Artificial Intelligence in Train Scheduling Problems

Kieran Molloy

Manchester Metropolitan University
May 29, 2020

Table of Contents

Project Motivation

Optimisation of train networks can provide return on all optimality factors; profit, timeliness or robustness. Train networks provide a unique problem where infrastructure expansions are complicated, expensive and disruptive. This leads to largely scheduling based problems, which this project is based upon.

Brief Introduction

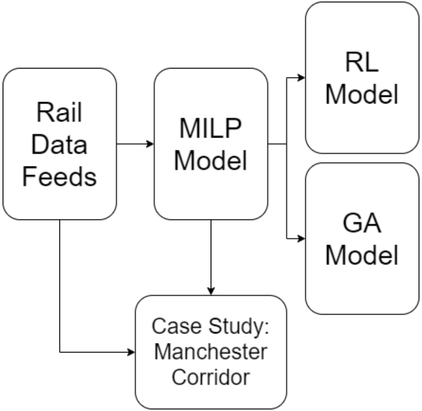


Figure: Project Outline

Brief Introduction

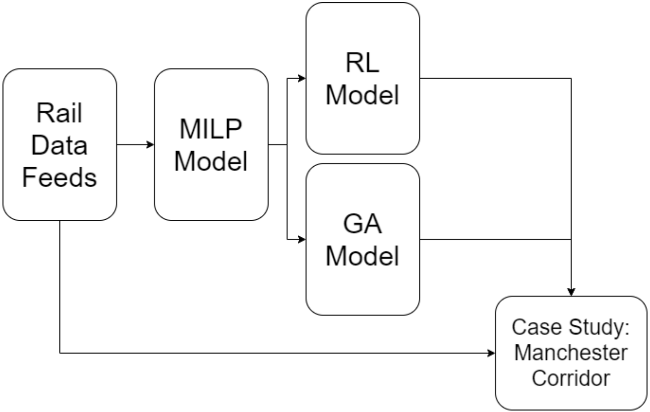


Figure: Intended Project Outline

Table of Contents

RDF - Implementation

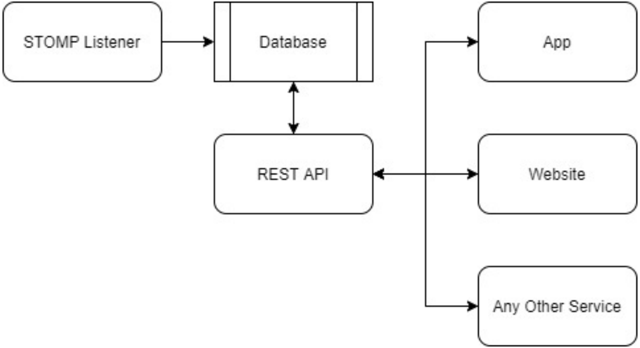


Figure: Rail Data Feeds Program Flow

RDF - Purpose

- View the Network Live
- Save Network Statistics
- Create Real World Test Sets

RDF - Outcomes

- Network Rail Listener
- Huxley Connector
- Network Rail Historic Database
- REST API
- Website

Table of Contents

MILP Model - TSP

Objective Function

$$\sum_{n=1}^{N} A^s_{n,M} - D^s_{n,0}$$

Example Constraints:

$$D^r_{n,m} \leq A^r_{n+1,m} - H$$

$$A^r_{n,m+1} \geq D^r_{n,m} + E_m$$

MILP Model - TRSP

Objective Function

$$\sum_{n=1}^{N} A^{r}_{n,M} - A^{o}_{n,M}$$

Example Constraints:

$$D^{r}_{n,m} = D^{0}_{n,m}$$

$$\delta_D > 1, \delta_t < D^{\delta_D - 1}_{\delta_n} + E_{\delta_D - 1} \rightarrow ...$$

$$A^{r}_{\delta_n,\delta_D} \geq D^{0}_{\delta_n,\delta_D - 1} + \delta_d + E_{\delta_d - 1}$$

MILP Model - TRSP (Passenger Weighted)

Objective Function

$$\sum_{n=1}^{N} P^{c}_{n,M} A^{r}_{n,M}$$

Example Constraints:

$$C - P^{l}_{n,m-1} < 100 \rightarrow P^{n,m}_{c} \leq 20\lambda_{n,m}$$
$$C - P^{l}_{n,m-1} \geq 100 \rightarrow P^{n,m}_{c} \leq 50\lambda_{n,m}$$
$$P^{c}_{n,m} \leq D + n, m^{r} - 50A^{r}_{n,m}$$

MILP Model - Outcomes

- MiniZinc TRSP Standard Model
- Series of Associated FlatZinc Data Files
- Minizinc TRSP Passenger Weighted Model
- Series of Assocaited FlatZinc Data Files

Table of Contents

GA - General Algorithm

```
P ← generatePopulation[100]                ▷ Get 100 random variables
while endCondition = false do
    P ← selection(P)
    P ← crossover(P)
    P ← mutation(P)
    B ← fitness(P)
    if B == optimal then
        endCondition ← true
return B,P
```

GA - TSP Framework

	Phenotype					
	toc	line	unit	origin	destination	intermediate
Bit Length	2	8	8	12	12	$12n$
Representations	4	256	256	4096	4096	$n(2^{12})$

GA - Basic Scenario

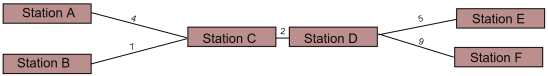
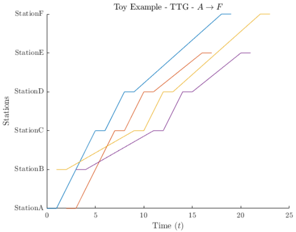
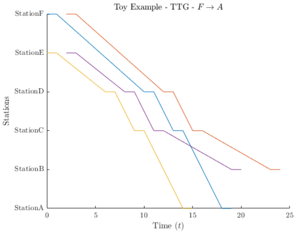


Figure: Toy Example Map

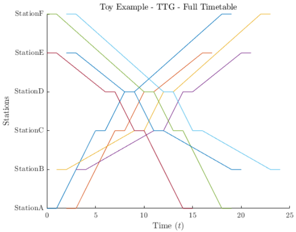
GA - Basic Scenario



(a) Stage 1



(b) Stage 2



(c) Stage 3

GA - Basic Scenario

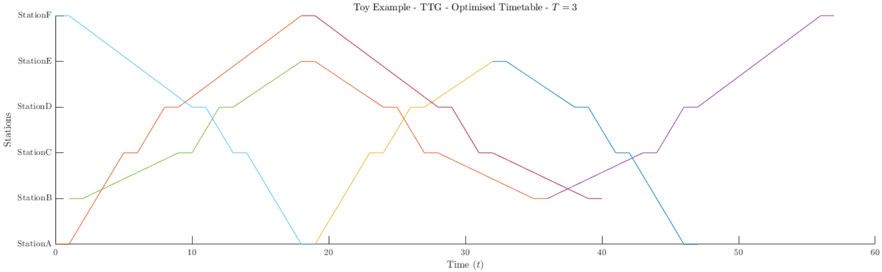


Figure: Toy Example Combined Stages

GA - Medium Scenario

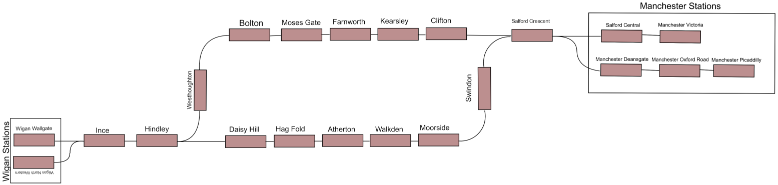


Figure: Small Example Map

Table of Contents

SBB Challenge

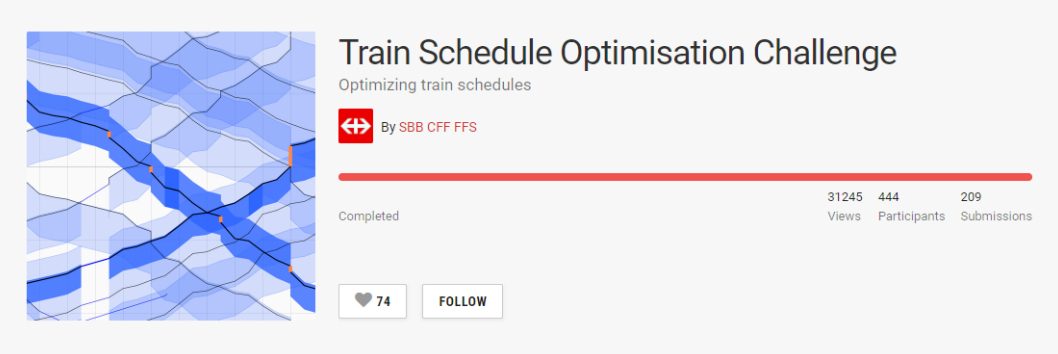


Figure: Crowd AI SBB Challenge

SBB Challenge

ID	Name	Trains	Routing	Difficulty		
01	dummy	4	V.Few	V.Simple		
02	a_little_less_dummy	58	Few	Simple		
03	FWA_0.125	143	Few	Simple		
04	V1.02_FWA_without_obstruction	148	Few	Medium		
05	V1.02_FWA_with_obstruction	149	Medium	Medium*		
06	V1.20_FWA	365	High	V.Hard		
07	V1.22_FWA	467	High	V.Hard		
08	V1.30_FWA	133	V.High	V.Hard		
09	ZUEZGCH_06001200			287	VV.High	VV.Hard

Table: Description of Problem Scenarios

RL - Delayed Q-Learning

Modelling a PACMDP
Update Function

$$Q_{t+1}(s,a) = \frac{1}{m}\sum_{i=1}^{m}(r_{k_i} + \gamma V_{k_i}(s_{k_i})) + \epsilon_1$$

Algorithm 1 Delayed Q-Learning

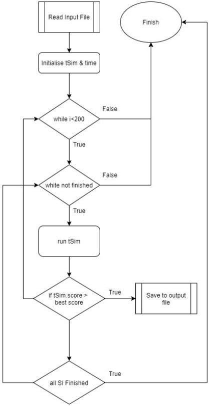
function DLQ(γ,S,A,M,ϵ_1)
    for all (s,a) **do**
        $Q(s,a) \leftarrow 1/(1-\gamma)$
        $U(s,a) \leftarrow 0)$
        $l(s,a) \leftarrow 0$
        $t(s,a) \leftarrow 0$
        $LEARN(s,a) \leftarrow TRUE$
    $t^* \leftarrow 0$
    while LEARN(s,a) =TRUE **do**
        $U(s,a) \leftarrow U(s,a) + r + \gamma \max_{a'} Q(s',a')$
        $l(s,a) \leftarrow l(s,a) + 1$
        if $l(s,a) = m$ **then**
            if $Q(s,a) - U(s,a)/m \geq 2\epsilon_1$ **then**
                $Q(s,a) - U(s,a)/m + \epsilon_1$
                $t^* \leftarrow t$
            else if $t(s,a) \geq t^*$ **then**
                LEARN$(s,a) \leftarrow FALSE$
            $t(s,a) \leftarrow t$
            $U(s,a) \leftarrow 0$
            $l(s,a) \leftarrow 0$
        if $t(s,a) < t^*$ **then** LEARN$(s,a) \leftarrow TRUE$

Figure: Delayed Q-Learning

RL SBB - Objective Function

$$\frac{1}{60} \times \left[\sum_{\mathcal{S},\mathcal{R},\mathcal{RS}} \mathit{weightIn}_{rs} \times \max\left(0, (t^{entry}_{s,r,rs} - \mathit{inLat}_{s,rs})\right) + \mathit{weightOut}_{rs} \times \max\left(0, (t^{exit}_{s,r,rs} - \mathit{outLat}_{s,rs})\right)\right] + \sum_{\mathcal{S},\mathcal{R},\mathcal{RS}} p_{s,rs} \times \beta_{s,rs}$$

RL SBB - Discrete Event Simulation



Algorithm 3 tSim Control Functions

procedure RUN(self)
    $t \leftarrow$minTime
    while $t <$ (maxTime+10) **do**
        for e **in** event[t] **do**
            runNextTrain[e]
        $t \leftarrow t + 1$
    return calculateScore()
procedure RUNNEXTTRAIN(self,e)
    if $e = type(Node)$ **then** runNode()
    else if $e = type(Resource)$ **then del** e.resource
    else if $e = type(Station)$ **then** $nextEvent \leftarrow$ **new**$event[e]$

RL SBB - Reinforcement Learning Applied Algorithm

Algorithm 5 tSim.QTable Class Functions

```
procedure GETACTION(self,o,s)          ▷ Getting Action for any option or state
    if length(o) = 1 return o[0] then
    if rand¡ϵ then                     ▷ Uniform Random Package, for better convergence
        a ← rand(o)
    else
        m ← −∞                         ▷ Maximum Q-Value
        a ← null                       ▷ Action to be performed
        for all o do
            v ← qval(s, o[i])          ▷ qval is predicted reward of action o[i]
            if v > m then
                a ← c
                m ← v
    return a
procedure UPDATE(self,S_n,S_{n−1},A,r) ▷ Updating Q-Values
    p ← qval(S_{n−1}, A)               ▷ Previous Value
    m ← 0
    if qval(S_n) > 0 then              ▷ If exists, update max
        m ← max(qval(S_n))
    v ← (1 − α) ∗ p + α ∗ (r + γ ∗ m)  ▷ New Value
    return v                           ▷ Update Q-Table with new Q-Value
```

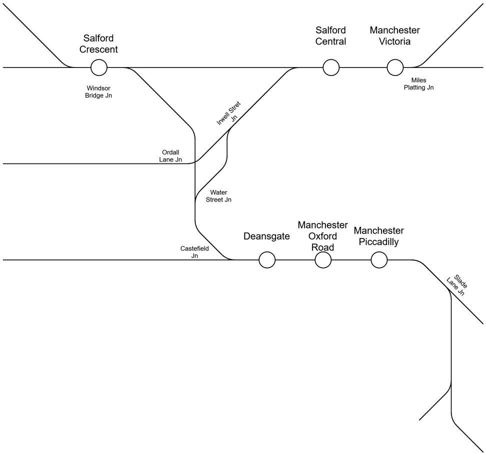
RL SBB - Results

	Solving System					
ID	01		02		03	
	ObjVal	Time	*ObjVal*	Time	*ObjVal*	Time
01	0	5.452	0	4.296	0	7.590
02	0.43	87	0.43	72.087	0.43	77.042
03	0.86	45	0.96	198.273	0.11	200.234
04	24.94	430	24.94	439.121	024.94	435.945
05	n/a	1800	n/a	1800	n/a	1800
06	207.12	1355	207.12	1174.690	207.12	1750.154
07	456.60	1800	442.33	1800	467.10	1800
08	153.23	1800	122.70	1800	233.6	1800
09	138.95	1800	38.25	1800	43.95	1800

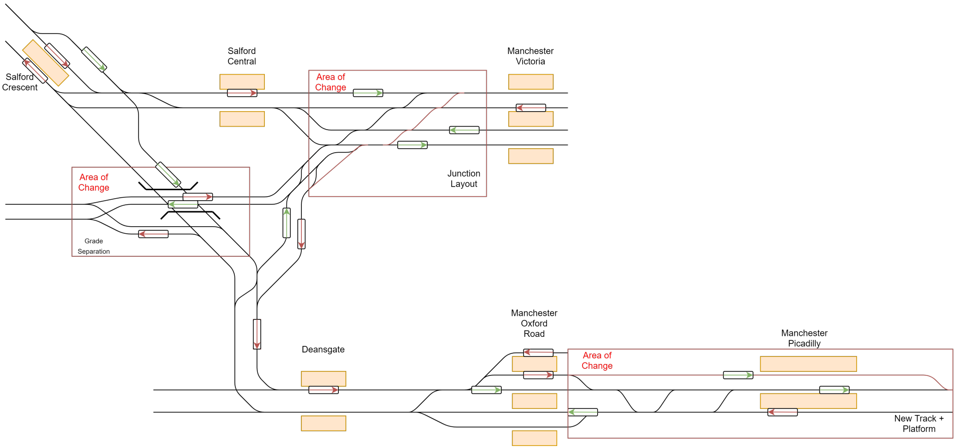
Table of Contents

Manchester Case Study - Method

- Junction TPH Analysis
- Station Timing Data
- Previous Case Studies



Manchester Case Study - Key Outcomes



Final Words - Questions

Brief Discussion and Q&A

(a) Full Research Paper

(b) Full Slides

(c) Full Poster