,3 ni naučil optimalne strategije, je pa agentovo znanje boljše, saj so pri enakih vhodnih podatkih in predlagani implementaciji učenja Q zamude manjše za 12 minut kot pri učenju Q z zakasnjeno nagrado.

b) Vlaka vozita v različnih smereh

V tem eksperimentu uspešnost predlagane implementacije učenja Q preverjamo na železniškem omrežju, podanem na prikazu Slika 24, in za vozni red, podan na prikazu Slika 29. Spomnimo, železniško infrastrukturo sestavljajo tri postaje, A, B in C, med postajama A in B je medpostajni odsek, dolg 5 km, med postajama B in C pa 8 km. Počasnejši vlak (Vlak 1) vozi s postaje A proti postaji B, hitrejši vlak (Vlak 2) pa s postaje C proti postaji A; po voznem redu se križata na postaji B. Obravnavamo tri scenarije zamud, in sicer po Scenariju 1 Vlak 1 na začetni postaji zamuja pet minut.

Analitično določena optimalna rešitev je devet minut, pri čemer se mesto križanja vlakov spremeni. Po Scenariju 2 Vlak 1 zamuja šest minut in je za optimalno rešitev (deset minut) treba zamenjati križanje vlakov. To pomeni, da je treba Vlaku 1 na postaji A podaljšati postanek; Vlak 1 gre z začetne postaje šele, ko Vlak 2 prispe na cilj (na postajo 1). Po Scenariju 3 Vlak 1 zamuja sedem minut, optimalna rešitev je enaka kot v Scenariju 2.

Rezultati učenja za vseh 125 kombinacij parametrov ($\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$), za različno število ponovitev učenja in za tri scenarije so podani v Prilogi E in povzeti v Preglednici 6, v nadaljevanju poglavja pa so podane le ključne ugotovitve.

Preglednica 6: Eksperiment b – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov α, γ in ε.
Table 6: Experiment b – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account.

$\sum d_{i,min}$	Scenarij 1			Scenarij 2			Scenarij 3		
	Št. ponovitev			Št. ponovitev			Št. ponovitev		
	50	100	150	50	100	150	50	100	150
9	98 %	98 %	100 %	69 %*	74 %*	77 %*	100 %*	100 %*	98 %*
10	2 %	2 %	/	7 %	6 %	3 %	/	/	2 %
11	/	/	/	24 %	20 %	20 %	/	/	/

* Pri analitičnem reševanju smo upoštevali, da Vlak 2 prispe na končno postajo v 12. minuti, torej gre lahko Vlak 1 s postaje A v $t = 12\ min$. Agent učenja Q upošteva, da vlak lahko zasede prosti odsek takoj, ko je to mogoče. V obravnavanem primeru Vlak 2 zapusti medpostajni odsek med postajama A in B v 11. minuti in se v začetku 12. minute ustavi pred signalnim znakom na postaji A. Sprostitev medpostajnega odseka v $t = 11\ min$ omogoča, da Vlak 1 zapusti začetno postajo že v 11. minuti, zato agent dobi optimalen rezultat zamud 9 min, analitično pa smo kot optimalno rešitev določili skupno zamudo v velikosti 10 min.

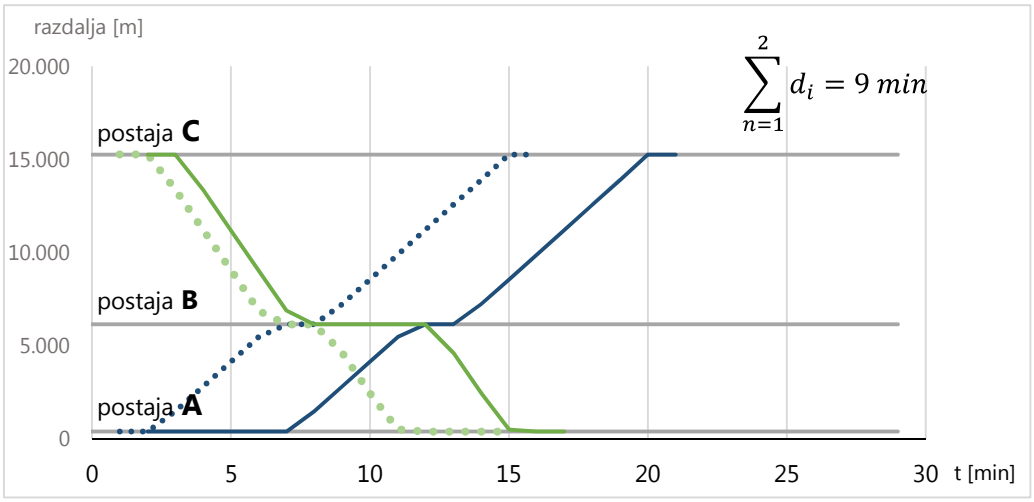
Iz Preglednice 6 je razvidno, da se je v primeru Scenarija 1 agent naučil optimalne strategije v 98 %, v primeru 100 ponovitev učenja pa v vseh kombinacijah parametrov α, γ in ε.

Pričakovano je agent nekoliko manj uspešen pri reševanju zamude po Scenariju 2, ko mora za optimalno rešitev podaljšati postanek Vlaku 1, kljub temu je optimalno rešitev predlagal v 69 %, če se je učil 50 ponovitev, 74 % pri 100 ponovitvah in 77 % pri 150 ponovitvah. Ostale rešitve se razlikujejo za največ dve minuti.

V primeru zamude po Scenariju 3, kjer je za optimalno rešitev treba Vlaku 1 podaljšati postanek na začetni postaji, vendar je takšnih akcij potrebnih manj kot v Scenariju 2, je bil agent zelo uspešen, saj je optimalno rešitev predlagal v 100 % oz. v 98 % v primeru, ko je imel na voljo 150 ponovitev učenja.

Rezultati, pridobljeni z učenjem Q z zakasnjeno nagrado in sledmi, so za vse scenarije zamud boljši, kot so pri učenju Q z zakasnjeno nagrado.

V nadaljevanju podajamo primer replaniranega voznega reda.



Slika 40: Eksperiment b – replanirani vozni red (α = 0,7; γ = 0,9; ε = 0,1, Scenarij 1)
Figure 40: Experiment b – Rescheduled timtable (α = 0.7; γ = 0.9; ε = 0.1, Scenario 1)

Na prikazu Slika 40 je rešitev pri enakih vhodnih podatkih enaka kot na prikazu Slika 31, ki je podan kot primer neuspešnega replaniranja po metodi učenja Q z zakasnjeno nagrado. Iz primerjave grafikonov voznih redov je razvidno, da je v Eksperimentu 3b vlakoma 1 in 2 podaljševal postanek, čeprav za to ni bilo prometnih ali varstvenih zahtev; skupne zamude v tem primeru so bile velike 22 minut. Z implementacijo predlaganega algoritma smo dosegli, da je agent izboljšal znanje, in tudi v primeru replaniranja voženj vlakov, ki vozita iz nasprotnih smeri, predlagal brezkonflikten vozni red, kjer je skupna zamuda obeh vlakov velika devet minut.

c) Kombinacija zaporednih voženj in voženj v različnih smereh

V tem poglavju bomo preverili uporabnost predlagane implementacije učenja Q za bolj kompleksen problem replaniranja vlakov. Ponovno upoštevamo železniško infrastrukturo med tremi postajami, kjer je odsek med postajama A in B razdeljen na tri prostorske odseke, odsek med postajama B in C pa na dva (glej prikaz Slika 32). Uspešnost agenta pri prepoznavanju različnih varnostnih načel smo preverili z voznim redom, v katerem so trije vlaki; dva vozita v isti smeri in eden v nasprotni. Vlaki se križajo na postaji B.

Uspešnost predlaganega algoritma smo preverili za tri scenarije zamud, in sicer za scenarij, ko na začetni postaji zamujajo enkrat Vlak 1, drugič Vlak 2 in tretjič Vlak 3. Velikost zamude v vseh scenarijih je pet minut.

Rezultati učenja za vseh 125 kombinacij parametrov ($\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, $\varepsilon \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$), za različno število ponovitev učenja in za tri scenarije so podani v Prilogi F in povzeti v Preglednici 7, v nadaljevanju poglavja pa so podane le ključne ugotovitve.

Preglednica 7: Eksperiment c – uspešnost algoritma. Upoštevane so minimalne vrednosti skupnih zamud, izračunane z desetimi semeni za generacijo naključnih spremenljivk za vseh 125 različnih kombinacij parametrov α, γ in ε.
Table 7: Experiment c – efficiency of algorithm. Minimal values of total delays obtained with 10 seeds for all 125 different combination of parameters α, γ and ε are taken into account.

	Scenarij 1			**Scenarij 2**			**Scenarij 3**		
$\sum d_{i,min}$	**Št. ponovitev**			**Št. ponovitev**			**Št. ponovitev**		
	50	**100**	**150**	**50**	**100**	**150**	**50**	**100**	**150**
6				94 %	96 %	95 %			
7				6 %	4 %	5 %			
8				/	/	/	94 %	96 %	96 %
9				/	/	/	6 %	4 %	4 %
10				/	/	/	/	/	/
11				/	/	/	/	/	/
12	70 %	72 %	71 %	/	/	/	/	/	/
13	28 %	22 %	25 %	/	/	/	/	/	/
14	2 %	6 %	4 %	/	/	/	/	/	/

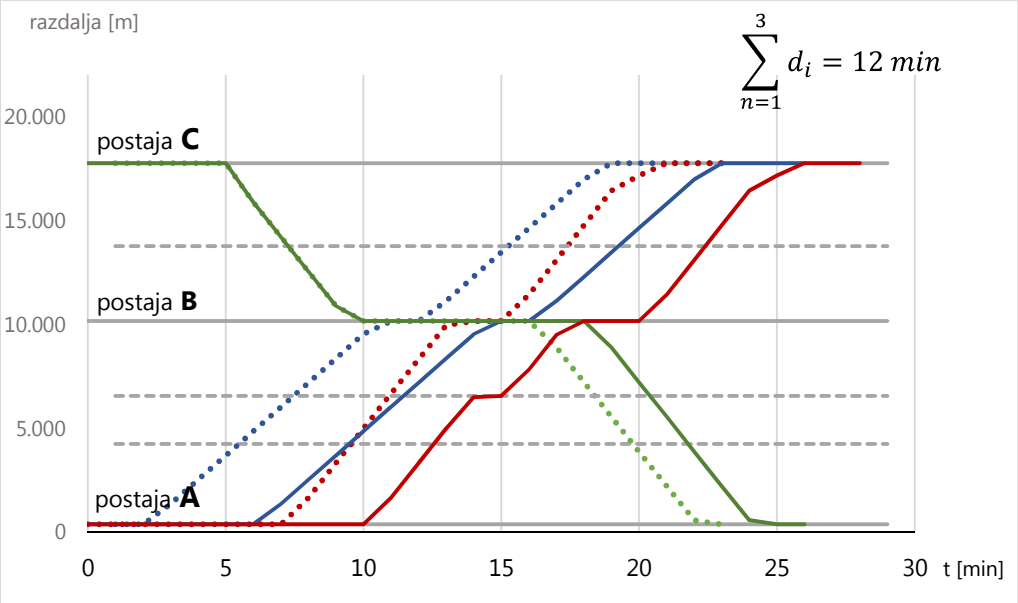
Tudi iz Preglednice 7 je hitro razvidno, da je predlagana implementacija učenja Q uspešna, saj agent v velikem deležu poskusov najde optimalno rešitev, hkrati pa je razlika med optimalno in najslabšo rešitvijo majhna; le v primeru Scenarija 1 je ta razlika velika dve minuti, sicer le eno minuto.

V primeru Scenarija 1 je agent našel v približno 70 % kombinacijah parametrov α, γ in ε, v primeru uporabe učenja Q je bil ta delež zanemarljivo majhen (10 % oz. samo 3 % in 1 %).

Pri reševanju problema zamud vlakov po Scenariju 2 in Scenariju 3 je bil agent še bolj uspešen, saj je optimalno rešitev našel v skoraj vseh kombinacijah parametrov (94–96 %).

Tudi na bolj kompleksnem primeru smo dokazali, da je učenje Q z zakasnjeno nagrado in sledmi uspešno pri reševanju problema replaniranja vlakov, kjer je kot uspešnost učenja upoštevana skupna zamuda vseh vlakov na končni postaji.

V nadaljevanju podajamo primer replaniranega voznega reda. Vhodni podatki so enaki kot v primeru, prikazanem na prikazu Slika 35, ki smo ga predstavili kot primer manj uspešnega učenja agenta.



Slika 41: Eksperiment c – replanirani vozni red (α = 0,7; γ = 0,3; ε = 0,3, Scenarij 1)
Figure 41: Experiment c – Rescheduled timtable (α = 0.7; γ = 0.3; ε = 0.3, Scenario 1)

Na prikazu Slika 41 je novi brezkonfliktni vozni red, ki ga je predlagal agent. S prikaza je razvidno, da je agent v 14. minuti Vlak 3 (trasa tega vlaka je na prikazu označena z rdečo) ustavil pred uvozom na tretji prostorski odsek, saj je bil le-ta zaseden z Vlakom 1 (trasa, označena z modro). Torej je agent pravilno prepoznal temeljno varnostno načelo in predlagal ustrezno rešitev. V kriterijski funkciji, ki jo uporabljamo v eksperimentih, upoštevamo skupne zamude vseh vlakov na njihovih končnih postajah. Ustavljanje vlakov pred prostorskimi signali vodi v povečano porabo energije, zato so takšni postanki nezaželeni. Za uspešno replaniranje tudi z vidika porabe energije bi dispečer pri implementaciji predlaganega voznega reda upošteval, da Vlak 3 zapusti postajo A eno minuto kasneje, in s tem preprečil ustavljanje vlaka.

3.3.11 Parametrična študija

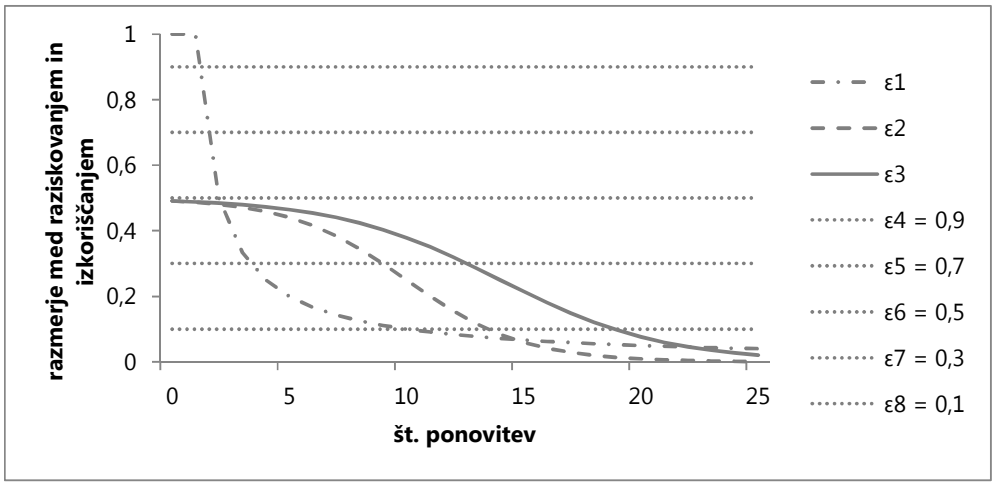
V poglavjih 3.3.8, 3.3.9 in 3.3.10 smo ugotavljali uspešnost različnih implementacij učenja Q za reševanje problema replaniranja voženj vlakov, pri čemer smo upoštevali, v kolikšnih kombinacijah parametrov α, γ in ε je bil agent uspešen. Pri uporabi učenja Q na realnih primerih se reševanje problema ne izvaja za različne vrednosti in kombinacije parametrov, temveč se za vsak problem, ki ga rešujemo z učenjem Q, določijo vrednosti parametrov, pri katerih je največja verjetnost dobre rešitve.

Analizo uspešnosti različnih kombinacij parametrov smo izvedli za železniško infrastrukturo s tremi postajami, kjer sta odseka med postajama razdeljena na tri oz. dva prostorska odseka (glej prikaz Slika 32) in tri vlake (glej vozni red na prikazu Slika 33).

V literaturi se za parametre α, γ in ε najpogosteje uporablja konstantna vrednost, zato smo tudi v študiji preverili uspešnost algoritma za konstantne vrednosti in različne kombinacije parametrov (pet različnih vrednosti vsakega parametra). Dodatno smo za parameter ε po analogiji z učenjem živih bitij (v otroštvu veliko raziskujejo in se učijo ter v odrasli dobi izkoriščajo pridobljeno znanje) vedenje agenta razdelili v fazo bolj aktivnega učenja in fazo izkoriščanja pridobljenega znanja. V parametrični študiji smo preverili uspešnost različnih kombinacij parametrov, kjer sta parametra α in γ v vseh ponovitvah učenja konstantna, parameter ε pa je konstanten ali pa se s časom spreminja (glej prikaz Slika 42). V parametrični študiji smo preverili naslednje kombinacije (skupaj 200 kombinacij):

- $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$;
- $\gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$;
- $\varepsilon(x) \in \left\{x^{-1}, \frac{0{,}5}{1+e^{(10*(x-0.4*25)/25))}}, \frac{0{,}5}{1+e^{(10*(x-0.4*35)/35))}}, 0.1, 0.3, 0.5, 0.7, 0.9\right\}$,
  X je zaporedna številka ponovitve.

Na prikazu v nadaljevanju podajamo krivulje ε-požrešnih funkcij v raziskavi.



Slika 42: Različne ε-požrešne funkcije, upoštevane v raziskavi
Figure 42: Different ε–greedy functions studied in the research

S prikaza Slika 42 je razvidno, da pri upoštevanju $\varepsilon 1(x) = x^{-1}$ agent v prvi ponovitvi samo raziskuje, nato pa delež raziskovanja hitro pade in v deseti ponovitvi agent raziskuje povprečno samo še v vsaki deseti iteraciji. Pri upoštevanju razmerja med raziskovanjem in izkoriščanjem po enačbi $\varepsilon 2(x) = \frac{0{,}5}{1+e^{(10*(x-0.4*25)/25))}}$ agent v začetnih ponovitvah raziskuje v vsaki drugi iteraciji, nato pa postopoma vedno bolj izkorišča znanje in v zadnjih ponovitvah sledi naučeni strategiji. Pri uporabi enačbe $\varepsilon 3(x) = \frac{0{,}5}{1+e^{(10*(x-0.4*35)/35))}}$ v začetnih ponovitvah agent prav tako raziskuje in izkorišča znanje v razmerju 1 : 1, vendar se raziskovanje zmanjšuje počasneje kot pri enačbi $\varepsilon 2(x)$.

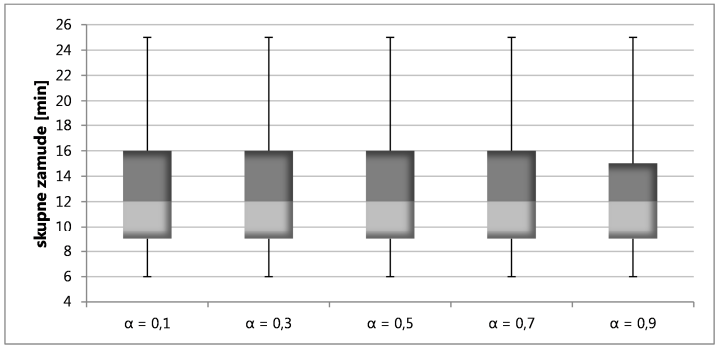
Ker izbira semena za generacijo naključnih spremenljivk vpliva na uspešnost učenja (glej prikaz Slika 26), smo učenje za vsako kombinacijo parametrov α, γ in ε izvedli z desetimi različnimi semeni za generacijo naključnih sprememb; 200 kombinacij parametrov, vsaka kombinacija parametrov je izračunana z desetimi semeni, skupaj torej 2000 poskusov.

Da zagotovimo čim večjo verjetnost dobre rešitve za različne strategije replaniranja voženj vlakov, torej za primere, ko je sprememba v vrstnem redu vlakov potrebna za optimalno rešitev, in za primere, ko le-ta ni potrebna, smo analizo uspešnosti različnih kombinacij parametrov izvedli za pet scenarijev zamud, in sicer:

- Scenarij 1: Vlak 1 na začetni postaji zamudi pet minut;
- Scenarij 2: Vlak 2 na začetni postaji zamudi pet minut;
- Scenarij 3: Vlak 3 na začetni postaji zamudi pet minut;
- Scenarij 4: Vlak 1 na začetni postaji zamudi tri minute in Vlak 2 na začetni postaji zamudi deset minut;
- Scenarij 5: Vlak 1 na začetni postaji zamudi tri minute in Vlak 3 na začetni postaji zamudi pet minut.

Analizo uspešnosti smo izvedli z metodo škatle z brki (ang. *Box plot* ali *whisker diagram*), ki na standardiziran način prikazuje porazdelitev podatkov, in sicer minimalno vrednost, prvi kvartil, mediano, tretji kvartil in največjo vrednost. V analizi smo upoštevali rezultate zadnje (testne) iteracije (glej str. 64).

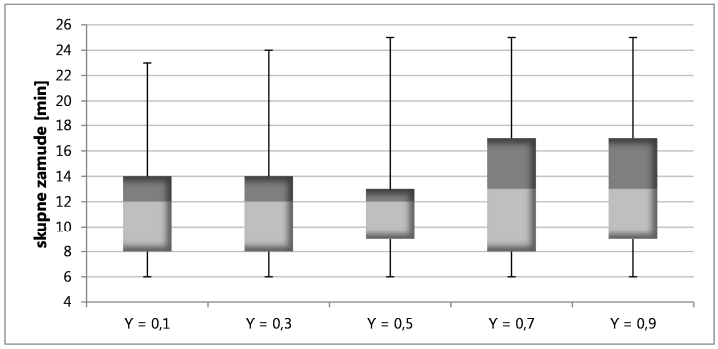
V prvem koraku smo preverili uspešnost algoritma pri različnih vrednostih parametra α.



Slika 43: Skupne zamude vlakov po α
Figure 43: Total train delay by α

Iz analize rezultatov (glej prikaz Slika 43), kjer smo preverjali uspešnost različnih vrednosti parametra α, je razvidno, da je v vseh primerih enaka minimalna vrednost skupnih zamud. Zaradi uteženosti diagrama navzdol je algoritem bolj verjetno uspešen pri izbiri $\alpha = 0{,}9$.

V nadaljevanju smo določili vrednost parametra γ, pri katerem lahko z največjo verjetnostjo pričakujemo kvaliteten rezultat učenja pri pogoju $\alpha = 0{,}9$.

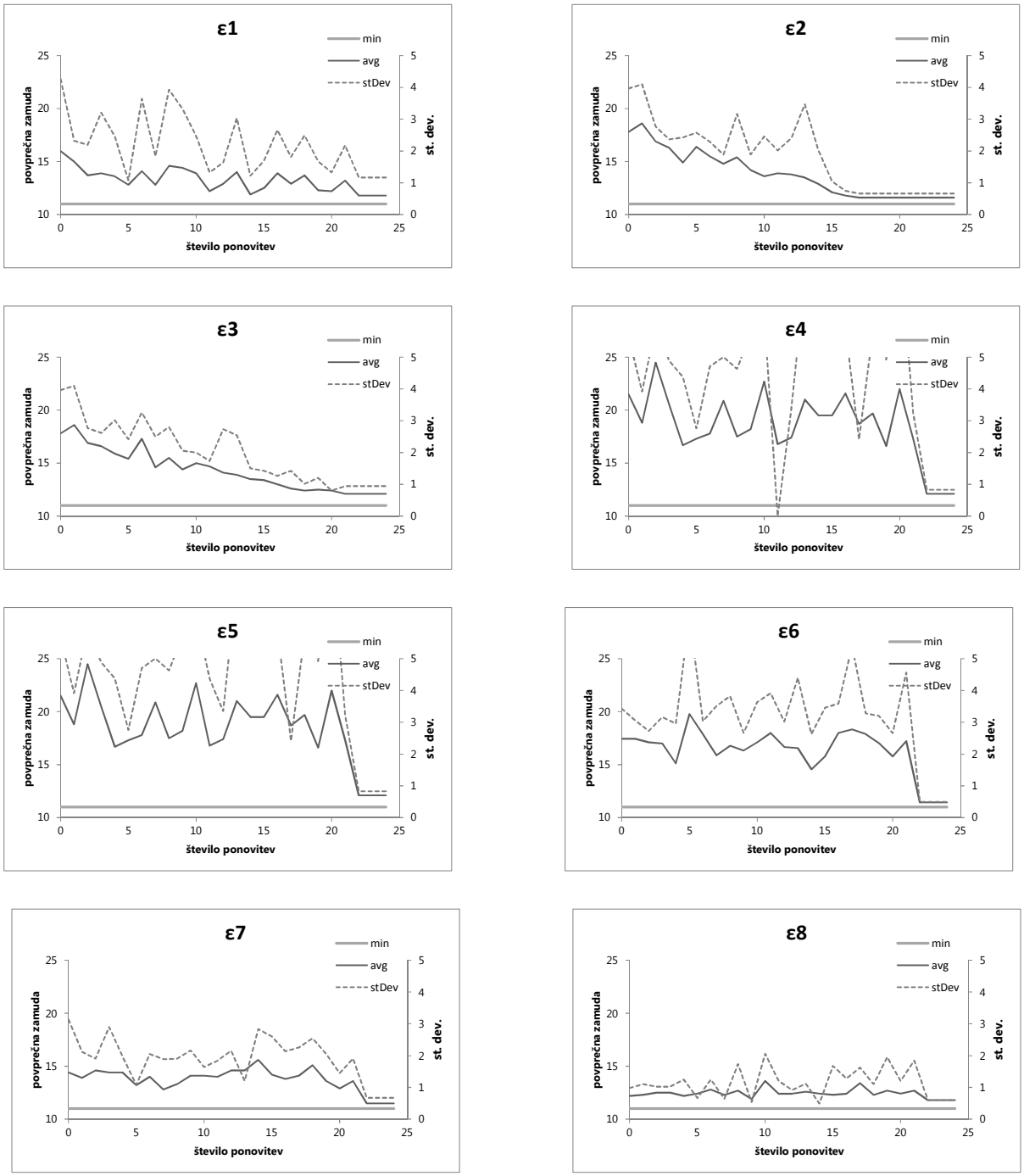


Slika 44: Skupne zamude vlakov po γ pri pogoju $\alpha = 0{,}9$
Figure 44: Total train delay by γ, where $\alpha = 0.9$

Iz analize rezultatov (glej prikaz Slika 44), kjer smo upoštevali uspešnost različnih vrednosti parametra γ pri pogoju $\alpha = 0{,}9$, je razvidno, da je v vseh primerih enaka minimalna vrednost skupnih zamud, vendar zaradi najnižje maksimalne vrednosti skupnih zamud in uteženosti diagrama navzdol izberemo vrednost $\gamma = 0{,}1$.

Vrednost (funkcijo) ε smo določili glede na konvergenco krivulje učenja Q. Krivulja učenja je povprečje skupnih zamud v posamezni ponovitvi, izračunanih z desetimi različnimi semeni

generacije naključnih spremenljivk. V nadaljevanju so podane samo krivulje učenja za Scenarij 5 (rezultati različnih scenarijev so primerljivi in so podani v prilogah G–J).

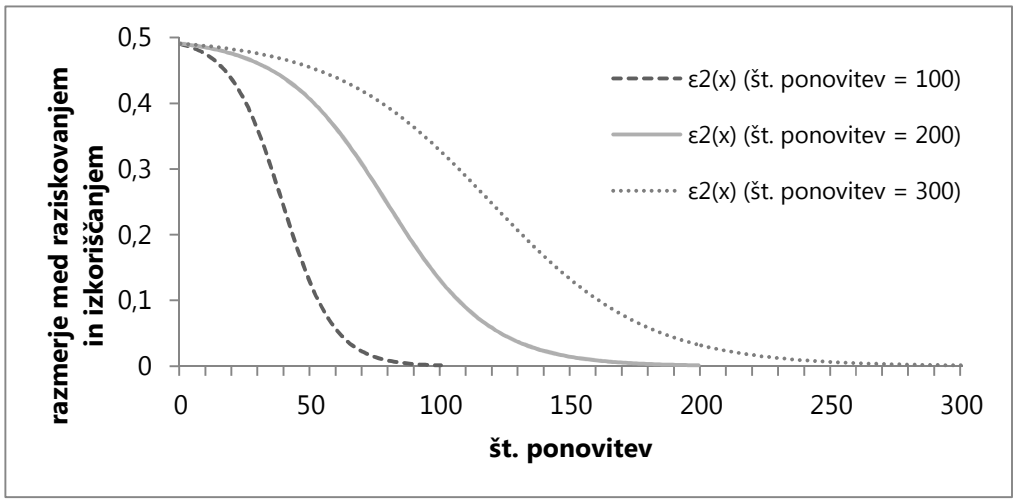


Slika 45: Krivulja učenja in krivulja standardne deviacije pri $\alpha = 0{,}9$, $\gamma = 0{,}1$ ter a) $\varepsilon 1 = x^{-1}$, b) $\varepsilon 2 = \frac{0{,}5}{1+e^{(10*(x-0.4*25)/25))}}$, c) $\varepsilon 3 = \frac{0{,}5}{1+e^{(10*(x-0.4*35)/35))}}$, d) $\varepsilon 4 = 0{,}9$, e) $\varepsilon 5 = 0{,}7$, f) $\varepsilon 6 = 0{,}5$, g) $\varepsilon 7 = 0{,}3$, h) $\varepsilon 8 = 0{,}1$

Figure 45: Learning and standard deviation curves for $\alpha = 0.9$, $\gamma = 0.1$, $\varepsilon 1$, and a) $\varepsilon 1 = x^{-1}$, b) $\varepsilon 2 = \frac{0{,}5}{1+e^{(10*(x-0.4*25)/25))}}$, c) $\varepsilon 3 = \frac{0{,}5}{1+e^{(10*(x-0.4*35)/35))}}$, d) $\varepsilon 4 = 0.9$, e) $\varepsilon 5 = 0.7$, f) $\varepsilon 6 = 0.5$, g) $\varepsilon 7 = 0.3$, h) $\varepsilon 8 = 0.1$

Iz primerjave krivulj učenja na prikazu Slika 45 je razvidno, da je v vseh primerih rezultat v testnih ponovitvah (v zadnjih treh ponovitvah) blizu optimalne rešitve, vendar se krivulje učenja razlikujejo v začetnih ponovitvah. Pri $\varepsilon 8$ je krivulja učenja ves čas najbližje optimalni rešitvi, vendar iz krivulje ni razvidno, da agent izboljšuje znanje. Sledi, da če agent v začetnih ponovitvah izbere slabo strategijo, le-te v nadaljnjih ponovitvah ne izboljša, zato izbira parametra $\varepsilon = 10$ ni priporočljiva. Zaradi kontinuiranega izboljšanja znanja (konvergence krivulje učenja proti optimalni vrednosti) predlagamo za določitev razmerja med raziskovanjem in izkoriščanjem v posamezni ponovitvi uporabo funkcije $\varepsilon 2(x) = \frac{0{,}5}{1+e^{(10*(x-0.4*\text{št.ponovitev})/\text{št.ponovitev})}}$.

V nadaljevanju je podan grafični prikaz predlagane funkcije razmerja med raziskovanjem in izkoriščanjem za različno število ponovitev učenja.



Slika 46: Predlagana ε-požrešna funkcija za različno število ponovitev učenj
Figure 46: Proposed ε-greedy function for different number of iterations

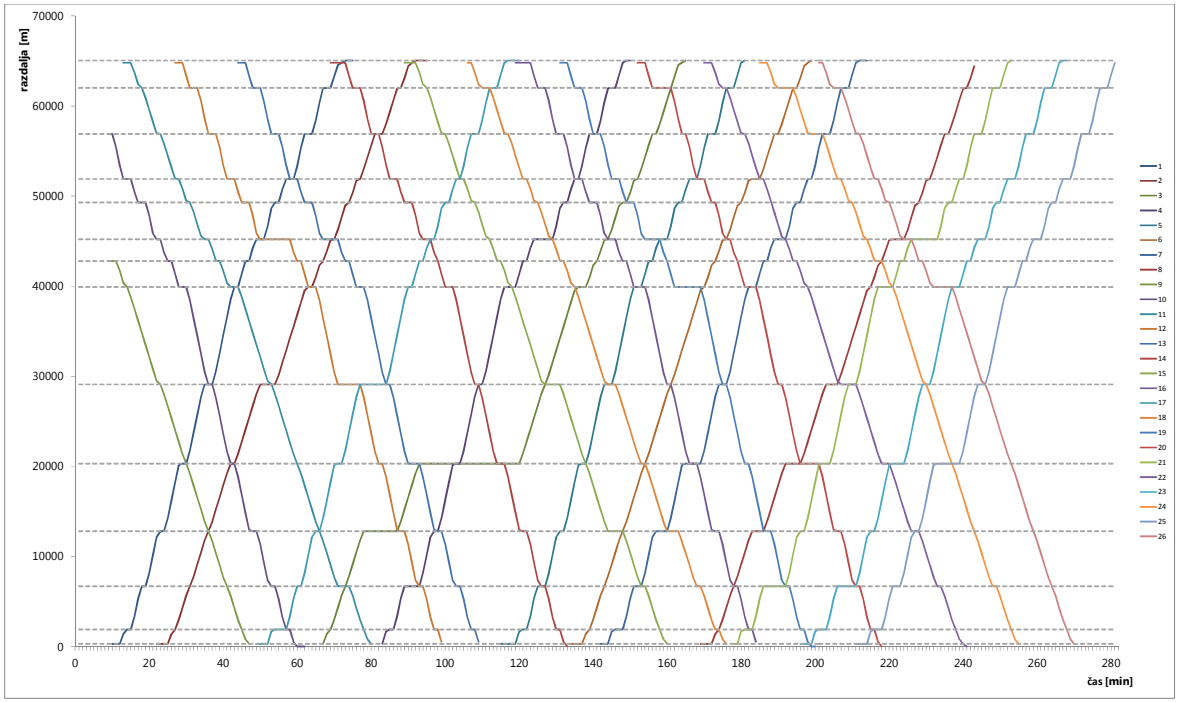
Iz prikaza Slika 46 je razvidno, da agent upošteva razmerje med raziskovanjem in izkoriščanjem glede na delež že izvedenih ponovitev. Agent v prvi epizodi, ne glede na število ponovitev, izbira akcijo z raziskovanjem ali izkoriščanjem v razmerju 1 : 1, nato pa se delež raziskovanja zmanjšuje. Agent na $2/5$ že izvedenih ponovitev raziskuje v povprečju samo še vsako četrto akcijo, na polovici učenja vsako osmo akcijo ter na koncu učenja vsako 800. akcijo.

4 UPORABA ALGORITMA UČENJA Q NA REALNEM PRIMERU ŽELEZNIŠKE INFRASTRUKTURE

Uporabnost in uspešnost predlagane implementacije algoritma učenja Q smo preverili za primer replaniranja voženj vlakov na realnem primeru visoko izkoriščene enotirne železniške proge Ljubljana–Jesenice. V modelu železniške proge so upoštevane razdalje med postajami ter število in dolžine blokovnih odsekov. Shema proge je podana na prikazu Slika 47.

Slika 47: Shema enotirne proge Ljubljana–Jesenice
Figure 47: The layout of the single-track railway between Ljubljana and Jesenice

V praksi je vozni red pripravljen z upoštevanjem progovne hitrosti. Na progi Ljubljana–Jesenice je progovna hitrost potniških vlakov med 75 km/h in 100 km/h ter tovornih vlakov med 75 km/h in 100 km/h. V predlaganem algoritmu pa predpostavimo, da vlaki na celotni progi vozijo z enako hitrostjo, zato v eksperimentu ne moremo upoštevati dejanskega voznega reda. Vozni red, ki ga upoštevamo v eksperimentu, je po izkoriščenosti kapacitete (26 vlakov v treh urah) primerljiv z dejanskim voznim redom in je podan na prikazu Slika 48.



Slika 48: Začetni vozni red za progo Ljubljana–Jesenice
Figure 48: Initial timetable for railway line between Ljubljana and Jesenice

Za izbrano infrastrukturo (proga Ljubljana–Jesenice), 26 vlakov in časovno okno treh ur je število možnih stanj po enačbi (4) 3 x 10^{47}, število možnih akcij pa po enačbi (5) 6,7 x 10^{7}.

Uspešnost predlaganega algoritma smo preverili za 20 scenarijev zamud, in sicer deset scenarijev zamud, kjer dva vlaka zamudita na začetni postaji (Ljubljana ali Jesenice) in en vlak zamudi na poljubni vmesni postaji, in deset scenarijev zamud, kjer zamudam vlakov iz prvega dela eksperimenta dodamo dva zamujena vlaka. Parametra, kateri vlak zamudi in velikost zamude, sta izbrana naključno. Velikosti zamud v posameznem scenariju so podane v Preglednici 8.

Preglednica 8: 20 scenarijev zamud, kjer so z S3_i označeni scenariji s tremi zamujenimi vlaki in z S5_i označeni scenariji s petimi zamujenimi vlaki.
Table 8: The twenty delay scenarios used in the experiments: the S3_i are scenarios with 3 delayed trains and S5_i are scenarios with 5 delayed trains.

Scenarij S3	Začetna zamuda [min]	Scenarij S5	Začetna zamuda [min]
S3_01	20, 16, 18	S5_01	20, 16, 18, 15, 19
S3_02	9, 20, 7	S5_02	9, 20, 7, 18, 8
S3_03	18, 15, 16	S5_03	18, 15, 16, 20, 17
S3_04	13, 9, 10	S5_04	13, 9, 10, 7, 11
S3_05	7, 18, 20	S5_05	7, 18, 20, 9, 6
S3_06	19, 10, 7	S5_06	19, 10, 7, 14, 16
S3_07	8, 7, 12	S5_07	8, 7, 12, 7, 13
S3_08	13, 19, 17	S5_08	13, 19, 17, 16, 10
S3_09	18, 16, 6	S5_09	18, 16, 6, 9, 7
S3_10	8, 13, 11	S5_10	8, 13, 11, 17, 20

V eksperimentih smo upoštevali v poglavju 3.3.11 določene vrednosti parametrov učenja Q, in sicer $\alpha = 0{,}9$, $\gamma = 0{,}1$ in $\varepsilon(x) = \frac{0{,}5}{1+e^{(10*(x-0.4*\text{št. ponovitev})/\text{št. ponovitev})}}$.

Uspešnost reševanja problema replaniranja voženj vlakov z algoritmom učenja Q smo primerjali z uspešnostjo replaniranja z upoštevanjem strategije FIFO (ang. *First-In-First-Out*). Pri uporabi strategije FIFO vodenje vlakov poteka po načelu *vlak, ki prvi pride na postajo, jo tudi prvi zapusti*. Takšen pristop onemogoča zamenjavo vrstnega reda na postaji, torej se lahko zgodi, da ostane hitrejši vlak ujet za počasnejšim. Strategija FIFO je kratkovidni požrešni pristop, saj minimizira skupne zamude z vodenjem vlakov v odvisnosti od časa postanka. Pravilo FIFO je uporabljeno samo ob nastanku konflikta, sicer vodenje vlakov sledi začetnemu voznemu redu.

Eksperiment smo glede na kriterijsko funkcijo razdelili na dva sklopa, in sicer:

a) kriterij uspešnosti je minimalna skupna zamuda vlakov na končnih postajah;
b) kriterij uspešnosti so minimalni stroški zamud vlakov na končnih postajah.

Delitev eksperimenta na dva sklopa omogoča, da ločeno preverimo, ali je algoritem učenja Q uporaben za reševanje problema z različnimi kriterijskimi funkcijami. V prvem eksperimentu je kriterij uspešnosti minimalna zamuda vlakov, pri čemer so zamude potniških in tovornih vlakov enakovredne. V drugem delu eksperimenta pa upoštevamo, da je strošek zamude potniškega vlaka višji od stroška zamude tovornega vlaka.

V obeh eksperimentih smo upoštevali enak začetni vozni red (Slika 48) in enake scenarije zamud (Preglednica 8).

a) Kriterij uspešnosti je minimalna skupna zamuda vlakov na končnih postajah

Rezultati replaniranja za 20 scenarijev zamud, izračunani z uporabo strategije FIFO in predlaganega algoritma učenja Q, so povzeti v preglednici v nadaljevanju. Algoritem učenja Q je bil izveden s 150 in 300 ponovitvami. V Preglednici 9 so za vsak scenarij zamud podani rezultati strategije FIFO in učenja Q. Za učenje Q so podane povprečne vrednosti in standardna deviacija testnih epizod za deset ponovitev učenja z različnimi semeni generacije naključnih spremenljivk ter rezultat učenja, torej minimalna vrednost testnih iteracij (glej str. 64). V preglednici so rezultati učenja Q, ki so boljši (nižje skupne zamude) od strategije FIFO, označeni krepko.

Preglednica 9: Rezultati, izračunani s strategijo FIFO in učenjem Q za 20 scenarijev zamud, podanih v Preglednici 8 – kriterij uspešnosti so minimalne skupne zamude.
Table 9: Results obtained with the FIFO strategy and with the Q-learning algorithm on the 20 delay scenarios from Table 8 – objective of minimizing the total delays.

		Učenje Q			
		Št. ponovitev = 150		**Št. ponovitev = 300**	
Scenarij	**FIFO**	**Skupna zamuda**	**Min. zamuda**	**Skupna zamuda**	**Min. zamuda**
S3_01	42,0	42,0 ± 0,0	42,0	42,0 ± 0,0	42,0
S3_02	50,0	49,6 ± 1,3	**46,0**	49,1 ± 1,4	**46,0**
S3_03	68,0	66,3 ± 3,6	**59,0**	68,0 ± 3,6	68,0
S3_04	52,0	52,0 ± 0,0	52,0	52,0 ± 0,0	52,0
S3_05	57,0	57,0 ± 0,0	57,0	57,0 ± 0,0	57,0
S3_06	31,0	31,0 ± 0,0	31,0	31,0 ± 0,0	31,0
S3_07	62,0	51,0 ± 5,6	**46,0**	54,0 ± 10,7	**43,0**
S3_08	/	94,0 ± 0,0	**94,0**	96,0 ± 0,0	**96,0**
S3_09	/	72,2 ± 7,4	**58,0**	68,5 ± 3,8	**63,0**
S3_10	48,0	48,0 ± 0,0	48,0	48,0 ± 0,0	48,0
S5_01	90,0	89,2 ± 2,4	**82,0**	86,5 ± 3,7	**80,0**
S5_02	65,0	64,7 ± 0,9	**62,0**	64,5 ± 1,0	**62,0**
S5_03	155,0	155,6 ± 14,3	**128,0**	162,0 ± 4,4	155,0
S5_04	76,0	76,0 ± 0,0	76,0	76,0 ± 0,0	76,0
S5_05	70,0	70,0 ± 0,0	70,0	69,2 ± 1,2	**67,0**
S5_06	58,0	58,0 ± 0,0	58,0	58,0 ± 0,0	58,0
S5_07	80,0	70,8 ± 6,9	**64,0**	68,7 ± 5,0	**64,0**
S5_08	/	99,0 ± 0,0	**99,0**	96,0 ± 0,0	**87,0**
S5_09	/	71,7 ± 3,5	**65,0**	76,3 ± 4,8	**69,0**
S5_10	102,0	96,2 ± 3,8	**88,0**	95,7 ± 4,9	**87,0**

Iz analize rezultatov je razvidno, da je učenje Q uspešnejše v primerjavi s strategijo FIFO (primerjamo rezultate, dobljene s strategijo FIFO, in minimalne zamude, dobljene z algoritmom učenja Q).

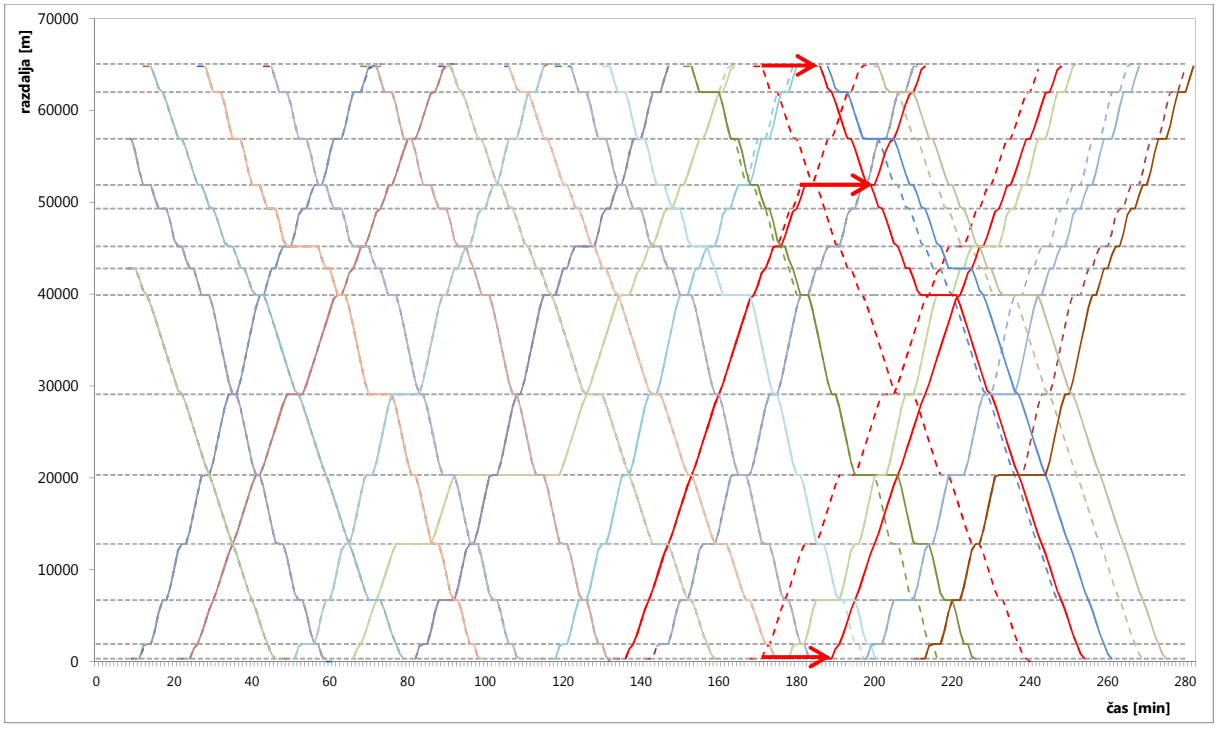
Pri reševanju problema replaniranja voženj vlakov s tremi zamujenimi vlaki so v petih primerih (S3_01, S3_04, S3_05, S3_06, S3_10) rezultati, izračunani po obeh metodah, enaki, kar nakazuje, da za optimalno rešitev pri izbrani kriterijski funkciji ni potrebna sprememba vrstnega reda vlakov. V treh primerih (S3_02, S3_03, S3_07) je rezultat, dobljen z algoritmom učenja Q, boljši kot rezultat, dobljen s strategijo FIFO – v primeru scenarija S3_07 celo za 26 % pri 150 ponovitvah in za 31 % pri 300 ponovitvah. Največja prednost učenja Q se izkaže v primeru scenarijev zamud S3_08 in S3_09, kjer uporaba strategije FIFO vodi v brezizhodno situacijo, algoritem učenja Q pa najde rešitev.

V primeru petih zamujenih vlakov so rezultati, dobljeni z obema metodama, enaki v treh primerih (S5_04, S5_05, S5_06) pri 150 ponovitvah in v treh primerih (S5_03, S5_04,

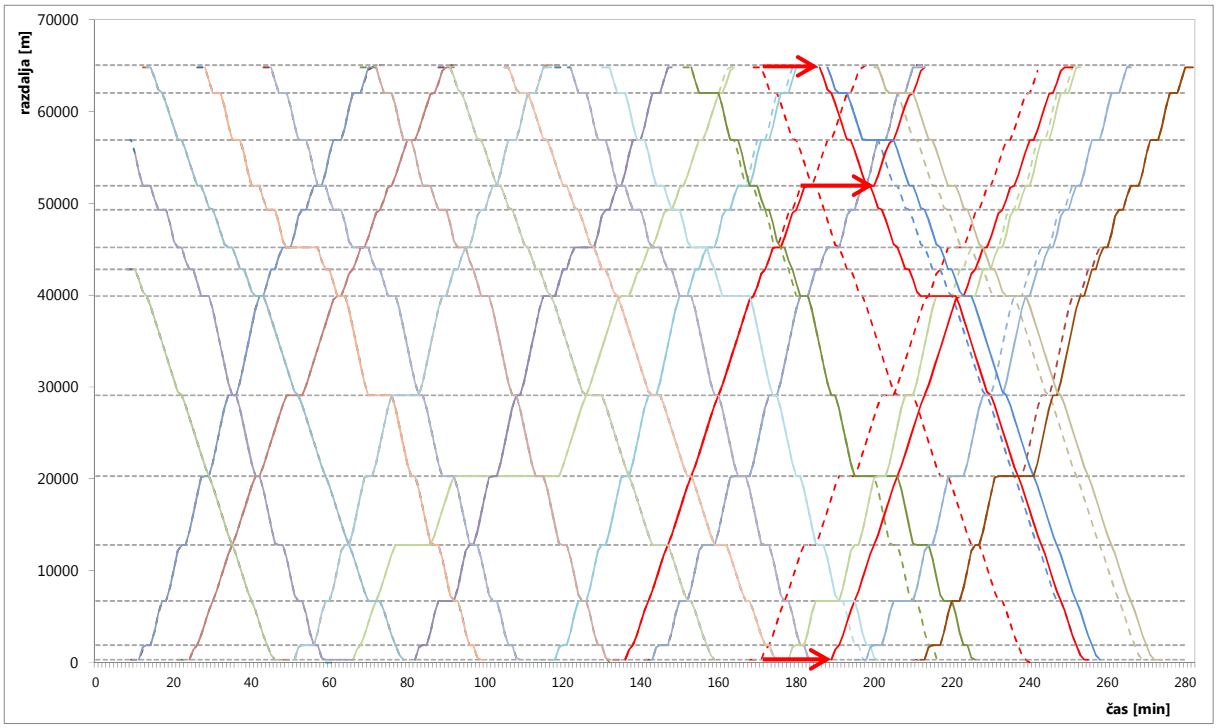
S5_06) pri 300 ponovitvah učenja, v vseh ostalih primerih pa je učenje Q uspešnejše, saj so rešitve nižje tudi do 17 % (S5_03 pri 150 ponovitvah). Tudi v primeru kompleksnejšega problema scenarijev zamud agent v scenarijih S5_08 in S5_09, ko strategija FIFO vodi v brezizhodno situacijo, algoritem učenja Q najde rešitev.

Iz primerjave rezultatov algoritma učenja Q za 150 in 300 ponovitev je razvidno, da agent v nekaterih primerih (S3_08, S3_09, S5_03 in S5_09) s povečanjem števila ponovitev poslabša znanje.

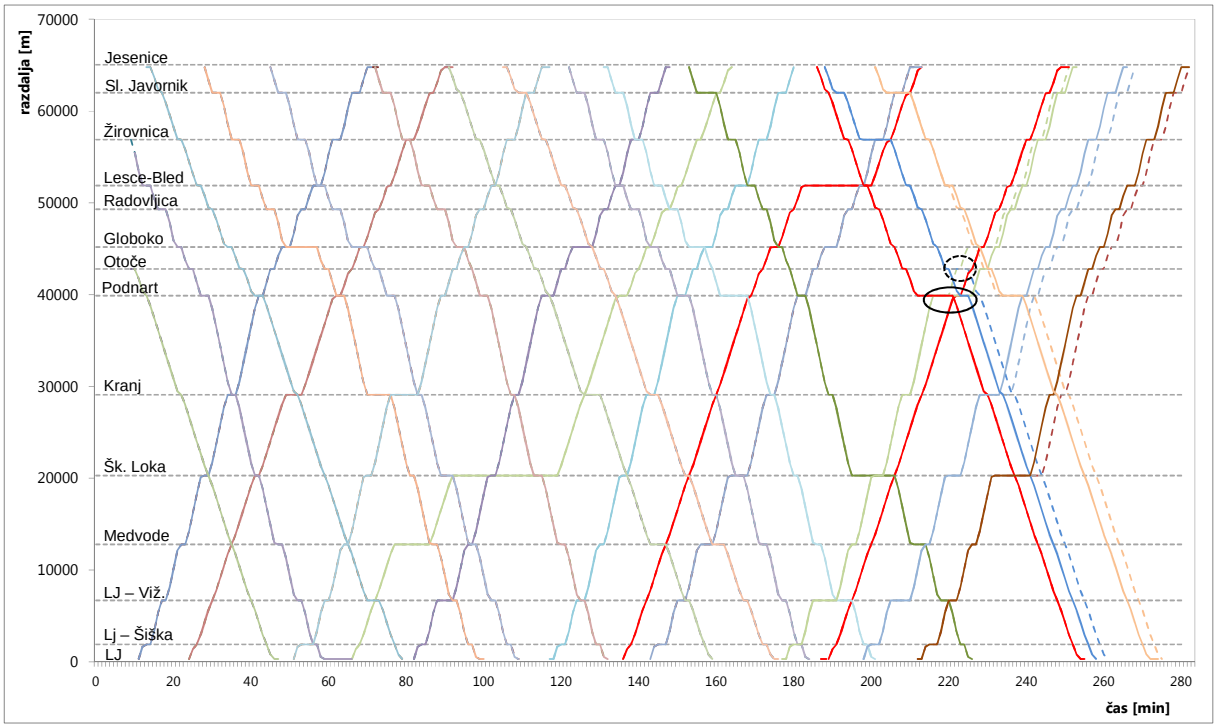
V nadaljevanju podajamo primerjavo med začetnim voznim redom ter s strategijo FIFO replaniranim voznim redom (Slika 49), začetnim voznim redom in z algoritmom učenja Q replaniranim voznim redom (Slika 50) ter primerjavo replaniranih voznih redov (Slika 51). Podan je primer replaniranja za scenarij zamude S3_03. Primer voznega reda, replaniranega z učenjem Q, je podan za seme, s katerim je rezultat replaniranja najbolj uspešen. Zamujeni vlaki so obarvani rdeče. Rdeče puščice nakazujejo mesto (postajo) in velikost zamude.



Slika 49: Začetni in s strategijo FIFO replanirani vozni red
Figure 49: Initial and with FIFO strategy rescheduled timetable



Slika 50: Začetni in z algoritmom učenja Q replanirani vozni red
Figure 50: Initial and with Q learning algorithm rescheduled timetable



Slika 51: S strategijo FIFO in z algoritmom učenja Q replanirani vozni red
Figure 51: With FIFO strategy and with Q learning algorithm rescheduled timetable

Na prikazu Slika 51 so s polno črto prikazane vlakovne poti, izračunane z algoritmom učenja Q, s črtkano črto pa vlakovne poti, izračunane s strategijo FIFO. Skupne zamude vlakov na

končnih postajah, izračunane s strategijo FIFO, znašajo 68 minut, z algoritmom učenja Q pa 59 minut. Iz primerjave replaniranih voznih redov je razvidno, da obe strategiji replaniranja voznega reda določita enake vlakovne poti za zamujene vlake. Ključna razlika med strategijama nastane v 220. minuti, ko se agent učenja Q odloči podaljšati postanek zelenemu vlaku na postaji Podnart, tako da se zeleni in modri vlak križata na postaji Podnart. Po strategiji FIFO pa zeleni vlak nadaljuje vožnjo takoj, in ker je odsek Podnart–Otoče zaseden, se modremu vlaku podaljša postanek na postaji Otoče. Iz primerjave podaljšanja postanka zelenega vlaka na postaji Podnart (polna črta) in podaljšanja postanka modrega vlaka na postaji Otoče (črtkana črta) je razvidno, da je strategija učenja Q lokalno slabša (daljši postanek vlaka), vendar so nadaljnje prilagoditve voznih poti glede na vozni red manjše. Iz primera je razvidno, kako odločitev o mestu križanja vlakov vpliva na domino efekt širjenja zamude.

b) Kriterij uspešnosti so minimalni stroški zamud vlakov na končnih postajah

Avtorji v prispevkih utemeljujejo izbiro različnih kriterijskih funkcij, zato smo v drugem delu eksperimenta preverili uspešnost algoritma učenja Q za primer, da upoštevamo drugačno kriterijsko funkcijo. Kriterijsko funkcijo (algoritem učenja je ostal nespremenjen) smo spremenili tako, da je strošek zamude potniškega vlaka večji od zamude tovornega vlaka, in sicer, da je razmerje stroškov minute zamude potniškega vlaka c_p in stroškov minute zamude tovornega vlaka c_f enako $c_p = 5 * c_f$. Strošek zamud na končni postaji izračunamo po enačbi $c = \sum_{p=1}^{n} d_{p,T} * c_p \ + \sum_{f=1}^{n} d_{f,T} * c_f$.

V Preglednici 10 so za vsak scenarij zamud podani rezultati strategije FIFO in učenja Q. Za učenje Q so podane povprečne vrednosti in standardna deviacija testnih epizod za deset ponovitev učenja z različnimi semeni generacije naključnih spremenljivk ter rezultat učenja, torej minimalna vrednost testnih iteracij (glej str. 64). V preglednici so rezultati učenja Q, ki so boljši (nižje skupne zamude) od strategije FIFO, označeni krepko.

Preglednica 10: Rezultati, izračunani s strategijo FIFO in učenjem Q za 20 scenarijev zamud, podanih v Preglednici 8 – kriterij uspešnosti so minimalni stroški zamud.
Table 10: Results obtained with the FIFO strategy and with the Q-learning algorithm on the 20 delay scenarios from Table 8 – objective of minimizing the delay costs.

		Učenje Q			
		Št. ponovitev = 150		**Št. ponovitev = 300**	
Scenarij	**FIFO**	**Skupna zamuda**	**Min. zamuda**	**Skupna zamuda**	**Min. zamuda**
S3_01	13,2	13,2 ± 0,0	13,2	13,2 ± 0,0	13,2
S3_02	34,0	33,9 ± 0,1	**33,6**	33,7 ± 1,1	**30,4**
S3_03	28,0	28,1 ± 0,3	28,0	27,1 ± 3,1	**18,4**
S3_04	21,6	21,6 ± 0,0	21,6	21,6 ± 0,0	21,6
S3_05	40,2	40,2 ± 0,0	40,2	40,2 ± 0,0	40,2
S3_06	19,0	19,0 ± 0,0	19,0	19,0 ± 0,0	19,0
S3_07	50,0	35,4 ± 2,2	**32,4**	33,6 ± 1,6	**31,0**
S3_08	/	94,0	**41,4**	96,0	**47,2**
S3_09	/	44,9	**38,0**	43,7	**35,6**
S3_10	12,0	12,0 ± 0,0	12,0	12,0 ± 0,0	12,0
S5_01	30,8	30,8 ± 0,0	30,8	30,8 ± 0,0	30,8
S5_02	51,4	51,4 ± 0,0	51,4	51,4 ± 0,0	51,4
S5_03	84,4	71,4 ± 5,7	**57,8**	66,4 ± 5,0	**55,0**
S5_04	44,0	44,0 ± 0,0	44,0	43,4 ± 1,7	**38,2**
S5_05	43,6	43,6 ± 0,0	43,6	43,1 ± 1,4	**39,0**
S5_06	37,2	37,2 ± 0,0	37,2	37,2 ± 0,0	37,2
S5_07	59,2	49,9 ± 7,8	**39,0**	44,1 ± 2,0	**40,6**
S5_08	/	99,0 ± 0,0	**98,8**	96,0 ± 0,0	**69,8**
S5_09	/	52,8 ± 21,9	**39,2**	45,1 ± 3,3	**38,4**
S5_10	53,2	49,9 ± 4,1	**40,6**	50,3 ± 4,5	**41,8**

Tudi v primeru uporabe drugačne kriterijske funkcije je učenje Q vsaj tako uspešno, kot je strategija FIFO, saj so skupne zamude, izračunane po obeh pristopih, enake oz. z uporabo algoritma učenja Q nižje.

Podobno kot v prvem delu eksperimenta je tudi pri upoštevanju različnih uteži zamude glede na vrsto vlaka učenje Q uspešneje rešilo scenarije zamud S3_02, S3_03, S3_07, S3_08 in S3_09. Pri scenarijih, kjer zamudi pet vlakov, je učenje Q uspešnejše pri reševanju scenarijev S5_03, S5_07, S5_08, S5_09, S5_10, pri večjem številu ponovitev pa tudi scenarijev S5_04 in S5_05.

Tudi v primeru drugačne kriterijske funkcije učenje Q najde rešitev za scenarije S3_08, S3_09, S5_08 in S5_09, ki pri uporabi strategije FIFO vodijo v brezizhodno situacijo.

Iz analize rezultatov je razvidno, da je predlagana implementacija algoritma učenja Q uspešna pri reševanju realnih problemov replaniranja vlakov. V eksperimentu smo upoštevali

časovno okno treh ur, pred vsakim učenjem inicializirali matriko Q. Pri implementaciji algoritma v realno situacijo bi bilo potrebno raziskati, kako določiti časovna okna, npr. tako, da dan razdelimo na več enako dolgih časovnih oken ali pa časovna okna definiramo v odvisnosti od števila vlakov ali do verjetnosti nastanka zamud in njihovih velikosti. Ne glede na izbiro velikosti časovnega okna bi se učenje vsak dan nadaljevalo (matrika Q se ne bi inicializirala). Agent bi z izkušnjami, ki bi jih pridobil v predhodnih učenjih, izboljševal znanje in rešitev bi konvergirala k optimalni vrednosti.

5 UGOTOVITVE RAZISKOVANJA

Cilj doktorske disertacije je bil razviti algoritem, ki omogoča replaniranje voženj vlakov v realnem času, in potrditi ali ovreči postavljene hipoteze. Z razumevanjem delovanja spodbujevanega učenja in principov vodenja vlaka smo učenje Q nadgradili tako, da smo razvili algoritem učenja Q z zakasnjeno nagrado in sledmi, ki smo ga uspešno uporabili na realnem primeru proge Ljubljana–Jesenice in dokazali v uvodnem poglavju podane hipoteze.

5.1 Preverjanje postavljenih hipotez

- H 1: Spodbujevano učenje je primerno za časovno replaniranje voženj vlakov

Replaniranje vlakov sodi v skupino NP-problemov, kar pomeni, da brez poenostavitev ali uporabe predznanja problem ni rešljiv v realnem času. Poenostavitve in predznanje so vezani na točno določeno železniško infrastrukturo, zato je treba takšne algoritme prilagoditi za uporabo na drugih omrežjih. Poglavitna prednost spodbujevanega učenja je, da agent raziskuje okolje in gradi svoje znanje iz povratnih informacij, ki jih pridobiva iz okolja. To pomeni, da je metoda uporabna v različnih okoljih, kar je tudi poglavitna prednost v primerjavi z ostalimi objavljenimi pristopi.

Z eksperimenti v poglavju 3 ter rezultati uporabe predlaganega algoritma spodbujevanega učenja na realnem primeru enotirne železniške proge Ljubljana–Jesenice, podanimi v poglavju 4, smo dokazali uspešnost pristopa. Za primere železniške infrastrukture v Eksperimentu 4 ter za železniško infrastrukturo Ljubljana–Jesenice je bil uporabljen isti algoritem, kar dokazuje, da je le-ta uporaben za različne strukture železniške infrastrukture ter za različno število vlakov brez prilagoditev algoritma.

- H 1.1: Metoda spodbujevanega učenja je uporabna za upravljanje zaporednih voženj in voženj v različnih smereh po istem tiru

Vsi eksperimenti za preverjanje uporabnosti in uspešnosti različnih formulacij učenja Q so bili zastavljeni tako, da se je ločeno preverila uporabnost za upravljanje zaporednih voženj, voženj v različnih smereh po istem tiru ter kombinacija obeh. Z eksperimenti smo dokazali, da agent upošteva v okolju definirani varnostni načeli (vožnjo v prostorskem razmiku ter upoštevanje pravila privolitve smeri vožnje) in predlaga brezkonfliken vozni red. Nadalje smo dokazali, da je metoda spodbujevanega učenja, natančneje učenja Q z zakasnjeno nagrado in sledmi, najbolj uspešen pristop k reševanju problema replaniranja vlakov, ki vozijo po enotirni progi.

- H 1.2: Metoda spodbujevanega učenja je primerna za različne vidike uspešnosti replaniranja

Uspešnost replaniranja se najpogosteje meri z velikostjo skupnih zamud vseh vlakov na končni postaji, vendar avtorji opozarjajo, da je treba upoštevati utežene vrednosti zamud glede na število vlakov ali celo vidik udobnosti potnikov ali vidik porabljene energije oz. kombinacijo različnih vidikov. Strokovnjaki še niso podali enotnega mnenja o kriterijski funkciji, ki bi bila enotno merilo uspešnosti replaniranja.

Pri uporabi učenja Q uporabnik definira nagrado, zato je nagrada lahko poljubno velika in upošteva različne kriteriji. V poglavju 4 so podane rešitve replaniranja za različna vidika uspešnosti; pri prvem vse vlake obravnavamo enakovredno, pri drugem pa potniškim vlakom pripišemo višjo vrednost zamude, torej morajo biti potniški vlaki čim bolj točni, da je replaniranje uspešno. Z rezultati v poglavju 4 smo dokazali, da se agent nauči različnih strategij za različne kriterijske funkcije.

- H 2: Izbira vrednosti parametrov algoritma učenja Q vpliva na uspešnost replaniranja

Učenje Q je zaradi splošnosti uporabno za reševanje različnih problemov. Od problema je odvisno, kakšno vrednost upoštevamo za parameter stopnje učenja in v kolikšni meri agent upošteva vrednost trenutne nagrade, predvsem pa je pomembno uravnotežiti razmerje med učenjem in izkoriščanjem znanja. S parametrično študijo smo dokazali, da je za predlagano implementacijo učenja Q z zakasnjeno nagrado in sledmi učenje najbolj uspešno pri izbiri parametrov $\alpha = 0{,}9$, $\gamma = 0{,}1$ in $\varepsilon(x) = \frac{0{,}5}{1+\mathrm{e}^{(10*(\mathrm{x}-0.4*\check{\mathrm{s}}\mathrm{t.ponovitev})/\check{\mathrm{s}}\mathrm{t.ponovitev}))}}$.

- H3: Način nagrajevanja

Značilnost replaniranja železniškega prometa je, da lahko z ustreznim podaljševanjem postanka vlaka na vmesni postaji zmanjšamo skupne zamude vlakov na končnih postajah. Pri implementaciji učenja Q, kjer agent prejme nagrado po vsaki izvedeni akciji, podaljševanje postankov na vmesnih postajah vedno vodi v slabšo uspešnost algoritma učenja Q, saj agent za daljši postanek prejme nižjo nagrado. Torej takšen pristop ni ustrezen. Z rezultati v poglavjih 3.3.8, 3.3.9, 3.3.10 smo dokazali, da je način implementacije nagrade odločilen za uporabnost učenja Q pri reševanju problema replaniranja ter da z uporabo sledi (prenosom informacije o nagradi od končnega proti začetnemu stanju po verigi obiskanih parov stanj in akcij) izboljšamo učinkovitost algoritma.

5.2 Izvirni znanstveni prispevek

V doktorski disertaciji je opisan in ovrednoten algoritem, ki je zasnovan na učenju Q in poišče (skoraj) optimalno strategijo vodenja prometa v realnem času v primeru nastanka zamud, ki jih ni mogoče kompenzirati s časovnimi dodatki. V okviru doktorske disertacije razviti algoritem išče optimalni brezkonfliktni vozni red ob upoštevanju načel vodenja železniškega prometa, predstavljenih v poglavju 2. Algoritem je bil testiran na enotirni progi, kjer obvoz oz. izbira drugačne vozne poti ni mogoča. Uspešnost algoritma je ovrednotena s kvaliteto rešitve in časom, potrebnim za določitev (skoraj) optimalne strategije vodenja vlakov.

Iz pregleda stanja na področju časovnega načrtovanja in replaniranja voženj vlakov je razvidno, da je predlaganih veliko različnih pristopov, vendar v strokovni in znanstveni literaturi nismo zasledili uporabe metode učenja Q v ta namen. Tako je poglaviten znanstveni prispevek širjenje uporabe metode učenja Q na področje optimizacije replaniranja voženj vlakov. V doktorski disertaciji smo prvi definirali opis okolja in njegovih stanj ter določili agenta in množico možnih akcij, med katerimi agent v procesu učenja izbira.

V procesu raziskovanja in vrednotenja uporabnosti metode učenja Q smo prišli do spoznanja, da sprotno dodeljevanje nagrad, kot je predvideno v osnovni ideji učenja Q, ni ustrezno za učinkovito replaniranje vlakov (glej poglavje 3.3.8). Uspešnost algoritma učenja Q nam je uspelo izboljšati z implementacijo dodeljevanja nagrade v končnem stanju, t. i. zakasnjene nagrade (ang. *delayed reward*). Izkazalo se je, da je takšen pristop učinkovitejši od učenja Q, vendar je potrebno veliko število ponovitev, da se agent nauči optimalne strategije (glej poglavje 3.3.8). Z vpeljavo zakasnjene nagrade s sledmi (ang. *eligibility traces*) smo še dodatno izboljšali uspešnost algoritma, ki vrne (skoraj) optimalno rešitev v zelo kratkem času (glej poglavje 3.3.10) in je primeren tudi za uporabo na večjih in bolj kompleksnih primerih železniške infrastrukture, za večje število vlakov in daljše časovno okno (glej poglavje 4).

Znotraj uporabe učenja Q smo z raziskovanjem določili vrednost parametra učenja (α), vrednost parametra upoštevanja nagrade (β), vrednost parametra razmerja med učenjem in izkoriščanjem znanja (ε).

Pomemben prispevek k znanosti je tudi razvoj makroskopskega simulacijskega orodja, ki za razliko od uveljavljenih simulacijskih orodij omogoča izredno hitro in učinkovito simuliranje železniškega prometa oz. preveritve različnih strategij agenta. Simulacijsko orodje deluje v realnem času in je prilagodljivo različnim železniškim infrastrukturam. Hitrost delovanja simulacijskega orodja smo dosegli z uvedbo nekaterih poenostavitev (glej poglavje 3.3.7), ki sicer vplivajo na uporabnost pri natančnem določevanju izkoriščenosti kapacitete železniške

infrastrukture, porabi energije ali profilov hitrosti in hkrati bistveno ne vplivajo na uporabno vrednost pri časovnem načrtovanju in replaniranju voženj vlakov, kar je primarni namen simulacijskega orodja.

5.3 Smernice za nadaljnje raziskovanje

Izkušnje in znanja, pridobljena s to raziskavo, so osnova za nadaljnja raziskovanja uporabnosti učenja Q v bolj kompleksnih problemih časovnega načrtovanja voženj vlakov.

V predlaganem algoritmu smo predpostavili, da vlaki lahko vozijo samo z v_{max}, s tem pa zanemarili, da se največja dovoljena progovna hitrost na odsekih razlikuje. Ker je hitrost vlakov ključni parameter pri replaniranju voženj vlakov, je v nadaljnjih raziskavah treba nadgraditi simulacijsko orodje tako, da bo uporabniku omogočalo definiranje različnih največjih dovoljenih hitrostih za različne odseke, predvsem omejitev hitrosti. S tem bi zagotovili uporabno vrednost sistema za pomoč dispečerju, saj veliko zamud nastane zaradi uvedbe počasne vožnji (npr. zaradi slabega stanja železniške infrastrukture ali del na sosednjem tiru). Z uvedbo (poenostavljenih) enačb za izračun odporov proge pa bi simulacijsko orodje omogočalo tudi uporabo kriterijskih funkcij, ki upoštevajo porabo energije.

V procesu odprave motnje (zamude) enega ali več vlakov nastanejo trije podproblemi, in sicer konflikti pri zagotavljanju posadke, konflikti pri zagotavljanju vlakov ter konflikti pri vodenju vlakov (Jespersen-Groth et al., 2006). Konflikti pri zagotavljanju posadke nastanejo takrat, ko je novi predvideni prihod vlaka na končno postajo kasnejši, kot je zaključek delovnika posadke, ali ko je število posadke premajhno, da bi lahko zagotovilo predpisani počitek. V obeh primerih je treba spremeniti predvideni plan delovnih ur. Konflikt pri zagotavljanju vlakov nastane, kadar pride do okvare in je potrebno replaniranje virov. Tako kot večina drugih raziskav smo se tudi mi osredotočili samo na konflikte, ki nastanejo pri vodenju vlakov. V nadaljnjih korakih bi bilo smiselno preveriti, kako pogosto nastanejo konflikti pri zagotavljanju virov (posadke in vlakov), ter razmisliti o implementaciji dodatnih kriterijev. Sklepamo pa, da se reševanje konfliktov zagotavljanja virov v posameznih državah rešuje na različne načine, zato bi z implementacijo takšnih pogojev izgubili na splošnosti algoritma.

Eden izmed kriterijev uspešnosti replaniranja je pogoj, da je nov vozni red izvedljiv. V doktorski disertaciji smo upoštevali, da je novi vozni red izvedljiv, če ni konfliktov, ki bi v naravi pomenili trk vlakov, in če ni presežena kapaciteta postajnih tirov. V algoritmu je implementiran zgolj kapacitetni pogoj, da je na postaji lahko toliko vlakov, kolikor je postajnih tirov. V nadaljnjih raziskavah je treba dodati pogoj, da se na postaji lahko hkrati ustavi le

toliko potniških vlakov, kolikor je na postaji peronov, in pogoj, da dolžina vlaka ne sme presegati uporabne dolžine postajnega tira, na katerem se vlak ustavi.

Na postajah sta s postajnim redom določena minimalni čas, potreben za odpravo dveh zaporednih vlakov, in minimalni čas, potreben za križanje vlakov. V algoritmu je trenutno implementiran le pogoj, da lahko vlak zasede prosti odsek – torej, takoj ko pride prvi vlak na postajo, lahko drugi postajo zapusti; čas, potreben za križanje vlakov v algoritmu, ni upoštevan. Neupoštevanje časa, potrebnega za odpravo dveh zaporednih vlakov, in časa, potrebnega za križanje vlakov, ne vpliva na dokaz o uporabnosti spodbujevanega učenja za reševanje replaniranja vlakov, vpliva pa na izvedljivost voznega reda v realnih situacijah. Pred implementacijo je treba za vse postaje implementirati omenjena pogoja.

V doktorski disertaciji uspešnost replaniranja vrednotimo z velikostjo skupnih zamud v končnem stanju, pri čemer smo končno stanje definirali kot trenutek, ko zadnji vlak prispe na končno postajo, kot trenutek, ko agent zazna nastanek brezizhodne situacije, oz. po preteku uporabniško definiranega časovnega okna. Na krajših in testnih primerih enostavno določimo, kdaj sistem doseže končno stanje, v realnih primerih pa se bo pojavil problem, kako te točke določiti.

Učenje Q je le ena izmed metod spodbujevanega učenja. V sklopu doktorske disertacije smo se osredotočili le na učenje Q in različne implementacije le-tega. V vseh eksperimentih smo uporabili enako definicijo nagrade, enak opis možnih stanj in enake možne akcije. V nadaljnjih raziskavah bi bilo smiselno preveriti tudi drugačne definicije in implementacije (npr. opis stanj brez časa in z upoštevanjem verjetnosti med prehodi stanj). Predlagamo, da se v prihodnjih raziskavah preveri tudi druge metode spodbujevanega učenja, npr. metodo SARSA.

6 RAZPRAVA IN ZAKLJUČEK

6.1 Razprava o uporabi »ukrojenega« simulacijskega orodja

Simulacijsko orodje, razvito v sklopu doktorske disertacije, upošteva nekaj poenostavitev, vendar le-te ne vplivajo na oceno uporabne vrednosti algoritma učenja Q za replaniranje voženj vlakov, hkrati pa močno vplivajo na čas trajanja simulacije. Tudi v voznih redih, ki so v uporabi v realnih situacijah, so vozni časi med postajama enaki ne glede na lokomotivo oz. težo vagonskega niza in konfiguracijo železniške infrastrukture. Poleg znanih dejavnikov, ki vplivajo na vozne čase, na le-te vplivajo tudi zunanji dejavniki, ki jih ni mogoče predvideti, zato natančnost, kot jo upoštevajo makroskopski modeli, nima uporabne vrednosti pri replaniranju. Pri reševanju problema replaniranja voženj vlakov v realnem času je pomembno, da kvalitetno rešitev dobimo v kratkem času, kar nam uporabljene poenostavitve v simulacijskem orodju tudi omogočajo. Kljub temu da je računanje dejanskih hitrosti vlakov in s tem zasedenosti posameznih odsekov poenostavljeno, se kompleksnost problema replaniranja ne zmanjša (Medanic in Dorfman, 2002a).

6.2 Razprava o (ne)upoštevanju predznanja

Problem časovnega načrtovanja voženj vlakov sodi v razred NP-problemov, zato so pristopi dinamičnega programiranja neuporabni za praktično uporabo v realnem času, če ne zmanjšamo prostora rešitev s poenostavitvami in z vnaprej določenimi pravili. Avtorji predlagajo različne poenostavitve, npr. uporabo niza pravil, ki so definirana skupaj z dispečerji in odražajo izkušnje dispečerjev, ne nujno pa tudi optimalne odločitve (Fay, 2000), upoštevanje pravila, da imajo najvišjo prednost vlaki, ki so točni, torej vozijo v skladu z voznim redom (Törnquist, 2006), pravila, da ima najvišjo prioriteto potniški vlak (Signalni pravilnik, 2007), ali pravila, da ima prednost hitrejši vlak (Jespersen-Groth et al., 2006). Uporaba hevrističnih pravil sicer zmanjša prostor možnih rešitev in zmanjša čas iskanja rešitve, vendar lahko nehote izloči globalni optimum. Priprava pravil je zahtevna naloga, rešitve pa so odvisne od železniške infrastrukture in hitrosti vlakov, zato običajno takšna pravila niso uporabna brez prilagajanj na drugačnih konfiguracijah železniških infrastruktur in upoštevanju drugačnih hitrosti vlakov. Ravno vzpostavljanje pravil ali baze znanja je ozko grlo sistemov za pomoč dispečerjem, ki temeljijo na hevrističnih pristopih.

Na proces učenja po principu spodbujevanega učenja lahko vplivamo s pravili, ki jih definiramo v okolju in s predznanjem. Princip definiranja elementov spodbujevanega učenja (glej poglavje 3.3) omogoča definiranje poljubnih pravil in poljubno število pravil v okolju. Tako bi lahko v okolju spodbujevanega učenja za vsako postajo posebej definirali poljubno pravilo, kot je npr. kateri vlak ima prednost. Takšna pravila lahko definiramo za stanja, ko sta

oba vlaka (npr. hitrejši in počasnejši) že na postaji, lahko pa tudi za stanja, ko je počasnejši vlak že na postaji, hitrejši pa se ji približuje: torej določimo časovni razmik med vlakoma, v katerem se agentu določi upoštevanje pravila. Kot smo že omenili, hevristična pravila lahko nehote izločijo raziskovanje tistega dela drevesa, ki vodi k optimalni rešitvi, zato jih v naš algoritem nismo implementirali.

Drugi način, s katerim lahko vplivamo na proces učenja, je podajanje predznanja agentu. Z vnaprej podanimi vrednostmi $Q(s_t, a_t)$ favoriziramo akcije, za katere vemo, da so bolj učinkovite. Zaradi kompleksnosti problema, velikega števila stanj in s tem matrike Q takšen pristop ni smiseln. S spremembo infrastrukture (dolžine in števila odsekov) in števila vlakov se spremeni število možnih stanj (in s tem matrika Q), zato je uporaba predznanja omejena na točno določen problem. Poleg tega na rešitev vplivata tudi hitrost vlakov in kriterijska funkcija, kar je še dodaten razlog, da agentu ne podamo predznanja.

V literaturi predstavljeni pristopi pogosto uporabljajo hevristiko ali algoritme, ki temeljijo na urejanju po prioritetah s strategijami, ki so značilne za določeno infrastrukturno okolje in praviloma niso prilagodljive na spremembe. Prednost naše formulacije problema je možnost uporabe algoritma tako za enotirne kot tudi dvotirne proge, različne dolžine in različno število odsekov med postajami, različno število postajnih tirov in za različno število vlakov.

6.3 Razprava o kriterijski funkciji

Pomemben element optimizacije problema je kriterij, na osnovi katerega ocenjujemo uspešnost pristopa. Kriterijsko funkcijo najpogosteje predstavlja matematična enačba, ki opisuje povezavo med stanjem z določenimi lastnostmi in vrednostmi, ki jih to stanje generira. Pri reševanju problema si lahko zastavimo različne kriterije, kot so npr. minimalni čas za opravilo naloge, minimalna zamuda pri opravljanju nalog, minimalna poraba energije. Optimiranje se lahko izvaja po enem ali več kriterijih. Kriteriji so med seboj primerljivi, pretvorijo se v isto valuto, običajno monetarno.

Proces replaniranja vlakov lahko optimiziramo z vidika potnikov, kjer želimo npr. povečati točnost vlakov, ali z vidika upravljavca, kjer želimo npr. zmanjšati porabo virov, lahko pa tudi kot kombinacijo obeh vidikov (glej poglavje 3.3). Kot smo že omenili, ne obstaja kriterijska funkcija, ki bi bila merodajna za vrednotenje učinkovitosti vodenja železniškega prometa.

V uvodu smo podali trditev, da se potniki odločajo o transportnem sredstvu predvsem na osnovi njegove točnosti. S tega vidika bi bila najbolj smiselna uporaba kriterijske funkcije, ki upošteva za potnike pomembne kriterije (npr. občutek gneče, zamujeno prestopanje, utežene zamude vlaka glede na število potnikov), vendar običajno število potnikov na vlaku

ni poznano, prav tako tudi ne število potnikov, ki so zamudili prestopanje. Za optimiziranje problema z vidika upravljavca, kadar želimo zmanjšati stroške energije ali optimizirati porabo virov (lokomotiv, vlakov, osebje), pa so prav tako potrebni natančni vhodni podatki.

V skladu z evropsko zakonodajo, UIC Objavo 450-2 (2009), se točnost vlaka izrazi z zamudo, ki je opredeljena kot odstopanje med načrtovanim in dejanskim časom na dogovorjenih merilnih mestih voženj vlaka. Zato smo za ilustracijo uporabnosti in učinkovitosti replaniranja voženj vlakov z uporabo metode spodbujevanega učenja uporabili enostavno kriterijsko funkcijo: minimalne skupne zamude vlakov na končni postaji.

Dodatna uporabna vrednost spodbujevanega učenja je možnost definiranja poljubnih kriterijskih funkcij, ki omogočajo optimiziranje problema za različne vidike uspešnosti brez prilagoditev algoritma.

6.4 Zaključek

V doktorski disertaciji obravnavamo zamude v železniškem prometu. Natančneje, obravnavali smo, kako zamude omejiti in jih čim prej odpraviti, pri tem pa se nismo spraševali po vzrokih za nastanek zamud. Časovno replaniranje voženj vlakov z namenom, da se vplivi zamude čim prej odpravijo, je velik in dinamičen optimizacijski problem, pri katerem je treba upoštevati veliko množico omejitev. Zaradi zahtevnosti problema še ni bil razvit sistem za pomoč pri odločanju, saj je kljub zmogljivi računalniški opremi problem replaniranja voženj vlakov v realnem času še vedno težko rešljiv v dovolj kratkem času za praktično uporabo, zato dispečerji problem replaniranja rešujejo z ekspertnim znanjem.

V disertaciji smo predstavili razvoj prilagodljivega algoritma, ki temelji na učenju Q, in razvoj simulacijskega orodja, ki je namenjen reševanju problema replaniranja voženj vlakov v realnem času. Pri uporabi izraza »prilagodljiv algoritem« pogosto prihaja do napačne uporabe oz. razlage. Treba je poudariti, da niso vsi algoritmi, ki delujejo v realnem času, res prilagodljivi. *Real-time* algoritmi so tisti, ki se odzovejo na vhodne podatke v realnem času, logika in parametri sistema pa ostanejo nespremenjeni. Bistvena lastnost prilagodljivih algoritmov pa je njihova sposobnost prilagajanja logike in parametrov kot odziv na večje spremembe v okolju. Ena izmed prednosti spodbujevanega učenja je ravno dejstvo, da so resnično prilagodljivi – ne samo v smislu, da so zmožni odgovora na dinamične vhodne podatke, temveč tudi na dinamične spremembe okolja. In ravno možnost uporabe predlaganega algoritma na različno velikih železniških omrežjih, za različno število vlakov in brez potrebnega predhodnega znanja je bistvena prednost predlaganega pristopa v primerjavi s pristopi, predstavljenimi v poglavju o pregledu znanstvenega področja.

V disertaciji smo predstavili koncept metode, ki služi razumevanju uporabljene metode, ter osnovne principe in tehnike, uporabljene v predlaganem algoritmu. Spodbujevano učenje oz. algoritem učenja Q smo prvi uporabili za problem replaniranja voženj vlakov, zato je možnosti za nadgradnjo in optimizacijo algoritma, predvsem pa simulacijskega orodja, veliko in verjamemo, da je prihodnost sistemov za pomoč dispečerjem v uporabi algoritmov spodbujevanega učenja. Naša raziskava je osredotočena na reševanje problema časovnega replaniranja vlakov z metodo spodbujevanega učenja, natančneje z učenjem Q. Uspešnost uporabljene metode smo vrednotili z dvema parametroma, in sicer s časovno zahtevnostjo ter s kvaliteto rešitve (ali je rešitev optimalna oz. ali se dovolj malo razlikuje od optimalne).

V poglavju 3.3.7 smo pokazali, da uporaba ukrojenega simulacijskega orodja za replaniranje voženj vlakov omogoča hitro reševanje problema in s tem uporabo v realnem času. V poglavju 3.3.8 je ponazorjen princip učenja Q, vendar je učenje, ki sledi osnovni ideji, neuspešno. Po analogiji z igro šaha smo v poglavju 3.3.9 uvedli drugačen princip nagrajevanja, in sicer z zakasnjeno nagrado, ki jo agent prejme v končnem stanju. Rezultati so bili nekoliko bolj spodbudni, saj smo dokazali, da agent prepozna različne varnostne zahteve in jih tudi pravilno rešuje, vendar je njegovo učenje premalo uspešno. Z raziskovanjem področja replaniranja vlakov z učenjem Q smo ugotovili, da je pomembno, da agent prejme nagrado v končnem stanju, ko ve, kako uspešna je bila posamezna strategija, ter da se informacija o uspešni strategiji čim hitreje prenese iz končnega proti začetnemu stanju, saj se mora agent že v začetnih stanjih pravilno odločiti o vrstnem redu vlakov in njihovih odhodih s postaj. V poglavju 3.3.10 smo dokazali, da se agent z uvedbo nagrade v končnem stanju (in upoštevanjem le-te v vseh posodobitvah matrike Q po sledeh izvedene strategije) ter posodabljanju matrike Q iz končnega proti začetnemu stanju uspešno uči, kar pomeni, da najde kvalitetne rešitve.

Učenje agenta je odvisno od parametra α, ki vpliva na stopnjo učenja, parametra γ, ki je faktor diskontiranja prihodnjih nagrad in razmerja med raziskovanjem in izkoriščanjem znanja. Vsi parametri so določijo ločeno za vsak problem, ki ga obravnavamo s spodbujevanim učenjem. V poglavju 3.3.11 smo določili vrednosti parametrov ($\alpha = 0{,}9$, $\gamma = 0{,}1$, $\varepsilon(x) = \frac{0{,}5}{1+e^{(10*(x-0.4*\text{št.ponovitev})/\text{št.ponovitev}))}}$), ki najpogosteje vrnejo optimalno rešitev.

Implementacijske detajle in rezultate raziskav, predstavljene v poglavju 3.3, smo uporabili pri reševanju problema replaniranja voženj vlakov na primeru obstoječe proge Ljubljana–Jesenice. Primer, ki smo ga obravnavali, je zelo kompleksen, zato z analitičnim pristopom ne moremo določiti optimalne rešitve. Prav tako ne moremo primerjati uspešnosti našega algoritma z ostalimi algoritmi, saj je rezultat odvisen od vseh vhodnih podatkov, ki pa v objavah niso podani. Uspešnost algoritma smo zato primerjali z rezultati, ki jih dobimo s

strategijo FIFO. V večini primerov (glej Preglednica 9 na strani 93 in Preglednica 10 na strani 97) je bil naš algoritem bolj uspešen oz. še več, v primerih, v katerih z uporabo strategije FIFO ne dobimo rešitve (nastane brezizhodna situacija), agent najde rešitev. V doktorski disertaciji smo dokazali, da z učenjem Q lahko uspešno rešujemo problem replaniranja vlakov, vendar je to prva raziskava na tem področju in zato obstajajo različne možnosti za izboljšave, ki so nakazane v poglavju 5.3.

Na kratko lahko povzamemo, da v doktorski disertaciji predlagani algoritem za replaniranje vlakov, ki temelji na učenju Q z zakasnjeno nagrado in sledmi, omogoča, da se agent uči veliko hitreje kot dispečer, si veliko bolj natančno zapomni rezultate in jih zna subjektivno interpretirati na različne načine, zato sklepamo, da je kvaliteta rešitve agenta boljša od kvalitete rešitve, ki bi jo podal dispečer.

7 POVZETEK

Železniški promet je zaradi specifičnih lastnosti železniške infrastrukture, načina vodenja prometa, velike medsebojne odvisnosti med vlaki natančno prostorsko in časovno načrtovan in poteka po vnaprej določenem voznem redu, v katerem so upoštevane vse zakonitosti tirno vodenega prometa. Tako so v doktorski disertaciji na kratko predstavljeni principi vodenja železniškega prometa s poudarkom na vodenju prometa v fiksnem prostorskem razmaku, saj je to trenutno najpogostejši princip vodenja prometa na evropskem železniškem omrežju, ter pogoji in omejitve, ki jih je treba zaradi varnostnih in tehnoloških omejitev upoštevati pri časovnem načrtovanju voženj vlakov. Vozni red se izdeluje in usklajuje z vsemi prosilci več mesecev, tako imajo konstruktorji dovolj časa, da pripravijo kvaliteten, izvedljiv in brezkonflikten vozni red, ki je osnova za izvajanje vlakovnega prometa.

Osebji prevoznika in upravljavca morata poskrbeti, da vlaki vozijo v skladu z voznim redom. Zaradi nepredvidenih situacij (npr. okvare vozil, signalno-varnostnih naprav, *ad-hoc* vlaka) lahko pride do motenj, nastanka zamud; v tem primeru je dispečer tisti, ki na osnovi poznavanja lokacije, hitrosti in ostalih podatkov vseh vlakov na obravnavanem območju določi novi, replanirani vozni red. Upravljavci železniških storitev želijo povečati konkurenčnost železniškega sistema s povečevanjem števila vlakov in z visoko kakovostjo storitev. Večja, kot je gostota vlakov, večja je verjetnost nastanka sekundarnih zamud, večje je vplivno območje zamude in dalj časa vpliva lokalna motnja na druge vlake, hkrati pa se poveča kompleksnost problema. Zato se dispečerji soočajo z izzivi, kako zagotoviti zanesljivost in točnost storitev ob dejstvih, da je železniška infrastruktura na meji izkoriščenosti kapacitete in da je na veliko odsekih uvedena počasna vožnja oz. so odseki celo zaprti.

Z matematičnega vidika je časovno načrtovanje voženj vlakov zahteven kombinatorični problem z veliko omejitvami, ki ga uvrščamo v razred NP-problemov. Dispečerji ga rešujejo z ekspertnim znanjem, njihove določitve so običajno suboptimalne, učinkovitost in stopnja optimizacije sta nizki. Dispečerji imajo pravila replaniranj, vendar brez tehnike optimiranja. Hitrost in gostota vlakov se povečujeta, kar povečuje kompleksnost problema. Pričakujemo, da bo kompleksnost problema presegla mejo, ko dispečerji še lahko relativno učinkovito rešujejo problem ročno, zato so za dosego cilja ohraniti in izboljšati zanesljivost (točnost) sistema potrebne nove metode, ideje in tehnologije.

V pregledu znanstvenega področja smo predstavili pristope, ki so jih v zadnjem obdobju predlagali raziskovalci. Pristopi se razlikujejo glede na velikost omrežja (ali obravnavajo samo manjše območje postaj, vozlišča ali večje območje), uporabljene kriterijske funkcije …

Vsem pristopom je skupno, da temeljijo na bazi znanja. Znanje je univerzalno za železniško omrežje ene države, pogosto celo samo za obravnavano omrežje. Omejitve in znanje je pri uporabi pristopa na drugem omrežju treba prilagoditi. Ločitev znanja in omejitev poenostavi vzdrževanje in izboljša kvaliteto podatkovne baze znanja. Zadnji dosežki na področju umetne inteligence nudijo temelj za obvladovanje tako velikih in kompleksnih problemov, ki pogosto presegajo zmožnosti tradicionalnih metod.

V doktorski disertaciji smo raziskali in analizirali najpomembnejše dejavnike konstrukcije voznega reda z upoštevanjem vseh varnostnih zahtev za varno odvijanje železniškega prometa. Na kratko so predstavljeni dejavniki, ki vplivajo na izvajanje in vodenje prometa, stabilnost voznega reda, dejavniki, ki vplivajo na nastanek zamud v železniškem prometu. Predstavljena je metoda spodbujevanega učenja, predvsem algoritem učenja Q, ki je tudi uporabljen za reševanje načrtovanja voženj vlakov v primeru, ko se pojavi odklon od voznega reda in je treba v realnem času pripraviti nov vozni red tako, da se zamuda vlakov ne razširi po omrežju in da je skupna zamuda vseh vlakov čim manjša. Glavni cilj algoritma je v čim krajšem času zajeziti prenos zamud med vlaki s pravilnimi odločitvami, usmerjenimi v minimizacijo negativnih vplivov zamud. V disertaciji je formuliran pristop učenja Q (stanja, akcije, nagrade), parametrična študija in na testnih primerih sistematično prikazana uspešnost algoritma.

8 SUMARY

Railway traffic is – due to railway infrastructure specifics, railway traffic management, and strong interdependencies between train routes – precisely defined in the temporal-spatial space and operates according to a timetable, whereby all safety principles of the railway operation are taken care of. In this thesis, different railway traffic management principles are presented, with focus on a signaled fixed block operation as the most common principle of railway traffic management on the European railway network, and on the conditions and restrictions related to safety and technological constraints that need to be considered in scheduling of the trains. Construction of the timetable takes several months, since it should be harmonized with all interested carriers. The timetable must be effective, feasible, and conflict-free; as it is to be the basis for railway traffic operation.

The drivers of the trains and the dispatchers must follow the timetable. Due to unpredictable situations (e.g. rolling stock breakdowns, interlocking breakdowns, ad-hoc trains), delays may occur. The train operation and rescheduling is supervised and regulated by dispatchers. The dispatchers know the actual position and speed of all trains, their timetable and railway infrastructure status, so they are responsible for real-time train traffic management. Infrastructure managers aim to increase the competitive position of the railway traffic with adding trains and with improving the quality of the service offered. Higher train density leads to greater possibility of secondary delays, the delay propagates on to more trains, the domino effects of the delay last longer, and at the same time the complexity of the problem increases. Thus, the dispatchers faced with the challenge of how to ensure reliable and punctual train service where the capacity utilization is very high.

The train rescheduling problem is a complex combinatorial and strongly constrained problem, classified as a NP-problem. Dispatchers reschedule trains using expert knowledge, their dispatching actions are usually suboptimal, efficiency and level of optimization are poor, since they follow dispatching rules without optimization of the problem. Speed and density of trains increase, and consequently the problem complexity increases. The railway traffic will come to the point where the complexity of the problem will be too high for the dispatchers to solve problem efficiently with their experience alone, therefore new methods, ideas, and technologies will be necessary to maintain high quality of railway traffic.

In the thesis, the state-of-the-art approaches proposed by researches in the last decade are shortly described. The approaches differ in the level of infrastructure (areas with few station vs. railway lines), regarding optimization objective etc. All approaches are based on predefined knowledge, where the knowledge is universal for a specific railway infrastructure.

For implementation on other infrastructures, the restrictions and knowledge should be adjusted. Separation of knowledge and restrictions simplify maintenance and improve the quality of the knowledge database. Recent achievements in the field of artificial intelligence provide a basis for managing complex problems that are beyond the capabilities of traditional methods.

In the thesis we investigate and analyze the most important factors influencing the process of timetable construction, where all safety requirements must be met to ensure safe railway traffic. Presented are the factors affecting the operation and traffic management, stability of the timetable, and formation of delays in rail traffic. The proposed reinforcement learning method, in particular the Q learning algorithm, which is used for train rescheduling when delays occur and a new operation plan (timetable) should be prepared in real time, in a way that the delay of trains does not extend over the network, and that the total delay of all trains is minimized. The main objective of the algorithm is to eliminate the propagation of the delay between the trains as soon as possible with the correct decisions aimed at minimizing the negative impacts of the delays. In the dissertation a Q learning algorithm for train rescheduling is formulated (states, actions, rewards, environment, agent), a parametric study is carried out, and test cases prove the efficiency of the algorithm.

LITERATURA IN VIRI

Abdulhai, B., Kattan, L. 2003. Reinforcement learning : Introduction to theory and potential for transport applications. Canadian Journal of Civil Engineering, 30(6): 981–991.

Abdulhai, B., Pringle, R., Karakoulas, G. J. 2003. Reinforcement learning for true adaptive traffic signal control. Journal of Transportation Engineering, 129(3): 278–285.

Abril, M., Barber, F., Ingolotti, L., Salido, M. A., Tormos, P., Lova, A. 2007. An Assessment of Railway Capacity. Department of Information Systems and Computation, Department of Applied Statistics and Operational Research, and Quality, Technical University of Valencia.

Acuna-Agost, R., Michelon, P., Feillet, D., Gueye, S. 2011. A MIP-based local search method for the railway rescheduling problem. Networks, 57(1): 69–86.

Arel, I., Liu, C., Urbanik, T., Kohls, A. G. 2010. Reinforcement learning-based multi-agent system for network traffic signal control. IET Intelligent Transport Systems, 4(2): p. 128.

Assad, A. A. 1980. Models for rail transportation. Transportation Research Part A: General, 14(3): 205–220.

Cacchiani, V. et al. 2014. An overview of recovery models and algorithms for real-time railway rescheduling. Transportation Research Part B: Methodological, 63: 15–37.

Caimi, G. et al. 2012. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. Computers Operations Research, 39(11): 2578–2593.

Chiu, C. K. et al. 2002. A Constraint-Based Interactive Train Rescheduling Tool 1 Introduction. Constraints, 7(2): 167–198.

Cordeau, J. F., Paolo, T., Vigo, D. 1998. A Survey of Optimization Models for Train Routing and Scheduling. Transportation Science, 32(4): 380–404.

Corman, F. et al. 2010a. A tabu search algorithm for rerouting trains during rail operations. Transportation Research Part B: Methodological, 44(1): 175–192.

Corman, F. et al. 2010b. Centralized versus distributed systems to reschedule trains in two dispatching areas. Public Transport, 2(3): 219–247.

Corman, F. et al. 2011. Dispatching and coordination in multi-area railway traffic management.

D'Ariano, A. 2008. Improving real-time train dispatching: Models, Algorithms and Applications. Doktorska disertacija. Delft, Faculty of Civil Engineering and Geosciences, Department of Transport and Planning: 240 f.

D'Ariano, A., Corman, F. et al. 2008. Reordering and local rerouting strategies to manage train traffic in real time. Transportation Science, 42(4): 405–419.

D'Ariano, A., Pacciarelli, D., Pranzo, M. 2007. A branch and bound algorithm for scheduling trains in a railway network. European Journal of Operational Research, 183(2): 643–657.

D'Ariano, A., Pacciarelli, D., Pranzo, M. 2008. Assessment of flexible timetables in real-time traffic management of a railway bottleneck. Transportation Research Part C: Emerging Technologies, 16(2): 232–245.

D'Ariano, A., Pranzo, M., Hansen, I. A. 2007. Conflict resolution and train speed coordination for solving real-time timetable perturbations. IEEE Transactions on Intelligent Transportation Systems, 8(2): 208–222.

Dollevoet, T., Corman, F., Huisman, D., D'Ariano, A. 2012. An iterative optimization framework for delay management and train scheduling: p. 23.

Dorfman, M. J., Medanic, J. 2004. Scheduling trains on a railway network using a discrete event model of railway traffic. Transportation Research Part B: Methodological, 38(1): 81–98.

Dotoli, M., Epicoco, N., Falagario, M., Piconese, A., Sciancalepore, F., Turchiano, B., Bari, P. 2013. A real time traffic management model for regional railway networks under disturbances. V: Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), Madison, USA, August 17–20, 2013: 892–897.

Fan, B., Roberts, C., Weston, P. 2012. A comparison of algorithms for minimising delay costs in disturbed railway traffic scenarios. Journal of Rail Transport Planning Management, 2(1–2): 23–33.

Fay, A. 2000. A fuzzy knowledge-based system for railway traffic control. Engineering Applications of Artificial Intelligence, 13: 719–729.

Ge, Y. 2009. Software Project Rescheduling with Genetic Algorithms. V: Proceedings of the International Conference on Artificial Intelligence and Computational Intelligence, Sanghaj, China, November 7–8, 2009: 439–443.

Gély, L., Dessagne, G., Lérin, C. 2006. Modelling train re-scheduling with optimization and operational research techniques: Results and Applications at SNCF, Paris, France: 9 f.

Geske, U. 2006. Railway scheduling with declarative constraint programming. Applications of Declarative Programming and Knowledge Management: 117–134.

Ghoseiri, K., Szidarovszky, F., Asgharpour, M. J. 2004. A multi-objective train scheduling model and solution. Transportation Research Part B: Methodological, 38(10): 927–952.

Ghosh, S. 2001. Understanding complex, real-world systems through asynchronous, distributed decision-making algorithms. Journal of Systems and Software, 58(2): 153–167.

Gregoire, P., Desjardins, C., Laumonier, J., Chaib-draa, B. 2007. Urban traffic control based on learning agents. V: proceedings of the IEEE Intelligent Transportation Systems Conference, Washington, USA, September 30–October 3, 2007: 916–921.

Hansen, I. A., Pachl, J. 2008. Railway timetable traffic, Hamburg, Germany: Eurailpress.

Hara, K., Kumazawa, K., Koseki, T. 2006. Efficient algorithm for evaluating and optimizing train reschedules by taking advantage of flexibility of quadruple track. V: Proceedings of the Third International Conference on Railway Traction Systems, Tokio, Japan, November 12–15, 2006: p. 7.

Ho, K. T., Yeung, H. T. 2001. Railway junction traffic control by heuristic methods. IEE Proceedings - Electric Power Applications, 148(1): 77–84.

Huisman, T., Boucherie, R. J. 2001. Running times on railway sections with heterogeneous train traffic. Transportation Research Part B: 35(3): 271–292.

Hulea, M., Avram, C., Letia, T., Muresan, D., Radu, S. 2007. Distributed real-time railway simulator.

Humphrys, M. 1996. Action Selection methods using Reinforcement Learning.

Jespersen-Groth, J. et al. 2006. Disruption management in passenger railway transportation. Robust and Online Large-Scale Optimization: 399–421.

Kecman, P., Potthoff, D., Clausen, J., Huisman, D., Maroti, G., Nyhave, N. M. 2012. Rescheduling models for network-wide railway traffic management. V: Proceedings of the Conference on Advanced Systems for Public Transport, Santiago, Cile, July 23–27, 2012: 1–31.

Ke-Ping, L. 2010. Scheduling trains on railway network using random walk method. Chinese Physics B, 19(3): p. 6.

König, A. 2002. Reability of transport system and its influence on modal split. V: Proceedings of the Swiss Transport Research Conference, Ascona, Swiss, March 20–22, 2002: 1–19.

Kroon, L., Romeijn, E., Zwaneveld, P. 1997. Routing trains through railway stations : complexity issues. European Journal of Operationa, 98: 485–498.

Kumazawa, K., Hara, K., Koseki, T. 2010. A Novel Train Rescheduling Algorithm for Correcting Disrupted Train Operation in a Dense Urban Environment. V: Hansen I. A. (ur.), Timetable Planning and Information Quality. Wit Press/Computational Mechanics.

Kuster, J., Jannach, D., Friedrich, G. 2008. Applying local rescheduling in response to schedule disruptions. Annals of Operations Research, 180(1): 265–282.

Landex, A. 2008. Passenger delays - Methods to estimate railway capacity and passenger delays.

Letia, T., Hulea, M., Miron, R. 2008. Distributed scheduling for real-time railway traffic control. V: Proceedings of the International Multiconference on Computer Science and Information Technology, Amsterdam, Netherlands, July 22–27, 200: 679–685.

Louwerse, I., Huisman, D. 2014. Adjusting a railway timetable in case of partial or complete blockades. European Journal of Operational Research, 235(3): 583–593.

Luethi, M. 2009. Improving the efficiency of heavily used railway networks through integrated real-time rescheduling. Doktorska disertacija. Zurich, Swiss Federal Institute of Technology: 286 f.

Luethi, M., Nash, A., Weidmann, U., Laube, F., Wuest, R. 2007. Increasing railway capacity and reliability through integrated real-time rescheduling. V: Proceedings of the International Seminar on Railway Operations Modelling and Analysis (IAROR), Hanover, Germany, March 28–30, 2007: p. 17.

Luethi, M., Laube, F., Medeossi, G., 2007. Rescheduling and train control: A new framework for railroad traffic control in heavily used networks. Proceedings of the 86th Transportation Research: p. 13.

Mannen, H. 2003. Learning to play chess using reinforcement learning with database games.

Mascis, A., Pacciarelli, D., Pranzo, M. 2004. Scheduling models for short-term railway traffic optimisation.

Medanic, J., Dorfman, M. J. 2002. Efficient scheduling of traffic on a railway line. Journal of Optimization Theory and Applications, 115(3): 587–602.

Medanic, J., Dorfman, M. J. 2002. Energy efficient strategies for scheduling trains on a line. V: Proceedings of the 15th IFAC World Congress, Barcelona, Spain, July 21–26, 2002: p. 6.

Min, Y., Park, M., Hong, S., Hong, S. 2011. An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network. Transportation Research Part B, 45: 409–429.

Mladenović, S., Čungalović, M. 2007. Heuristic Approach to Train Rescheduling. Yugoslav Journal of Operations Research, 17(1): 9–29.

Nagasaki, Y., Eguchi, M., Koseki, T. 2003. Automatic generation and evaluation of urban railway rescheduling plan. V: International Symposium on Speed-up and Service Technology for Railway and Maglev Systems. Tokyo, Japan: 26–31.

Narayanaswami, S. & Rangaraj, N. 2013. Modelling disruptions and resolving conflicts optimally in a railway schedule. Computers & Industrial Engineering, 64(1): 469–481.

Norio, T., Yoshiaki, T., Noriyuki, T., Chikara, H. 2005. Train rescheduling algorithm which minimizes passengers' dissatisfaction. V: proceedings of the Innovations in Applied Artificial Intelligence, Bari, Italy, June 22-24, 2005: 829–838.

Olivera, E. S. 2001. Solving single-track railway scheduling problem using constraint programming. Doktorska disertacija. Leeds, Univeristy of Leeds, School of Computing: 129 f.

Pachl, J. 2011. BRAUNSCHWEIG Deadlock Avoidance in Railroad Operations Simulations Jörn Pachl Braunschweig : Institute of Railway Systems Engineering and Traffic Safety, 2011 Textfassung eines Vortrages auf dem 90th Annual Meeting des Transportation Research Board in Washington, (11): 23–27.

Pellegrini, P., Marlière, G., Rodriguez, J. 2014. Optimal train routing and scheduling for managing traffic perturbations in complex junctions. Transportation Research Part B: Methodological, 59: 58–80.

Ping, L., Axin, N., Limin, J., Fuzhang, W. 2001. Study on intelligent train dispatching. V: Proceedings of the IEEE Intelligent Transportation Systems Conference, Oakland, USA, August 25–29, 2001: 949–953.

Prashanth, L. A., Bhatnagar, S., Member, S. 2011. Reinforcement learning with function approximation for traffic signal control. Intelligent Transportation Systems, IEEE Transactions on ITS, 12(2): 412–421.

Rodriguez, J. 2007. A constraint programming model for real-time train scheduling at junctions. Transportation Research Part B: Methodological, 41(2): 231–245.

Russell, S. J., Norvig, P. 2003. Artificial Intelligence, A Modern Approach 3rd ed., Pearson Education ©2003.

Sajedinejad, A., Mardani, S., Hassannayebi, E., Mohammadi, A. R. M. 2011. SIMARAIL: Simulation based optimization software for scheduling railway network. V: Proceedings of the 2011 Winter Simulation Conference, Phoenix, USA, December 11–14, 2011: 3730–3741.

Shoufeng, L., Ximin, L., Shiqiang, D. 2008. Q-learning for adaptive traffic signal control based on delay minimization strategy. V: Proceedings of the IEEE International Conference on Networking, Sensing and Control, Hainan, China April 6–8, 2008: 687–691.

Strotmann, C. 2007. Railway scheduling problems and their decomposition at Osnabr uck. Doktorska disertacija. Osnabruck, Osnabrück University: 124 f.

Sutton, R. S., Barto, A. G. 1998. Reinforcement Learning: An Introduction, MIT Press Cambridge, USA.

Tazoniero, A., Gonçalves, R., Gomide, F. 2007. Decision making strategies for real-time train dispatch and control. Analysis and Design of Intelligent Systems using Soft Computing Techniques. Springer Berlin Heidelberg: 195–204.

Tazoniero, A., Gonçalves, R., Gomide, F. 2005. Fuzzy algorithm for real-time train dispatch and control. Fuzzy Information Processing Society, 2005. Annual Meeting of the North American: 332–336.

Theeg, G., Vlasenko, S. 2009. Railway Signalling Interlocking: International Compendium 1st ed., Hamburg: DVV Media Group.

Tokic, M. 2010. Adaptive ε-greedy exploration in reinforcement learning based on value differences. V: Dillmann R.et al. (ur.), Advances in Artificial Intelligence, 33rd Annual German Conference on AI. Karlsruhe, Germany: 203–210.

Tornquist, J. 2006. Railway traffic disturbance. Doktorska disertacija. Karlskrona, School of Engineering, Blekinge Institute of Technology, Department of Software Engineering (založba: Blekinge Institute of Technology): 206 f.

Törnquist, J. 2006. Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. Algorithmic Methods and Models for Optimization of Railways. Palma de Mallorca, Spain: 23 f.

Törnquist, J. 2007. Railway traffic disturbance management-An experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. Transportation Research Part A: Policy and Practice, 41(3): 249–266.

Törnquist, J., Davidsson, P. 2002. A Multi-Agent System Approach to Train Delay Handling. V: Proceedings of Agent Technologies in Logistics Work-shop, the 15th European Conference on Artificial Intelligence, Lyon, France, July 21–26, 2002: 4 f.

Törnquist, J., Persson, J. A. 2007. N-tracked railway traffic re-scheduling during disturbances. Transportation Research Part B: Methodological, 41(3): 342–362.

Törnquist, J. Persson, J. A. 2005b. Train traffic deviation handling using tabu search and simulated annealing. V: Proceedings of the 38th Hawaii International Conference on System Sciences, Hawaii, USA, January 3–6, 2005: p. 1–10.

Törnquist Krasemann, J. 2012. Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. Transportation Research Part C: Emerging Technologies, 20(1): 62–78.

Watkins, C. J. C. H., Dayan, P. 1992. Q-Learning, Technical Note. Machine Learning, (3): 279–292.

Wegele, S., Schnieder, E. 2004. Dispatching of train operations using genetic algorithms. V: Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport, San Diego, USA, August 9–11: p. 1–9.

Xu, X., Li, K., Yang, L. 2015. Scheduling heterogeneous train traffic on double tracks with efficient dispatching rules. Transportation Research Part B: Methodological, 78: 364–384.

Zgonc, B. 2012. Železniška infrastruktura, Portorož: Univerza v Ljubljani, Fakulteta za pomorstvo in promet.

Zhan, S., Kroon, L. G., Veelenturf, L. P., Wagenaar, J. C. 2015. Real-time high-speed train rescheduling in case of a complete blockage. Transportation Research Part B: Methodological, 78: 182–201.

Zou, L., Xu, J., Zhu, L. 2006. Light rail intelligent dispatching system based on reinforcement learning. V: Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Pittsburgh, USA, June 25–29, 2006: 2493–2496.

Signalni pravilnik. Uradni list RS, št. 123/2007: 18085

Prometni pravilnik. Uradni list RS, št. 50/2011: 6824

Priloga A: Učenje Q z zakasnjeno nagrado, Eksperiment a – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,1	0,1	10	12	12	12	14	17	14	13	18	12
0,1	0,1	30	12	12	14	14	13	12	12	14	14
0,1	0,1	50	12	12	14	14	18	18	12	12	16
0,1	0,1	70	12	12	12	12	14	14	12	14	14
0,1	0,1	90	12	12	12	12	14	14	12	14	14
0,1	0,3	10	12	12	12	14	12	12	14	18	22
0,1	0,3	30	12	14	13	15	14	14	12	16	12
0,1	0,3	50	12	14	15	13	12	14	16	15	18
0,1	0,3	70	12	12	12	12	13	12	14	16	16
0,1	0,3	90	12	12	12	12	13	12	14	16	16
0,1	0,5	10	12	12	12	16	12	17	18	16	16
0,1	0,5	30	12	12	17	12	15	16	20	12	16
0,1	0,5	50	14	12	12	12	14	16	14	19	22
0,1	0,5	70	12	12	12	12	12	12	15	16	16
0,1	0,5	90	12	12	12	12	12	12	15	16	16
0,1	0,7	10	14	12	12	14	16	15	16	17	16
0,1	0,7	30	12	14	18	13	18	22	17	22	12
0,1	0,7	50	13	13	14	12	18	18	14	17	22
0,1	0,7	70	12	15	14	12	12	16	12	17	18
0,1	0,7	90	12	15	14	12	12	16	12	17	18
0,1	0,9	10	14	17	14	15	15	12	12	15	14
0,1	0,9	30	12	12	12	14	16	17	12	15	13
0,1	0,9	50	13	12	14	13	16	16	20	20	21
0,1	0,9	70	12	16	15	12	12	16	12	19	19
0,1	0,9	90	12	16	15	12	12	16	12	19	19
0,3	0,1	10	14	16	17	14	14	12	15	12	14
0,3	0,1	30	16	15	17	15	17	14	12	14	12
0,3	0,1	50	12	12	16	13	16	16	12	18	17
0,3	0,1	70	12	16	12	12	12	12	12	14	14
0,3	0,1	90	12	16	12	12	12	12	12	14	14
0,3	0,3	10	14	12	16	14	12	18	14	18	12
0,3	0,3	30	17	14	20	14	16	12	13	21	16
0,3	0,3	50	12	16	14	12	18	18	16	20	19
0,3	0,3	70	12	12	12	12	12	16	13	19	18
0,3	0,3	90	12	12	12	12	12	16	13	19	18
0,3	0,5	10	15	13	14	18	12	18	12	12	18
0,3	0,5	30	12	16	21	19	20	20	14	21	13
0,3	0,5	50	12	12	17	14	16	20	18	18	22
0,3	0,5	70	12	12	12	12	17	17	12	17	19
0,3	0,5	90	12	12	12	12	17	17	12	17	19
0,3	0,7	10	15	13	15	19	17	16	14	19	18
0,3	0,7	30	16	17	20	19	18	26	19	13	25
0,3	0,7	50	13	14	18	14	22	23	18	24	23

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,3	0,7	70	12	12	12	12	18	16	12	19	21
0,3	0,7	90	12	12	12	12	18	16	12	19	21
0,3	0,9	10	14	18	20	18	14	18	12	13	17
0,3	0,9	30	19	17	23	18	25	24	17	25	27
0,3	0,9	50	13	12	16	14	18	22	18	20	24
0,3	0,9	70	12	12	12	13	16	18	12	18	19
0,3	0,9	90	12	12	12	13	16	18	12	18	19
0,5	0,1	10	12	14	14	17	18	20	12	12	20
0,5	0,1	30	15	16	20	15	18	20	16	23	24
0,5	0,1	50	12	16	16	14	18	17	19	22	19
0,5	0,1	70	14	12	15	12	16	16	16	16	16
0,5	0,1	90	14	12	15	12	16	16	16	16	16
0,5	0,3	10	16	21	16	16	16	14	18	16	12
0,5	0,3	30	18	20	21	19	17	22	14	22	17
0,5	0,3	50	12	12	16	16	19	20	20	18	21
0,5	0,3	70	12	12	15	14	14	18	16	17	18
0,5	0,3	90	12	12	15	14	14	18	16	17	18
0,5	0,5	10	12	20	19	14	12	15	13	13	12
0,5	0,5	30	15	19	21	14	23	26	16	18	13
0,5	0,5	50	12	16	17	18	18	22	18	22	22
0,5	0,5	70	13	13	14	12	16	16	12	17	21
0,5	0,5	90	13	13	14	12	16	16	12	17	21
0,5	0,7	10	16	17	12	14	16	14	14	12	12
0,5	0,7	30	14	20	23	14	23	24	18	15	22
0,5	0,7	50	12	16	19	18	19	21	20	23	23
0,5	0,7	70	12	12	12	14	14	18	15	18	18
0,5	0,7	90	12	12	12	14	14	18	15	18	18
0,5	0,9	10	14	19	17	13	13	13	13	18	14
0,5	0,9	30	15	19	24	20	22	22	21	27	23
0,5	0,9	50	12	14	20	12	18	23	20	23	23
0,5	0,9	70	12	12	12	12	16	16	16	18	18
0,5	0,9	90	12	12	12	12	16	16	16	18	18
0,7	0,1	10	14	15	12	12	19	13	19	12	14
0,7	0,1	30	18	20	18	15	17	18	16	17	21
0,7	0,1	50	12	15	18	15	16	17	16	18	22
0,7	0,1	70	14	16	14	12	16	16	16	16	16
0,7	0,1	90	14	16	14	12	16	16	16	16	16
0,7	0,3	10	14	20	13	16	14	14	14	14	15
0,7	0,3	30	17	18	18	14	20	23	18	21	14
0,7	0,3	50	14	18	14	14	19	19	19	19	21
0,7	0,3	70	12	13	16	12	14	14	17	17	18
0,7	0,3	90	12	13	16	12	14	14	17	17	18
0,7	0,5	10	18	12	17	15	14	18	16	16	19
0,7	0,5	30	15	17	24	19	18	17	21	18	25
0,7	0,5	50	14	13	20	18	16	20	18	22	20

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,7	0,5	70	12	12	14	12	17	16	18	18	18
0,7	0,5	90	12	12	14	12	17	16	18	18	18
0,7	0,7	10	20	15	14	16	17	15	14	17	18
0,7	0,7	30	16	21	13	19	23	21	18	24	17
0,7	0,7	50	14	15	16	19	21	21	21	22	21
0,7	0,7	70	12	12	12	12	16	16	16	19	20
0,7	0,7	90	12	12	12	12	16	16	16	19	20
0,7	0,9	10	15	16	16	13	21	12	17	19	21
0,7	0,9	30	13	21	22	16	18	24	21	24	21
0,7	0,9	50	12	12	16	13	17	22	18	22	23
0,7	0,9	70	12	12	12	16	16	18	17	20	18
0,7	0,9	90	12	12	12	16	16	18	17	20	18
0,9	0,1	10	16	14	12	13	14	14	21	13	18
0,9	0,1	30	17	19	17	16	24	18	22	15	15
0,9	0,1	50	14	16	17	16	18	19	19	22	20
0,9	0,1	70	12	12	12	14	14	15	18	18	18
0,9	0,1	90	12	12	12	14	14	15	18	18	18
0,9	0,3	10	12	16	14	14	18	16	16	13	16
0,9	0,3	30	12	19	18	21	18	17	15	18	15
0,9	0,3	50	12	15	16	13	20	21	19	18	19
0,9	0,3	70	12	12	12	16	14	16	17	17	18
0,9	0,3	90	12	12	12	16	14	16	17	17	18
0,9	0,5	10	14	14	17	13	14	22	17	19	22
0,9	0,5	30	12	17	18	12	22	20	17	18	14
0,9	0,5	50	16	18	19	13	19	21	18	19	16
0,9	0,5	70	12	12	12	12	16	17	17	22	17
0,9	0,5	90	12	12	12	12	16	17	17	22	17
0,9	0,7	10	17	13	14	15	13	20	17	16	12
0,9	0,7	30	14	17	16	20	17	22	20	16	18
0,9	0,7	50	16	12	16	16	19	18	16	15	18
0,9	0,7	70	12	12	12	12	14	18	14	19	18
0,9	0,7	90	12	12	12	12	14	18	14	19	18
0,9	0,9	10	21	14	15	15	14	19	20	14	16
0,9	0,9	30	12	21	17	21	13	19	16	22	17
0,9	0,9	50	12	18	18	14	18	18	19	20	19
0,9	0,9	70	12	20	13	12	16	16	14	17	17
0,9	0,9	90	12	20	13	12	16	16	14	17	17

Priloga B: Učenje Q z zakasnjeno nagrado, Eksperiment b – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,1	0,1	10	9	9	11	11	12	11	9	13	13
0,1	0,1	30	9	12	9	11	11	13	11	14	9
0,1	0,1	50	9	10	11	11	14	14	9	13	13
0,1	0,1	70	9	9	9	11	11	11	9	9	9
0,1	0,1	90	9	9	9	11	11	11	9	9	9
0,1	0,3	10	11	11	9	11	12	12	16	10	13
0,1	0,3	30	11	10	13	11	11	15	12	14	16
0,1	0,3	50	11	11	11	12	11	14	13	14	14
0,1	0,3	70	9	11	11	11	11	11	9	9	9
0,1	0,3	90	9	11	11	11	11	11	9	9	9
0,1	0,5	10	12	10	9	11	11	12	13	9	15
0,1	0,5	30	11	10	12	12	15	12	13	14	17
0,1	0,5	50	10	11	11	13	14	13	14	14	15
0,1	0,5	70	9	9	12	11	12	12	9	11	13
0,1	0,5	90	9	9	12	11	12	12	9	11	13
0,1	0,7	10	9	10	10	13	12	12	13	10	13
0,1	0,7	30	11	9	11	12	16	17	13	16	18
0,1	0,7	50	11	10	10	12	14	15	10	16	16
0,1	0,7	70	9	11	13	11	13	13	9	13	13
0,1	0,7	90	9	11	13	11	13	13	9	13	13
0,1	0,9	10	9	12	11	11	14	12	13	14	15
0,1	0,9	30	11	10	12	14	15	18	15	14	19
0,1	0,9	50	11	14	15	13	14	12	13	17	14
0,1	0,9	70	11	12	13	11	12	14	9	13	13
0,1	0,9	90	11	12	13	11	12	14	9	13	13
0,3	0,1	10	12	13	9	12	11	15	9	13	13
0,3	0,1	30	11	12	13	12	13	14	11	14	15
0,3	0,1	50	11	12	10	12	12	14	13	14	14
0,3	0,1	70	10	11	11	11	12	12	9	13	13
0,3	0,1	90	10	11	11	11	12	12	9	13	13
0,3	0,3	10	10	10	12	13	11	11	15	14	14
0,3	0,3	30	12	16	11	14	15	13	13	18	13
0,3	0,3	50	11	11	12	14	15	14	16	15	14
0,3	0,3	70	9	13	13	11	13	13	9	13	15
0,3	0,3	90	9	13	13	11	13	13	9	13	15
0,3	0,5	10	13	16	14	11	12	12	12	13	17
0,3	0,5	30	11	10	13	14	16	14	15	18	16
0,3	0,5	50	9	11	13	13	13	15	13	16	18
0,3	0,5	70	9	11	11	11	11	14	13	13	14
0,3	0,5	90	9	11	11	11	11	14	13	13	14
0,3	0,7	10	10	13	14	11	13	15	15	14	15
0,3	0,7	30	11	11	12	13	16	13	15	17	15
0,3	0,7	50	11	11	10	15	15	18	13	16	17

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,3	0,7	70	11	11	11	12	13	15	13	13	15
0,3	0,7	90	11	11	11	12	13	15	13	13	15
0,3	0,9	10	12	16	16	13	16	14	15	14	21
0,3	0,9	30	13	11	19	14	13	20	15	19	19
0,3	0,9	50	11	11	15	12	17	17	15	18	15
0,3	0,9	70	10	13	13	13	14	15	13	13	15
0,3	0,9	90	10	13	13	13	14	15	13	13	15
0,5	0,1	10	10	10	12	13	13	12	16	14	14
0,5	0,1	30	13	11	10	13	15	15	16	14	15
0,5	0,1	50	11	11	10	12	13	14	15	14	16
0,5	0,1	70	9	10	11	11	11	11	13	13	13
0,5	0,1	90	9	10	11	11	11	11	13	13	13
0,5	0,3	10	11	11	11	14	12	11	13	16	14
0,5	0,3	30	11	10	11	11	13	13	13	16	15
0,5	0,3	50	11	11	11	15	13	13	13	16	17
0,5	0,3	70	11	11	11	11	12	12	13	14	13
0,5	0,3	90	11	11	11	11	12	12	13	14	13
0,5	0,5	10	14	10	16	13	14	13	15	13	14
0,5	0,5	30	10	15	19	16	12	12	14	17	15
0,5	0,5	50	11	12	12	12	15	12	16	15	16
0,5	0,5	70	11	13	13	12	14	15	14	14	14
0,5	0,5	90	11	13	13	12	14	15	14	14	14
0,5	0,7	10	11	12	17	12	13	15	9	18	14
0,5	0,7	30	13	15	18	14	16	15	15	16	17
0,5	0,7	50	11	11	13	13	14	13	16	18	19
0,5	0,7	70	11	13	14	11	13	13	13	14	16
0,5	0,7	90	11	13	14	11	13	13	13	14	16
0,5	0,9	10	11	18	13	16	15	11	14	14	15
0,5	0,9	30	13	14	12	13	14	15	15	15	19
0,5	0,9	50	11	16	14	14	17	16	16	17	18
0,5	0,9	70	12	12	12	13	12	15	13	14	15
0,5	0,9	90	12	12	12	13	12	15	13	14	15
0,7	0,1	10	10	11	11	12	11	11	13	14	13
0,7	0,1	30	12	15	11	13	12	11	15	16	14
0,7	0,1	50	11	11	10	14	12	13	13	15	14
0,7	0,1	70	11	12	10	11	15	12	13	13	13
0,7	0,1	90	11	12	10	11	15	12	13	13	13
0,7	0,3	10	15	14	10	11	11	11	14	13	13
0,7	0,3	30	10	13	16	13	12	14	15	15	15
0,7	0,3	50	11	11	11	13	12	12	16	16	14
0,7	0,3	70	10	11	11	12	13	12	14	13	13
0,7	0,3	90	10	11	11	12	13	12	14	13	13
0,7	0,5	10	13	12	11	16	11	11	18	14	13
0,7	0,5	30	10	13	12	15	13	12	15	15	14
0,7	0,5	50	10	11	11	14	12	13	15	16	15

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,7	0,5	70	11	12	12	11	13	14	15	15	15
0,7	0,5	90	11	12	12	11	13	14	15	15	15
0,7	0,7	10	14	13	13	13	13	11	17	16	13
0,7	0,7	30	13	16	14	14	16	15	16	17	20
0,7	0,7	50	12	11	11	14	17	13	15	15	14
0,7	0,7	70	12	13	14	13	13	14	13	16	16
0,7	0,7	90	12	13	14	13	13	14	13	16	16
0,7	0,9	10	15	12	22	13	21	17	15	16	18
0,7	0,9	30	16	19	14	13	13	14	15	20	21
0,7	0,9	50	13	12	11	13	14	13	13	18	17
0,7	0,9	70	11	13	13	12	15	13	13	15	16
0,7	0,9	90	11	13	13	12	15	13	13	15	16
0,9	0,1	10	11	13	12	16	12	11	14	13	11
0,9	0,1	30	13	13	12	12	12	12	15	14	14
0,9	0,1	50	11	11	11	12	12	12	13	15	14
0,9	0,1	70	11	11	11	11	13	14	13	14	15
0,9	0,1	90	11	11	11	11	13	14	13	14	15
0,9	0,3	10	11	16	9	12	11	11	14	10	13
0,9	0,3	30	11	13	11	13	12	11	15	15	14
0,9	0,3	50	10	10	10	15	12	12	14	15	14
0,9	0,3	70	11	11	10	13	13	13	13	13	15
0,9	0,3	90	11	11	10	13	13	13	13	13	15
0,9	0,5	10	10	15	11	12	11	12	15	13	9
0,9	0,5	30	12	12	11	13	15	11	14	13	15
0,9	0,5	50	11	10	11	13	12	12	13	13	15
0,9	0,5	70	9	11	11	13	14	13	13	17	16
0,9	0,5	90	9	11	11	13	14	13	13	17	16
0,9	0,7	10	12	10	14	16	11	11	14	13	13
0,9	0,7	30	11	12	12	13	13	13	14	17	17
0,9	0,7	50	12	12	11	12	12	13	15	15	14
0,9	0,7	70	11	11	11	13	14	12	13	15	16
0,9	0,7	90	11	11	11	13	14	12	13	15	16
0,9	0,9	10	13	11	11	16	14	14	12	14	14
0,9	0,9	30	13	11	11	13	13	11	15	19	13
0,9	0,9	50	11	13	12	13	15	12	14	15	19
0,9	0,9	70	13	13	11	13	15	15	13	13	17
0,9	0,9	90	13	13	11	13	15	15	13	13	17

Priloga C: Učenje Q z zakasnjeno nagrado, Eksperiment c – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,1	0,1	10	14	15	14	6	9	9	11	10	10
0,1	0,1	30	15	14	12	9	7	8	9	12	13
0,1	0,1	50	12	14	14	6	10	8	9	9	12
0,1	0,1	70	13	13	13	6	8	7	9	10	10
0,1	0,1	90	13	13	13	6	8	7	9	10	10
0,1	0,3	10	15	13	16	9	11	8	11	11	8
0,1	0,3	30	12	12	20	8	6	13	12	12	13
0,1	0,3	50	15	14	14	9	10	11	10	13	15
0,1	0,3	70	13	13	13	6	7	6	9	9	12
0,1	0,3	90	13	13	13	6	7	6	9	9	12
0,1	0,5	10	13	15	16	8	15	8	13	10	16
0,1	0,5	30	14	15	13	11	9	9	14	14	14
0,1	0,5	50	13	13	15	8	8	10	9	13	15
0,1	0,5	70	13	13	13	6	6	8	10	11	11
0,1	0,5	90	13	13	13	6	6	8	10	11	11
0,1	0,7	10	12	13	13	7	7	6	11	15	15
0,1	0,7	30	16	15	14	11	13	13	13	13	12
0,1	0,7	50	13	17	16	10	10	12	9	15	15
0,1	0,7	70	13	12	13	6	6	7	9	13	13
0,1	0,7	90	13	12	13	6	6	7	9	13	13
0,1	0,9	10	14	12	15	9	10	9	9	16	11
0,1	0,9	30	14	19	15	12	13	16	12	15	16
0,1	0,9	50	14	13	16	12	13	13	13	13	14
0,1	0,9	70	13	13	13	7	8	7	9	13	12
0,1	0,9	90	13	13	13	7	8	7	9	13	12
0,3	0,1	10	14	20	13	6	13	11	9	16	14
0,3	0,1	30	16	15	13	10	13	11	10	14	14
0,3	0,1	50	14	13	15	9	9	10	11	14	14
0,3	0,1	70	13	13	13	6	7	8	9	13	12
0,3	0,1	90	13	13	13	6	7	8	9	13	12
0,3	0,3	10	13	14	19	9	7	18	12	14	15
0,3	0,3	30	12	13	13	12	13	13	12	12	19
0,3	0,3	50	15	14	18	8	12	13	10	14	13
0,3	0,3	70	13	13	13	6	7	8	10	14	14
0,3	0,3	90	13	13	13	6	7	8	10	14	14
0,3	0,5	10	15	17	13	15	10	17	8	12	14
0,3	0,5	30	12	14	16	11	15	18	12	16	20
0,3	0,5	50	14	14	14	9	13	13	11	12	17
0,3	0,5	70	12	15	13	7	8	9	10	10	14
0,3	0,5	90	12	15	13	7	8	9	10	10	14
0,3	0,7	10	14	14	15	15	17	19	12	15	16
0,3	0,7	30	14	15	21	12	13	13	14	19	20
0,3	0,7	50	13	16	14	11	13	13	12	17	15

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,3	0,7	70	13	16	16	7	9	9	10	11	12
0,3	0,7	90	13	16	16	7	9	9	10	11	12
0,3	0,9	10	18	13	19	7	7	15	11	13	17
0,3	0,9	30	14	20	18	13	15	18	14	13	20
0,3	0,9	50	15	16	13	11	15	13	12	13	18
0,3	0,9	70	14	17	15	8	9	8	10	12	12
0,3	0,9	90	14	17	15	8	9	8	10	12	12
0,5	0,1	10	14	15	13	9	16	10	10	11	13
0,5	0,1	30	13	15	15	10	14	19	13	16	17
0,5	0,1	50	15	18	15	9	10	12	12	15	12
0,5	0,1	70	12	16	17	6	7	10	9	10	13
0,5	0,1	90	12	16	17	6	7	10	9	10	13
0,5	0,3	10	15	15	20	13	15	18	15	14	20
0,5	0,3	30	13	18	17	12	13	19	15	18	21
0,5	0,3	50	14	14	15	9	12	13	12	13	13
0,5	0,3	70	15	13	13	6	6	8	11	12	12
0,5	0,3	90	15	13	13	6	6	8	11	12	12
0,5	0,5	10	13	21	18	15	15	15	18	16	18
0,5	0,5	30	15	16	19	9	13	13	14	19	21
0,5	0,5	50	13	15	15	8	12	13	13	15	17
0,5	0,5	70	13	16	17	6	8	11	12	10	12
0,5	0,5	90	13	16	17	6	8	11	12	10	12
0,5	0,7	10	12	16	18	11	14	12	13	19	18
0,5	0,7	30	15	15	16	13	16	14	17	21	18
0,5	0,7	50	15	13	19	6	13	13	13	13	15
0,5	0,7	70	13	14	16	7	8	11	12	13	13
0,5	0,7	90	13	14	16	7	8	11	12	13	13
0,5	0,9	10	14	15	17	9	11	15	15	19	16
0,5	0,9	30	15	16	22	13	14	14	14	20	17
0,5	0,9	50	17	13	19	12	15	13	12	12	16
0,5	0,9	70	14	13	15	7	9	11	12	12	12
0,5	0,9	90	14	13	15	7	9	11	12	12	12
0,7	0,1	10	14	17	13	12	17	13	13	18	17
0,7	0,1	30	16	14	14	11	13	16	12	13	17
0,7	0,1	50	14	14	18	9	10	14	12	13	14
0,7	0,1	70	15	15	15	6	6	8	10	12	12
0,7	0,1	90	15	15	15	6	6	8	10	12	12
0,7	0,3	10	13	16	18	14	14	19	17	18	10
0,7	0,3	30	19	21	13	11	14	14	14	20	18
0,7	0,3	50	17	14	18	12	13	13	13	15	13
0,7	0,3	70	13	16	13	6	8	8	11	12	13
0,7	0,3	90	13	16	13	6	8	8	11	12	13
0,7	0,5	10	16	18	20	11	18	20	11	16	23
0,7	0,5	30	16	14	18	13	14	14	16	21	22
0,7	0,5	50	15	15	15	12	13	13	12	15	15

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,7	0,5	70	13	14	14	6	9	8	10	12	12
0,7	0,5	90	13	14	14	6	9	8	10	12	12
0,7	0,7	10	14	22	20	13	14	11	16	17	21
0,7	0,7	30	15	15	19	13	14	16	15	21	20
0,7	0,7	50	12	14	22	13	12	13	13	12	16
0,7	0,7	70	13	13	13	6	8	10	11	12	12
0,7	0,7	90	13	13	13	6	8	10	11	12	12
0,7	0,9	10	19	15	20	12	15	15	12	17	26
0,7	0,9	30	14	14	15	13	14	13	15	20	23
0,7	0,9	50	12	15	15	10	13	13	15	15	17
0,7	0,9	70	16	13	13	6	10	11	11	11	13
0,7	0,9	90	16	13	13	6	10	11	11	11	13
0,9	0,1	10	14	21	19	15	18	22	15	15	19
0,9	0,1	30	14	15	18	13	14	16	15	17	19
0,9	0,1	50	13	16	13	11	12	13	14	15	12
0,9	0,1	70	13	15	14	6	9	11	11	12	14
0,9	0,1	90	13	15	14	6	9	11	11	12	14
0,9	0,3	10	13	21	17	14	17	10	14	14	16
0,9	0,3	30	15	15	18	13	15	17	18	19	17
0,9	0,3	50	14	14	18	11	13	13	12	15	15
0,9	0,3	70	13	15	19	6	9	10	12	14	12
0,9	0,3	90	13	15	19	6	9	10	12	14	12
0,9	0,5	10	14	16	17	8	9	11	15	17	20
0,9	0,5	30	14	17	17	13	14	13	16	19	21
0,9	0,5	50	13	13	16	10	13	13	15	17	16
0,9	0,5	70	13	14	16	8	9	12	12	12	12
0,9	0,5	90	13	14	16	8	9	12	12	12	12
0,9	0,7	10	13	15	20	14	18	15	13	16	21
0,9	0,7	30	15	15	15	13	15	16	15	21	20
0,9	0,7	50	14	14	14	11	12	13	15	14	15
0,9	0,7	70	13	18	15	7	10	11	10	12	13
0,9	0,7	90	13	18	15	7	10	11	10	12	13
0,9	0,9	10	13	17	17	11	14	13	16	15	16
0,9	0,9	30	16	15	19	13	15	13	18	18	21
0,9	0,9	50	13	15	17	11	13	13	13	15	18
0,9	0,9	70	13	14	14	7	9	11	11	13	13
0,9	0,9	90	13	14	14	7	9	11	11	13	13

Priloga D: Učenje Q z zakasnjeno nagrado in sledmi, Eksperiment a – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,1	0,1	10	12	12	12	13	13	13	12	12	12
0,1	0,1	30	12	12	12	13	12	12	12	12	12
0,1	0,1	50	12	12	12	12	13	12	12	12	12
0,1	0,1	70	12	12	12	13	12	12	12	12	12
0,1	0,1	90	12	12	12	13	12	12	12	12	12
0,1	0,3	10	12	12	12	13	13	13	12	12	12
0,1	0,3	30	12	12	12	13	12	12	12	12	12
0,1	0,3	50	12	12	12	12	12	12	12	12	12
0,1	0,3	70	12	12	13	12	12	12	12	12	13
0,1	0,3	90	12	12	13	12	12	12	12	12	13
0,1	0,5	10	12	12	12	13	13	13	12	12	12
0,1	0,5	30	12	12	12	12	12	12	12	12	12
0,1	0,5	50	12	12	12	12	12	12	12	12	12
0,1	0,5	70	12	12	12	12	12	12	12	12	12
0,1	0,5	90	12	12	12	12	12	12	12	12	12
0,1	0,7	10	12	12	12	14	14	14	12	12	12
0,1	0,7	30	12	12	12	14	14	14	12	12	12
0,1	0,7	50	12	12	12	14	14	14	12	12	12
0,1	0,7	70	12	12	12	14	14	14	13	13	13
0,1	0,7	90	12	12	12	14	14	14	13	13	13
0,1	0,9	10	12	12	12	14	14	14	12	12	12
0,1	0,9	30	12	12	12	14	14	14	12	12	12
0,1	0,9	50	12	12	12	14	14	14	12	12	12
0,1	0,9	70	12	12	12	14	14	14	12	12	12
0,1	0,9	90	12	12	12	14	14	14	12	12	12
0,3	0,1	10	12	12	12	13	12	12	12	12	12
0,3	0,1	30	12	12	12	12	12	12	12	12	12
0,3	0,1	50	12	12	12	12	12	12	12	12	12
0,3	0,1	70	12	12	12	12	12	12	12	12	12
0,3	0,1	90	12	12	12	12	12	12	12	12	12
0,3	0,3	10	12	12	12	12	12	12	12	12	12
0,3	0,3	30	12	12	12	12	12	12	12	12	12
0,3	0,3	50	12	12	12	12	12	12	12	12	12
0,3	0,3	70	12	12	12	12	12	12	12	12	12
0,3	0,3	90	12	12	12	12	12	12	12	12	12
0,3	0,5	10	12	12	12	13	12	12	12	12	12
0,3	0,5	30	12	12	12	12	12	12	12	12	12
0,3	0,5	50	12	12	12	12	12	12	12	12	12
0,3	0,5	70	12	12	12	12	12	12	12	12	12
0,3	0,5	90	12	12	12	12	12	12	12	12	12
0,3	0,7	10	12	12	12	14	14	14	12	12	12
0,3	0,7	30	12	12	12	14	14	14	12	12	12

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,3	0,7	50	12	12	12	14	14	14	12	12	12
0,3	0,7	70	12	12	12	14	14	14	13	13	13
0,3	0,7	90	12	12	12	14	14	14	13	13	13
0,3	0,9	10	12	12	12	14	14	14	12	12	12
0,3	0,9	30	12	12	12	14	14	14	12	12	12
0,3	0,9	50	12	12	12	14	14	14	12	12	12
0,3	0,9	70	12	12	12	14	14	14	12	12	12
0,3	0,9	90	12	12	12	14	14	14	12	12	12
0,5	0,1	10	12	12	12	12	12	12	12	12	12
0,5	0,1	30	12	12	12	12	12	12	12	12	12
0,5	0,1	50	12	12	12	12	12	12	12	12	12
0,5	0,1	70	12	12	12	12	12	12	12	12	12
0,5	0,1	90	12	12	12	12	12	12	12	12	12
0,5	0,3	10	12	12	12	12	12	12	12	12	12
0,5	0,3	30	12	12	12	12	12	12	12	12	12
0,5	0,3	50	12	12	12	12	12	12	12	12	12
0,5	0,3	70	12	12	12	12	12	12	12	12	12
0,5	0,3	90	12	12	12	12	12	12	12	12	12
0,5	0,5	10	12	12	12	12	12	12	12	12	12
0,5	0,5	30	12	12	12	12	12	12	12	12	12
0,5	0,5	50	12	12	12	12	12	12	12	12	12
0,5	0,5	70	12	12	12	13	12	13	12	12	12
0,5	0,5	90	12	12	12	13	12	13	12	12	12
0,5	0,7	10	12	12	12	14	14	14	12	12	12
0,5	0,7	30	12	12	12	14	14	14	12	12	12
0,5	0,7	50	12	12	12	14	14	14	12	12	12
0,5	0,7	70	12	12	12	14	14	14	13	13	13
0,5	0,7	90	12	12	12	14	14	14	13	13	13
0,5	0,9	10	12	12	12	14	14	14	12	12	12
0,5	0,9	30	12	12	12	14	14	14	12	12	12
0,5	0,9	50	12	12	12	14	14	14	12	12	12
0,5	0,9	70	12	12	12	14	14	14	12	12	12
0,5	0,9	90	12	12	12	14	14	14	12	12	12
0,7	0,1	10	12	12	12	12	12	12	12	12	12
0,7	0,1	30	12	12	12	12	12	12	12	12	12
0,7	0,1	50	12	12	12	12	12	12	12	12	12
0,7	0,1	70	12	12	12	12	12	12	12	12	12
0,7	0,1	90	12	12	12	12	12	12	12	12	12
0,7	0,3	10	12	12	12	12	12	12	12	12	12
0,7	0,3	30	12	12	12	12	12	12	12	12	12
0,7	0,3	50	12	12	12	12	12	12	12	12	12
0,7	0,3	70	12	12	12	12	12	12	12	12	12
0,7	0,3	90	12	12	12	12	12	12	12	12	12
0,7	0,5	10	12	12	12	12	12	12	12	12	12
0,7	0,5	30	12	12	12	12	12	12	12	12	12

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,7	0,5	50	12	12	12	12	12	12	12	12	12
0,7	0,5	70	12	12	12	12	12	12	12	12	12
0,7	0,5	90	12	12	12	12	12	12	12	12	12
0,7	0,7	10	12	12	12	12	12	12	12	12	12
0,7	0,7	30	12	12	12	12	12	12	12	12	12
0,7	0,7	50	12	12	12	12	12	12	12	12	12
0,7	0,7	70	12	12	12	12	12	12	13	13	13
0,7	0,7	90	12	12	12	12	12	12	13	13	13
0,7	0,9	10	12	12	12	14	14	14	12	12	12
0,7	0,9	30	12	12	12	14	14	14	12	12	12
0,7	0,9	50	12	12	12	14	14	14	12	12	12
0,7	0,9	70	12	12	12	14	14	14	12	12	12
0,7	0,9	90	12	12	12	14	14	14	12	12	12
0,9	0,1	10	12	12	12	12	12	12	12	12	12
0,9	0,1	30	12	12	12	12	12	12	12	12	12
0,9	0,1	50	12	12	12	12	12	12	12	12	12
0,9	0,1	70	12	12	12	12	12	12	12	12	12
0,9	0,1	90	12	12	12	12	12	12	12	12	12
0,9	0,3	10	12	12	12	12	12	12	12	12	12
0,9	0,3	30	12	12	12	12	12	12	12	12	12
0,9	0,3	50	12	12	12	12	12	12	12	12	12
0,9	0,3	70	12	12	12	12	12	12	12	12	12
0,9	0,3	90	12	12	12	12	12	12	12	12	12
0,9	0,5	10	12	12	12	12	12	12	12	12	12
0,9	0,5	30	12	12	12	12	12	12	12	12	12
0,9	0,5	50	12	12	12	12	12	12	12	12	12
0,9	0,5	70	12	12	12	12	12	12	12	12	12
0,9	0,5	90	12	12	12	12	12	12	12	12	12
0,9	0,7	10	12	12	12	12	12	12	12	12	12
0,9	0,7	30	12	12	12	12	12	12	12	12	12
0,9	0,7	50	12	12	12	12	12	12	12	12	12
0,9	0,7	70	12	12	12	12	12	12	12	12	12
0,9	0,7	90	12	12	12	12	12	12	12	12	12
0,9	0,9	10	12	12	12	14	14	14	12	12	12
0,9	0,9	30	12	12	12	14	14	14	12	12	12
0,9	0,9	50	12	12	12	14	14	14	12	12	12
0,9	0,9	70	12	12	12	14	14	14	12	12	12
0,9	0,9	90	12	12	12	14	14	14	12	12	12

Priloga E: Učenje Q z zakasnjeno nagrado in sledmi, Eksperiment b – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,1	0,1	10	9	9	9	10	10	10	9	9	9
0,1	0,1	30	9	9	9	11	10	10	9	9	9
0,1	0,1	50	9	9	9	9	9	9	9	9	9
0,1	0,1	70	9	9	9	11	9	9	9	9	9
0,1	0,1	90	9	9	9	11	9	9	9	9	9
0,1	0,3	10	9	9	9	10	10	10	9	9	9
0,1	0,3	30	9	9	9	11	10	9	9	9	9
0,1	0,3	50	9	9	9	9	9	9	9	9	10
0,1	0,3	70	9	9	9	9	9	9	9	9	9
0,1	0,3	90	9	9	9	9	9	9	9	9	9
0,1	0,5	10	9	9	9	10	10	10	9	9	9
0,1	0,5	30	9	9	9	11	9	9	9	9	9
0,1	0,5	50	9	9	9	9	9	9	9	9	10
0,1	0,5	70	9	9	9	9	9	9	9	9	9
0,1	0,5	90	9	9	9	9	9	9	9	9	9
0,1	0,7	10	9	9	9	9	9	9	9	9	9
0,1	0,7	30	9	9	9	9	9	9	9	9	9
0,1	0,7	50	9	9	9	9	9	9	9	9	9
0,1	0,7	70	9	9	9	9	9	9	9	9	9
0,1	0,7	90	9	9	9	9	9	9	9	9	9
0,1	0,9	10	9	9	9	11	11	11	9	9	9
0,1	0,9	30	9	9	9	11	11	11	9	9	9
0,1	0,9	50	9	9	9	11	11	11	9	9	9
0,1	0,9	70	9	9	9	11	11	11	9	9	9
0,1	0,9	90	9	9	9	11	11	11	9	9	9
0,3	0,1	10	9	9	9	10	10	9	9	9	9
0,3	0,1	30	9	9	9	9	9	9	9	9	9
0,3	0,1	50	9	9	9	9	9	9	9	9	9
0,3	0,1	70	9	9	9	9	9	9	9	9	9
0,3	0,1	90	9	9	9	9	9	9	9	9	9
0,3	0,3	10	9	9	9	10	10	9	9	9	9
0,3	0,3	30	9	9	9	9	9	9	9	9	9
0,3	0,3	50	9	9	9	9	9	9	9	9	9
0,3	0,3	70	9	9	9	9	9	9	9	9	9
0,3	0,3	90	9	9	9	9	9	9	9	9	9
0,3	0,5	10	9	9	9	10	10	9	9	9	9
0,3	0,5	30	9	9	9	9	9	9	9	9	9
0,3	0,5	50	9	9	9	9	9	9	9	9	9
0,3	0,5	70	9	9	9	9	9	9	9	9	9
0,3	0,5	90	9	9	9	9	9	9	9	9	9
0,3	0,7	10	9	9	9	9	9	9	9	9	9
0,3	0,7	30	9	9	9	9	9	9	9	9	9

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,3	0,7	50	9	9	9	9	9	9	9	9	9
0,3	0,7	70	9	9	9	9	9	9	9	9	9
0,3	0,7	90	9	9	9	9	9	9	9	9	9
0,3	0,9	10	9	9	9	11	11	11	9	9	9
0,3	0,9	30	9	9	9	11	11	11	9	9	9
0,3	0,9	50	9	9	9	11	11	11	9	9	9
0,3	0,9	70	9	9	9	11	11	11	9	9	9
0,3	0,9	90	9	9	9	11	11	11	9	9	9
0,5	0,1	10	9	9	9	10	9	9	9	9	9
0,5	0,1	30	9	9	9	9	9	9	9	9	9
0,5	0,1	50	9	9	9	9	9	9	9	9	9
0,5	0,1	70	9	9	9	9	9	9	9	9	9
0,5	0,1	90	9	9	9	9	9	9	9	9	9
0,5	0,3	10	9	9	9	10	9	9	9	9	9
0,5	0,3	30	9	9	9	9	9	9	9	9	9
0,5	0,3	50	9	9	9	9	9	9	9	9	9
0,5	0,3	70	9	9	9	9	9	9	9	9	9
0,5	0,3	90	9	9	9	9	9	9	9	9	9
0,5	0,5	10	9	9	9	10	9	9	9	9	9
0,5	0,5	30	9	9	9	9	9	9	9	9	9
0,5	0,5	50	9	9	9	9	9	9	9	9	9
0,5	0,5	70	10	10	9	9	9	9	9	9	9
0,5	0,5	90	10	10	9	9	9	9	9	9	9
0,5	0,7	10	9	9	9	9	9	9	9	9	9
0,5	0,7	30	9	9	9	9	9	9	9	9	9
0,5	0,7	50	9	9	9	9	9	9	9	9	9
0,5	0,7	70	9	9	9	9	9	9	9	9	9
0,5	0,7	90	9	9	9	9	9	9	9	9	9
0,5	0,9	10	9	9	9	11	11	11	9	9	9
0,5	0,9	30	9	9	9	11	11	11	9	9	9
0,5	0,9	50	9	9	9	11	11	11	9	9	9
0,5	0,9	70	9	9	9	11	11	11	9	9	9
0,5	0,9	90	9	9	9	11	11	11	9	9	9
0,7	0,1	10	9	9	9	9	9	9	9	9	9
0,7	0,1	30	9	9	9	9	9	9	9	9	9
0,7	0,1	50	9	9	9	9	9	9	9	9	9
0,7	0,1	70	9	9	9	9	9	9	9	9	9
0,7	0,1	90	9	9	9	9	9	9	9	9	9
0,7	0,3	10	9	9	9	9	9	9	9	9	9
0,7	0,3	30	9	9	9	9	9	9	9	9	9
0,7	0,3	50	9	9	9	9	9	9	9	9	9
0,7	0,3	70	9	9	9	9	9	9	9	9	9
0,7	0,3	90	9	9	9	9	9	9	9	9	9
0,7	0,5	10	9	9	9	9	9	9	9	9	9
0,7	0,5	30	9	9	9	9	9	9	9	9	9

Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,7	0,5	50	9	9	9	9	9	9	9	9	9
0,7	0,5	70	9	9	9	9	9	9	9	9	9
0,7	0,5	90	9	9	9	9	9	9	9	9	9
0,7	0,7	10	9	9	9	9	9	9	9	9	9
0,7	0,7	30	9	9	9	9	9	9	9	9	9
0,7	0,7	50	9	9	9	9	9	9	9	9	9
0,7	0,7	70	9	9	9	9	9	9	9	9	9
0,7	0,7	90	9	9	9	9	9	9	9	9	9
0,7	0,9	10	9	9	9	11	11	11	9	9	9
0,7	0,9	30	9	9	9	11	11	11	9	9	9
0,7	0,9	50	9	9	9	11	11	11	9	9	9
0,7	0,9	70	9	9	9	11	11	11	9	9	9
0,7	0,9	90	9	9	9	11	11	11	9	9	9
0,9	0,1	10	9	9	9	9	9	9	9	9	9
0,9	0,1	30	9	9	9	9	9	9	9	9	9
0,9	0,1	50	9	9	9	9	9	9	9	9	9
0,9	0,1	70	9	9	9	9	9	9	9	9	9
0,9	0,1	90	9	9	9	9	9	9	9	9	9
0,9	0,3	10	9	9	9	9	9	9	9	9	9
0,9	0,3	30	9	9	9	9	9	9	9	9	9
0,9	0,3	50	9	9	9	9	9	9	9	9	9
0,9	0,3	70	9	9	9	9	9	9	9	9	9
0,9	0,3	90	9	9	9	9	9	9	9	9	9
0,9	0,5	10	9	9	9	9	9	9	9	9	9
0,9	0,5	30	9	9	9	9	9	9	9	9	9
0,9	0,5	50	9	9	9	9	9	9	9	9	9
0,9	0,5	70	9	9	9	9	9	9	9	9	9
0,9	0,5	90	9	9	9	9	9	9	9	9	9
0,9	0,7	10	9	9	9	9	9	9	9	9	9
0,9	0,7	30	9	9	9	9	9	9	9	9	9
0,9	0,7	50	9	9	9	9	9	9	9	9	9
0,9	0,7	70	9	9	9	9	9	9	9	9	9
0,9	0,7	90	9	9	9	9	9	9	9	9	9
0,9	0,9	10	9	9	9	11	11	11	9	9	9
0,9	0,9	30	9	9	9	11	11	11	9	9	9
0,9	0,9	50	9	9	9	11	11	11	9	9	9
0,9	0,9	70	9	9	9	11	11	11	9	9	9
0,9	0,9	90	9	9	9	11	11	11	9	9	9

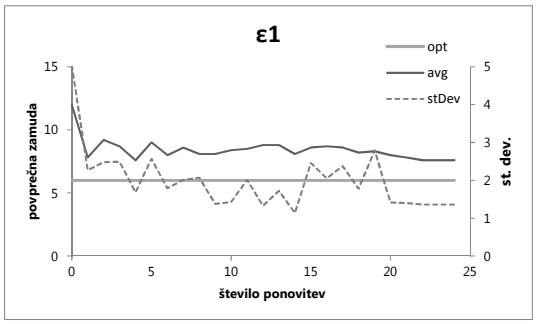
Priloga F: Učenje Q z zakasnjeno nagrado in sledmi, Eksperiment c – rezultati učenja za vse kombinacije parametrov, za različno število ponovitev učenja in za tri scenarije zamud

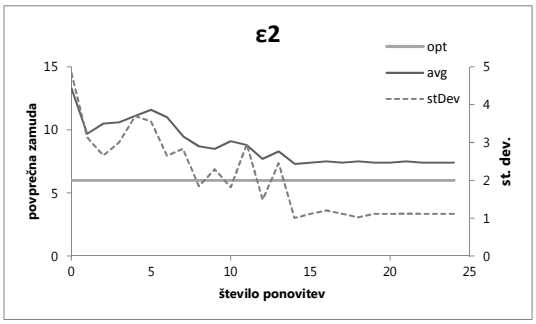
Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,1	0,1	10	13	13	13	6	6	6	8	8	8
0,1	0,1	30	12	12	12	6	6	6	8	8	8
0,1	0,1	50	12	12	13	6	6	6	8	8	8
0,1	0,1	70	12	13	13	6	6	6	8	8	8
0,1	0,1	90	12	13	13	6	6	6	8	8	8
0,1	0,3	10	13	13	13	6	6	6	8	8	8
0,1	0,3	30	12	12	12	6	6	6	8	8	8
0,1	0,3	50	12	12	12	6	6	6	8	8	8
0,1	0,3	70	13	12	12	6	6	6	8	8	8
0,1	0,3	90	13	12	12	6	6	6	8	8	8
0,1	0,5	10	12	12	12	6	6	6	8	8	8
0,1	0,5	30	12	12	12	6	6	6	8	8	8
0,1	0,5	50	12	12	12	6	6	6	8	8	8
0,1	0,5	70	12	12	12	6	6	6	8	8	8
0,1	0,5	90	12	12	12	6	6	6	8	8	8
0,1	0,7	10	12	12	12	6	6	6	8	8	8
0,1	0,7	30	12	12	12	6	6	6	8	8	8
0,1	0,7	50	12	12	12	6	6	6	8	8	8
0,1	0,7	70	13	13	13	6	6	6	8	8	8
0,1	0,7	90	13	13	13	6	6	6	8	8	8
0,1	0,9	10	12	12	12	7	7	7	8	8	8
0,1	0,9	30	12	12	12	6	6	6	9	9	9
0,1	0,9	50	12	12	12	6	6	6	8	8	8
0,1	0,9	70	13	13	13	6	6	6	8	8	8
0,1	0,9	90	13	13	13	6	6	6	8	8	8
0,3	0,1	10	13	12	12	6	6	6	8	8	8
0,3	0,1	30	12	12	12	6	6	6	8	8	8
0,3	0,1	50	13	12	12	6	6	6	8	8	8
0,3	0,1	70	12	12	12	7	6	6	8	8	8
0,3	0,1	90	12	12	12	7	6	6	8	8	8
0,3	0,3	10	13	12	12	6	6	6	8	8	8
0,3	0,3	30	12	12	12	6	6	6	8	8	8
0,3	0,3	50	12	12	12	6	6	6	8	8	8
0,3	0,3	70	12	12	12	6	6	6	8	8	8
0,3	0,3	90	12	12	12	6	6	6	8	8	8
0,3	0,5	10	12	12	12	6	6	6	8	8	8
0,3	0,5	30	12	12	12	6	6	6	8	8	8
0,3	0,5	50	12	12	12	6	6	6	8	8	8
0,3	0,5	70	12	12	12	6	6	6	8	8	8
0,3	0,5	90	12	12	12	6	6	6	8	8	8
0,3	0,7	10	12	12	12	6	6	6	8	8	8
0,3	0,7	30	13	13	13	6	6	6	8	8	8

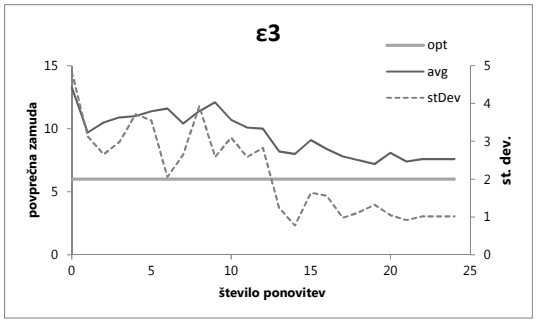
Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,3	0,7	50	12	12	12	6	6	6	8	8	8
0,3	0,7	70	13	13	13	6	6	6	8	8	8
0,3	0,7	90	13	13	13	6	6	6	8	8	8
0,3	0,9	10	12	12	12	7	7	7	8	8	8
0,3	0,9	30	12	12	12	6	6	6	9	9	9
0,3	0,9	50	12	12	12	6	6	6	8	8	8
0,3	0,9	70	13	13	13	6	6	6	8	8	8
0,3	0,9	90	13	13	13	6	6	6	8	8	8
0,5	0,1	10	12	12	12	6	6	6	8	8	8
0,5	0,1	30	12	12	12	6	6	6	8	8	8
0,5	0,1	50	12	12	12	6	6	6	8	8	8
0,5	0,1	70	12	13	13	6	6	6	8	8	8
0,5	0,1	90	12	13	13	6	6	6	8	8	8
0,5	0,3	10	12	12	12	6	6	6	8	8	8
0,5	0,3	30	12	12	12	6	6	6	8	8	8
0,5	0,3	50	12	12	12	6	6	6	8	8	8
0,5	0,3	70	12	12	12	6	6	6	8	8	8
0,5	0,3	90	12	12	12	6	6	6	8	8	8
0,5	0,5	10	12	12	12	6	6	6	8	8	8
0,5	0,5	30	12	12	12	6	6	6	8	8	8
0,5	0,5	50	12	12	12	6	6	6	8	8	8
0,5	0,5	70	13	12	13	6	6	6	8	8	8
0,5	0,5	90	13	12	13	6	6	6	8	8	8
0,5	0,7	10	12	12	12	6	6	6	8	8	8
0,5	0,7	30	12	12	12	6	6	6	8	8	8
0,5	0,7	50	12	12	12	6	6	6	8	8	8
0,5	0,7	70	13	14	14	6	6	6	8	8	8
0,5	0,7	90	13	14	14	6	6	6	8	8	8
0,5	0,9	10	12	12	12	7	7	7	8	8	8
0,5	0,9	30	12	12	12	6	6	6	9	9	9
0,5	0,9	50	12	12	12	6	6	6	8	8	8
0,5	0,9	70	13	13	13	6	6	6	8	8	8
0,5	0,9	90	13	13	13	6	6	6	8	8	8
0,7	0,1	10	12	12	12	6	6	6	8	8	8
0,7	0,1	30	12	12	12	6	6	6	8	8	8
0,7	0,1	50	12	12	12	6	6	6	8	8	8
0,7	0,1	70	12	12	12	6	6	6	8	8	8
0,7	0,1	90	12	12	12	6	6	6	8	8	8
0,7	0,3	10	12	12	12	6	6	6	8	8	8
0,7	0,3	30	12	12	12	6	6	6	8	8	8
0,7	0,3	50	12	12	12	6	6	6	8	8	8
0,7	0,3	70	12	13	13	6	6	6	8	8	8
0,7	0,3	90	12	13	13	6	6	6	8	8	8
0,7	0,5	10	12	12	12	6	6	6	8	8	8
0,7	0,5	30	12	12	12	6	6	6	8	8	8

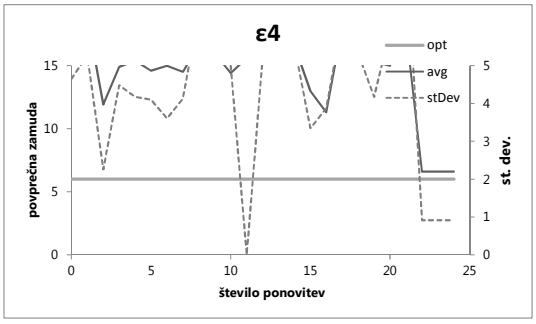
Parameter			Scenarij 1			Scenarij 2			Scenarij 3		
			Število ponovitev			Število ponovitev			Število ponovitev		
α	γ	ε	50	100	150	50	100	150	50	100	150
0,7	0,5	50	12	12	12	6	6	6	8	8	8
0,7	0,5	70	12	12	12	7	6	6	8	8	8
0,7	0,5	90	12	12	12	7	6	6	8	8	8
0,7	0,7	10	14	14	14	6	6	6	8	8	8
0,7	0,7	30	12	12	12	6	6	6	8	8	8
0,7	0,7	50	12	13	13	6	6	6	8	8	8
0,7	0,7	70	13	13	13	6	6	6	8	8	8
0,7	0,7	90	13	13	13	6	6	6	8	8	8
0,7	0,9	10	12	12	12	7	7	7	8	8	8
0,7	0,9	30	12	12	12	6	6	6	9	9	9
0,7	0,9	50	12	12	12	6	6	6	8	8	8
0,7	0,9	70	13	13	13	6	6	6	8	8	8
0,7	0,9	90	13	13	13	6	6	6	8	8	8
0,9	0,1	10	12	12	12	6	6	6	8	8	8
0,9	0,1	30	12	12	12	6	6	6	8	8	8
0,9	0,1	50	12	12	12	6	6	6	8	8	8
0,9	0,1	70	12	12	12	6	6	6	8	8	8
0,9	0,1	90	12	12	12	6	6	6	8	8	8
0,9	0,3	10	12	12	12	6	6	6	8	8	8
0,9	0,3	30	12	12	12	6	6	6	8	8	8
0,9	0,3	50	12	12	12	6	6	6	8	8	8
0,9	0,3	70	13	12	12	6	6	6	8	8	8
0,9	0,3	90	13	12	12	6	6	6	8	8	8
0,9	0,5	10	12	12	12	6	6	6	8	8	8
0,9	0,5	30	12	12	12	6	6	6	8	8	8
0,9	0,5	50	12	12	12	6	6	6	8	8	8
0,9	0,5	70	12	12	12	6	6	6	9	8	8
0,9	0,5	90	12	12	12	6	6	6	9	8	8
0,9	0,7	10	14	14	14	6	6	6	8	8	8
0,9	0,7	30	12	12	12	6	6	6	8	8	8
0,9	0,7	50	13	13	13	6	6	6	8	8	8
0,9	0,7	70	13	13	12	6	6	6	8	8	8
0,9	0,7	90	13	13	12	6	6	6	8	8	8
0,9	0,9	10	14	14	14	6	7	7	8	8	8
0,9	0,9	30	13	13	13	6	6	7	9	9	9
0,9	0,9	50	12	12	12	6	6	6	8	8	8
0,9	0,9	70	13	14	13	6	6	6	8	8	8
0,9	0,9	90	13	14	13	6	6	6	8	8	8

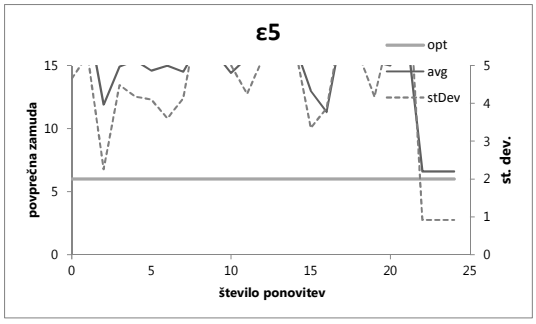
Priloga G: Parametrična študija – Krivulje učenja za Scenarij zamud 1

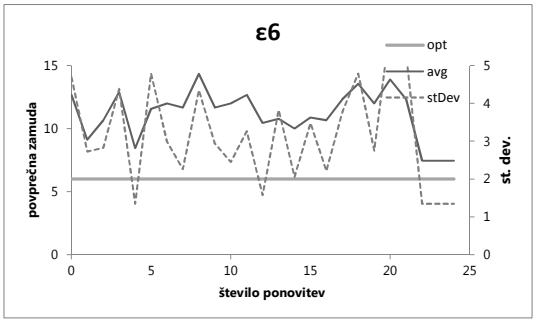


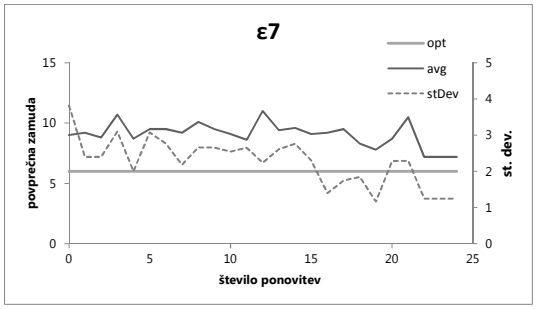


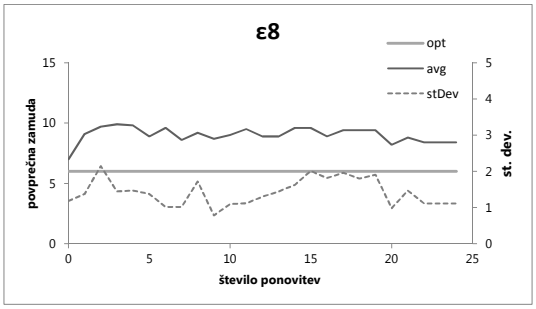








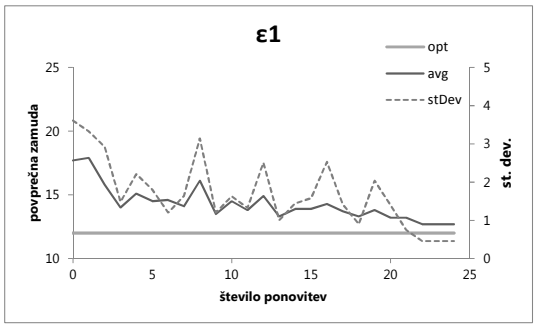


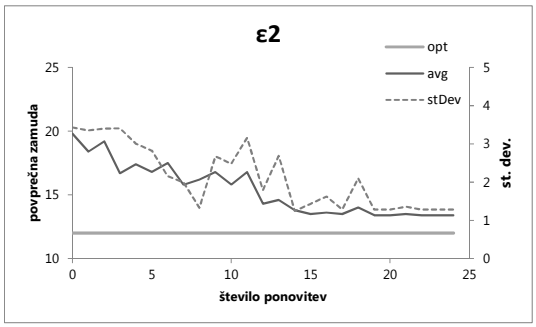


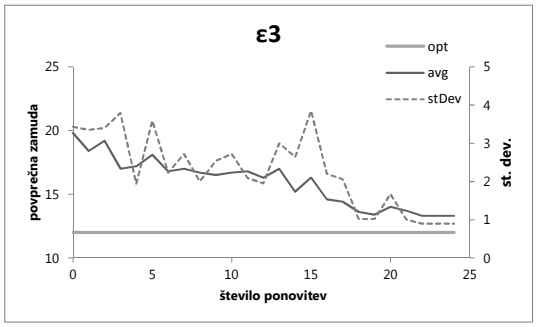
Krivulje učenja in krivulje standardne deviacije pri $\alpha = 0{,}9$, $\gamma = 0{,}1$ ter

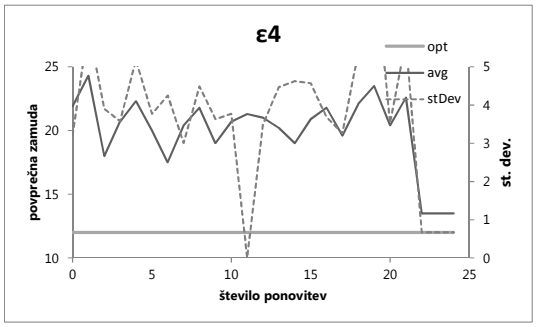
$$\varepsilon 1 = x^{-1},$$
$$\varepsilon 2 = \frac{0{,}5}{1+e^{(10*(x-0.4*25)/25))}},$$
$$\varepsilon 3 = \frac{0{,}5}{1+e^{(10*(x-0.4*35)/35))}},$$
$$\varepsilon 4 = 0{,}9,$$
$$\varepsilon 5 = 0{,}7,$$
$$\varepsilon 6 = 0{,}5,$$
$$\varepsilon 7 = 0{,}3,$$
$$\varepsilon 8 = 0{,}1$$

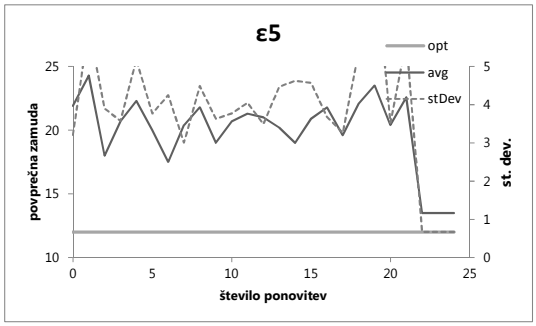
Priloga H: Parametrična študija – Krivulje učenja za Scenarij zamud 2

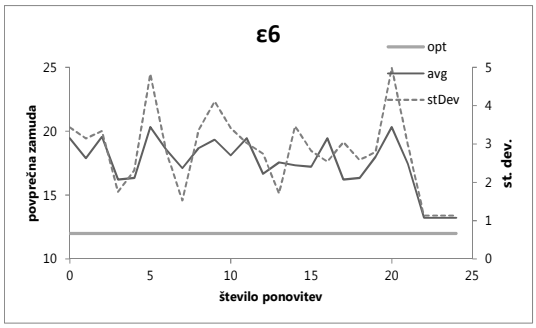


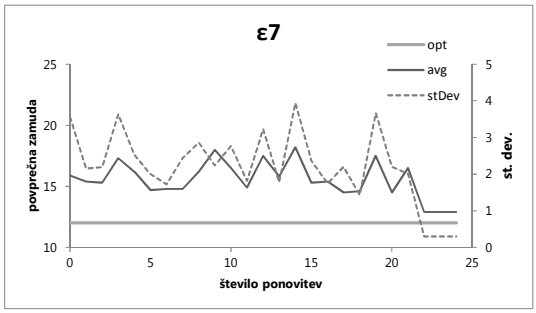


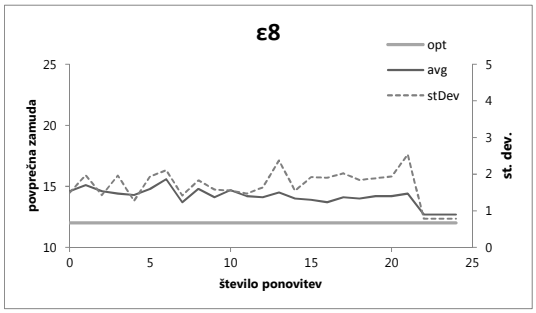








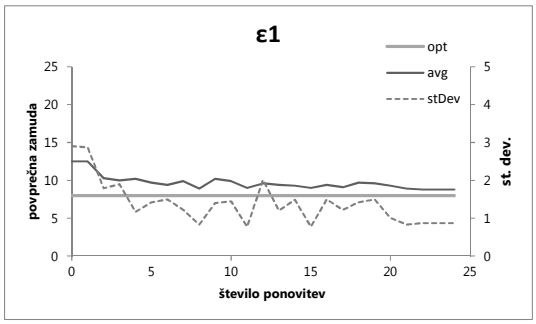


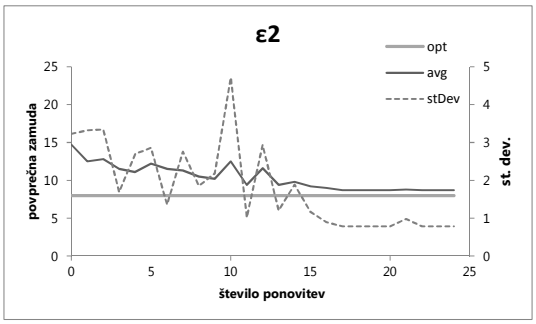


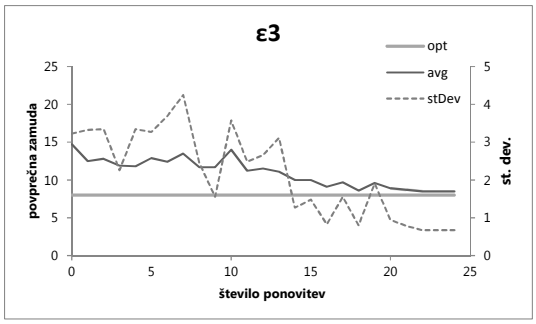
Krivulje učenja in krivulje standardne deviacije pri $\alpha = 0{,}9$, $\gamma = 0{,}1$ ter

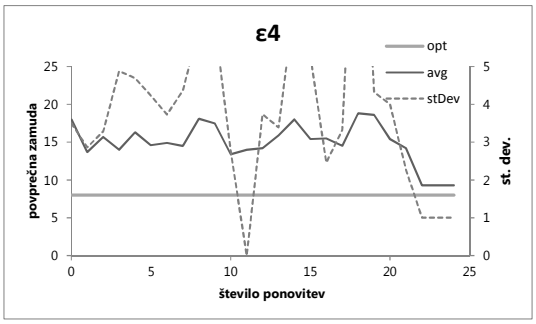
$$\varepsilon 1 = x^{-1},$$
$$\varepsilon 2 = \frac{0{,}5}{1+\mathrm{e}^{(10*(\mathrm{x}-0.4*25)/25))}},$$
$$\varepsilon 3 = \frac{0{,}5}{1+\mathrm{e}^{(10*(\mathrm{x}-0.4*35)/35))}},$$
$$\varepsilon 4 = 0{,}9,$$
$$\varepsilon 5 = 0{,}7,$$
$$\varepsilon 6 = 0{,}5,$$
$$\varepsilon 7 = 0{,}3,$$
$$\varepsilon 8 = 0{,}1$$

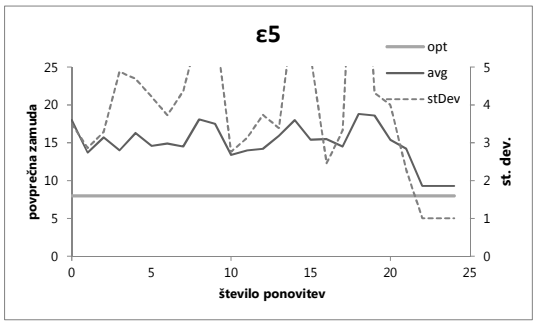
Priloga I: Parametrična študija – Krivulje učenja za Scenarij zamud 3

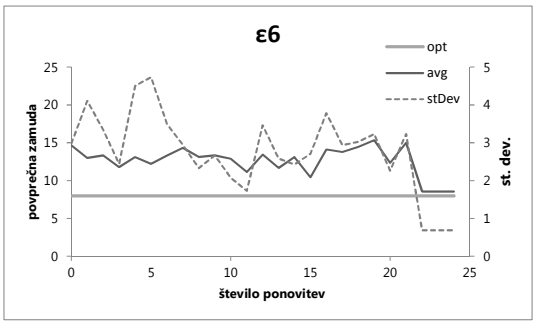


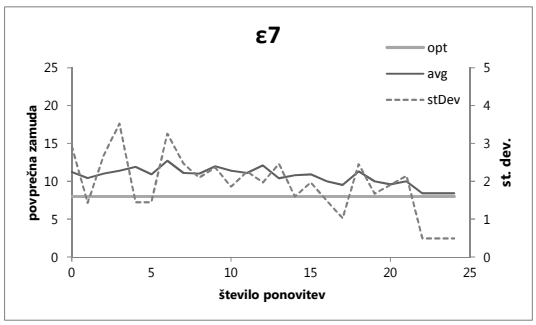


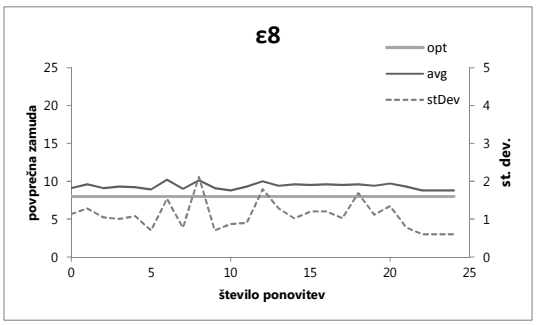








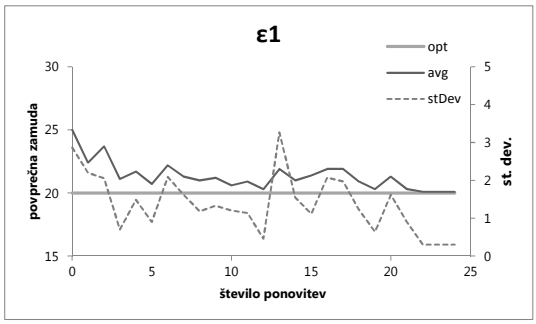


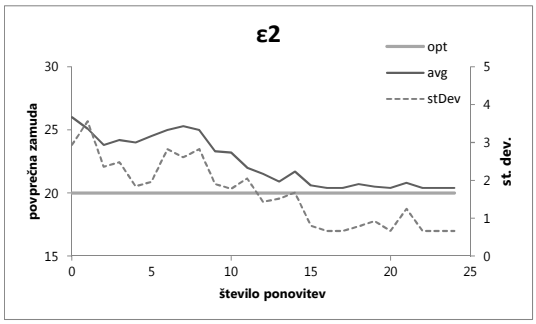


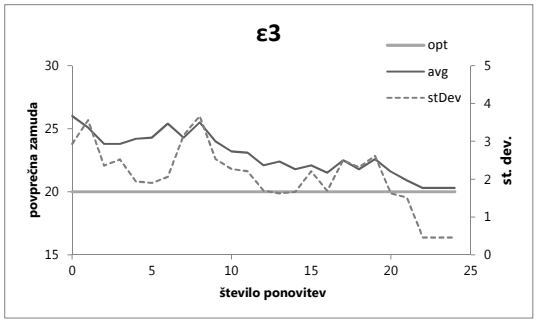
Krivulje učenja in krivulje standardne deviacije pri $\alpha = 0{,}9$, $\gamma = 0{,}1$ ter

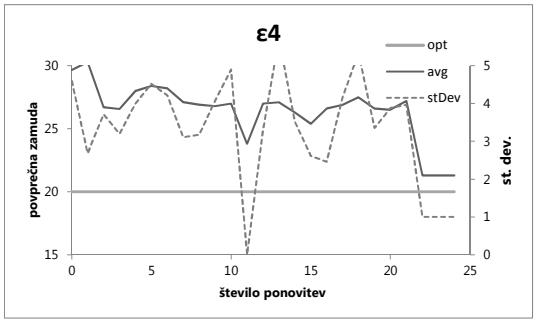
$$\varepsilon 1 = x^{-1},$$
$$\varepsilon 2 = \frac{0{,}5}{1+e^{(10*(x-0.4*25)/25))}},$$
$$\varepsilon 3 = \frac{0{,}5}{1+e^{(10*(x-0.4*35)/35))}},$$
$$\varepsilon 4 = 0{,}9,$$
$$\varepsilon 5 = 0{,}7,$$
$$\varepsilon 6 = 0{,}5,$$
$$\varepsilon 7 = 0{,}3,$$
$$\varepsilon 8 = 0{,}1$$

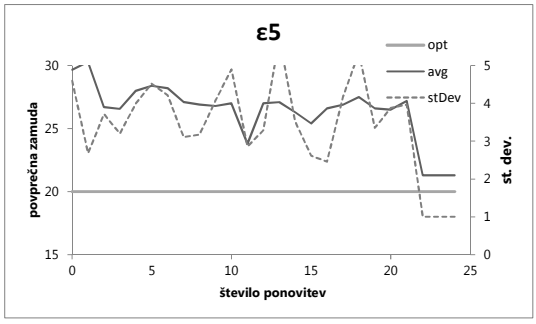
Priloga J: Parametrična študija – Krivulje učenja za Scenarij zamud 4

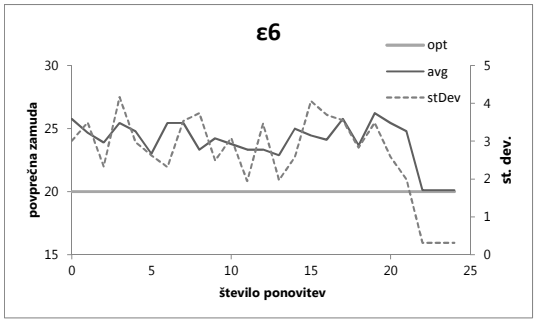


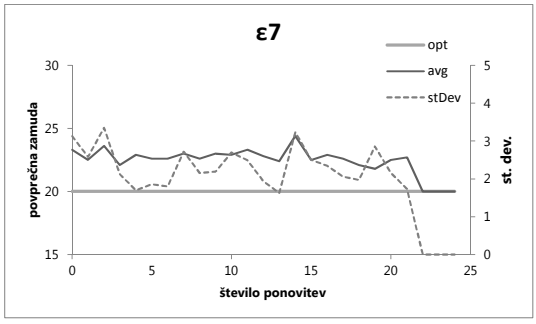


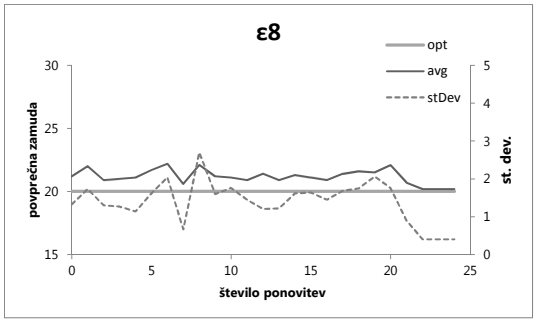












Krivulje učenja in krivulje standardne deviacije pri $\alpha = 0{,}9$, $\gamma = 0{,}1$ ter

$$\varepsilon 1 = x^{-1},$$
$$\varepsilon 2 = \frac{0{,}5}{1+\mathrm{e}^{(10*(\mathrm{x}-0.4*25)/25))}},$$
$$\varepsilon 3 = \frac{0{,}5}{1+\mathrm{e}^{(10*(\mathrm{x}-0.4*35)/35))}},$$
$$\varepsilon 4 = 0{,}9,$$
$$\varepsilon 5 = 0{,}7,$$
$$\varepsilon 6 = 0{,}5,$$
$$\varepsilon 7 = 0{,}3,$$
$$\varepsilon 8 = 0{,}1$$