Sub Optimality-Runtime Trade off Analysis on a Single Track Train Scheduling Problem

*Mustafa Mehmet Bayar¹, Abdullah Süreyya Ersoy²

¹(Business Administration, Gazi University, Turkey)
²(Business Administration, Gazi University, Turkey)
Corresponding Author: Mustafa Mehmet Bayar

Abstract: A known complex problem: Single Track Train Scheduling Problem is formulated as a mixed-integer program in which, unmet demand was minimized as the objective. The model was solved for various runtime limitations. “Runtime-Distance From Optimal Solution” tuples was used to form an efficient frontier and via Data Envelopment Analysis the trade-off decision identified as the most productive scale size was suggested to the decision maker as the most efficient trade-off between runtime and distance from optimal solution.

Keywords: Single Track Train Scheduling Problem, Complexity, Data Envelopment Analysis, Most Productive Scale Size

I. INTRODUCTION

The Single Track Train Scheduling Problem is a special type of Public Transport Scheduling Problem, which is defined as a system that operates on a single line, generally to minimize the total waiting time [1].

The first optimization study on train timetabling was introduced by Amit and Goldfarb in 1971 [2]. The following studies focused on different objectives on the problem: trip time[3], total waiting time[4], demand[5], delay time[6] and reliability[7]. There are studies[8] that use multiple objectives. Some studies [9] focused on non-linear models. And, some [10] focused on heuristic methods. This paper has a single objective and that is to minimize unmet demand using a mixed integer programming model and implementing a second objective which is optimizing the solution time-sub optimality trade off.

The size of a decision problem is defined by the number of interacting variables, by the number and the structure of the constraints that limit the range of the values that those variables can take. The solution time of the single-track train scheduling problem does increase by a non-deterministic polynomial function as the size of the problem increase. In other words the problem is a non-deterministic polynomial time (NP) class decision problem. [11]

Due to the complexity of the problem previous studies often adopted meta heuristic methods[12]. This paper offers a dual process approach. First a mixed integer program is used to obtain different optimality gap results under different runtime limitations. And then, the runtime-optimality gap tuples are used to form a best practice frontier[13] and the trade off performance is analyzed via data envelopment analysis. Finally the most productive scale size is identified and declared as the best trade off decision.

This study consists of six sections considering the introduction as the first section. Second section is the methodology part which includes the three models used in this paper. In the third section results of the models are presented. Forth section is the conclusion part where the results are evaluated. Fifth section is the discussion section in which the limitations of this study was discussed and suggestions for further research in the area of decision making are given. The sixth and the last section is dedicated for the references that fueled this study.

II. METHODOLOGY

Single track train scheduling problem, here in this paper, aims to identify the optimal train schedule that minimize the total unmet demand [14]. The problem is formulated both as a linear programming model (here after to be referred as LP) and a mixed-integer programming model (here after to be referred as MIP).

Due to the complexity of the problem MIP optimal solution is not attainable within reasonable runtimes of the solver algorithm. So the MIP is solved under various runtime limitations, hence various sub-optimal results with various optimality gaps are obtained. To define the optimality gap of MIP results LP optimal solution is assumed to be the MIP optimal solution.

It is observed that the Optimality Gap/Runtime tuples constitute a trade-off decision: “How much time does a public transport system has to invest to achieve a higher service level with a schedule?”. To performs a trade off analysis Data Envelopment Analysis served as the right tool. Input Oriented CCRmodel[15] is solved to
identify the Most Productive Scale Size (MPSS) [16] within the best practice frontier [17] which is represented by the Optimality Gap-Runme tuples, the Decision Making Units (DMUs).

2.1. The Linear Programming Model
The LP model constitutes 284 parameters, 376 variables and 6 classes of constraints and a single objective.

2.1.1. Sets
Set of stations:
\[ D = \{ n; \forall n \in D \mid n \in [1, 2N], N = 3, \quad n \in \mathbb{Z}_+ \} \]
Set of train runs:
\[ S = \{ i; \forall i \in S \mid i \in [1, I], \quad I = 25, \quad i \in \mathbb{Z}_+ \} \]

2.1.2. Parameters
\( t_{i,n} \): travelling time of the \( i^{th} \) train from the \( n^{th} \) station to the \( n + 1^{th} \) station
\( k_i^1 \): incoming passenger rate at the \( n^{th} \) station (number of people per sec.)
\( k_i^2 \): ratio of on board passengers that would leave at the \( n^{th} \) station to total passengers on board
\( k_i \): rate of departing from the train (number of people per sec.)
\( \kappa \): maximum passenger capacity for each wagon

2.1.3. Hypothetical Data Set
\( t_{i,n} \) is assumed to be homogeneous and 120 for all train runs and between every station.
Incoming passenger rate for the stations are respectively \( 4/3 \cdot 2/3 \cdot 0 \cdot 8/3 \cdot 4/3 \cdot 0 \).
The ratio of on board passengers that would leave at the \( n^{th} \) station to total passengers on board is a random number between 0 and 1 for all stations.
Values of \( k_i \), \( k_i^1 \) and, \( \kappa \) are respectively 18, 27 and, 90.

2.1.4. Variables
\( t_{i,n}^x \): waiting time for passenger departures at \( i^{th} \) train run and at \( n^{th} \) station
\( t_{i,n}^y \): waiting time for passenger loading at \( i^{th} \) train run and at \( n^{th} \) station
\( x_{i,n} \): total waiting time at \( i^{th} \) train run and at \( n^{th} \) station
\( h \): average headway time
\( t_{i,n}^1 \): the time \( i^{th} \) train leaves \( n^{th} \) station
\( t_{i,n}^2 \): arrival time of the \( i^{th} \) train to the \( n^{th} \) station
\( Q_{i,n} \): number of customers waiting at the \( n^{th} \) station for the \( i^{th} \) train
\( Q_{i,n}^0 \): number of left over customers at the \( n^{th} \) station waiting for the next train
\( w_i \): number of wagons attached to the \( i^{th} \) train
\( y_i^1 \): binary variable for deciding to attach an additional wagon to the \( i^{th} \) train
\( y_i^2 \): binary variable for deciding to attach a second additional wagon to the \( i^{th} \) train
\( y_{i,n} \): number of passengers boarding the \( i^{th} \) train at the \( n^{th} \) station
\( P_{i,n} \): number of passengers on board at the \( i^{th} \) train between the \( n^{th} \) and the \( n + 1^{th} \) stations

2.1.5. Constraints
The linear programming model has six group of constraints.

2.1.5.1. Waiting Constraints
\[ x_{i,n} \geq t_{i,n}^x + t_{i,n}^y \quad \forall i \in S, \quad \forall n \in D \] (1)

2.1.5.2. Headway Constraint
\[ h = \frac{\left( \sum_{n=1}^{2N-1} \sum_{i=1}^{I} x_{i,n} \right) + \sum_{n=1}^{2N-1} \sum_{i=1}^{I} y_{i,n}^1 t_{i,n}^x}{I} \] (2)

2.1.5.3. Time Constraints:
for \( n < 2N \)
\[ t_{i,n}^1 = \begin{cases} i.h, & \text{if } n = 1 \\ (i - 1).h + \sum_{n=2}^{n} x_{i,n'} + \sum_{n=2}^{n-1} t_{i,n'}, & \text{if } n > 1 \end{cases} \quad \forall i \in S, \quad \forall n, n' \in D \] (3)

for \( n < 2N \)
\[ t_{i,n+1}^4 = t_{i,n}^3 + t_{i,n} \quad \forall i \in S, \quad \forall n \in D \] (4)

www.ijbmi.org 5 | Page
for $i > i'$
$t_{i,n}^4 \geq t_{i',n}^3 + 10$

for $i > i' + 3$
$t_{i,2N}^4 + x_{i',2N} + 120 \leq t_{i,1}^3$

$\forall i, i' \in S$ \hspace{1cm} (6)

2.1.5.4. Wagon Constraints

$$\frac{1}{2N} \sum_{n=1}^{2N} Q_{i,n}^0 \leq M y_i^1$$

$$\forall i \in S$$ \hspace{1cm} (7)

$$50 - \frac{1}{2N} \sum_{n=1}^{2N} Q_{i,n}^0 \leq M y_i^2$$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (8)

$w_i = y_i^1 + y_i^2 + 3$

$$\forall i \in S$$ \hspace{1cm} (9)

$y_i^1 \leq 1$

$$\forall i \in S$$ \hspace{1cm} (10)

$y_i^2 \leq 1$

$$\forall i \in S$$ \hspace{1cm} (11)

2.1.5.5. Conservation of Flow Constraints

$y_{LN} = 0$

$$\forall i \in S$$ \hspace{1cm} (12)

$y_{L2N} = 0$

$$\forall i \in S$$ \hspace{1cm} (13)

$y_{i,1} \leq w_{i-} - \delta_{i,1}$

$$\forall i \in S$$ \hspace{1cm} (14)

$n > 1 i' in$;

$y_{i,n} \leq w_{i, -} - \delta_{i,n} - p_{i,n-1}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (15)

$y_{i,n} \leq Q_{i,n}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (16)

$y_{i,n} \leq t_{i,n}^2 k_{y}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (17)

$\delta_{i,1} = 0$

$$\forall i \in S$$ \hspace{1cm} (18)

$\delta_{i,N+1} = 0$

$$\forall i \in S$$ \hspace{1cm} (19)

$p_{i,1} = y_{i,1} - \delta_{i,1}$

$$\forall i \in S$$ \hspace{1cm} (20)

$n > 1 i' in$;

$\delta_{i,n} \leq k_{\delta}^2 p_{i,n-1}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (21)

$\delta_{i,n} \leq t_{i,n}^2 k_{\delta}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (22)

$p_{i,n} = 0$

$$\forall i \in S$$ \hspace{1cm} (23)

$p_{i,2N} = 0$

$$\forall i \in S$$ \hspace{1cm} (23)

$n > 1 i' in$;

$p_{i,n-1} - \delta_{i,n} + y_{i,n} \leq w_{i, -}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (24)

$n > 1 i' in$;

$p_{i,n} = p_{i,n-1} - \delta_{i,n} + y_{i,n}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (25)

$Q_{i,n} = t_{i,n}^3 k_{n}$

$$\forall n \in D$$ \hspace{1cm} (26)

$Q_{i,n}^0 = Q_{i,n} - y_{i,n}$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (27)

$i > 1 i' in$;

$Q_{i,n} = (t_{i,n}^4 - t_{i-1,n}^3) k_{n}^4 + Q_{i,n}^0$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (28)

2.1.5.6. Non-negativity Constraints

$x_{i,n}, t_{i,n}^4, t_{i,n}^5 \geq 0$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (29)

$h \geq 0$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (30)

$t_{i,n}^4, t_{i,n}^5 \geq 0$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (31)

$w_i, y_i^1, y_i^2 \geq 0$

$$\forall i \in S$$ \hspace{1cm} (32)

$p_{i,n}, y_{i,n}, \delta_{i,n}, Q_{i,n}, Q_{i,n}^0 \geq 0$

$$\forall i \in S, \forall n \in D$$ \hspace{1cm} (33)
2.1.6. Objective Function

\[ \min Z = \sum_{i=1}^{S} \sum_{n=1}^{D} Q_{i,n}^{0} \]  \hspace{1cm} (34)

2.2. The Mixed-Integer Programming Model

The LP model is impractical due to unrealistic fractional values of the variables. Despite this handicap the LP model is also advantageous, since the problem size increases the solution time by a polynomial function and thus the solver algorithm can very effectively find the solution in insignificantly small times.

On the other hand, the MIP model is more realistic and applicable. But the optimal solution is not attainable in reasonable times. With some runtime limitations to the solver algorithm, some sub-optimal solutions to the problem can be attained.

The MIP model will not be openly represented in this paper due to the similarity with the LP model. Technically the MIP and the LP models have the same parameters, same constraints and the same objective functions. They also have similar variables with different domains. Variables of the LP can be assigned non-negative real numbers, but the variables of the MIP must be assigned non-negative integers.

2.3. The CCR Model

Since there are eight trade off decision alternatives to be compared the CCR model is run eight times. Each model has 11 variables, 12 constraints and a single objective.

2.3.1. Sets

Set of trade off decisions (the optimality gap-runtime tuples, DMUs)

\[ \text{DMUs} = \{ o, j \mid \forall o, j \in \text{DMUs} | o, j \in [1, n], n = 8; \ o, j \in \mathbb{Z}_+ \} \]

Set of runtimes (Inputs):

\[ \text{Inputs} = \{ i \mid \forall i \in \text{Inputs} | i \in [1, m], m = 1, \ i \in \mathbb{Z}_+ \} \]

Set of optimality gap scores (Outputs):

Optimality gap is the ratio of the MIP solution to the LP optimal solution and thus, less optimality gap is better. In other words optimality gap is an input type criterion. In this analysis the reciprocal of the optimality gap \((LP \text{ optimal solution}/MIP \text{ solution})\) is used as the output criterion and still referred as the optimality gap for simplicity.

\[ \text{Outputs} = \{ r \mid \forall r \in \text{Outputs} | r \in [1, s], s = 1, \ r \in \mathbb{Z}_+ \} \]

2.3.2. Parameters

\[ x_{ij} : \text{amount of input } i \text{ consumed by the } j^{th} \text{ DMU while producing outputs} \]
\[ y_{rj} : \text{amount of output } r \text{ produced by the } j^{th} \text{ DMU using the inputs} \]

2.3.3. Data Set

<table>
<thead>
<tr>
<th>Table 1: Runtime and Optimality gap values of DMUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime</td>
</tr>
<tr>
<td>Optimality gap</td>
</tr>
</tbody>
</table>

2.3.4. Variables

\[ \lambda_{ij} : \text{intensity factor that defines the efficient projection of the } o^{th} \text{ DMU in terms of the } j^{th} \text{ DMU} \]
\[ \theta_{o} : \text{efficiency score of the } o^{th} \text{ DMU} \]

2.3.5. Constraints

The CCR model has three group of constraints.

2.3.5.1. Input Related Constraints

\[ \sum_{j=1}^{n} \lambda_{ij} x_{ij} \leq \theta_{o} x_{1o} \hspace{1cm} \forall i \in \text{Inputs} \hspace{1cm} (35) \]

2.3.5.2. Output Related Constraints

\[ \sum_{j=1}^{n} \lambda_{ij} y_{ij} \geq y_{ro} \hspace{1cm} \forall r \in \text{Outputs} \hspace{1cm} (36) \]

2.3.5.3. Non-negativity Constraints

\[ \lambda_{ij} \in \mathbb{R}_+ \hspace{1cm} \forall j \in \text{DMUs} \hspace{1cm} (37) \]

\[ \theta_{o} = \theta_{o}' - \theta_{o}' \mid \theta_{o} \in \mathbb{R}, \ \theta_{o}', \theta_{o}'' \in \mathbb{R}_+ \hspace{1cm} (38) \]

2.3.6. Objective Function

\[ \min \theta_{o} \hspace{1cm} \forall o \in \text{DMU} \hspace{1cm} (39) \]
III. RESULTS
The LP optimal solution, various MIP solutions under various runtime limitations and the MIP constraint satisfaction model solution are summarized below in Table2.

Table 2: LP and MIP Results Summarized

<table>
<thead>
<tr>
<th></th>
<th>Objective Function Value</th>
<th>Runtime (in seconds)</th>
<th>Optimality Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP optimal solution</td>
<td>71336</td>
<td>0.39</td>
<td>-</td>
</tr>
<tr>
<td>MIP (constraint satisfaction)</td>
<td>909900</td>
<td>0.64</td>
<td>0.078400</td>
</tr>
<tr>
<td>MIP 1second</td>
<td>71435</td>
<td>1</td>
<td>0.998614</td>
</tr>
<tr>
<td>MIP 2seconds</td>
<td>71418</td>
<td>2</td>
<td>0.998852</td>
</tr>
<tr>
<td>MIP 3seconds</td>
<td>71412</td>
<td>3</td>
<td>0.998936</td>
</tr>
<tr>
<td>MIP 6seconds</td>
<td>71412</td>
<td>6</td>
<td>0.998936</td>
</tr>
<tr>
<td>MIP 7seconds</td>
<td>71399</td>
<td>7</td>
<td>0.999118</td>
</tr>
</tbody>
</table>

It can be observed from Table 2 that the MIP solutions form a clear trade-off between the solver algorithm runtime (i.e. the decision makers' agility, speed to adapt to the changes in the conditions). The optimality gap-runtime tuples are different decisions the decision maker has to choose among. The data envelopment analysis provides the decision maker a tool to compare these alternatives.

Table 3: Performance Analysis Results of the DMUs

<table>
<thead>
<tr>
<th></th>
<th>DMU1</th>
<th>DMU2</th>
<th>DMU3</th>
<th>DMU4</th>
<th>DMU5</th>
<th>DMU6</th>
<th>DMU7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency Score</td>
<td>12.46%</td>
<td>100.00%</td>
<td>50.01%</td>
<td>33.34%</td>
<td>25.01%</td>
<td>20.01%</td>
<td>16.67%</td>
</tr>
<tr>
<td>λ_{DMU1}</td>
<td>0.07851</td>
<td>1</td>
<td>1.00024</td>
<td>1.00032</td>
<td>1.00032</td>
<td>1.00032</td>
<td>1.00032</td>
</tr>
</tbody>
</table>

The sole efficient DMU is DMU1. The efficient DMUs of CCR models represent the most efficient trade-off between the inputs and the outputs. In other words investing 1 second in the solver algorithm yields the most efficient reduction in the objective function and thus, DMU2 is the most productive scale size (i.e. the best choice).

IV. CONCLUSION
When decision makers face complex problems that they are required to solve repeatedly as the decision making environment change continuously, agility to respond become an objective. In such situations decision makers need a tool to have control over the solution time.

Re-defining the problem as a trade-off between sub-optimality and the solution time lets the decision maker to control the solution time and thus, the level of decision making agility.

This paper suggests a dual approach for the situation. The complex problem is solved under some runtime limitation and the different levels of suboptimalities are recorded in the first phase. In the second phase the results obtained from phase one are compared using data envelopment analysis to identify the most efficient trade-off decision.

V. DISCUSSION
The data of this study was generated randomly and the problem size is deliberately kept small for simplicity. It is assumed that this compact decision environment is scalable to real life problems. It is also assumed that there are no efficient algorithms for the problem at hand.

This study aims to expand the research in decision making area by composing a multiple criteria decision making technique and optimization together to empower the decision maker in solution time control for complex problems. The dual approach this paper suggest can be taken as an alternative to the meta-heuristic approaches.

REFERENCES
Sub Optimality/Runtime Trade off Analysis on a Single Track Train Scheduling Problem


