PREVENTIVE AND CORRECTIVE MAINTENANCE SCHEDULING FOR VEHICLES AT A SINGLE DEAD-END TRACK UNDER SERVICE LEVEL AGREEMENT

by

Murat Elhüseyni

M.S., Industrial Engineering, Istanbul Technical University, 2012

B.S., Industrial Engineering, Yıldız Technical University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Industrial Engineering

Boğaziçi University

2018

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Assoc. Prof. Ali Tamer Ünal. He has believed in me from beginning till the end. His vision has helped me think creatively and holistically. Idealizing a practical problem in a theoretical way has been very challenging for me and it would never have been possible to complete this study successfully without his patience, guidance and support. I have always kept my motivation thanks to his positive attitude towards me even at hard times during this study. It has been a great pleasure to work with him.

I am particularly grateful for the critics given by my thesis committee members, Prof. Gülgün Kayakutlu and Prof. Kuban Altınel. Their suggestions have really improved my work. Besides, I deeply appreciate Prof. Altinel's guidance as an academic advisor before PhD thesis stage. I would also like to thank Prof. Semra Ağralı and Assoc. Prof. Gönenç Yücel for taking part in my thesis committee.

My sincere thanks go to my colleagues Murat Güngör, Pınar Dursun and my old friend Burak Kocuk for their friendship, answering my questions and their support. Furthermore, I am also very thankful to Bogazici IE department for ensuring a peaceful and friendly work environment during my teaching assistantship career. I am indebted to Prof. Zeki Caner Taşkın and Prof. Necati Aras for generously sharing their work stations.

This thesis would not have a solid practical support if I had not learnt vehicle maintenance system in MetroIstanbul. I owe thanks to Dr. Hayri Baraçlı, secretary general of Istanbul Metropolitan Municipality, and Ali Fırat who is a deputy general manager in MetroIstanbul for directing me to maintenance department. I am very thankful to Hasan Şen, LRV maintenance workshop chief, for showing and telling me how they handle maintenance elaborately. I am indebted to LRT tram vehicle workshop manager, Orhan Ustaoğlu. He told me their maintenance strategy in MetroIstanbul and provided me crucial documents.

Words cannot express how grateful I am to my dear parents for their unconditional and endless support in my work and me. I am so lucky to have such a family and I always feel indebted to their sacrifices for me in my life. Thank you for giving me a chance to prove myself through all stages of my life.

I wish to thank TÜBİTAK for providing a financial support during my graduate study under 2211-A scholarship program.

ABSTRACT

PREVENTIVE AND CORRECTIVE MAINTENANCE SCHEDULING FOR VEHICLES AT A SINGLE DEAD-END TRACK UNDER SERVICE LEVEL AGREEMENT

We introduce the problem of vehicle maintenance scheduling under service level agreement (SLA) and preventive maintenance cycles at a single dead-end track. We show that the problem is NP-Hard. We build MILP model to solve the problem and propose formulation improvements based on problem structure. Besides, we develop a heuristic that generates an initial feasible solution to the MILP solver. As a result of computational experiments, we show that improved model, which is a combination of formulation improvements, CPLEX parameter fine-tuning and the heuristic drastically heightens the solution quality compared to the MILP model under given time limits. We select the improved model as a solution method. We create a discrete-event simulation environment to determine effects of problem parameters on key performance indicators (KPI). We build two alternative methodologies, as corrective jobs worsen tardiness related KPIs of preventive jobs. One of them is the buffer method which reschedules the result of the solution method, the other one is anticipation method in which we alter objective function of the solution method to favor earliness in the model. In conclusion, these methods diminish tardiness related KPIs of preventive jobs but they increase preventive earliness compared to the solution method.

ÖZET

HİZMET SEVİYE ANLAŞMASI ALTINDA TEK UCU KAPALI RAYDA ARAÇLAR İÇİN ÖNLEYİCİ VE DÜZELTİCİ BAKIM ÇİZELGELEMESİ

Hizmet seviye anlaşması (HSA) ve planlı bakım çevrimleri altında tek ucu kapalı rayda araç bakım çizelgelemesi problemini ele aldık. Problemin NP-Zor olduğunu gösterdik. Problemi çözmek için Karma Tamsayılı Program (KTP) modeli ve formulasyon iyileştirmeleri önerdik. Ayrıca KTP çözücüsüne başlangıç çözümü veren bir sezgisel geliştirdik. Bilgisayısal deneyler sonucunda formulasyon iyileştirmesi, CPLEX parametre ince ayarı ve sezgiselin birleşimi olan gelişmiş model, verilen zaman sınırları altında KTP modeline kıyasla çözüm kalitesini ciddi şekilde arttırmıştır. Gelişmiş modeli çözüm yöntemi olarak seçtik. Problem parametrelerinin anahtar performans göstergesine (APG) etkisini belirlemek için kesikli olay benzetimi ortamı yarattık. Düzeltici işler önleyici işlerin geçlik bazlı APGlerini kötüleştirdiği için iki alternatif metodoloji önerdik. Birincisi çözüm yönteminin verdiği sonucu yeniden çizelgeleyen tampon yöntemi, diğeri ise modelde erkenliği gözetmek için çözüm yönteminin amaç fonksiyonunu değiştirdiğimiz beklenti yöntemidir. Sonuç olarak bu yöntemler çözüm yöntemine göre önleyici işlerin geçlik bazlı APGlerini düşürmekte fakat önleyici erkenliğini yükseltmektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . v
ÖZET . . . vi
LIST OF FIGURES . . . x
LIST OF TABLES . . . xv
LIST OF SYMBOLS . . . xvii
LIST OF ACRONYMS/ABBREVIATIONS . . . xxiii
1. INTRODUCTION . . . 1
2. LITERATURE REVIEW . . . 5
2.1. Train, Tram and Bus Maintenance . . . 5
2.2. Aviation Maintenance . . . 6
2.3. Corrective Maintenance . . . 10
2.4. Parking . . . 10
3. PROBLEM DEFINITION AND MOTIVATIONS . . . 13
4. MATHEMATICAL FORMULATIONS AND THE SOLUTION APPROACH 25
4.1. MILP Model . . . 25
4.1.1. Assumptions . . . 26
4.1.2. Sets . . . 26
4.1.3. Parameters . . . 26
4.1.4. Decision variables . . . 27
4.1.5. Mathematical Model . . . 28
4.2. Model Improvements . . . 32
4.2.1. Optimality Conditions . . . 32
4.2.2. Model II . . . 36
4.2.3. Optimality Cuts . . . 45
4.2.4. Valid Inequalities . . . 45
4.3. Heuristic Algorithm . . . 46
4.3.1. Sets . . . 47

4.3.2. Parameters . . . 47
4.3.3. Dynamic Sets . . . 48
4.3.4. Some Variables . . . 48
4.4. Alternative Methods for Handling Corrective Jobs . . . 49
4.4.1. Introduction of Buffer Time . . . 49
4.4.2. Anticipation of future events . . . 53
5. COMPUTATIONAL EXPERIMENT DESIGN . . . 54
5.1. Service Level Agreement Generation . . . 54
5.2. Preventive Job Generation . . . 58
5.3. Corrective Job Generation . . . 59
5.4. Job Duration Generation . . . 60
5.5. Determination of Critical Jobs . . . 61
5.6. Warmup Period Determination . . . 65
5.7. Single Run Length and Batch Size . . . 67
5.8. Cplex Parameter Fine-Tuning . . . 68
5.9. Computational Study . . . 69
5.9.1. 2 mins Time Limit . . . 70
5.9.2. 10 mins Time Limit . . . 74
5.10. Experiment Plan . . . 79
6. RESULTS AND DISCUSSIONS . . . 82
6.1. Preventive Maintenance . . . 82
6.1.1. Effect of Track Capacity . . . 83
6.1.2. Effect of Service Level Agreement . . . 83
6.1.3. Effect of Preventive Interarrival Times . . . 84
6.1.4. Effect of Variability Factor in Preventive Intervals . . . 84
6.1.5. Regression Analysis . . . 85
6.1.6. Marginal Analysis . . . 86
6.1.7. Multifactor Effect . . . 87
6.2. Preventive and Corrective Maintenance . . . 88
6.2.1. Effect of Track Capacity . . . 88
6.2.2. Effect of Service Level Agreement . . . 89

6.2.3. Effect of Preventive Interarrival Times 89
6.2.4. Effect of Variability Factor in Preventive Intervals 90
6.2.5. Effect of Mean Time Between Failures 91
6.2.6. Regression Analysis 92
6.2.7. Multifactor Effect 93
6.2.8. Comparison at High Load 95
6.3. Comparison of Solution Methods 97
7. CONCLUSION 101
REFERENCES 105

LIST OF FIGURES

Figure 3.1. Two Configurations of a Shunt Track [1] 13

Figure 3.2. Crossing Example [2] 14

Figure 3.3. Various Block Schedules 16

Figure 3.4. SLA Schedule Example 16

Figure 3.5. Spare Vehicles and SLA 17

Figure 3.6. Preventive Maintence Interval and Scheduling Due time 18

Figure 3.7. Rolling Horizon Scheme 19

Figure 3.8. Weekly Schedule based on Critical Jobs 19

Figure 3.9. Extended Rolling Horizon Scheme 20

Figure 3.10. Weekly Schedule based on Preventive Jobs 20

Figure 3.11. Relationship between SLA and Parking Capacity without Corrective Jobs 22

Figure 3.12. Relationship between SLA and Parking Capacity with Corrective Jobs 22

Figure 4.1. Schedule and its Assignment Plot 25

Figure 4.2. Case 1 Change in Parent-Child Relationship when $dist = 0$ 35

Figure 4.3. Case 1 Change in Parent-Child Relationship when $dist > 0$ 35

Figure 4.4. Case 2 Change in Parent-Child Relationship 36

Figure 4.5. Possible Assignments to Track Locations with respect to Lemma 4.5 38

Figure 4.6. Possible Parent-Child Assignments based on b' and b'' 39

Figure 4.7. Possible Parent-Child Assignments based on a 39

Figure 4.8. Maximum Job Assignments to Second Track Position Example . . 41

Figure 4.9. Maximum Job Assignments to Third Track Position Example . . . 41

Figure 4.10. Possible Job Assignments to Track Position 3 42

Figure 4.11. Dimensionality Reduction Example 42

Figure 4.12. General Heuristic Algorithm 49

Figure 4.13. Core Heuristic Algoritm 50

Figure 4.14. Buffer Time Method Example 52

Figure 4.15. Anticipation Solution of SP 53

Figure 5.1. Core SLA Schedule 55

Figure 5.2. Case 1 Schedule 56

Figure 5.3. Case 2 Schedule 57

Figure 5.4. Weekly Case 1 Schedule 57

Figure 5.5. Weekly Case 2 Schedule 58

Figure 5.6. First Two Preventive Maintenance Intervals for a Vehicle 59

Figure 5.7. Ongoing Jobs Case I Example 62

Figure 5.8. Ongoing Jobs Case II Example 62

Figure 5.9. Ongoing Jobs Case III Example 62

Figure 5.10. SP and Preventive Maintenance Intervals 63

Figure 5.11. Overlapping of Maintenance Jobs for a Vehicle 64

Figure 5.12. Conflict Free Maintenance Job Asssignments for a Vehicle 64

Figure 5.13. Preventive Interval and Stay 66

Figure 5.14. Corrective Interval and Stay 66

Figure 5.15. Average Cumulative Stays in Test System 67

Figure 5.16. Autocorrelation Graph for 10 Batch Size 68

Figure 5.17. Gap(%) Normality Plot 70

Figure 5.18. Low Load System Comparison under Z_f with 2 mins Time Limit . 71

Figure 5.19. Low Load System Comparison under Gap(%) with 2 mins Time Limit 71

Figure 5.20. Middle Load System Comparison under Z_f with 2 mins Time Limit 72

Figure 5.21. Middle Load System Comparison under Gap(%) with 2 mins Time Limit 73

Figure 5.22. High Load System Comparison under Z_f with 2 mins Time Limit 73

Figure 5.23. High Load System Comparison under Gap(%) with 2 mins Time Limit 74

Figure 5.24. Low Load System Comparison under Z_f with 10 mins Time Limit 76

Figure 5.25. Low Load System Comparison under Gap(%) with 10 mins Time Limit 76

Figure 5.26. Middle Load System Comparison under Z_f with 10 mins Time Limit 77

Figure 5.27. Middle Load System Comparison under Gap(%) with 10 mins Time Limit 77

Figure 5.28. High Load System Comparison under Z_f with 10 mins Time Limit 78

Figure 5.29. High Load System Comparison under Gap(%) with 10 mins Time Limit 78

Figure 6.1. Four Performance Indicators under Preventive Maintenance 87

Figure 6.2. Slaviol under Preventive Maintenance 87

Figure 6.3. Shop Load % under Preventive Maintenance 88

Figure 6.4. Four Performance Indicators under Preventive and Corrective Maintenance 93

Figure 6.5. Slaviol under Preventive and Corrective Maintenance 94

Figure 6.6. Shop Load % under Preventive and Corrective Maintenance 94

Figure 6.7. Four Performance Indicators under $\Pi = 480$ vs $\tau = 2508$ 95

Figure 6.8. Slaviol under $\Pi = 480$ and $\tau = 2508$ 96

Figure 6.9. Shop Load % under $\Pi = 480$ vs $\tau = 2508$ 96

LIST OF TABLES

Table 5.1. Warmup period experimental factors 65

Table 5.2. Time comparisons of methods under 2 mins time limit 75

Table 5.3. Time comparisons of methods under 10 mins time limit 79

Table 5.4. Simulation experimental factors 80

Table 6.1. KPI values depend on m change 83

Table 6.2. KPI values depend on SLA change 84

Table 6.3. KPI values depend on Π change 84

Table 6.4. KPI values depend on v change 85

Table 6.5. KPI values depend on Π change 86

Table 6.6. KPI values depend on v change under preventive and corrective maintenance 88

Table 6.7. KPI values depend on SLA change under preventive and corrective maintenance 89

Table 6.8. KPI values depend on Π change under preventive and corrective maintenance 90

Table 6.9. KPI values depend on v change under preventive and corrective maintenance 90

Table 6.10. Comparison of KPIs under preventive maintenance and preventive and corrective maintenance 92

Table 6.11. Comparison of two systems at high load 97

Table 6.12. Method comparison 98

Table 6.13. Model based KPIs and Z values of methods 100

LIST OF SYMBOLS

a	A coefficient to determine SLA_b
a_{dj}	$\|T\| - d_j$
a_{jt}	1 if job j is in maintenance hangar at time period t, else 0
$\hat{a_{jt}}$	Realized value of a_{jt}
A	$\lceil \frac{num_b}{c_{ub}} \rceil$
b_{dj}	$\|T\| - p_j - 1$
b_i	Base job of block i
b_j	Maintenance begins time of job j
b_{jt}	1 if job j's maintenance work begins at time period t, else 0
beg_i	$(i-1)\,\|CR\|$
B	Bases
B_{bj}	Linearization of $x_{bj}S_b$
B_i	Bases at track i
$B_i^{'}$	$beg_i....end_i - (i-1)$
$B_{i+1}^{'}$	$beg_{i+1}....end_{i+1} - pos_b$
$B_{i-1}^{'}$	$beg_{i-1}....end_{i-1} - (num_a + i - 2)$
$Base$	Set of bases where each base contains info like start,length and finish time along with appointed job j
Bl_i	ith block
c_i	Last arriving child of b_i
c_{ub}	Maximum possible number of children for any base b
C	Track positions
CM_j	Maintenance finish time of job j
C_j	Exit time of job j from maintenance hangar
CR	Critical Jobs
$C(S)$	Finish time of block schedule S
d	A coefficient to determine δ_2
df	Degree of freedom in t-test

d_j	Due time of job j
D	A set where jobs in J are ordered in descending order of their processing times
e	A coefficient to determine n
e_j	Maintenance finish time of job j
e_{jt}	1 if job j exits from maintenance hangar at the end of time period t, else 0
end_i	$i\|CR\| - 1$
ev_j	Number of corrective events for every job
E_j	Earliness time of job j
$Exit$	Jobs that have abandoned maintenance hangar so far
$Exp(\lambda)$	Exponential distribution with λ parameter
f	A coefficient to determine δ_1
F_b	Finish time of base b
Gap(%)	Optimality gap obtained from CPLEX given time limit
$hangarslack$	Unused capacity in hangar in any time instant
H_t	Jobs at maintenance hangar at time t
$Hangarcandid$	Jobs that are candidates to be pulled to hangar
I_i	Idle time of second track position at block i
IB	$\{b \in B \| b = 0, 0 + \|CR\|, , , (k-1)\|CR\|\}$ where $k = min\left(\|C\|, \|J_{tard}\|\right)$
IS_t	$\left\{j \in PM \| t < Y_j^{min}\right\}$
J	Vehicle maintenance jobs
J_i	Jobs in Bl_i
J_k	kth set that is composed of jobs where each of them share same due times with others such that $\|J_k\| \geq 2$, $J_k \subseteq J$, $k \in K$
J_t	Jobs at time t
J_{tard}	$\left\{j \in CR \| j \in U \bigcup j \in PM \quad s.t \quad Y_j^{max} \leq wk_b\right\}$
K_b	Upper bound value for b's number of children such that $K_b > 0$ and $K_b = \|CR\| - pos_b$
L	Planning horizon length
L_{bj}	Linearization of $x_{bj} F_b$

m	Number of track locations
M	Large number
n	Number of vehicles in our system
$neigh$	Maximum allowed slack size for jobs to pull them to maintenance hangar
$nextper$	Number of jobs skipped to next period
np_j	1 if job j is delayed to next period, else 0
$numbertohangar$	Number of railcars that can be appointed to maintenance hangar at time t
num_a	k th number of base at track $i+1$, $num_a = a - beg_{i+1} + 1$
num_b	k th number of base at track $i < m$, $num_b = b - beg_i + 1$
NC	Noncritical jobs
NP	$\left\{j \in PM \mid \quad Y_j^{max} \geq wk_e\right\}$
obj	Objective function value
o_{jt}	1 if job j is out of service at time t, else
$Ongoings$	Ongoing jobs in heuristic
OB	Bases where ongoing jobs reside
OJ	Ongoing jobs
p	Preventive job Time
p_{cor}	Corrective job Time
$pinc$	Best incumbent solution so far
p_j	Maintenance duration of job j
$pnew$	New solution of *CoreHeuristic*
p_{max}	$max_{j \in CR} p_j$
pos_b	Least number of assigned jobs till base b beginning from base 0 where $pos_b = (b - beg_i + 1) + (i - 1)$
$ParentChild$	Set that keeps *child* and *parent* pairs
P_{asc}	Set where jobs in CR are ordered in nondecreasing order of their processing times
P_b	Park time of base b
P_i	Makespan of ith block
PM	Only preventive jobs

PW	$\{j \in CR \mid \quad Y_j^{min} \leq wk_b + p_{max}\}$
$Pmax$	Schedule makespan
r_j^{max}	$\max\left(Y_j^{max}, 0\right)$
r_j^{min}	$\max\left(Y_j^{min}, 0\right)$
$slack_j$	Distance of jobs to their corresponding due date
S	Schedule S
S_b	Start time of base b
S_j	Maintenance start time of job j
$\hat{S}_j$	Realized value of S_j
SLA_b	Daily total SLA demand
SLA_c	Core SLA schedule
SLA_c^{max}	Maximum value in SLA_c
SLA_t	Number of required vehicles to satisfy SLA at time t
t_b	Threshold for possible number of children of base b such that $t_b = num_b c_{ub}$
T_j	Tardiness time of job, $j \in PM$
	Time between breakdown of a vehicle and its maintenance start time, $j \in U$
T	Time Horizon
TS	Test statistic in t-test
u_z	Uniform number between 0 and 1 to generate z_{th} corrective job
U	Corrective jobs
$Urgents$	Preventive jobs that out of service at time t and all corrective jobs
v	Variability factor in preventive intervals
V	Vehicle set
w_b	1 if $F_b = S_b + \sum_{j \in CR} x_{bj} p_j$, else 0
wk_b	Begin time of the current week
wk_e	End time of the current week
wn_b	Number of jobs assigned till base b in the worst case
W_i	Waiting time of f_i at block i

x_{bj}	1 if job j is the base job of block b, else 0
X_z	zth corrective job interarrival time
y_{ab}	1 if a is a child of b where $b \in B_i$, $a \in B_{i+1}$, else 0
Y_j^{max}	End time of maintenance interval of job j
Y_{jz}^{max}	z_{th} end time of maintenance interval of job j
Y_{jz}^{min}	z_{th} begin time of maintenance interval of job j
Y_j^{min}	Begin time of maintenance interval of job j
z	z_{th} maintenance interval
z_{ab}	1 if $F_a > S_b + \sum_{j \in CR} x_{bj} p_j$ given that $y_{ab} = 1$, else 0
Z	Weighted sum of model based KPIs according to objective function coefficients of model
Z_f	Objective value of feasible solution of model given time limit
Z_S	Objective value of schedule S
α	Statistical significance threshold
β	Begin time upper limit for preventive job generation
$\triangle_i$	Difference between parking stays of parent job and child block at a specific block i
$\triangle SLA_t^-$	Number of tram fails to satisfy SLA_t demand
$\triangle Y_j^{max+}$	Number of time units that Y_j^{max} is violated
χ_k	$\lceil \frac{\chi_{k-1}}{c_{ub}} \rceil$
δ_1	$\lceil SLA_c'^{max} f \rceil$
δ_2	$\lceil SLA_c'^{max} d \rceil$
ϵ	Penalty per one unit violation of $d_j, \quad j \in U$
γ	Penalty per one unit violation of SLA schedule
κ	Preventive and corrective job time variability
μ_1	True population mean of first sample
μ_2	True population mean of second sample
ω	T_j coefficient in objective function of MILP model for anticipation of future events method
ϕ	Expected number of vehicle fleet breakdowns within first 24 hours
σ	Penalty per one unit violation of $Y_j^{max}, \quad j \in PM$

τ	Corrective job mean interarrival time
θ_1	Latest start coefficient of preventive job
θ_2	Latest start coefficient of corrective job
$\mathit{\Pi}$	Preventive Job Interarrival Time
ϱ	Due time coefficient of preventive job
ζ	Buffer time in terms of hours

LIST OF ACRONYMS/ABBREVIATIONS

C++	C Plus Plus
DI	Daily Inspection
FMP	Flight and Maintenance Problem
KPI	Key Performance Indicator
LIFO	Last In First Out
MI	Monthly Inspection
MILP	Mixed Integer Linear Programming
OSR	Operating Spare Ratio
SLA	Service Level Agreement
SP	Scheduling Period
TUSP	Train Unit Shunting Problem
UNIF	Uniform Distribution
VLNS	Variable Large Neighborhood Search

1. INTRODUCTION

Due to urbanization and increasing population size, tram service operations have become a challenging task day by day for tram service providers. They should operate vehicles efficiently so that they satisfy passenger demands with a reasonable fleet size and cost. The preventive maintenance plays an important role to fulfill this aim, as a poor preventive maintenance policy results in too much breakdowns for vehicles. This threatens transportation safety, diminishes availability of vehicles, incurs major repair costs and damages the reputation of service providers. Service providers control various service lines that each one of them has their own path. In this work, we engage in one type of short term preventive maintenance operations in one service line while ignoring heavy maintenances like overhauls and multi-types in any of these service lines.

Tram service providers need to provide efficient services to satisfy increasing passenger demands to their services. While doing that, they must also consider planning their services successfully through using their resources in full potential. Hence, they should manage their system not only considering only preventive maintenance cycles, but also their resources like manpower and hangar tracks. Besides, they need to run their system according to demand pattern to their services. Therefore, all these aspects of the vehicle maintenance system need a holistic approach, as focusing on one of these features while disregarding another one would lead to suboptimal results in the long term. Yet, usually the provider is not aware of this suboptimality in their plan because they confront daily operational problems like satisfaction of demanded vehicle quantity. Since it is costly to see the effects of such an approach in an existing system, we build an abstract environment to see long term effects of factors that have an impact upon the system.

Our maintenance hangar consists of one single dead-end track. Hence, vehicles are parked to track according to LIFO rule. It entails that park time of each vehicle at this track does not equal to maintenance or intended stay time at all cases. Let us consider two vehicles. One of them is already parked and while its stay, another vehicle parks into that track. After a while, already parked vehicle tries to leave the track but it is obstructed by later incoming vehicle. This case is called *crossing*. This makes the problem harder to solve. Preventive maintenance interval is a time window that a vehicle can undergo a maintenance. Since one vehicle contains many parts and each part has its own maintenance time window, actually there are several intervals per vehicle. Nevertheless, we employ *one stop principle* which is a combination of maintenance jobs in one stop. Dekker [3] names it as *opportunistic maintenance* in which neighboring and deteriorated components of the main failed components undergo preventive action. Therefore, we assume that every vehicle has one maintenance interval and thus one maintenance job.

Service Level Agreement (SLA) is an agreement between the service provider and the customer that provides a guarantee for satisfaction of determined service levels [4]. We are interested in commitments between a city and a tram/train service provider. In this regard, the customer is people living in a city and the tram service provider aims at fulfilling their travel demands via assigning convenient type and number of vehicles [5]. We consider the commitment from the maintenance planner of the tram service provider's point of view. For them, operation department of the provider is a customer such that their SLA targets must be met. Number of demanded vehicles vary by time so it results in a temporal SLA schedule. In order to prevent SLA breach, providers utilize spare vehicles to diminish the effect of breakdowns and preventive jobs.

In this thesis, we introduce the problem of vehicle maintenance scheduling under service level agreement (SLA) and preventive maintenance cycles in a single dead-end track. To the best of our knowledge, there is no work that deals with all of these aspects of the problem simultaneously. Furthermore, we show that even only hangar parking scheduling at the track is NP-Hard in the strong sense. During maintenance scheduling, we consider two operational scenarios. We define a due time in maintenance time window. It is a target point that vehicle maintenance should begin. If maintenance begins before it, preventive job becomes early and tardiness is occured if we pass the point. Corrective job tardiness is defined as time between breakdown event and maintenance start time. We build MILP model to solve the problem in which we minimize penalized SLA and latest maintenance start point violations as well as earliness and tardiness of preventive jobs and weighted tardiness of corrective jobs.

We derive a number of valid inequalities and some formulation improvements so as to speed up computational time. We also construct a heuristic to generate an initial feasible solution for the MILP solver. We propose two alternative methodologies to tackle negative effects of stochastic arrivals on performance indicators of preventive jobs. We create a discrete-event simulation environment to test the solution method. Throughout tests, we employ a rolling horizon method to schedule jobs using our methodology on each day by the end of the planning horizon for each parameter combination. We report results of our numerical experiments that measure for performance of our methodology under various solution parameters. As a result of the simulation study, we aim to draw conclusions about the general system and make general recommendations to assure a solid vehicle maintenance scheduling.

Rest of this work is organized as follows: We explain the problem in detail in Chapter 3. In Chapter 4, we elaborate on our solution methodology. In Chapter 5, we explain instance generation, warmup period and CPLEX parameter fine-tuning. Chapter 6 points out computational study, discussion of operational scenarios and alternative method comparisons with the MILP model. We draw conclusions from tests in Chapter 7.

Contributions of this work are as follows:

- It investigates the effect of SLA on a vehicle maintenance schedule at the single dead-end track. Other works might take various service demands like flight hours, flight legs or train duties into account but none of them are regarded as SLA.
- It incorporates single dead-end track into vehicle maintenance scheduling.

- It is one of the few studies in the literature that handles corrective maintenance.
- It considers a due time in maintenance time window.
- We make use of a heuristic, model improvements and CPLEX parameter fine-tunings to come up with efficient solutions during rolling horizon.
- It takes proactive methods into account to come up with robust schedules in the existence of corrective jobs.

2. LITERATURE REVIEW

There are vast number of studies that treats maintenance operations and transportation together in different manners. We generally examine preventive maintenance in terms of application areas. After that, since our work includes also corrective maintenance, we refer works how they treat stochasticity in their problem. Finally, our review ends with works that deal with parking in railtracks.

2.1. Train, Tram and Bus Maintenance

Haghani and Shafahi [6] handle bus maintenance scheduling problem such that they minimize the time that bus is out of service to maximize service reliability. Jing [7] concerns the impact of vehicle planning according to depots and depot capacity on deadhead km (non-value added trips) and average service delay. He uses a discrete event simulation for modelling interactions between different components of the system. He concludes that vehicles should be distributed to lines appropriately and new depot should be opened on a location where there is no depot in the neighborhood exists.

In trains, we encounter vehicle routing problems that are comprised of either rolling stock assignment or rolling stock circulations. Giacco *et al.* [8] take circulation into account with considering empty rides and maintenance issues so as to minimize number of trams required. Cadarso [9], Maróti and Kroon ([10], [11]) build models to route each individual rolling stocks into maintenance depot. Lai *et al.* [12] is different from them by concerning two maintenance types and deadhead movements. They handle routing and two maintenance types which are daily inspection(DI) and monthly inspection(MI). They minimize replacement cost of wrong assigned trains to trips, deadhead kms along with maintenance costs while ensuring feasible assignments to trips and depot capacity limits. They derive a heuristic to attack the problem. Since it has two maintenance types and replacement issue like Sriram and Haghani [13], they are similar in this regard but they treat these maintenances in a different way. Lai *et* *al.* [14] is an extension of it in which they find out optimal DI times in the presence of MIs. At last, Jacobsen and Pisinger [15] schedule trains in workshops and depot tracks in order to finish repairs while avoiding blockings in LIFO track and deadlines which causes tardiness.

Few works in preventive maintenance deal with high level maintenance issues. Lin and Lin [16] turn km requirements into days which is same conversation method we apply in our problem. They dispatch every train within their maintenance time window unlike us because we allow its violation as a soft constraint. They minimize unutilized remaining millaege by taking maintenance capacity and rate related constraints into account and solve the problem simulated annealing based solution approach. Sriskindajarah *et al.* [17] establish a common due date for each train which is composed of multiple railcars where each has its own due date. They minimize distance of each due date to common due date through genetic algorithm. Vernooij [18] handles boogie maintenance and diminishes work in progress of spare parts between maintenance depot and overhaul center by smoothing preventive maintenance calls to depot. Jiravanstit and Tharmmaphornphilas [19] consider monthly overhaul of vehicles from different service lines by diminishing of number of machines used by different tasks. They develop a MILP model for it. Then, given monthly plans, they devise a second MILP model to come up with detailed schedule of tasks to check whether number of machines at each month found by first model is feasible or not.

2.2. Aviation Maintenance

In aviation, we observe many diverse works. One group handles multi-commodity formulations to model vehicle maintenance routing. We examine these works in two classes. In the first class, problem is not daily, that is, solved for k-days or periodically. Feo and Bard [20] introduces origin-destination pairs for each flights and considers navigation of airlines on network in an infinite horizon while also considering a new maintenance facility construction in their set partitioning model. Gopalan and Talluri [21] treat the problem different than Feo and Bard [20] and introduce a polynomial time algorithm for aircraft maintenance routing problem in which aircrafts in fleets are directed to maintenance station at every 3-days provided that flown routes during the day is fixed. Sriram and Haghani [13] adopt Feo's approach over existing facilities but they also take Type B checks and cyclic weekly schedule with heteregeneous fleet into account. El Moudani and Camino [22], by contrast, split this problem into two parts. In the first part, airlines are alloted to flights with dynamic programming then maintenance reassignments are made via greedy heuristic. Keysan *et al.* [23] deal with maintenance timing of critical aircrafts. Because critical aircraft may overburden a maintenance facility, they appoint critical and noncritical aircrafts to flights in k-day period to ensure smooth hangar workload. Second class is about keeping track of remaining maintenance hour or km calculations. Sarac *et al.* [24] employ a set-partitioning model to employ branch-and-price algorithm. They minimize number of unused legal flying hours for high-time or critical airplanes in one day horizon without violating legal remaining flight hours, man-hour and slot availability of maintenance stations. Basdere and Bilge [25] handle similar objective but make use of weekly planning horizon and concern maintenance capacity in terms of number of aircrafts as an extension. They propose a heuristic based on compressed annealing. Al-Thani *et al.* [26] utilize almost same model with Basdere and Bilge [25] but they allow that an aircraft can undergo more than one maintenance check within a week unlike Basdere and Bilge [25]. Besides, they introduce graph preprocessing and valid inequalities and solve the remaining instance by Variable Large Neighborhood Search(VLNS).

Second group of works do not cover routing. In terms of a relationship between maintenance and flights, different approaches have been utilized in the literature. Some of them focus on aircraft utilization to fulfill this criteria. Afsar *et al.* [27] succeed in minimizing unused flight time of critical aircrafts that must undergo maintenance check and mark other aircrafts as uncritical to maximize general aircraft utility. When we look flight load maximization from a different side, we can see that it also corresponds to minimization of lost flying hours between maintenance checks. Thus, Boere [28] recommends a priority based scheduling reliant on a discrete event simulation for this objective. If one considers maintenance costs, flight assignments that have already been made might be changed. Sriraram and Haghani [13] combat this issue via minimizing costs during reassignment of already assigned flights. If more than one maintenance base and other aircraft resources are taken into consideration, Yang and Yang [29] minimize flight, opportunity and maintenance cost.

We generally detect studies on second group in military, especially in fighter aircrafts. Verhoeff *et al.* [30] state that military part of the problem distinct itself from civil counterpart due to having central maintenance base and mission readiness instead of profitability. Mission readiness is a flight availability optimization for a given budget. Mostly observed problems in this part can be named as *Flight and Maintenance Problem* (FMP). According to Gavranidis and Kozanidis [31], FMP decides on assignment of maintenance and flight hours to maximize unit fleet availability on a planning horizon under certain flight and maintenance requirements. Verhoeff *et al.* [30] state that sufficient training flight hours or loads are needed to keep aircrew in mission capable position. Pippin [32] determines flight load addressing this issue. He copes with minimization of deviations of flight hours till maintenance from ideal phase flow line while satisfying several flight hour bounds using MILP and another model which has a quadratic objective function while ignoring maintenance phase. Rosenzweig *et al.* [33] also consider Pippin [32]'s problem and model it with a MILP model that penalizes weighted deviations from ideal line in a civil training aircrafts. First study that tackles both maintenance and flight loads is Sgaslik [34]. He treats helicopter fleet maintenance and mission assignment problem. Maintenance planning part is considered in yearly planning model that penalizes violation of load and flight related constraints. This model yields monthly maintenance schedule and required flight hours for each helicopter. The short term model takes these parameters into account and brings about mission assignments of helicopter and spare helicopters. Steiner [35]'s work is similar to Sgaslik [34]'s yearly plan. In his heuristic based approach, maintenance activities are combined in master plan by shifting. Afterwards, optimization heuristic is executed to obtain optimal maintenance schedule and so as to minimize variability in flying hours per aircraft per time under maintenance and flight related constraints. The last two papers generally study flight load and maintenance schedule generation but fail to handle residual flight load of aircrafts through time. In Kozanidis and Skipis [36], residual flight time is defined as the total remaining time that an aircraft can fly until it has to undergo a maintenance check. Furthermore, a total deviation index is used to describe smoothness of residual flight times. It is the first paper that also takes residual flight times into consideration. They handle maintenance and flight load constraints in their bi-objective mathematical programming model which reckons with total number of available aircraft for customer service and total residual flight time. Kozanidis *et al.* [37] develop three heuristics for Kozanidis and Skipis [36] 's model since their model is quite limited to yield a nondominated solution in a reasonable number of time on large problems in terms of computational effort. Kozanidis *et al.* [38] is an extended version of Kozanidis *et al.* [37] regarding theoretical, computational and heuristics improvements. Kozanidis *et al.* [39] handle only minimization of least squares of total deviation of individual aircraft flight and maintenance times from their target values via built mixed integer nonlinear model. Finally, Gavranidis and Kozanidis [31] tackle this problem by virtue of maximizing residual flight time. It is the latest paper in their series of works and comes up with an exact solution algorithm for the FMP problem that find optimal solution for realistic problems in a reasonable time to fill the gap in FMP literature where we see heuristics with poor solution quality like Kozanidis *et al.* [37] or Kozanidis and Skipis [36] with computational intractable times. These works in this field have generally common constraints except their different kinds of objectives.

Other works take residual maintenance time in constraints into account but use different objective functions. Given sortie and maintenance requirement, Cho [40] minimizes maximum number of aircraft in maintenance at any time to smooth maintenance demand variability of fleet while distributing flights of aircrafts through time evenly. They decompose their MILP into two subproblem in their solution method. At first, they solve maintenance assignment problem and thereafter tackle flying assignments given solution of first maintenance problem to get rid of incurred computer tractability because of large datasets. Yet, it yields a suboptimal solution. Finally, Verhoeff *et al.* [30] present a model which is unique in terms of considering three key components of operational readiness at the same time. First one is availability that is total functioning duration of a fleet. Second one is serviceability that considers number of functioning aircrafts at a specific instant of time. These conditions are concerned while maximizing minimum sustainability which is a third factor and related with total remaining functioning duration of entire fleet at a particular time.

2.3. Corrective Maintenance

There is also a problem type that is similar to FMP problem but military aircraft fleets undergo a maintenance check not after certain flight hours like FMP but after part failures and scheduled maintenance intervals. Safaei *et al.* [41] treat a problem of execution of a healthy military flight program where major probabilistic failures are considered before and after daily flight mission checks. Their aim is to schedule repair shop under workforce availability constraints to maximize available aircrafts in flight fleet. Bajestani and Back [42] concern similar problem to Safaei *et al.* [41] in repair shop but they schedule maintenance jobs in shop to determine start-time of maintenance activities so that wave/fleet which comprises of various airplanes for military mission covered best in the presence of aircraft failure probabilities and maintenance capacity. Lastly, Lai *et al.* [12] and Lai *et al.* [14] consider stochastic breakdowns of trains as an addition to their preventive maintenance problem.

2.4. Parking

In daytime, tram operations are executed according to a timetable that tram is either in service or maintenance. Nevertheless, especially in nights, it is temporarily out of operation so it is convenient to park it on shunting area near one of the stations in the railway network. The main purpose is to park the rolling stock on shunt tracks in such a way that railway process at the next morning can embark on without any problem [43–45].

Some works in the literature treat train unit shunting problem (TUSP). Freeling *et al.* [46] strive to solve train unit shunting problem (TUSP) that is composed of matching arriving and departing train units and parking these units to minimize total shunting costs. However, they solve the problem in a separate way. Several papers attack TUSP problem as an integrated approach. Fons [47] uses Freeling *et al.* [46]'s parking model except allowing a parking at platform track. Kroon *et al.* [43] combine two separate steps of TUSP in one model. Akker [48]'s work takes platform waiting and moving train units between tracks into consideration which is different than Kroon *et al.* [43]'s model, on the other hand Jekkers [49] makes use of flexible shunt times in addition to Kroon *et al.* [43]'s approach. Gallo and Di Miele [50] stretch Winter and Zimmermann [51]'s Quadratic Assigment Model by way of using on bus dispatch schedule and Lübbecke and Zimmermann [52]'s through handling different vehicle lengths. Cardonha and Börndörfer [53] derive a three-index variable that combines arrival, departure and their assigned shunt track and solve the problem as a set-partitioning model whereas Borndörfer and Cardonha [54] devise quadratic programming approach and come up with a better model through relaxation this model.

Common assumption in these papers is that blocks which are result of matching problem are given as an input to a train parking problem and parking is handled over these blocks. However, they are different with regard to their focused parts in parking problem. Lübbecke and Zimmermann [52] and Blasum *et al.* [55] deal with departure parking problem of train unit vehicles but Winter and Zimmermann [51] investigate this problem in detail that they not only handle departure parking but also deal with online dispatching of incoming arrivals. Cornelsen and Di Stefano [56] treat dispatching trams to free track and Hamdouni *et al.* [57] engage in dispatching buses to lanes in order to minimize shunting efforts in the departure time of vehicles. Nevertheless, Cornelsen and Di Stefano [56] opt for graph algorithms to this problem whereas Hamdouni *et al.* [57] develop a robust formulation that is unlikely to be affected by incoming arrivals. Beygo [2] develops a formulation that includes many real life constraints.

Some papers make use of routing as well as parking and matching in depot. Yet, they may utilize different approach while handling these problems. Lentink *et al.* [58] decompose this problem into separate components while Hartog [59] tackles this problem with an integrated approach.

3. PROBLEM DEFINITION AND MOTIVATIONS

We introduce an integrated maintenance scheduling problem that incorporates single dead-end track parking in hangar into vehicle maintenance scheduling under SLA. Since this integrated problem contains three parts, which are single dead-end track parking, vehicle maintenance scheduling considering preventive cycles and SLA, we explain them separately. We formulate the problem and then we show that even only the hangar parking scheduling at the track is NP-Hard in the strong sense.

Our maintenance hangar consists of one single dead-end track. Hence, vehicles are parked to track according to the LIFO rule. In general, this type of track is called *LIFO track*, while tracks without a dead-end is named *free track* [1]. An example of it along with free tracks can be seen in Figure 3.1. In the figure, dead-end track and free track represented as stack and deque, respectively.

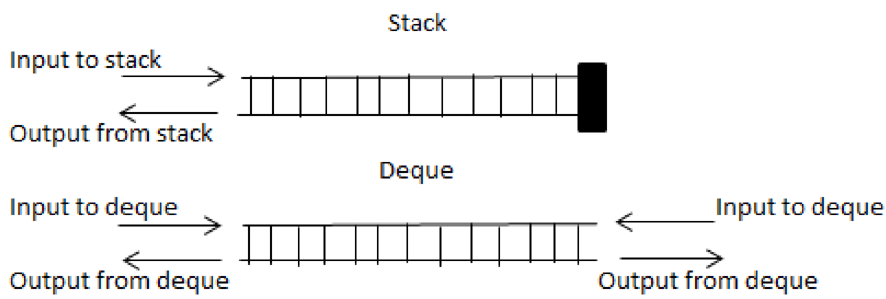

Figure 3.1. Two Configurations of a Shunt Track [1]

.

As we mentioned earlier, parking stay time of each vehicle could be longer due to crossing. Its example can be seen in Figure 3.2. Vehicle n prevents departure of vehicle m which prolongs block m's stay time. Therefore, the time a vehicle occupies a capacity on the track does not only depend on the maintenance time. It distinguishes this problem from train unit shunting problem literature in which crossings are not allowed. Yet, parking is treated differently in this study because safety measures that are taken during vehicle maintenance allow crossings. It follows that, we can not assign an exact stay time for each vehicle beforehand which makes it harder to schedule maintenance operations.

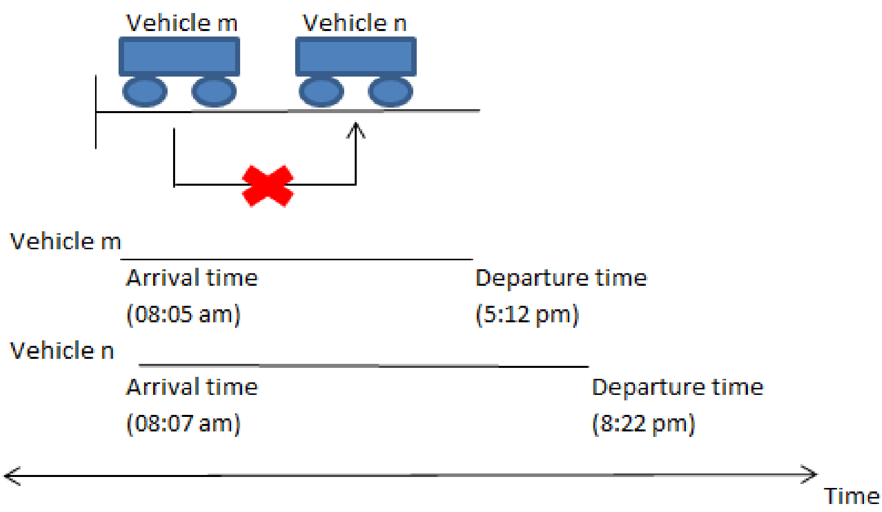

Figure 3.2. Crossing Example [2]

We work on LIFO track, so we make some definitions to transform situations arise in parking into our problem structure. We assume that vehicles are of equal length. Hence there are fixed positions on the track. Suppose a vehicle is parked in a track location k. During its stay, possible arriving and departing vehicles form a structure. We specify it as a *block*. Each later incoming vehicle has also its own block, so the structure is recursive. If no other vehicle visits the track during the parked vehicle's maintenance time, the vehicle leaves the track after its stay time. We call this case as a *simple block*. Simple block structure naturally occurs once the vehicle is parked into a last track position, m. Because last parked vehicle leaves the track after its maintenance time. Each vehicle has one maintenance job, so we name the vehicle at the root of the block as a *base job*. When any vehicle is parked into location $k+1$ during base job's maintenance duration, we define the vehicle as a *child job*. It results in a parent-child relationship where base job is a parent of the child job. In general, for two consequent track positions, we define some terms as follows:

p_j : Maintenance duration of job j

b_i : Base job of block i

c_i : Last arriving child of b_i

P_i : Makespan of ith block

$Pmax$: Schedule makespan

I_i : Idle time of second track position at block i

W_i : Waiting time of b_i at block i

S_j : Begin time of job j

CM_j : Maintenance finish time of job j

$C(S)$: Finish time of block schedule S.

Z_S : Objective value of schedule S

$$\triangle_i = CM_{b_i} - CM_{c_i} \tag{3.1}$$

Three cases arise in block schedule with regard to $\triangle_i$ value. They are given as below:

a. $\triangle_i > 0$: Second track position remains idle $\triangle_i$ time then $I_i = \triangle_i$.
b. $\triangle_i < 0$: As soon as b_i's processing time is finished, it has to wait $-\triangle_i$ time owing to the fact that the job at the second track position blocks the departure of b_i. Hence, $W_i = -\triangle_i$. This is called crossing.
c. $\triangle_i = 0$: No idle time of second track position or waiting time for b_i.

One can see a block example where $m = 2$ and base job and first child job begins at the same time in Figure 3.3.

Block makespan, P_i, is dependent on whether a block is simple or not. If it is simple, then it equals to the processing time of the base job. Otherwise, it has a parent-child relationship. In this case, P_i is determined by the value of $\triangle_i$. Provided that $\triangle_i \geq 0$, P_i is equal to p_{b_i}. Otherwise, P_i equals to sum of p_{b_i} and W_i.

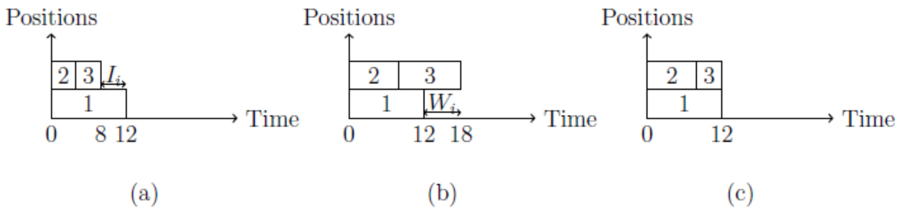

Figure 3.3. Various Block Schedules

SLA satisfaction is crucial and its breach necessitates penalty. Number of demanded vehicles vary by time so it results in a temporal SLA schedule. An instance for this can be seen in Figure 3.4. In this depiction, x-axis indicates time points whereas y-axis states the number of vehicles at t, that is, SLA_t.

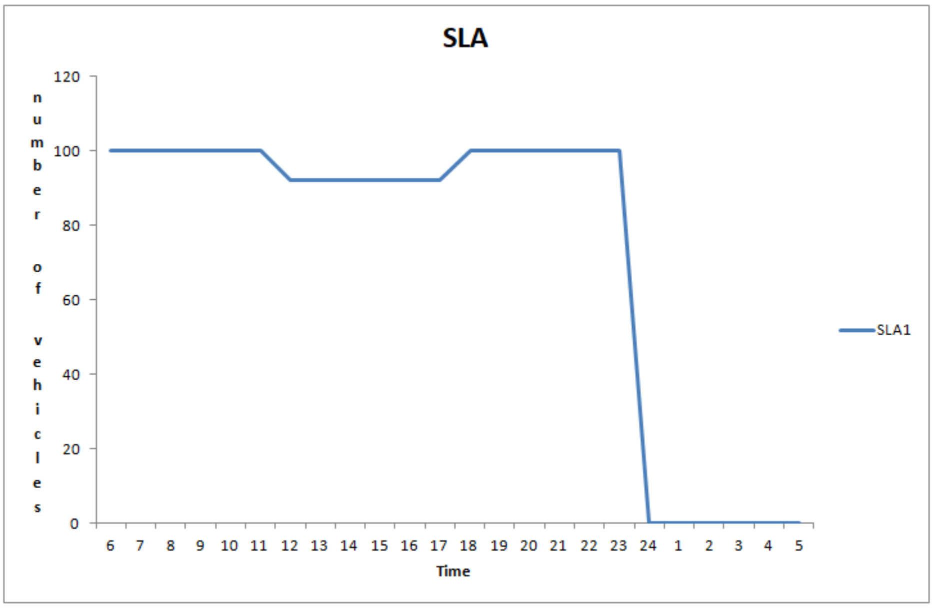

Figure 3.4. SLA Schedule Example

Maintenance departments make use of spare vehicles so as to relieve the effect of unscheduled breakdowns and planned maintenance on SLA. Spare vehicles are number of vehicles that are not required for maximum service. They can be measured by operating spare ratio (OSR). It is the ratio of number of spare vehicles to vehicles that are used in a peak service [60]. For example, consider an agency that has 100 vehicles, keep 80 of them are needed in the peak service and the remaining 20 are kept for possible breakdowns. The spare ratio for the agency is 20 / 80 or 25%. The spare ratio changes from one system to another. Oregon public maintenance department [61] uses 15% whereas Metrofleet [62], which is a Washington train service provider, utilizes 20%. Generally rail transit systems utilize OSR ranging from 10% to 30% that depends on the age and condition of vehicles, vehicle fleet size, and the effectiveness of the maintenance program [63]. OSR can be considered as a measure of distance of peak SLA demand to total number of vehicles. We show an example of it in Figure 3.5. In the figure, dotted line denotes the number of vehicles while solid line corresponds to SLA. OSR is the ratio of spare vehicles and peak SLA value which is shown at the end of lower part of the arrow. When we consider spare vehicles in a temporal fashion, the number of available spare vehicles at each time changes as a result of distance between upper and lower lines.

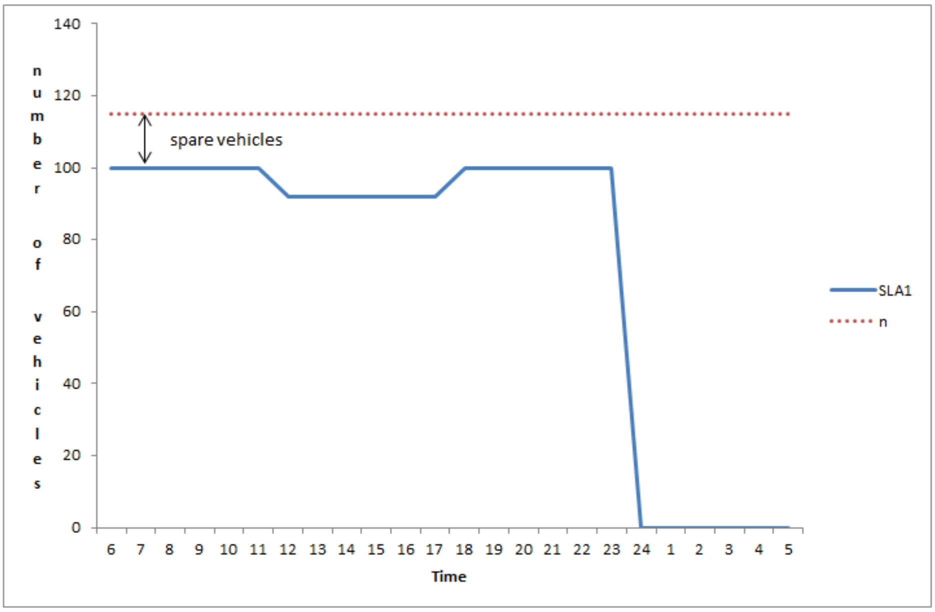

Figure 3.5. Spare Vehicles and SLA

Each vehicle has regular maintenance cycles because it is performed at certain mileages. As an example, at every x km, a vehicle must be maintained. Yet, a maintenance can be processed within interval of $x + -10\%$ of x. If one assumes that, on average a vehicle runs y km/hr, we can turn lower and upper limits of the interval into hour by dividing them to y. Lower and upper limits of maintenance intervals depend on both the vehicle and the maintenance type. Executing maintenance too early leads to lots of excessive maintenance in total. Besides, it fails to minimize unused km's till maintenance, which is a common objective in aircraft related papers ([24], [25], [27]). On the contrary, we cannot surpass the upper limit as it yields the rolling stock out of service. In order to avoid from that situation, we define a *due time* which is the time that a vehicle should be taken to maintenance. Because once that time is passed, we are susceptible to a danger that we might violate a latest maintenance start time. Hence, we aim at calling vehicles from the service as closest extent to it. An example of the maintenance interval is shown in Figure 3.6.

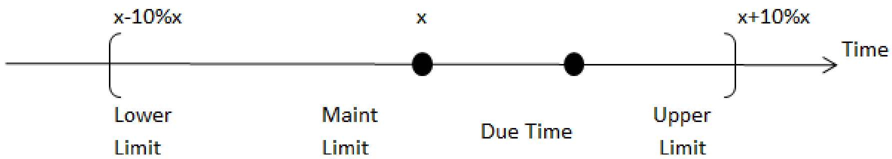

Figure 3.6. Preventive Maintence Interval and Scheduling Due time

In real life, a planner builds a schedule for a current scheduling period (SP). Yet, some events might occur at previous time that disrupted latest schedule. In our case, these events are corrective maintenance jobs. In order to handle these deteriorating effects, we need a reactive schedule. Besides, even if there is no corrective job, we should consider that upcoming preventive jobs could alter the current schedule. Addressing this issue, we utilize a *rolling horizon*. To do so, we implement current solution, update job data for consequent time periods, and perform a reschedule again, and so forth [64]. In other words, we employ a rescheduling strategy in which we perform a periodic update with complete regeneration [65]. In our work, we ascertain implementation time of the schedule as one day and decision period or SP as seven days, which is used in [12]. It means that we make a reschedule on every day from scratch with updated data.

In detail, we determine a maintenance list that is comprised of preventive jobs and corrective jobs at the beginning of SP. We name it as *critical jobs*. We only deal with critical jobs and rest of vehicles are not in the maintenance list. So, they are uncritical. In addition, if any job operation begins on previous day but not have finished yet, we continue its processing on current day. We specify them as *ongoing jobs*. We schedule critical jobs according to some priority. During rolling horizon implementation, we only execute first day of the schedule, slide the SP by one day and update critical jobs with respect to first day realization and new SP. Changes over SP is depicted in Figure 3.7. Arrows denote corrective job arrivals. First roll ranges from day $t-1$ to day $t+6$. When we produce a schedule, we defer roll one day in time. Thus, second roll begins from day t and ends in $t+7$. In this roll, we also take corrective jobs that arrive in day $t-1$ into account. From day t's point of view, current SP covers day t to $t+7$. Current schedule and its relation with past day is shown in Figure 3.8. Jobs 1,11,7,5,9 and 2 are corrective jobs whereas rest of them are preventive jobs. Shaded parts of 7 and 5 are completed on the very beginning of the current schedule. Afterwards, we consider corrective jobs 2 and 9 which are shown with arcs and comes on previous day. We allocate them to appropriate positions in current SP.

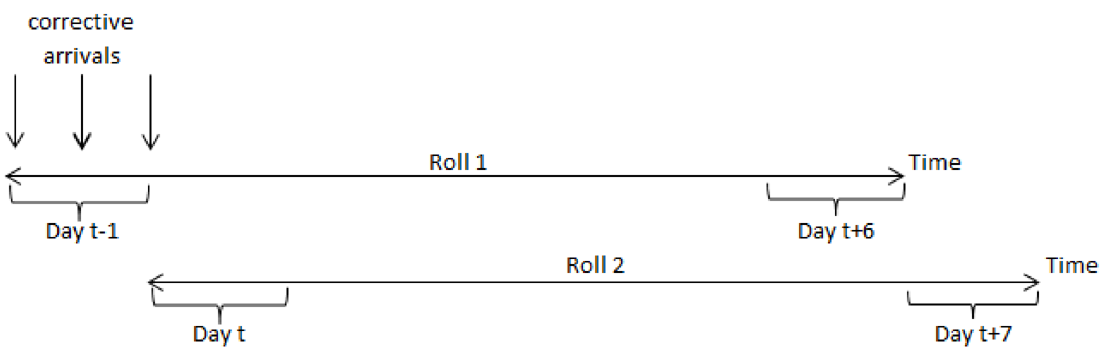

Figure 3.7. Rolling Horizon Scheme

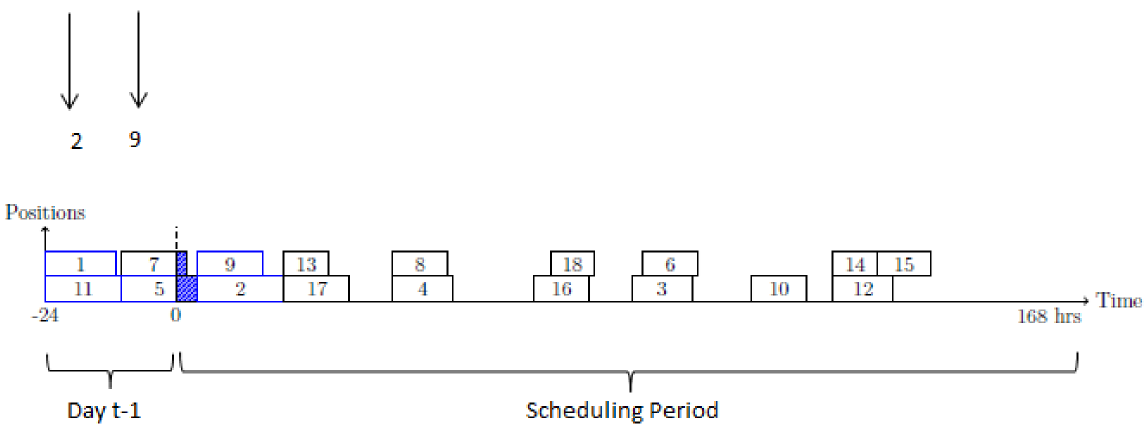

Figure 3.8. Weekly Schedule based on Critical Jobs

If we extend two consecutive day example which is mentioned above, it can be displayed as in Figure 3.9. Here, we have already executed first two days. Current SP corresponds to roll 3 and covers day $t + 1$ to $t + 8$. In this work, we propagate this scheme till end of the planning horizon.

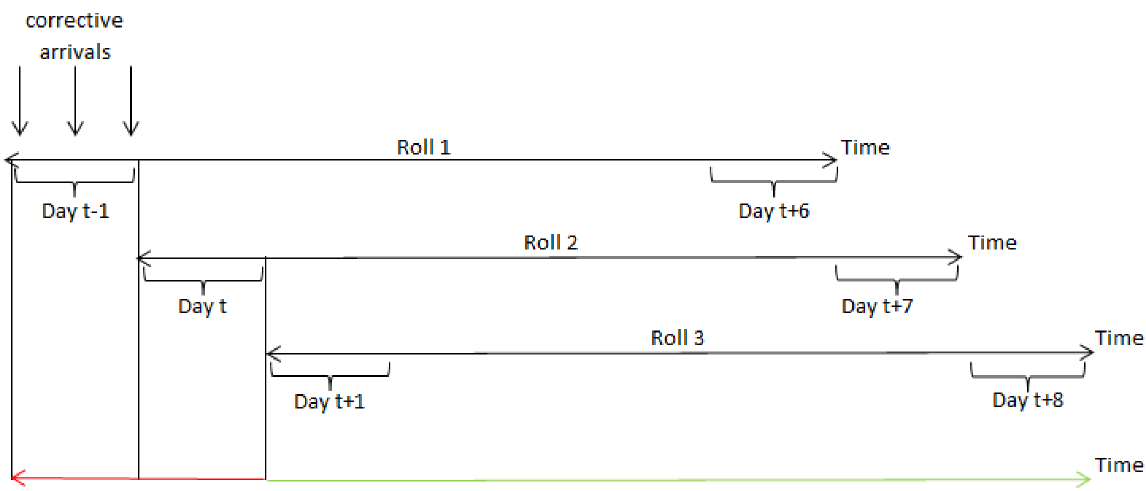

Figure 3.9. Extended Rolling Horizon Scheme

If we only focus on preventive jobs, critical jobs consist of preventive jobs. In Figure 3.10, since there are no corrective jobs, 2 and 9, which we mention in Figure 3.8, they do not push preventive jobs 13 and 17 further in time. So, they can be scheduled right after ongoing jobs. Besides, rolling horizon scheme is the same but we do not take arrows in Figures 3.7 and 3.9 into account.

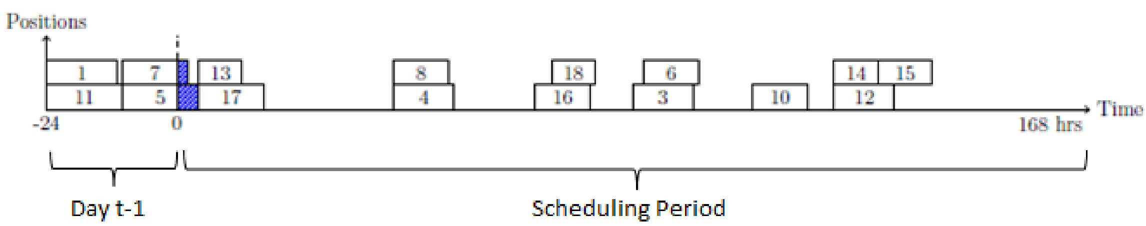

Figure 3.10. Weekly Schedule based on Preventive Jobs

Fulfillment of SLA depends heavily on maintenance scheduling. In preventive maintenance, the number of vehicles at the hangar track diminishes available number of vehicles for service, i.e., net capacity. At first, we study the case in which we only take preventive jobs into account. The relationship between parking capacity and SLA can be seen in Figure 3.11. If we look at the figure, solid line represents the capacity, m, that remains the same during the planning horizon. Let n denote fleet size. Dotted line stands for net capacity, that is, $n - SLA$ assuming that there is no out of service preventive jobs at the beginning of SP. So, it is a function of SLA and seems as if we looked at the SLA in reverse direction. Dyed region is the net parking capacity such that its value at each time instant is the minimum of dotted and solid line values. Because we do not want to violate SLA and parking capacity.

Second, we extend first case by attaching corrective jobs and out of service preventive jobs at the beginning of SP. They are called as out of service jobs. The case is shown in Figure 3.12. If we look at the picture, dotted line states a little bit different meaning than before. Let us denote n' as the number of *operational vehicles*, which is the number of vehicles that are not out of service, such that $n' < n$. Since we satisfy SLA from operational vehicles, number of vehicles remaining after SLA satisfaction is depicted as dotted line. Whenever it goes below zero in y-axis, we encounter SLA violation which indicates that we could not manage to satisfy SLA at the particular point. In the figure, we observe such situations in [6-11] and [18-20] time intervals. We also append a dashed line that portrays number of out of service jobs in our parking track. As we process them, it declines to zero. So, it goes till a predetermined point (23) parallel with dyed region. After that point, dyed region deviates from the dashed line due to the fact that we assign preventive jobs that are not out of service. Consequently, this case is reduced to first case. It can be seen from time 24 to 5 in the picture. As a summary, dyed area is not only affected by solid straight and dotted lines, but also by dashed line, because out of service jobs tend to be scheduled at the earliest times of SP.

Given n vehicles, m track locations in a single dead-end track, maintenance job list, preventive job maintenance intervals that comprises earliest and latest maintenance start times with a due time in between them, maintenance processing times, breakdown time of corrective jobs and the SLA, the problem is to find a schedule on a single dead-

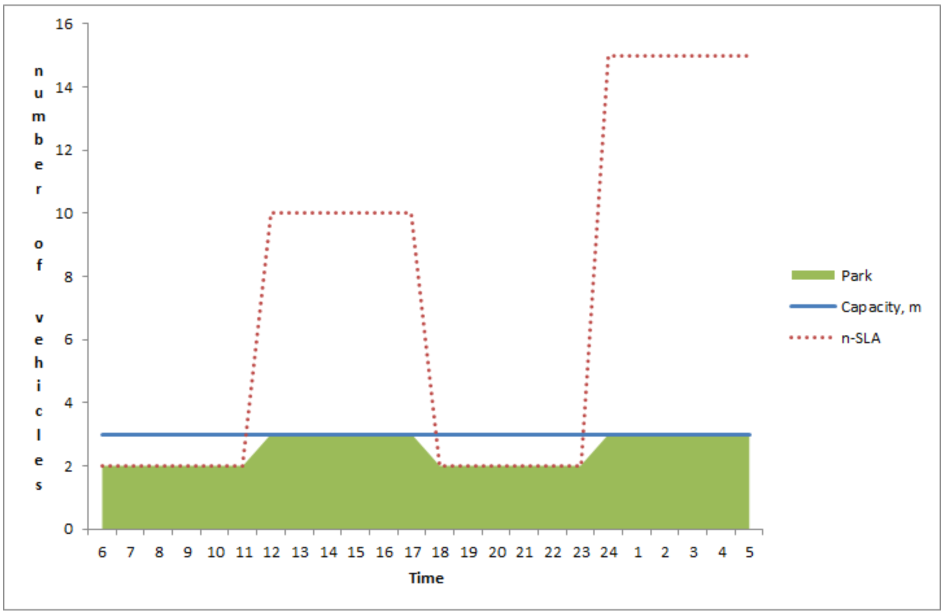

Figure 3.11. Relationship between SLA and Parking Capacity without Corrective Jobs

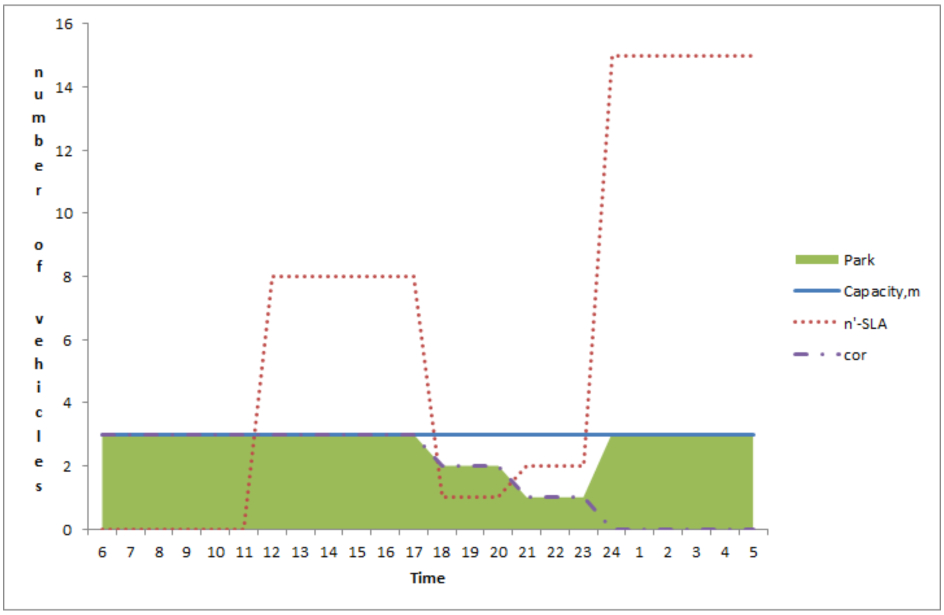

Figure 3.12. Relationship between SLA and Parking Capacity with Corrective Jobs

end track that minimizes deviations of maintenance begin time of preventive jobs from their due time, weighted tardiness of corrective jobs and penalized SLA and latest maintenance start point breaches.

Finally, we prove complexity of the problem. Simplest version of the problem is the minimum makespan of all vehicles at hangar parking track. The following Theorem 3.1 shows that it is NP-Hard in strong sense in terms of optimization version.

Theorem 3.1. *P2 || LIFO track || Pmax problem is NP-Hard in strong sense.*

Proof. *3-Partition* problem reduces to P2 || LIFO track || Pmax problem. Let's write *3-Partition* problem as follows [66]:

$$\frac{b}{4} < a_j < \frac{b}{2} \quad j = 1, ..., 3t \tag{3.2}$$

$$\sum_{j=1}^{3t} a_j = tb \tag{3.3}$$

do there exist three element subset $S_i \subset 1, ..., 3t$ such that

$$\sum_{j \in S_i} a_j = b \quad i = 1, ..., t? \tag{3.4}$$

Reduce this problem into our problem in polynomial time by taking

$$p_j = a_j \qquad j = 1, ..., 3t \tag{3.5}$$

$$p_j = b \qquad j = 3t + 1, ..., 4t \tag{3.6}$$

$$\sum_{j=1}^{3t} p_j = tb \tag{3.7}$$

and assume $m = 2$ for track capacity. There exists a schedule with $Pmax = tb$ if and only if for each i block, we can find three jobs from 3.5 and assign them to second track position such that sum of their job processing time equal to b while allocating any job with $p_j = b$ as a base job. It is possible if and only if *3-Partition* has a solution. □

4. MATHEMATICAL FORMULATIONS AND THE SOLUTION APPROACH

In this chapter, we present MILP model that solves weekly vehicle maintenance scheduling problem. We discuss how we formulate and improve its time efficiency. We devise a polynomial time heuristic algorithm. Thereafter, we introduce two approaches to handle corrective jobs efficiently. They also utilize the MILP model but recommend extra adjustments considering anticipated stochastic arrivals in future time periods.

4.1. MILP Model

In any block, base job constitutes root job of the block. Inspiring from this, we open bins at each track position as number of critical jobs and call each bin as *base*. We try to allocate a job into it in the model. Once a job is allocated, it becomes the base job of the block. So in our model, we will define each block with reference to its base number. According to job assignments to bases, we form an *assignment plot*. In this plot, each box means a specific base and numbers in bases point out allocated job indices to relevant base while empty boxes indicate no job allocation to that base. Should there exists a parent-child relationship, we show it via an undirected arc. An example schedule and its assignment plot can be seen in Figure 4.1. In the figure, we observe some portion of the schedule in part a whereas part b demonstrates its assignment plot. In part a, 1 is the base job and jobs 5,4 are its children. Hence, we represent this relationship in part b through undirected arcs between 1 and 4,5.

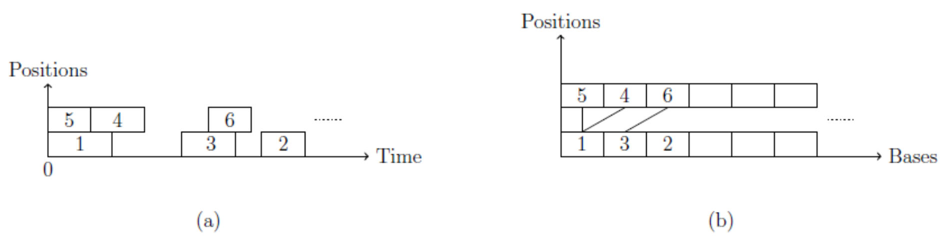

Figure 4.1. Schedule and its Assignment Plot

4.1.1. Assumptions

(i) Scheduling period covers only *critical jobs*.
(ii) Each vehicle has unit length.
(iii) Each vehicle has one combined maintenance job during the scheduling period.
(iv) Maintenance hangar comprises of one dead-end track so trams can only approach it according to LIFO.
(v) Jobs are taken from customer service to maintenance track immediately.
(vi) Preemption is not allowed.

4.1.2. Sets

J Vehicle maintenance jobs, $j \in J$

PM Preventive jobs

U Corrective jobs

CR Critical jobs, $CR = PM \bigcup U$

NC Noncritical jobs, $NC = J \backslash CR$

T Time Horizon, $t \in T$

C Track positions, $i \in C$

B Bases, $b \in B \quad s.t \quad |B| = |CR||C|$

OB Bases where ongoing jobs reside, $b \in OB$

OJ Ongoing jobs, $j \in OJ$

B_i Bases at track i s.t $B_i = \{(i-1)|CR|, ..., i|CR| - 1\}$

4.1.3. Parameters

SLA_t Number of required vehicles to satisfy SLA at time t

Y_j^{min} Begin time of maintenance interval of job j

Y_j^{max} End time of maintenance interval of job j

p_j Maintenance duration of job j

$r_j^{min} = \max\left(Y_j^{min}, 0\right)$

$r_j^{max} = \max\left(Y_j^{max}, 0\right)$

d_j Due time of job j

M Large number

ϵ Penalty per one unit violation of $d_j, \quad j \in U$

σ Penalty per one unit violation of $Y_j^{max}, \quad j \in PM$

γ Penalty per one unit violation of SLA schedule

4.1.4. Decision variables

y_{ab} 1 if a is a child of b where $b \in B_i$, $a \in B_{i+1}$, else 0

z_{ab} 1 if $F_a > S_b + \sum_{j \in CR} x_{bj} p_j$ given that $y_{ab} = 1$, else 0

w_b 1 if $F_b = S_b + \sum_{j \in CR} x_{bj} p_j$, else 0

x_{bj} 1 if job j is assigned to base b, else 0

T_j Tardiness time of job, $j \in PM$

Time between breakdown of a vehicle and its maintenance start time, $j \in U$

E_j Earliness time of job j

e_{jt} 1 if job j exits from maintenance hangar at the end of time period t, else 0

b_{jt} 1 if job j's maintenance work begins at time period t, else 0

a_{jt} 1 if job j is at maintenance hangar at time period t, else 0

o_{jt} 1 if job j is out of service at time t, else 0

np_j 1 if job j is delayed to next period, else 0

$\triangle SLA_t^-$ Number of trams fails to satisfy SLA_t demand

$\triangle Y_j^{max+}$ Number of time units that Y_j^{max} is violated

P_b Park time of base b

S_b Start time of base b

F_b Finish time of base b

L_{bj} Linearization of $x_{bj} F_b$

B_{bj} Linearization of $x_{bj} S_b$

S_j Maintenance start time of job j

C_j Exit time of job j from maintenance hangar

4.1.5. Mathematical Model

$$Min \sum_{j \in PM} \left(T_j + E_j + \sigma \triangle Y_j^{max+}\right) + \epsilon \sum_{j \in U} T_j + \gamma \sum_{t=1}^{|T|} \triangle SLA_t^- \tag{4.1a}$$

s.t.

$$\sum_{b \in B} x_{bj} + np_j = 1 \qquad j \in CR \tag{4.1b}$$

$$\sum_{j \in CR} x_{bj} \leq 1 \qquad b \in B \tag{4.1c}$$

$$\sum_{j \in CR} x_{bj} p_j \leq P_b \qquad b \in B \tag{4.1d}$$

$$\sum_{b \in B_i} y_{ab} \leq 1 \qquad a \in B_{i+1}, i{<}|C| \tag{4.1e}$$

$$y_{ab} \leq \sum_{j \in CR} x_{bj} \qquad b \in B_i, a \in B_{i+1}, i{<}|C| \tag{4.1f}$$

$$y_{ab} \leq \sum_{j \in CR} x_{aj} \qquad b \in B_i, a \in B_{i+1}, i{<}|C| \tag{4.1g}$$

$$x_{aj} \leq \sum_{b \in B_i} y_{ab} \qquad j \in CR, a \in B_{i+1}, i{<}|C| \tag{4.1h}$$

$$\sum_{j \in CR} x_{aj} \geq \sum_{j \in CR} x_{(a+1)j} \qquad a \in B_i \tag{4.2a}$$

$$F_b \leq S_b + \sum_{j \in CR} x_{bj} p_j \qquad b \in B_i, i = |C| \tag{4.2b}$$

$$F_b \leq S_b + \sum_{j \in CR} x_{bj} p_j + M \left(\sum_{a \in B_{i+1}} y_{ab} \right) \qquad b \in B_i, a \in B_{i+1}, i < |C| \tag{4.2c}$$

$$F_b \geq S_b + P_b \qquad b \in B \tag{4.2d}$$

$$F_b \geq \sum_{j \in PM | Y_j^{min} > 0} x_{bj} Y_j^{min} + P_b \qquad b \in B \tag{4.2e}$$

$$z_{ab} \leq y_{ab} \qquad b \in B_i, a \in B_{i+1}, i < |C| \tag{4.2f}$$

$$F_b \geq F_a - M(1 - y_{ab}) \qquad b \in B_i, a \in B_{i+1}, i < |C| \tag{4.2g}$$

$$F_b \leq F_a + M(1 - z_{ab}) \qquad b \in B_i, a \in B_{i+1}, i < |C| \tag{4.2h}$$

$$F_b \leq S_b + \sum_{j \in CR} x_{bj} p_j + M(1 - w_b) \qquad b \in B_i, i < |C| \tag{4.2i}$$

$$\sum_{a \in B_{i+1}} z_{ab} + w_b = 1 \qquad b \in B_i, i < |C| \tag{4.2j}$$

$$S_a \leq S_b + \sum_{j \in CR} x_{bj} p_j - 1 + M(1 - y_{ab}) \qquad b \in B_i, a \in B_{i+1}, i < |C| \tag{4.2k}$$

$$S_b \leq S_a + M(1 - y_{ab}) \qquad b \in B_i, a \in B_{i+1}, i < |C| \tag{4.2l}$$

$$S_{b+1} \geq F_b \qquad b \in B_i, i \in C \tag{4.2m}$$

$$B_{bj} \leq S_b \qquad b \in B, j \in CR \tag{4.2n}$$

$$B_{bj} \geq S_b - M(1 - x_{bj}) \qquad b \in B, j \in CR \tag{4.2o}$$

$$B_{bj} \leq M x_{bj} \qquad b \in B, j \in CR \tag{4.2p}$$

$$L_{bj} \leq F_b \qquad b \in B, j \in CR \tag{4.2q}$$

$$L_{bj} \geq F_b - M(1 - x_{bj}) \qquad b \in B, j \in CR \tag{4.2r}$$

$$L_{bj} \leq M x_{bj} \qquad b \in B, j \in CR \tag{4.2s}$$

$$S_j \geq Y_j^{min}(1 - np_j) \qquad j \in PM | Y_j^{min} > 0 \tag{4.2t}$$

$$S_j = \sum_{b \in B} B_{bj} \qquad j \in CR \tag{4.2u}$$

$$C_j = \sum_{b \in B} L_{bj} \qquad j \in CR \tag{4.2v}$$

$$T_j \geq S_j + |T| - d_j - |T|(1 - np_j) \qquad j \in CR \tag{4.3a}$$

$$E_j \geq d_j - S_j - |T|np_j \qquad j \in PM \tag{4.3b}$$

$$T_j - \triangle Y_j^{max+} \leq Y_j^{max} - d_j \qquad j \in PM \tag{4.3c}$$

$$\sum_{t \in T} te_{jt} = C_j \qquad j \in CR \tag{4.3d}$$

$$\sum_{t \in T} a_{jt} = C_j - S_j \qquad j \in CR \tag{4.3e}$$

$$\sum_{t \in T} b_{jt} + np_j = 1 \qquad j \in CR \tag{4.3f}$$

$$\sum_{t \in T} e_{jt} + np_j = 1 \qquad j \in CR \tag{4.3g}$$

$$\sum_{t \in T} tb_{jt} = S_j \qquad j \in CR \tag{4.3h}$$

$$a_{jt} \leq 1 - \sum_{l=1}^{t} e_{jl} \qquad j \in CR, t \in T \tag{4.3i}$$

$$a_{jt} \leq \sum_{l=1}^{t} b_{jl} \qquad j \in CR, t \in T \tag{4.3j}$$

$$\sum_{j \in CR} 1 - (a_{jt} + o_{jt}) + \triangle SLA_t^- \geq SLA_t - |NC| \qquad t \in T \tag{4.3k}$$

$$o_{jt} = 1 - \sum_{i=r_j^{min}}^{t} b_{ji} \qquad j \in PM, t \geq r_j^{max} \tag{4.3l}$$

$$o_{jt} = 1 - \sum_{i=r_j^{min}}^{t} b_{ji} \qquad j \in U, t \in T \tag{4.3m}$$

$$x_{bj} = 1 \qquad b \in OB, j \in OJ \tag{4.3n}$$

$$B_{bj} = 0 \qquad b \in OB, j \in OJ \tag{4.3o}$$

$$L_{bj} \geq p_j \qquad j \in OJ \tag{4.3p}$$

$$a_{jt}, e_{jt}, b_{jt}, np_j = \{0,1\} \qquad j \in CR, t \in T \tag{4.4a}$$

$$x_{bj} = \{0,1\} \qquad j \in CR, b \in B \tag{4.4b}$$

$$y_{ab} = \{0,1\} \qquad b \in B_i, a \in B_{i+1}, i<|C| \tag{4.4c}$$

$$S_b, F_b, P_b \geq 0 \qquad b \in B \tag{4.4d}$$

$$T_j, C_j, S_j \geq 0 \qquad j \in CR \tag{4.4e}$$

$$B_{bj}, L_{bj} \geq 0 \qquad b \in B, j \in CR \tag{4.4f}$$

The objective function (4.1a) minimizes sum of tardiness, earliness, penalize Y_j^{max} violations for planned jobs, tardiness of corrective jobs and failure of SLA_t satisfaction. (4.1b) states that base assignment is performed for each job as long as $np_j = 0$. (4.1c) accounts for at most one job assignment to a given base. (4.1d) yields a lower bound for a block makespan in terms of base job j. Each child has at most one parent and this is satisfied by (4.1e) constraints. (4.1f) - (4.1h) constraints are used for linking parent-child variable y_{ab} to x_{bj}. A base number assignment for each job does not matter once they belong to same track position i and preserve their order among themselves. This issue brings about a symmetry. (4.2a) constraint is used to eliminate symmetry at each track position i. (4.2b)-(4.2e) constraints ensure base finish time. In detail, (4.2b) and (4.2c) determine F_b given that block is a simple block. (4.2d)-(4.2e) identify a lower bound for F_b regarding maximum of S_b and Y_j^{min} of assigned job. (4.2f) clarifies that z_{ab} could be employed once $y_{ab} = 1$. (4.2g) guarantees that child of a block finishes before its parent. (4.2h) - (4.2j) make sure that F_b is maximum of child block's finish time and $S_b + p_j$. (4.2k) specifies that child begin time is no later than $S_b + p_j - 1$. (4.2l) assures that begin time of a child base cannot start earlier than its parent. (4.2m) identifies the following base can start after its immediately before base is finished. (4.2n)-(4.2p) are linearization constraints of $x_{bj}S_b$. (4.2q)-(4.2s) are linearization of $x_{bj}F_b$. (4.2t) certifies that job begin time must be after than its ready time provided that $np_j = 0$. (4.2u) - (4.2v) deal with maintenance job start and finish time, respectively.

(4.3a) and (4.3b) are tardiness and earliness constraints, respectively and they are reliant on np_j value. (4.3c) identifies an upper bound for a tardiness and is a *soft constraint* for Y_j^{max+} point violation. (4.3e) makes sure that sum of a_{jt} times of a given job must be equal to its stay time at maintenance hangar. (4.3d) links exit time of a given job to maintenance finish time of the base that this specific job is assigned to. On the condition that $np_j = 0$, (4.3f) guarantees the initialization of maintenance decision whereas (4.3g) assures that maintenance of each job must end somewhere in the scheduling period. Otherwise, they cannot be scheduled on hangar. (4.3h) links maintenance start time period of a job to maintenance begin time of the base that this specific job is assigned to. (4.3i) states that a job cannot be at maintenance hangar after its maintenance is finished. (4.3j) specifies that a job cannot be at maintenance center before its maintenance starts. (4.3k) is a *soft constraint* that satisfies SLA demand. A vehicle cannot be at passenger service if it is in $a_{jt} = 1$ or $o_{jt} = 1$ state. Having $\triangle SLA_t^-$ larger than zero is penalized in objective function. (4.3l) makes sure that if the job maintenance has not begun till $t \geq r_j^{max}$, job remains out of service state. The same point is employed at (4.3m) where $t \geq r_j^{min}$. (4.3n) - (4.3p) schedule remaining maintenance process times of ongoing jobs at pertinent tracks. (4.4a) - (4.4c) are binary variables while (4.4d) - (4.4f) are continuous variables. Finally, big M is selected as $|T|$ as it is the tightest parameter that all variables are defined until this value.

4.2. Model Improvements

4.2.1. Optimality Conditions

Definition 4.2.1. P_{asc} is a set where jobs in CR are ordered in nondecreasing order of their processing times

wk_b: Begin time of the current week

m: Number of track locations

wk_e: End time of the current week

beg_i: $(i-1)\,|CR|$, $i \in C$

end_i: $i|CR| - 1$, $i \in C$

J_{tard}: $\left\{j \in CR | j \in U \bigcup j \in PM \quad s.t \quad Y_j^{max} \leq wk_b\right\}$

NP: $\left\{j \in PM | \quad Y_j^{max} \geq wk_e\right\}$

p_{max}: $max_{j \in CR} p_j$

PW: $\left\{j \in CR | \quad Y_j^{min} \leq wk_b + p_{max}\right\}$

IB: $\{b \in B | b = 0, .., beg_k\}$ where $k = min\,(m, |J_{tard}|)$

pos_b: Least number of assigned jobs till base b beginning from base 0 where $pos_b = (b - beg_i + 1) + (i - 1)$

c_{ub}: Maximum possible number of children for any base $b \in B_i, i < m$ based on processing times

K_b: Upper bound value for b's number of children such that $K_b > 0$ and $K_b = |CR| - pos_b$

Lemma 4.1. *Optimum schedule is composed of distinct successive blocks.*

Proof. Assume a moment in time frame such that both track positions are empty given that $m = 2$. Then while a base job is parked on first track position, we obtain either a block with a parent-child relationship or a simple block. After P_i time, the vehicle(s) at the block leaves the track. As a result, we again have a situation that both track positions are unoccupied. Therefore, all jobs can be parked into track as successive blocks. Since optimum schedule is a feasible schedule, it is also formed from successive blocks. □

Lemma 4.2. *There exists an optimal schedule where every $b \in B_i, i < m$ can have a child job at bases ranging from beg_{i+1} to $beg_{i+1} + K_b - 1$.*

Proof. As a result of inequality (4.2a) in Model, we deduce that at least pos_b jobs are allocated to the track where $pos_b = (b - beg_i + 1) + (i - 1)$. First term in parenthesis states number of jobs have been assigned till $b \in B_i$ beginning from beg_i whereas second parenthesis explains that b is filled as a child of any $b' \in B_{i-1}$ given that $i > 1$. So, there remains $K_b = |CR| - pos_b$ unassigned jobs which are candidate to be allocated as a children of b. It follows that if $K_b > 0$, children base numbers begin from beg_{i+1} and end with $beg_{i+1} + K_b - 1$. □

Lemma 4.3. *There exists an optimal schedule where every $b \in B_i, i < m$ can have c_{ub} children at the most adverse case.*

Proof. At the most extreme case, assume at any base $b \in B_i, i < m$, there exists a job j such that $p_j = p_{max}$. Besides, its children are allocated to the track in the nondecreasing order of their p_j times in such a way that it follows the index order of P_{asc} set. We thereby define c_{ub} as follows:

$$\sum_{i=1}^{c_{ub}-1} p_i < p_{max}$$
$$\sum_{i=1}^{c_{ub}} p_i \geq p_{max}$$

□

Corollary 4.4. *Lemma 4.2 and 4.3 lead to the conclusion that number of children for any base $b \in B_i, i < m$ equal to min (K_b, c_{ub}) in an optimal schedule.*

Lemma 4.5. *There exists an optimal schedule where number of bases at each track $i > 1$ is at most $|CR| - (i - 1)$.*

Proof. Every base $b \in B_i, i > 1$ is a child of any base $b' \in B_{i-1}$. It means that $i - 1$ jobs are already assigned till track i. Thereby we can allocate at most $|CR| - (i - 1)$ bases at track position i for possible job assignments. □

Lemma 4.6. *For jobs $j, j' \in J_{tard}$ such that $p'_j \geq p_j$, if they form a parent-child relationship in the same block, then there exists an optimal schedule where j is a child job of j'.*

Proof. Let we have two jobs $j, j' \in J_{tard}$ such that $p'_j > p_j$ in B_i and $m = 2$. Assume that we have an optimal schedule S where j is a base job, j' is a child job, $S_{j'} \geq S_j, W_i > 0$ in block B_i. Note that $W_i > 0$ always occurs as $p'_j > p_j$. According to $S_{j'}$ and S_j values in S, we define $dist = S_{j'} - S_j$. If we interchange j and j', we have a schedule S' where j' is a parent job of j such that $S_j \geq S_{j'}$. Since both jobs are displaced $dist$ in time and they are identical, Z value is not affected by new begin times of these jobs. Yet, there may be free space changes between these schedules. Let we introduce a new term $space = \{(C(S) - C(S'))\, m\}$. Two cases can occur in S' with respect to $\triangle_i$ value:

- Case 1: $I_i > 0$ in S'

  If $dist = 0$, then we obtain a schedule S' where $C(S) = C(S')$ thus $space = 0$. Since $I_i > 0$, we attain a free space to schedule other jobs in CR. One can see an example of it in Figure 4.2. If not, both $space > 0$ and I_i creates an ample space to schedule other remaining jobs. An instance of it is depicted in Figure 4.3. So the change results in $Z'_S \leq Z_S$ with regard to both $dist$ values which means that we obtain a schedule at least as good as S.

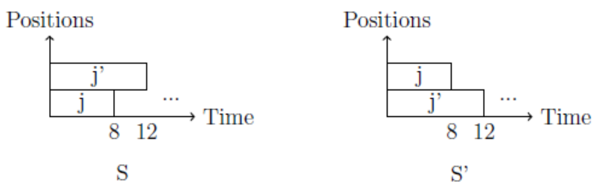

Figure 4.2. Case 1 Change in Parent-Child Relationship when $dist = 0$

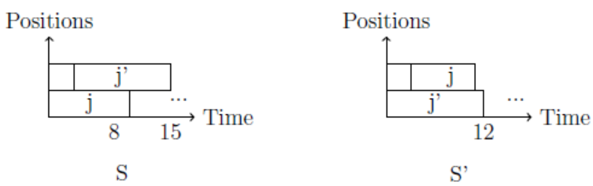

Figure 4.3. Case 1 Change in Parent-Child Relationship when $dist > 0$

- Case 2: W_i is diminished in S'

  We get a schedule S' in which $C(S') < C(S)$. This results in *space* > 0 which increase free space as we mentioned in Case 1. Thus, S' is at least as good as S. See Figure 4.4 to see an instance of this issue.

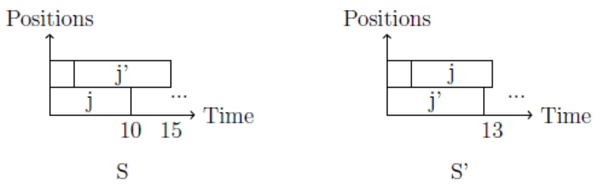

Figure 4.4. Case 2 Change in Parent-Child Relationship

□

Lemma 4.7. *In an optimal schedule,* $k = min\,(m, |J_{tard}|)$ *of jobs* $j \in PW$ *are allocated to blocks* $b \in IB$ *unless there exists an ongoing job from previous day*

Proof. J_{tard} jobs are out of service such that they are required to be scheduled as early as possible. Since we have m track capacity, first bases at each track are employed so as to minimize tardiness of jobs in J_{tard}. Yet, we cannot guarantee that all of them are assigned to first bases as $\exists j \in PW, j \notin J_{tard}$ such that it can also be located in these bases. Provided that $|J_{tard}| < m$, $|J_{tard}|$ of them are employed. Else, we utilize m as we have more J_{tard} jobs than capacity. Therefore, number of such bases are $min\,(m, |J_{tard}|)$ and these bases are indicated by IB. Should there exists a job $j \in OJ$, it fills one of IB blocks so this allocation could not happen. □

4.2.2. Model II

With a further analyze, we update the MILP model presented in Section 4.1.5. The changes and their proofs are given below.

- We remove $F_b \geq \sum_{j \in CR} x_{bj} Y_j^{min} + P_b$ constraint (4.2e) and insert

$S_b \geq \sum_{j \in PM | Y_j^{min} > 0} x_{bj} Y_j^{min}$ instead of it.

This constraint specifies begin time of the block provided that $x_{bj} = 1$. We come into conclusion due to preliminary computational experiment results that this constraint diminishes computational time more if we replace old one with this.

- We replace (4.2b) constraint with $L_{bj} \leq B_{bj} + x_{bj} p_j$, which is a disaggregation of it.
- We substitute proposition 4.8 equalities for (4.3e), (4.3i) and (4.3j).

  Proposition 4.8.

$$a_{jt} = \sum_{l=r_j^{min}}^{t} b_{jl} \qquad r_j^{min} < t < r_j^{min} + p_j, \quad j \in CR$$

$$a_{jt} = \sum_{l=r_j^{min}}^{t} b_{jl} - \sum_{l=r_j^{min}+p_j}^{t} e_{jl} \qquad t \geq r_j^{min} + p_j, \quad j \in CR$$

  are valid equalities.

  Proof. We know that job cannot exit from maintenance hangar if it has not started. Therefore, $a_{jt} = 0$ so long as $\sum_{l=r_j+p_j}^{t} b_{jl} = 0$. Once $b_{jt'} = 1$ at some t', then from t' till $C_j - 1$, $a_{jt} = 1$ because the job is still at hangar and not finished yet. Earliest time that e_{jt} can become 1 is $r_j + p_j$, so second equality starts right after that threshold. When $C_j = k$, then $e_{jk} = 1$ that leads to $a_{jt} = 0$ where $t \geq k$ which is in consistent with (4.3i). □

- We delete (4.1e) from our model.
- We replace (4.1g) - (4.1h) inequalities with following equalities:

$$\sum_{j \in CR} x_{aj} = \sum_{b \in B_i} y_{ab} \tag{4.5}$$

It is true because we can talk about a parent-child relationship only if there exists an assigned job in any of the candidate children bases.

- *Dimensionality Reduction I*

  We employ lemma 4.5 to decrease number of bases at every track. Regular candidate child job allocations can be seen on left side of Figure 4.5. Once lemma is employed, it results in candidate children assignments as shown in right part of the picture. Colored bases denote that it is impossible to locate any job there.

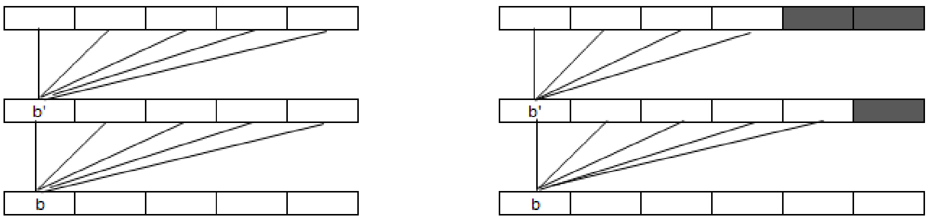

Figure 4.5. Possible Assignments to Track Locations with respect to Lemma 4.5

In order to diminish redundant relations between parent and children bases, we define following additional terms:

num_b: k th number of base at track $i < m$, $num_b = b - beg_i + 1$

num_a: k th number of base at track $i+1$, $num_a = a - beg_{i+1} + 1$

t_b = Threshold for possible number of children of base b such that $t_b = num_b c_{ub}$

$B_i^{'} : beg_i....end_i - (i-1)$

$B_{i+1}^{'} : beg_{i+1}....end_{i+1} - pos_b$

$B_{i-1}^{'} : beg_{i-1}....end_{i-1} - (num_a + i - 2)$

$B_i^{'}$ and $B_{i+1}^{'}$ follow from lemma 4.5 and lemma 4.2, respectively. $B_{i-1}^{'}$ is true because $end_{i-1} - ((i-1)-1)$ is different version of $B_i^{'}$. num_a is subtracted from that value because it is ath candidate children with respect to possible base at $i-1$.

By means of Corollary 4.4, for almost every parent-child relationship in the model, we select $b \in B_i^{'}$, $a \in B_{i+1}^{'}$ such that $num_a \leq t_b$. Furthermore, we define following equalities so as to rule out cases beyond these sets.

$$y_{ab} = 0 \qquad b \in B_i^{'}, a \in B_{i+1}^{'}, num_a > t_b \tag{4.6}$$

$$z_{ab} = 0 \qquad b \in B_i^{'}, a \in B_{i+1}^{'}, num_a > t_b \tag{4.7}$$

We present Figure 4.6 to clarify equations 4.6 and 4.7. We take $c_{ub} = 2$ and gray areas denote the locations that we cannot assign any job since pos_b jobs have already been assigned till any base $b \in B_i$. Minus sign is for bases that we are not eligible to assign any jobs. In left picture, $t_{b'} = 2$, consequently for third and fourth bases in $i = 3$, we put a minus sign. On the right, we take b'' as a root base. Since $t_{b''} = 4$, it seems that we could appoint a job with respect to equations 4.6 and 4.7 but as $K_b = 3$, we suspend three bases beginning from last base in $i + 1 = 3$, so the minus sign follows.

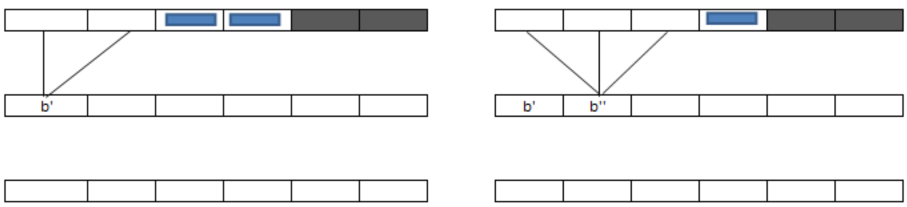

Figure 4.6. Possible Parent-Child Assignments based on b' and b''

Only for our newly defined equation 4.5, for every $a \in B_i^{'}, i > 1$ and we sum y_{ab} over $b \in B_{i-1}^{'}$ provided that there exists a job in a. In order to clarify this, for a given a in $i = 3$, four bases from b $\in B_{i-1}^{'}$ can be a parent of a. It is denoted in the left side of Figure 4.7. However, once we move onto a' in $B_i^{'}$, number of candidate parent bases diminishes to three as depicted in right side of Figure 4.7.

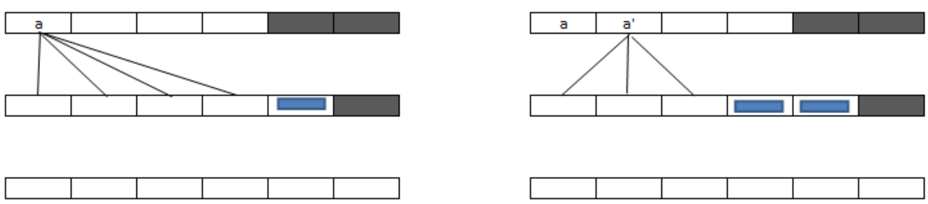

Figure 4.7. Possible Parent-Child Assignments based on a

- *Dimensionality Reduction II*

  Here, we intend to find $b', b^* \in B_i$ such that it becomes impossible to dispatch a job to bases $b' > b^*$. We update aforementioned pos_b according to track indices and name it as wn_b. We give equations for each base b as below where $A = \lceil \frac{num_b}{c_{ub}} \rceil$.

$$wn_b = num_b + A \qquad b \in B_i, i = 2 \qquad (4.8)$$

$$wn_b = num_b + A + \lceil \frac{A}{c_{ub}} \rceil \qquad b \in B_i, i = 3 \qquad (4.9)$$

The point in equation 4.8 is that b is parked above num_Ath base in $i-1$th track. Because if each parent has c_{ub} children at worst case, there must have been A parents which is the minimum number of bases at track location $i-1$ to reach base b in track location i. Using that logic, we extend it to equation 4.9. In equation 4.9, b in track 3 is above A_{th} base in track 2 which is also on the top of $\lceil \frac{A}{c_{ub}} \rceil_{th}$ base in track position 1. By means of this method, we can generalize these equalities as follows:

$$wn_b = num_b + \sum_{k=1}^{i-1} \chi_k$$

where $\chi_0 = A$ and $\chi_k = \lceil \frac{\chi_{k-1}}{c_{ub}} \rceil$ for a given $i > 1$

Afterwards, we determine b^* such that $wn_b^* \leq |CR|$ and $wn_{b^*+1} > |CR|$. Note that base numbers begin from 0. For instance, we have $|CR| = 8$ and $c_{ub} = 2$. So, we depict in Figure 4.8 that $b^* = 12$ is the last point for assignment. Because $num_{b^*} = 5$ which makes $wn_{b^*} = 8$ with reference to equation 4.8. It follows that from $b' = 13$ to $b'' = 14$, we fix $x_{bj} = 0$ which are shown as minus sign.

At track position 3, seven jobs have already been allocated till $b = 19$ with regard to equations 4.9. Hence, we do not allocate any jobs to bases with minus sign which are shown at Figure 4.9. Note that gray areas show the places where it is

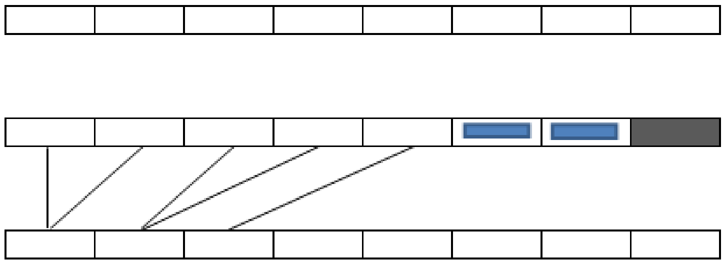

Figure 4.8. Maximum Job Assignments to Second Track Position Example

impossible to allot any kind of job. In conclusion, we derive equality 4.10 so as to prevent job allocations to these bases.

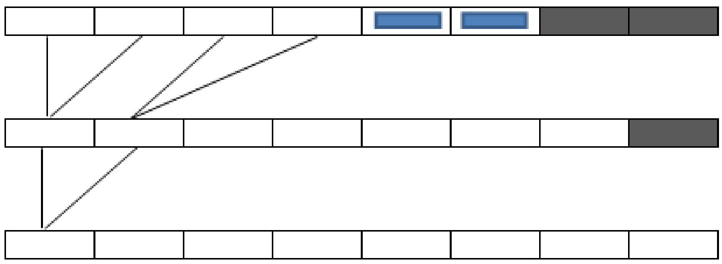

Figure 4.9. Maximum Job Assignments to Third Track Position Example

$$x_{bj} = 0 \qquad\qquad b \in B_i', b > b^*, j \in CR \tag{4.10}$$

- *Dimensionality Reduction III*

  In the worst case, we have already appointed wn_b jobs till block $b, b \in B_i$ due to Dimensionality Reduction II (equations 4.8 and 4.9). Hence, we can at most assign $|CR| - wn_b$ jobs as children of that particular base. Yet, we can build a child arc from that base till base $b', b' \in B_{i+1}$ such that $b' \leq t_b$ because of Dimensionality Reduction I. Therefore, we call this amalgam rule as Dimensionality Reduction III. For a depiction, in the left part of Figure 4.10, two base positions in the third track are candidate to be children of the base b. When we apply dimensionality reduction III for b, $wn_b = 5$ so $6 - 5 = 1$ base in third track position lefts to be a potential child candidate. It leads to right part of the picture.

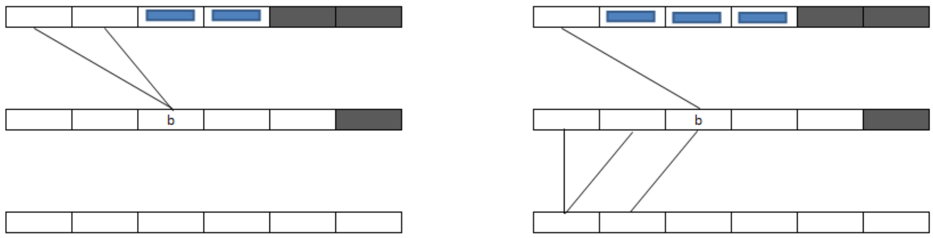

Figure 4.10. Possible Job Assignments to Track Position 3

We have shown techniques to diminish parent-child relations in our model. We give a simple instance in Figure 4.11 to see effects of these techniques. In picture a, as we modelled, all relations exist between parent-child bases. Once we apply Lemma 4.5 to decrease relations, our relation map results in picture b. Thereafter, we can employ dimensionality reduction I and II to come up with a relation map as picture c. All in all, we diminish number of relations from 17 to 7 and finally 3. Therefore, these reductions help us reduce problem size substantially.

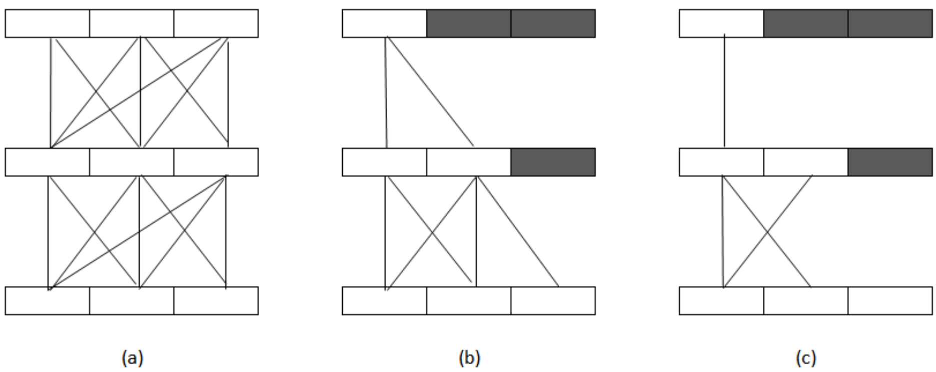

Figure 4.11. Dimensionality Reduction Example

- *Other inequalities*

Proposition 4.9. *The following equalities are valid:*

$$e_{jt} = 0 \qquad t<r_j^{min} + p_j, j \in CR \tag{4.11}$$

$$a_{jt} = 0 \qquad t<r_j^{min}, j \in PM \tag{4.12}$$

$$b_{jt} = 0 \qquad t<r_j^{min}, t \geq |T| - p_j, j \in PM \tag{4.13}$$

$$b_{jt} = 0 \qquad t \geq |T| - p_j \quad j \in U \tag{4.14}$$

$$o_{jt} = 0 \qquad t<r_j^{max}, j \in PM \tag{4.15}$$

Proof. Since earliest end time for a job j is $r_j + p_j$, e_{jt} values before that point can be set to zero like (4.11). In (4.12), earliest time that job can enter to track is r_j^{min}. So a_{jt} and b_{jt} values are fixed to zero prior to it. Furthermore, latest start for a job is $|T| - p_j - 1$. So, from that point onwards, $b_{jt} = 0$ for jobs in CR. Finally, (4.15) maintains that a preventive job cannot be out of service before r_j^{max}. □

We replace (4.3d) - (4.3h) inequalities in Model with preposition 4.10.

Proposition 4.10. *Following inequalities are valid.*

$$\sum_{t \geq r_j^{min} + p_j} te_{jt} = C_j \qquad j \in CR \tag{4.16}$$

$$\sum_{t \geq r_j^{min}} tb_{jt} = S_j \qquad j \in PM \tag{4.17}$$

$$\sum_{t \geq r_j^{min}} b_{jt} + np_j = 1 \qquad j \in PM \tag{4.18}$$

$$\sum_{t \geq r_j^{min} + p_j} e_{jt} + np_j = 1 \qquad j \in CR \tag{4.19}$$

Proof. This preposition depends on preposition 4.9. Since e_{jt} and b_{jt} values are zero before $r_j^{min} + p_j$ and r_j^{min}, respectively, deletion of these terms from model does not change anything. □

Proposition 4.11. *Given that $d_j \geq |T|$, $np_j = 1$ is a valid inequality.*

Proof. Since $d_j \geq |T|$, $T_j = 0$ is trivially correct. Yet, $E_j > 0$ in this case. In objective function, E_j is minimized. It follows that, start time of maintenance job must be as close as possible to d_j which is ensured if upkeep operation begins at next period. Thus, $np_j = 1$. □

Proposition 4.12. *Let us define following terms:*

$a_{dj} = |T| - d_j$

$b_{dj} = |T| - p_j - 1$

Given that $d_j > b_{dj}$ and $d_j - b_{dj} > a_{dj}$, then $np_j = 1$ is a valid equality.

Proof. Model minimizes sum of earliness and tardiness in objective function for preventive jobs, so we focus on the vicinity of d_j. If $d_j \leq b_{dj}$, we cannot know exactly that whether the job is scheduled in this period or not. Yet, if $d_j > b_{dj}$, the model either chooses to schedule jobs in this SP or delay to next period. Latest time that a job is scheduled within this period is b_{dj} that results in earliness. Beyond this point, model delays it to next period so which causes tardiness as a_{dj}. Therefore, model selects minimum of them and if $a_{dj} < d_j - b_{dj}$, job is delayed to next period. So, $np_j = 1$. □

Proposition 4.13. *Should $Y_j^{min} + p_j \geq |T|$, then $np_j = 1$ is a valid inequality.*

Proof. The earliest time for C_j is $Y_j^{min} + p_j$ in our block schedule. C_j relies on t value w.r.t (4.3d). Since largest value that t can have is $|T| - 1$, a job cannot assigned to maintenance hangar. This produces that $np_j = 1$. □

Proposition 4.14. *Lemma 4.7 results in the inequalities given below:*

$$\sum_{j \in PW} x_{bj} = 1 \qquad b \in IB \qquad (4.20)$$

$$y_{bb'} = 1 \qquad b, b' \in IB, b \in B_{i+1}, b' \in B_i \qquad (4.21)$$

Proposition 4.15. *For each t, define $IS_t = \left\{ j \in PM | t < Y_j^{min} \right\}$*

If $|NC| + |IS_t| \geq SLA_t$ then $\triangle SLA_t^- = 0$.

Proposition 4.16. *Given that* $j \in NP$*, then* $\triangle Y_j^{max+} = 0$*.*

Proposition 4.17. *Given that* $j \in PM$ *such that* $j \in J_{tard}$*, then* $E_j = 0$*.*

4.2.3. Optimality Cuts

Proposition 4.18. *Following inequality follows from lemma 4.6*

$$\sum_{j \in J_{tard}} x_{bj} p_j \geq \sum_{j \in J_{tard}} x_{aj} p_j - p_{max}(1 - y_{ab}) - p_{max}(1 - \sum_{j \in J_{tard}} x_{bj}) \qquad b \in B_i, a \in B_{i+1}, i < |C| \tag{4.22}$$

is an optimality cut.

4.2.4. Valid Inequalities

Proposition 4.19.

$$C_j \geq S_j + p_j(1 - np_j), \quad j \in CR$$

is a valid inequality.

Proof. Given that $np_j = 0$, job is scheduled. Hence $S_j + p_j$ yields a lower bound for C_j because job may stay longer after $S_j + p_j$ due to hangar parking conditions. If $np_j = 1$, this is trivial because $C_j \geq S_j$ as $C_j = S_j = 0$. This completes the proof. □

Proposition 4.20.

$$L_{bj} \geq B_{bj} + p_j x_{bj}, \quad j \in CR, b \in B$$

is a valid inequality.

Proof. Once $x_{bj} = 0$, $L_{bj} = B_{bj} = 0$ according to linearization constraints in our model I. Suppose that $x_{bj'} = 1$ at some b and j'. It means that job j' is in block b. Since P_b is bounded below by p_j regarding (4.1d), given inequality satisfies the model. □

Proposition 4.21. *Corollary 4.4 brings about the inequalities given below:*

$$\sum_{a \in B_{i+1}} y_{ab} \leq c_{ub} \quad c_{ub} < K_b, b \in B_i, i < |C|$$

is a valid inequality.

4.3. Heuristic Algorithm

We develop a general heuristic scheme in Figure 4.12. Our main intention is to find a feasible and efficient heuristic solution. If we succeed in finding such a solution, we provide CPLEX the solution as a starting point. Because it diminishes computational time of the exact approach [67]. This heuristic has $O(|T||CR|^2)$ time complexity. It stems from line 57 of Core Heuristic Algorithm. Note that $|Set|$ on algorithms means a cardinality of pertinent set.

Heuristic algorithm consists of two nested loops. In the inner loop, we run Core Heuristic algorithm. In this algorithm, we iterate over t. At every t, we keep distance of every job j to their due time and name it as $slack_j$. We also define a neighborhood parameter, $neigh$. $neigh$ is given by outer loop. We at first select urgent jobs that are either corrective or out of service. For other jobs, we select jobs such that $slack_j \leq neigh$ and order other jobs in the order of increasing $slack_j$ because our aim is to minimize deviation from due time. We combine these sets and form a candidate set, J_t. We keep track of idle capacity of hangar and name it as $hangarslack$. At every t, we pick either $hangarslack$ jobs from J_t or all jobs from J_t if $|J_t| < hangarslack$ and form another set, say J'_t and order it in the decreasing order of p_j times because we would like to allocate job with bigger p_j first. At $t = 0$, we assign ongoing jobs first and then dispatch jobs from J'_t if there still exists idle track position. For $t > 0$, if

$hangarslack > 0$, we send jobs in J'_t to track one by one. Yet, if a candidate job j in J'_t leads to crossing with job at track position $m-1$ and $|J'_t| \geq m$, we do not allocate the job. Because it means that lots of critical jobs could be delayed later in time if we assigned that job that causes crossing. We obtain a feasible solution by means of inner loop. In the outer loop, we increase *neigh* by five and reapply core algorithm again. If objective value of feasible solution decreases, we keep increasing *neigh* by five until we see no improvement in the incumbent objective value. This way, we devise a best feasible solution in a very short amount of time.

In summary, we give priority to jobs that are out of service first and then jobs that are close to their due time over other jobs at every time t. We assign them as long as we have an idle capacity. We select jobs with bigger p_j first during assignment. Yet, by changing *neigh* parameter in the outer loop and implementing core heuristic again, we try to dispatch jobs earlier and improve objective value of the heuristic algorithm.

4.3.1. Sets

T Time
CR Critical Jobs

4.3.2. Parameters

d_j Due time of job j to pull it to maintenance hangar
p_j Maintenance duration of job j
Y_j^{min} Begin time of maintenance interval
$neigh$ Maximum allowed slack value for jobs to pull them to maintenance hangar
m Number of track locations

4.3.3. Dynamic Sets

H_t Jobs at maintenance hangar at time t

J_t Jobs at time t

$Exit$ Jobs that have abandoned maintenance hangar so far

$slack_j$ Distance of jobs to their corresponding due date, $slack_j = t - d_j$

$Hangarcandid$ Jobs that are candidates to be pulled to hangar

$Base$ Set of bases where each base contains info like start,length and finish time along with appointed job j

$ParentChild$ Set that keeps *child* and *parent* pairs

$Urgents$ Preventive jobs that out of service at time t and all corrective jobs

$Ongoings$ Ongoing jobs in heuristic

4.3.4. Some Variables

e_j Maintenance finish time of job j

b_j Maintenance begin time of job j

$numbertohangar$ Number of railcars that can be appointed to maintenance hangar at time t

obj Objective function value

$pinc$ Best incumbent solution so far

$pnew$ New solution of *CoreHeuristic*

$nextper$ Number of jobs skipped to next period

$hangarslack$ Unused capacity in the track in any time instant

$neigh = 0$
$pinc \Leftarrow CoreHeuristic(neigh)$
repeat
  Increase $neigh$ by 5 and $pnew \Leftarrow CoreHeuristic(neigh)$
  if $pnew.obj < pinc.obj$ **then**
    $pinc \Leftarrow pnew$
    continue
  else
    break
  end if
until break
Final: Feed *pinc* to *MILPStart*

Figure 4.12. General Heuristic Algorithm

4.4. Alternative Methods for Handling Corrective Jobs

In the existence of stochastic breakdowns, schedule that is generated by model could be undermined by future corrective job arrival. So, it results in a decrease in proactive planning efficiency [68]. Hence, we can say that model yields a myopic schedule. It may have a negative impact on key performance indicators (KPI) in the long term. In order to alleviate the effect, we propose two methods to tackle this issue.

4.4.1. Introduction of Buffer Time

First, we introduce a buffer method to protect our system against uncertainty in the future as well as utilizing our track in an efficient way. By virtue of this method, we act proactively which makes sure a robust schedule. In order to implement it, we assume that we already have a solution. We define $\hat{S}_j$ as realized maintenance begin time in weekly schedule. Since we execute solutions where $\hat{S}_j < 24$, we take solutions into consideration in which $\hat{S}_j < 24+\zeta$. Then we reschedule them by pulling jobs back in time at most ζ hours. This displacement could be deemed as buffer time, that is ζ.

```
Initialize Sets and values
H_0 ⇐ ∅
J_t ⇐ CR   ∀t
nextper = 0
for t = 1 to T do
  Initialization of sets at the beginning of time t
  for all j ∈ H_t do
    if  e_j == t then
      H_{t−1} ⇐ H_{t−1} − j
      Exit ⇐ Exit ⋃ j
    end if
  end for
  H_t ⇐ H_{t−1}
  J_t ⇐ J_t − H_t − Exit
  Decrease slack of every job by 1
  for all j ∈ J_t do
    if slack_j ≤ neigh then
      if t + p_j ≥ T then
        J_t ⇐ J_t − j
        nextper + +
      else if t < Y_j^{min} then
        J_t ⇐ J_t − j
      end if
    else
      J_t ⇐ J_t − j
    end if
  end for
```

Figure 4.13. Core Heuristic Algoritm

if $|Exit| + nextper == |CR|$ **then**
  break
else
  $nextper = 0$
end if
$numbertohangar = min\{|J_t|, hangarslack\}$
if $t == 0$ **then**
  $numbertohangar = max\{numbertohangar, |Ongoings|\}$
end if
if $numbertohangar == 0$ **then**
  continue
end if
  Order jobs in J_t
  Select $Urgents$ jobs and remove them from J_t
  Order rest of the jobs in this set in the order of decreasing *slack* times and name this set as J_t'
  Sort jobs in $Urgents$ in the order of increasing p_j times and name this set as $Urgents'$.
  $J_t' \Leftarrow J_t' \bigcup Urgents'$
if $|J_t'| \geq m \& |H_t| \geq m - 1$ & last job of J_t' blocks departure of last job of H_t **then**
  continue
end if
Select last $numbertohangar$ jobs from J_t' and form a set $Hangarcandid$
Sort jobs in $Hangarcandid$ w.r.t decreasing order of p_j's and set this new set as $Hangarcandid'$

Figure 4.13. Core Heuristic Algoritm(cont.)

```
    if t == 0 then
        Perform assignments of jobs in Ongoings to Base and Parentchild then
        appoint jobs in Hangarcandid' if hangarslack > 0
    else
        Perform assignments of jobs in Hangarcandid' to Base and Parentchild
    end if
end for
Finalize assignments: Allocate finish and lengths of jobs in Base
Calculate obj
```

Figure 4.13. Core Heuristic Algoritm(cont.)

We say at most because jobs may not be able to be driven left in time due to schedule feasibility.

In Figure 4.14, we show an example. In part a, $\zeta = 0$ because it is a solution in which all jobs are preventive obtained by CPLEX solver. Since we set $\zeta = 20$ we focus on jobs where $\hat{S}_j < 44$. Next, we pull jobs back which leads to part b in the figure. Jobs 0 and 12 are dragged till zero and remaining jobs which were not planned to be executed before on first day is now processed within the day. Note that jobs 1 and 13 could have been driven back more but they could not be pulled back further because jobs 12 and 0 already occupies first 15 hours after movement. Therefore, we assure a feasibility. Furthermore, job 24 is moved back in time 20 hrs, which is the maximum quantity that it can be dislocated.

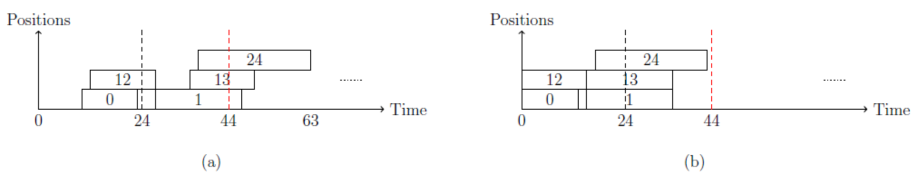

Figure 4.14. Buffer Time Method Example

4.4.2. Anticipation of future events

We introduce another method to diminish preventive tardiness KPIs in the presence of corrective events. In this method, model remains same but we change objective function of main model to assign jobs earlier in our model using MILP approach. To do so, we insert ω coefficient to T_j in equation (57). For instance, if we apply this technique, solution in part a turns into part b as given in Figure 4.15.

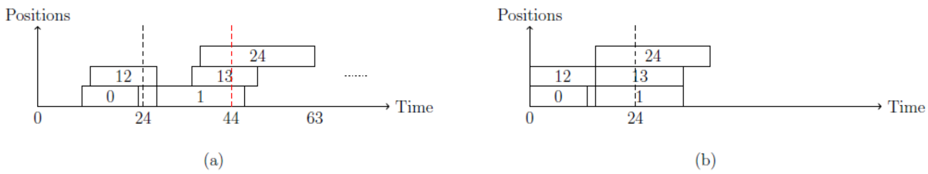

Figure 4.15. Anticipation Solution of SP

5. COMPUTATIONAL EXPERIMENT DESIGN

In this chapter, we point out how we randomly generate problem parameters and determine critical jobs. After that, we specify warmup period, ascertain single run length along with batch size. We fine-tune CPLEX parameters. We compare MILP model and its extensions computationally. Finally, we present an experiment plan for comprehensive simulations.

5.1. Service Level Agreement Generation

In this work, SLA is an agreement between city and tram service provider. In order to produce SLA, we investigate different kinds of examples from all over the world. For Istanbul, we requested SLA schedules from tram service provider of Istanbul, MetroIstanbul. We also take a look at a timetable of tram service lines of other cities which are London-Victoria line [69] and Berlin-U7 line [70] and other relevant works ([71], [72], [62]). We generally observe that most of these schedules are comprised of two peak times and one off-peak time in between them. After some point (usually after 24.00) SLA value declines to very small number or zero. Therefore, we devise a schedule to represent such an SLA and name it as a core schedule, that is SLA_c. One can see it in Figure 5.1. Without loss of generality, we set maximum value of it to 100. By looking at the distribution of the schedule over time, we dissect time in four parts as below:

- $Morning : [6, 12]$
- $Afternoon : [12, 18]$
- $Evening : [18, 24]$
- $NoServive : [24, 6]$

We will use this schedule as a base to produce random daily SLA schedules. In order to do so, let us define following terms:

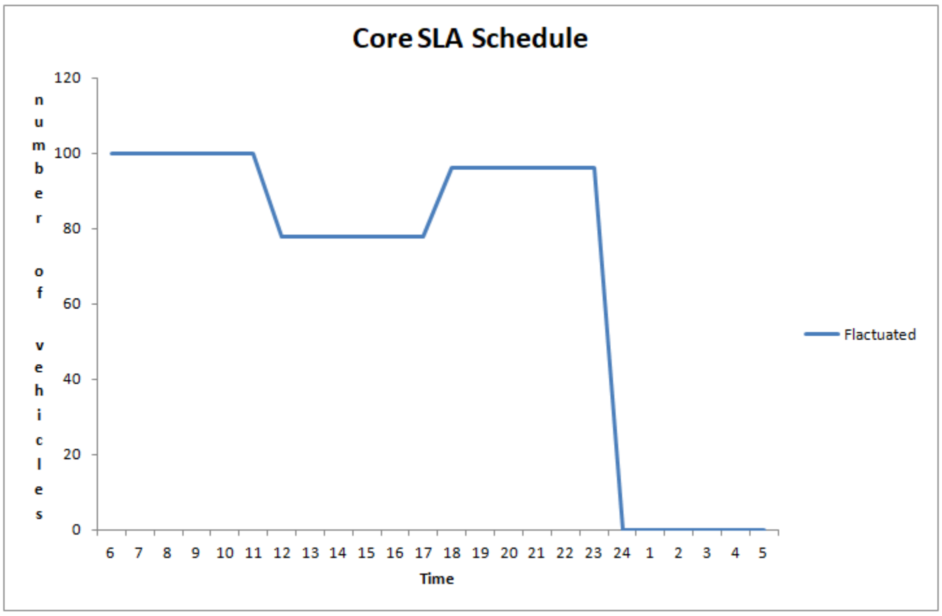

Figure 5.1. Core SLA Schedule

n : Number of vehicles in our system

e : A coefficient to determine n

SLA_c^{max} : Maximum value in SLA_c

SLA_b: Daily total SLA demand

a : A coefficient to determine SLA_b

$\delta_1 = \lceil SLA_c^{max} f \rceil$

$\delta_2 = \lceil SLA_c^{max} d \rceil$

We have two SLA types. We differentiate one type from another by virtue of a variability. It is a distance of off-peak time to maximum peak time, namely $Afternoon$ to $Morning$ in our instance. Since SLA_c includes a variability, we actually make use of it to produce a Case 2 schedule while preserving SLA_c pattern. To do so, we perturb $Morning$ and $Afternoon$ sections by - $UNIF\,[0, \delta_1]$ whereas $Evening$ is perturbed by - $UNIF\,[\delta_1, 2\delta_1]$. On the other hand, Case 1 schedule has a different pattern, so we just utilize SLA_c^{max} to derive a new SLA. We treat $Morning$ and $Evening$ same because they constitute two peaks of the schedule and we require our schedule as plain as possible. Addressing this issue, we perturb SLA_c^{max} by - $UNIF\,[0, \delta_1]$. In contrast to it, we change SLA_c^{max} by - $UNIF\,[0, \delta_2]$ in $Afternoon$. As a consequence, we obtain a *temporary schedule*. It is either Case 1 or Case 2 schedule.

SLA_b is obtained by multiplication of total number of vehicle in SLA_c and a. So, for every produced *temporary schedule*, we harness adjustment algorithm to specify final schedules such that daily SLA demand equals to SLA_b. Adjustment algorithm works on $Morning$, $Afternoon$ and $Evening$ intervals and gives priority in the order of $Afternoon$, $Evening$ and $Morning$ ones so as to calibrate temporary schedule. In conclusion, we make sure that total SLA demand of every final random daily SLA schedule equals to SLA_b. To determine n, we need to set a fixed value which is independent from random SLA. Therefore, we set $n = SLA_c^{max}(1 + e)$.

In our case, we fix parameters as follows: $a = 0.95, d = 0.10, e = 0.05, f = 0.05$. An example for Case 1 and Case 2 schedules can be seen in Figure 5.2 and Figure 5.3, respectively.

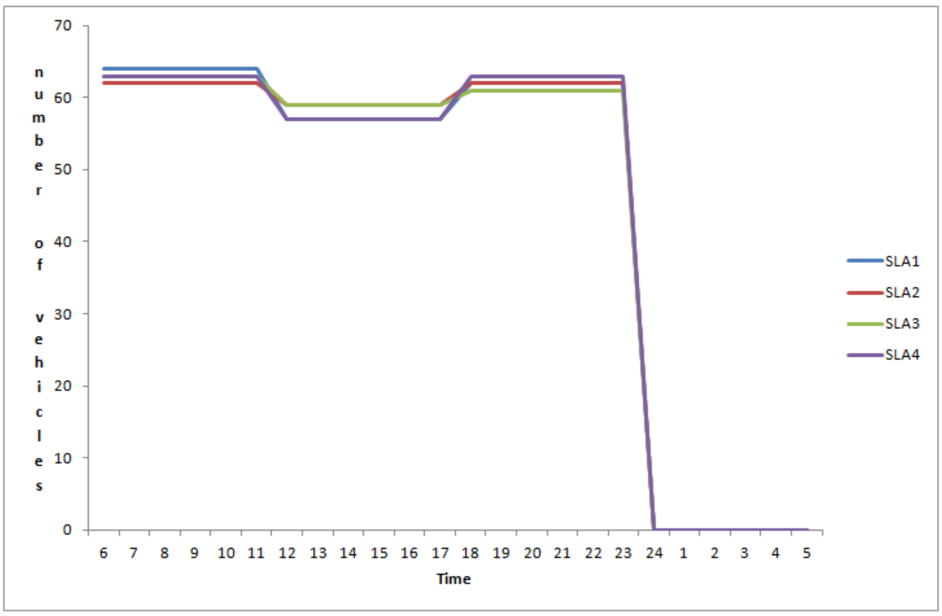

Figure 5.2. Case 1 Schedule

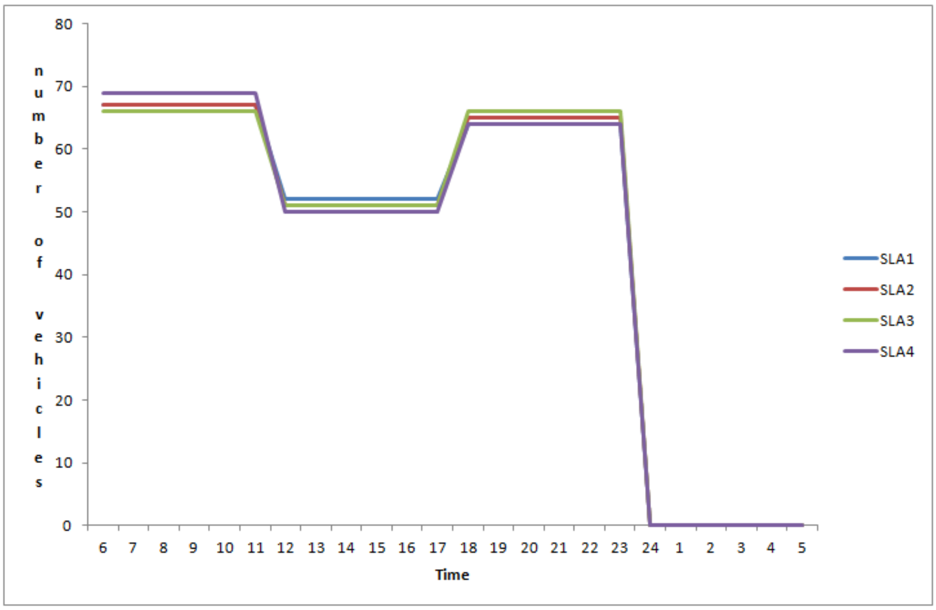

Figure 5.3. Case 2 Schedule

Finally, we extend these daily schedules throughout scheduling period via repeating them every day (see Figure 5.4 and 5.5).

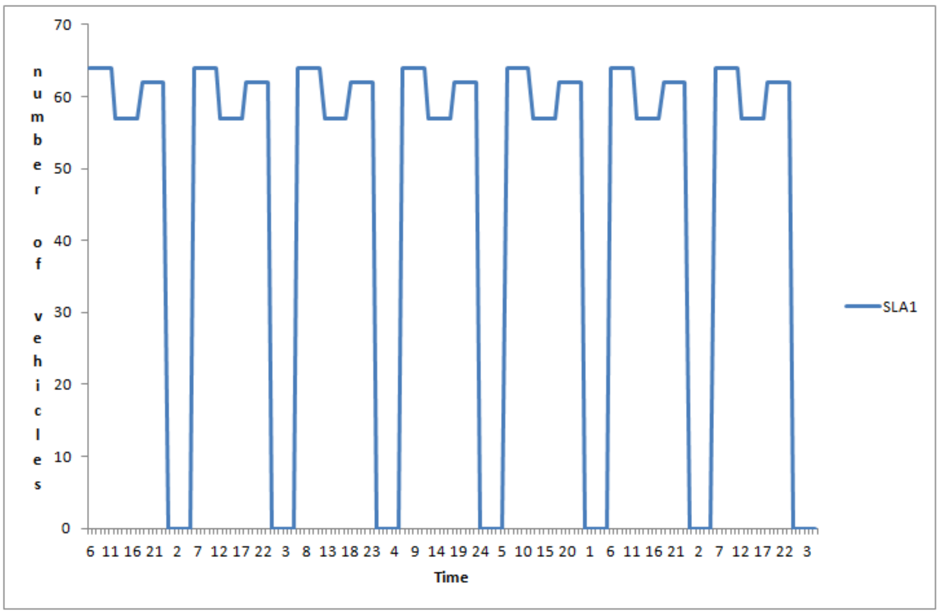

Figure 5.4. Weekly Case 1 Schedule

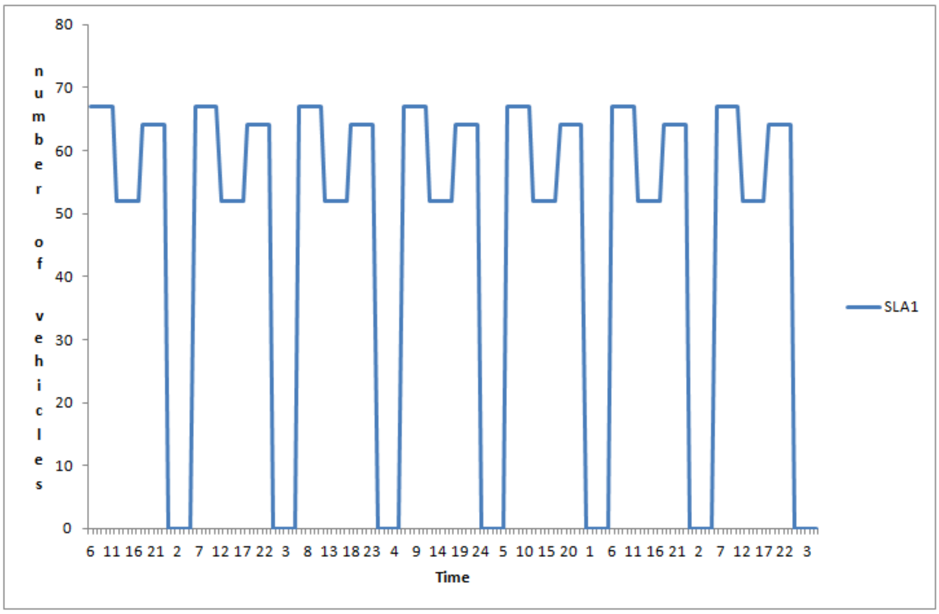

Figure 5.5. Weekly Case 2 Schedule

5.2. Preventive Job Generation

For each vehicle in the fleet, there exists critical maintenance intervals. Begin time (Y_{j1}^{min}) of first preventive interval is created according to UNIF[0, β]. Generally, vehicles should be maintained after target Π time that depends on maintenance type. However, it is very hard to achieve it due to dynamic passenger service and maintenance hangar conditions. Hence, we define v as positive and negative deviation coefficient from our target level Π inspiring from [23]. Besides, taking v variability into account during vehicle arrivals, Y_j^{min} of the next interval can be between Π and $(1+v)\Pi$ which is generated by $UNIF[\Pi, (1+v)\Pi]$. Let z be zth maintenance interval of a specific vehicle, θ_1 be the coefficient that determines the length of maintenance interval, ϱ be the coefficient to ascertain due time and V be the vehicle set. Aforementioned equations and other interval parameter calculations are given below:

$$Y_{j(z+1)}^{min} = Y_{jz}^{min} + UNIF[\varPi, (1+v)\varPi] \qquad j \in V, z = 1, , l \qquad (5.1)$$

$$Y_{jz}^{max} = \lceil Y_{jz}^{min} + \theta_1 v \varPi \rceil \qquad j \in V, z = 1, , l \qquad (5.2)$$

$$d_{jz} = \lceil Y_{jz}^{min} + \varrho v \varPi \rceil \qquad j \in V, z = 1, , l \qquad (5.3)$$

If we set $\varPi = 720, v = 0.1, \theta_1 = 2$ and $\beta = 672$, we obtain Figure 5.6. Left side of the figure matches Y_{j1}^{min} generation. We generate it by UNIF[0, β]. From possible intervals, one of them is realized where $Y_{j1}^{min} = 500$. The point is denoted with cross. Next, we produce second interval through 5.1. Second cross is realized at $Y_{j2}^{min} = 1115$. Y_j^{max} of intervals are calculated by means of 5.2.

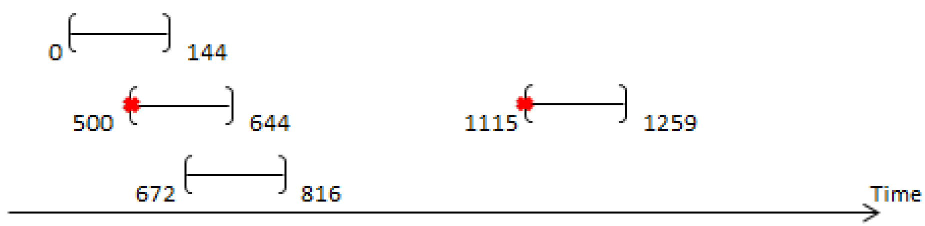

Figure 5.6. First Two Preventive Maintenance Intervals for a Vehicle

5.3. Corrective Job Generation

Corrective maintenance arrival is a time when vehicle arrives to the hangar due to a breakdown. In our problem, we assume that vehicles come to system according to *Poisson process*. It can be assumed that preventive maintenance eliminates wear-out failures but random ones still exist [73]. Let τ be a mean time between failures, L be a planning horizon length. For any vehicle, let $u_z \sim UNIF[0,1]$, X_z be a zth time between failures, Y_{jz}^{min} is the time of zth breakdown. Poisson process entails that $X_z \sim Exp(\frac{1}{\tau})$. Using inverse transform technique for exponential distribution, Banks *et al.* [74] conclude that $X_z = -\tau \ln(1 - u_z)$. Thus, we produce events accordingly as long as $Y_{jz}^{min} < L$. As a result, we obtain number of events for every vehicle and it is denoted by ev_j. In fact, corrective event is just one time incident and do not have an interval like preventive job but we insert an interval so as to create maintenance stays in our test system for warmup period determination. Y_j^{min} and other parameters related to corrective job maintenance interval length are found as below:

$$Y_{jz}^{min} = X_z + Y_{j(z-1)}^{min} \qquad j \in V, z = 1, , ev_j \qquad (5.4)$$

$$Y_{jz}^{max} = Y_{jz}^{min} + \theta_2 \qquad j \in V, z = 1, , ev_j \qquad (5.5)$$

Since we use Poisson process, we can derive τ according to desired expected number of corrective job arrivals to our system. Let us denote $Pr\{X_z < 24\}$ as failure probability of vehicle in first 24 hours and $E[fail|X_z < 24]$ as the expected number of vehicle fleet breakdowns within first 24 hours. For any vehicle, $Pr\{X_z < 24\} = 1 - exp\left(\frac{-1}{\tau}24\right)$ follows from exponential distribution. By binomial expectation, $E[fail|X_z < 24] = nPr\{X_z < 24\}$. We equate $E[fail|X_z < 24]$ to ϕ. Using equations given below, we derive the result as follows:

$$n(1 - exp(\frac{-1}{\tau}24)) = \phi \qquad (5.6)$$

$$exp(\frac{-1}{\tau}24) = 1 - \frac{\phi}{n} \qquad (5.7)$$

$$\frac{-24}{\tau} = \ln(1 - \frac{\phi}{n}) \qquad (5.8)$$

$$\Rightarrow \tau = \frac{-24}{\ln(1 - \frac{\phi}{n})} \qquad (5.9)$$

5.4. Job Duration Generation

Job durations are variable as standard times are difficult to be identified according to Kelroy Solutions [75] and jobs deviate from standard times due to some reasons.

Besides, some extra defects are observed after maintenance stop [76]. It may necessitate the need for checking neighboring and deteriorated components of main failed component [3]. In addition to it, we may encounter predictive maintenance issues like replacement and overhaul of some parts depending on their condition [68]. Thus, we take κ as a possible noise to standard times of preventive and corrective jobs. Uniform random value is appropriate for these job length distributions, and they are generated after corrective or preventive job is produced. Maintenance time depends on its type. Let us define p as minimum time required to perform the preventive maintenance task and p_{cor} as a minimum time to execute corrective maintenance task. Depending on these definitions, preventive jobs are distributed as UNIF $[p, p+\kappa]$ while corrective jobs have UNIF$[p_{cor}, p_{cor}+\kappa]$ hrs.

5.5. Determination of Critical Jobs

During critical job determination, we at first inherit jobs from previous day's maintenance list if they are failed to be allocated to track on yesterday. Sometimes, some jobs could be appointed a track but their job has not been finished by the beginning of the current day. In other words, they are ongoing jobs. Let us we have jobs A and B such that they have a parent-child relation. Addressing this issue, three cases occur on two adjacent track positions as given below:

(i) Case I: A finishes maintenance on previous day whereas B goes on maintenance process. It is displayed part a in Figure 5.7. On current day, we observe part b. B continues its process two more hours on second day while A waits it till that point. It is denoted by arrow in the right side of the figure.

(ii) Case II: A and B ends maintenance on current day. It is shown part a in Figure 5.8. It follows that only remaining p_j times are fulfilled in current SP.

(iii) Case III: A ends maintenance on current day but B is not an ongoing job. We show it in part a of Figure 5.9. So on current day, we only take remaining part of p_A into account and fix it on our model.

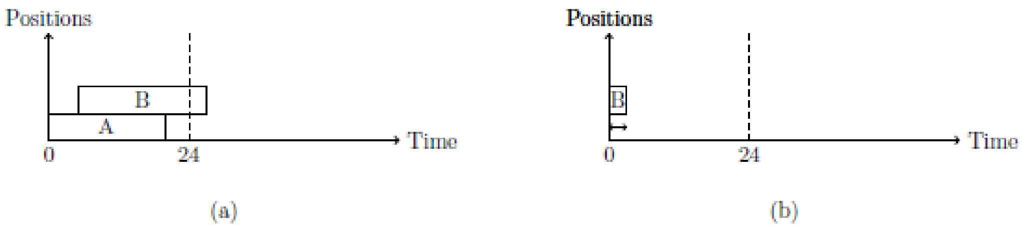

Figure 5.7. Ongoing Jobs Case I Example

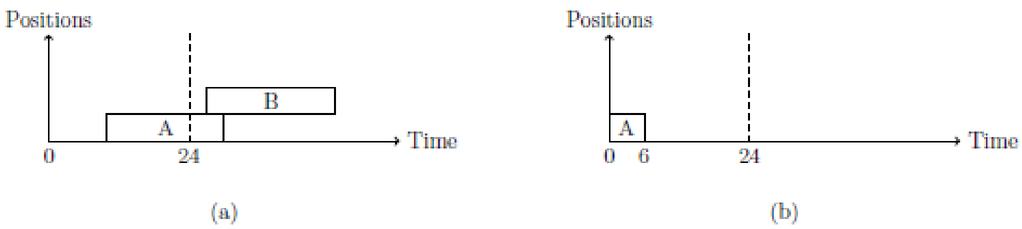

Figure 5.8. Ongoing Jobs Case II Example

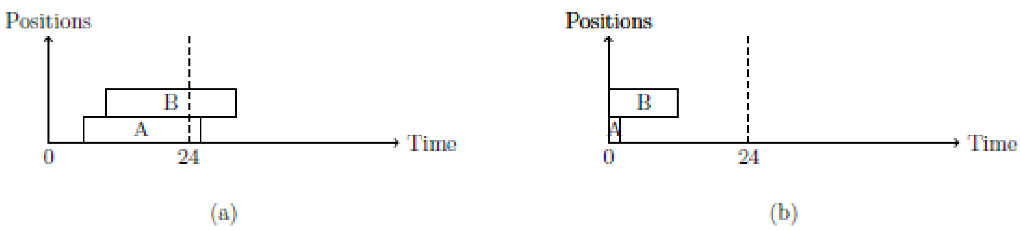

Figure 5.9. Ongoing Jobs Case III Example

Besides, we look at maintenance intervals of other jobs such that their maintenance due times lies in current SP. If a job satisfies preposition 4.12, we do not append the job to critical jobs. Otherwise, job is inserted into critical jobs. Figure 5.10 shows a good demonstration that how we select these jobs. If we look at the figure from above, we fetch first two jobs in this SP because their due time lies away from end point of SP. For third job, even though its due time lies in SP, we do not append it to critical jobs because it satisfies preposition 4.12. Since due time of fourth job is out of SP, we do not consider it either.

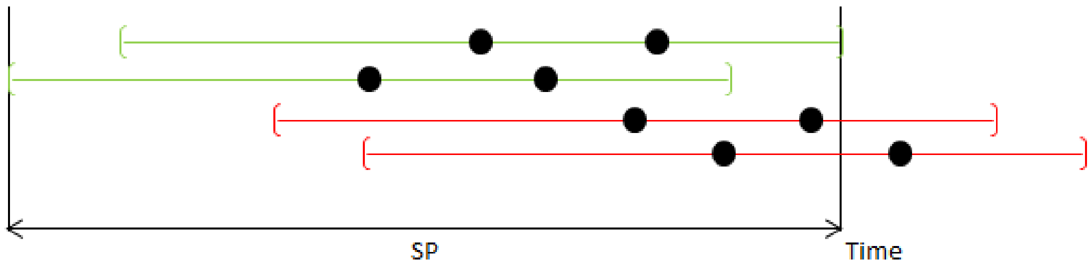

Figure 5.10. SP and Preventive Maintenance Intervals

During corrective job selection, we only consider breakdowns that has occurred the last day. Because these vehicles have already become out of service and stay at somewhere in hangar to be maintained on the track.

In real life, conflict of preventive and corrective job maintenances are inevitable because any planned vehicle is prone to breakdown. Therefore, this issue causes confusion such that a vehicle may exist at the same scheduling period as preventive and corrective job. You can see an example in Figure 5.11 where preventive intervals are depicted as lines with brackets, vehicle breakdowns are shown as crosses and a preventive ongoing job at the hangar is shown as a rectangle. This figure portrays four various conflict cases.

In order to solve these cases, we look at the picture from planner's standpoint. In day t, the planner is aware of corrective jobs due to rolling horizon. For vehicle 1, it plans to be processed as a preventive job in SP but broke down on yesterday.

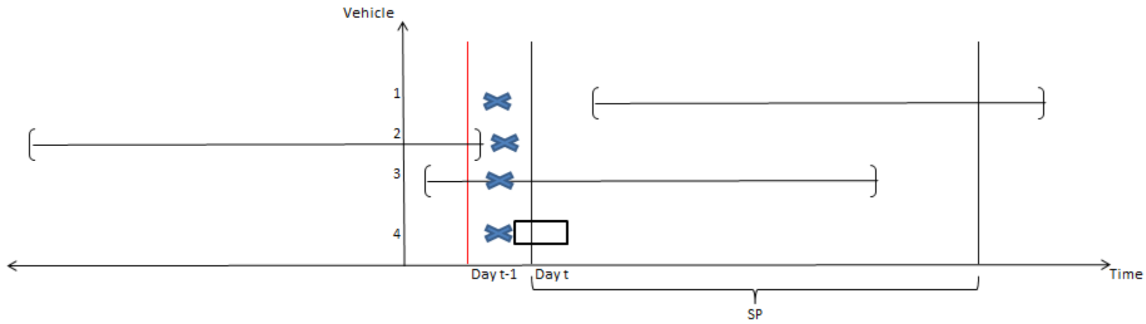

Figure 5.11. Overlapping of Maintenance Jobs for a Vehicle

So, we combine preventive job with a corrective one and regard this job as corrective in our maintenance list. For vehicle 3, we again apply the same procedure because the vehicle has not been managed to be taken into consideration until t. Vehicle 2 has likewise failed to be allocated to track till its latest maintenance start point, thus become out of service. It implies that it cannot be prone to breakdown. Hence, we discard the corrective event. Vehicle 4 is a little bit different than the previous cases. Because it is an ongoing job so its operation has already begun. Thus, we ignore the corrective event and continue its process on day t. These changes result in Figure 5.12. Furthermore, when a vehicle is in corrective job list, a new corrective event might occur for it on account of independent job production process. In this regard, we cancel the generated event. Because a vehicle that stays on hangar cannot undergo extra corrective maintenance action. In conclusion, we obtain conflict free vehicle maintenance operations for every vehicle.

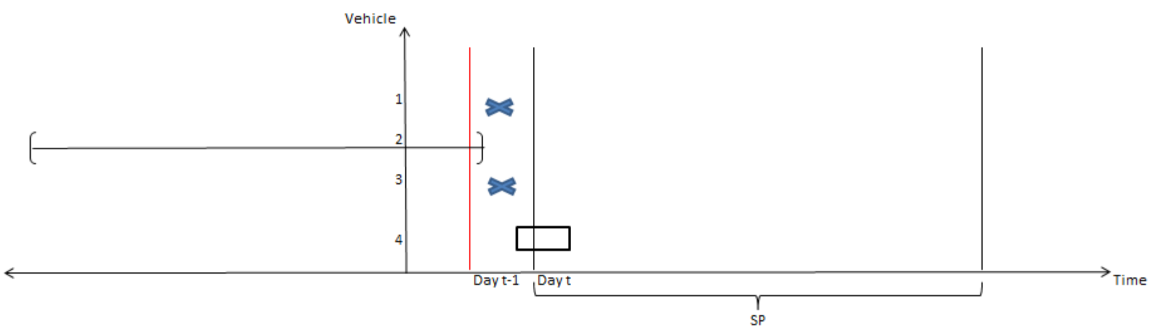

Figure 5.12. Conflict Free Maintenance Job Asssignments for a Vehicle

5.6. Warmup Period Determination

We are interested in long term behavior of the system, so we ignore the time till the system reaches a steady state. This time is warmup period and we gather statistics after that time. We build another test system to fetch the time where the system gets saturated. Experimental design parameters and their values can be seen in Table 5.1. We fix corrective and preventive job generation parameters as follows:

$\theta_2 = 24$

$\beta = \mathit{\Pi}$

$\varrho = 1.5$

$\theta_1 = 2$

Table 5.1. Warmup period experimental factors

	Levels	
Factors		
$\mathit{\Pi}$	600	720
τ	2508	10068
v	0.10	0.20
Maintenance	Prev	Prev+Cor

In warmup analysis, we specify an auxiliary performance indicator. Our performance indicator is a *cumulated average jobs* on maintenance hangar where hangar has no capacity. We set planning horizon length as one year, i.e 8760 hrs. We select time unit as hour. During this analysis, we have two creation processes. Let z be the z_{th} interval. In preventive arrivals, Y_{jz}^{min} and Y_{jz}^{max} are determined w.r.t (5.1) and (5.2), respectively. So we generate an arrival event at time tz_{arr} between Y_{jz}^{min} and Y_{jz}^{max} and a random processing time, p. Then, a departure time, tz_{dep} from maintenance system equals to $tz_{dep} = tz_{arr} + p$. Same logic is also made use of in corrective job stays where Y_{jz}^{min} is produced with respect to (5.4) and Y_{jz}^{max} is created with respect to (5.5). If we fix $p = 8, \mathit{\Pi} = 600, v = 0.10$, $\beta = 672$, this give rises to an example interval like in Figure 5.13. Between $Y_{jz}^{min} = 360$ and $Y_{jz}^{max} = 432$, an event generated uniformly at time $tz_{arr} = 365$ and the vehicle departs the system at $tz_{dep} = 373$.

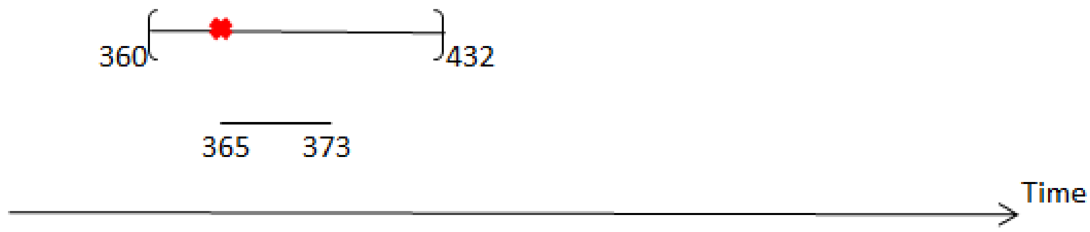

Figure 5.13. Preventive Interval and Stay

Given that we set $\theta_2 = 24, p_{cor} = 10$, we produce a corrective interval at Figure 5.14. Let we are at k_{th} interval of the corrective job. It follows that $Y_{jk}^{min} = 720$ and $Y_{jk}^{max} = 744$, an event generated uniformly at time $tz_{arr} = 732$ and the vehicle departs the system at $tz_{dep} = 742$.

Figure 5.14. Corrective Interval and Stay

Yet, as creation processes are independent, we delete intersections between them. In detail, if their stay times overlap with each other, we pick the job which comes first and discard other one. It is similar to the method we utilize in Section 5.5 but we implement it over maintenance stays instead. Thereafter, we calculate number of jobs present in the system at each t by looking their visit intervals, and get *cumulated average number of jobs* in the system till that time.

We pick three representative systems with same random seed to experiment. First two systems are related with minimum and maximum loaded preventive systems, respectively. In the presence of corrective jobs, System 3 causes most congested system.

They are presented below:

(i) System 1: $\Pi = 720$, $\upsilon = 0.2$

(ii) System 2: $\Pi = 600$, $\upsilon = 0.1$

(iii) System 3: $\Pi = 600$, $\tau = 2508$, $\upsilon = 0.1$

If we take a closer look at the graphs in Figure 5.15, system converges to steady state after the point from which stems a perpendicular line. We detect it 1440 hrs which equals to $1440/24 = 60$ days. We make this transformation because on each day of the real system, we solve an instance. Hence, at each parameter combination during our experiments, we do not collect statistics till it. After that, we will record statistics by the end of planning horizon.

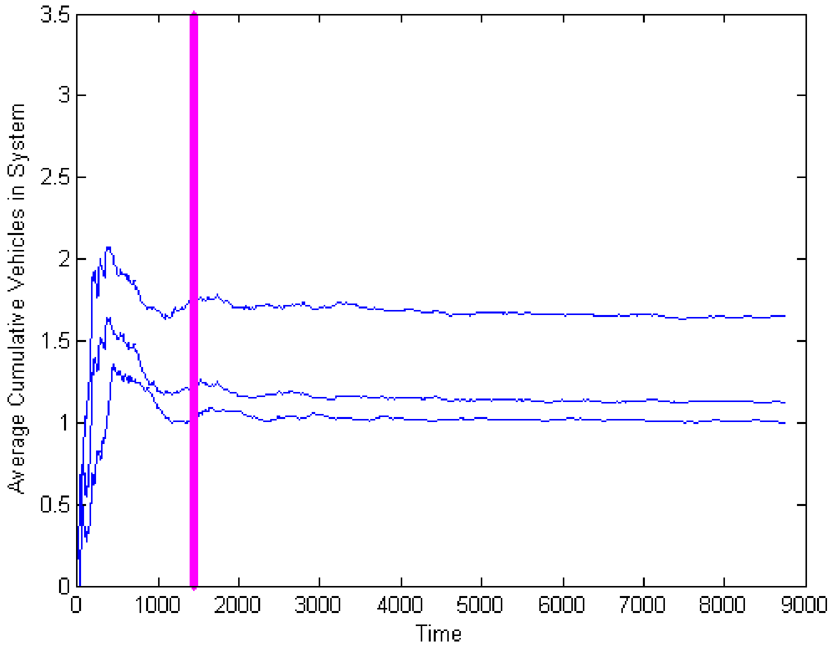

Figure 5.15. Average Cumulative Stays in Test System

5.7. Single Run Length and Batch Size

In our system, we want to find out effects of problem parameters throughout the planning horizon. Due to rolling horizon, we solve the problem on every day.

It follows that reaching to 60 days takes much time. In this circumstance, we can make a very long single run and batch some parts of the run instead of performing independent replications [77]. Hence, we conduct some experiments to ascertain batch size. In a single batch run, each batch is regarded as a single replication. Our aim is to find uncorrelated batches by looking at autocorrelation of every batch mean. For our case, System 1 with $m = 2$, $SLA = Case1$ is picked for examination. Again, we ignore corrective arrivals and only design our system with reference to preventive maintenance cycles. We fix total number of days to 160 and perform a single run. We record outcomes after 60th day(warmup period). We choose batches which are comprised of 10 day outcomes. We plot its correlation graph in Minitab 16 statistical package software which yields a graph in Figure 5.16. It shows us that bars stay within limits. It means that correlation between batches are eliminated. Therefore, we will conduct statistical tests on 10 different batch means after the warmup period.

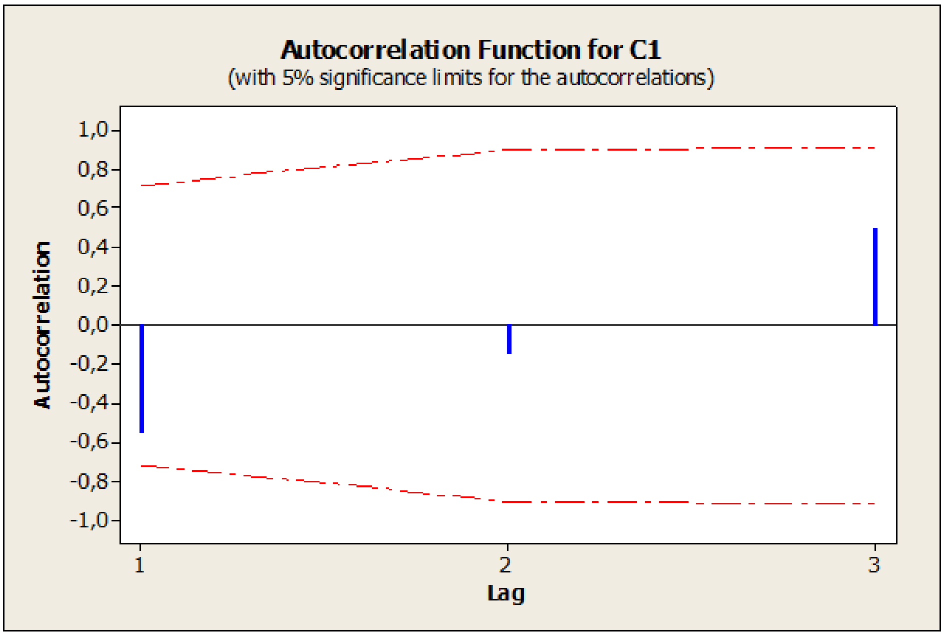

Figure 5.16. Autocorrelation Graph for 10 Batch Size

5.8. Cplex Parameter Fine-Tuning

Since the problem is computationally intractable for large instances, we propose here to change CPLEX parameters to diminish computational time. First, we implement a priority rule for variable branching in CPLEX. Prioritized variables which are presented in the decreasing order of their importance are as follows: x_{bj}, y_{ab}, z_{ab}. Second, we apply preprocessing and probing on a moderate level by means of CPLEX parameter fine-tuning options.

5.9. Computational Study

In this section, we compare three methods in terms of time. Model corresponds to our main model. Model2 represents Model with all model improvements and fine-tuned CPLEX parameters. Lastly, Heur+Model2 denotes the methodology where we utilize heuristic to feed an initial heuristic solution to the model. We hypothesize that Heur+Model2 is the best method. Hence, we need to perform some tests to prove it. We select three preventive systems for testing. Their settings are shown below:

- System 1: $\Pi = 720, v = 0.2, m = 3, SLA = S$
- System 2: $\Pi = 600, v = 0.1, m = 2, SLA = F$
- System 3: $\Pi = 480, v = 0.1, m = 2, SLA = F$

Note that S denotes Case 1 SLA while F corresponds to Case 2. We execute Heur+Model2 on these systems till the end of planning horizon where we utilize 2 mins time limit for each day in the horizon. We collect results after warmup period and find average load of 100 days where each day represents a problem instance. As a result, system 1,2 and 3 yield 52.88%, 81.64% and 95.31% Load, respectively. Hence, we classify them as low, medium and high load systems, respectively. Since we even have to solve 160 problems for one system, we set 2 mins time limit to solve each problem. Yet, we also would like to observe how computational performance of these methods would change under more time limit. Thus, we set 10 mins time limit for alternative scenario.

For every system, we record 100 instances after warmup period. We pick 10 random day instances out of 100. We compare methods under Z_f and Gap(%) performance measures. Z_f means objective value of feasible solution of model given time limit whereas Gap(%) implies optimality gap obtained from CPLEX given time limit. At first, we examine that whether Model outcomes follow a normal distribution under low load with 2 mins time limit. H_0 hypothesis states that data follow a normal distribution in $\alpha = 0.05$. We reject it if $p < \alpha$ [78]. We plot the gap using Minitab with Anderson-Darling test in Figure 5.17. It exhibits that data points are not centered around normal fit line and $p < \alpha = 0.05$. We conclude that data is not normal. So, we do not execute t-test while comparing methods and use box-plots instead.

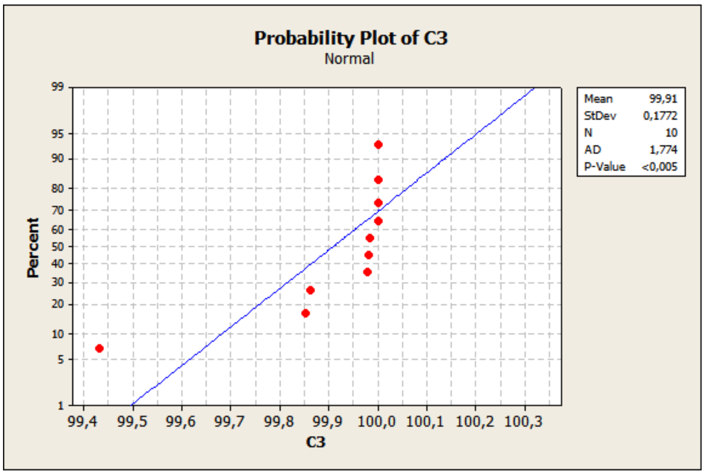

Figure 5.17. Gap(%) Normality Plot

5.9.1. 2 mins Time Limit

We examine low load system. According to Z_f results in Figure 5.18, Model2 displays a little bit more variability than Model. However, there does not seem a significant difference between two treatments. Yet, Heur+Model2 has almost no dispersion and yields dramatically significantly small Z_f values. Concerning gaps, we plot results in Figure 5.19. Model and Model2 have similar intervals and close to 100% whereas Heur+Model2 diminishes gaps below 90% if we disregard outliers. Hence, it gives the best gap.

We now study middle load system. Z_f results in Figure 5.20 suggests that last fourth quartile of Model is less than minimum whisker point of Model2. Hence, Model

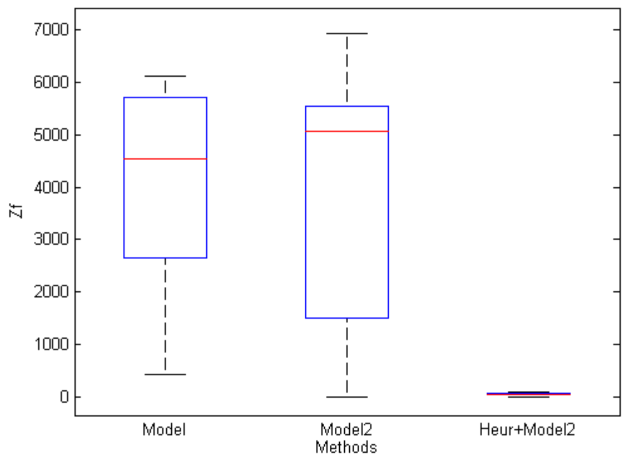

Figure 5.18. Low Load System Comparison under Z_f with 2 mins Time Limit

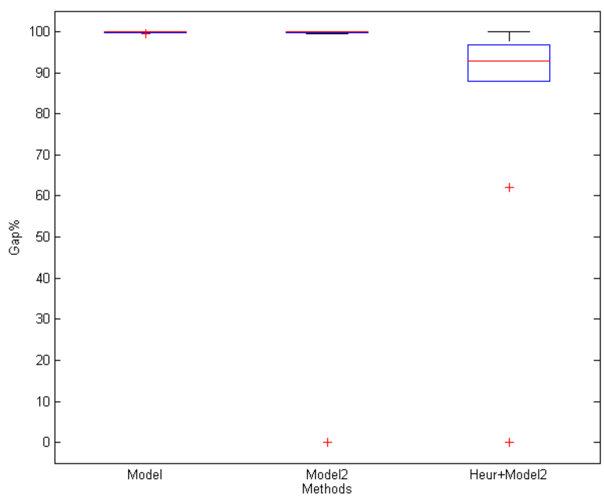

Figure 5.19. Low Load System Comparison under Gap(%) with 2 mins Time Limit

performs a little bit better in Z_f. Should we append an initial heuristic to Model2, we come up with profoundly lower Z_f outcomes than other two methods. Regarding gaps, we portray results in Figure 5.21. For Model and Model2, we observe outcomes that are so close to 100%. Oppositely, Heur+Model2 has a considerably wider range from 100% to 74% and almost all data points are significantly lower than other ones. In conclusion, it yields the best result in terms of gap and Z_f

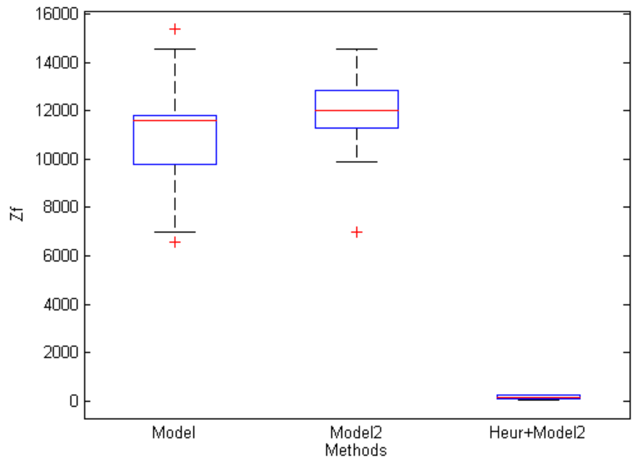

Figure 5.20. Middle Load System Comparison under Z_f with 2 mins Time Limit

Lastly, we investigate results under high load. Z_f outcomes could be seen in Figure 5.22. Median of Model2 is significantly lower than Model. Besides, its range is less than Model. It indicates that Model2 is better than Model. Heur+Model2 produces best outcome because its first quartile matches with minimum whisker points of Model and Model2, so 75% of data is lower than them. We portray gap results in Figure 5.23. If we compare Model and Model2, Model2 could reduce gap till 85% including outlier whereas Model does the same thing until 95%. So, Model2 performs better. Heur+Model2 brings about best solution because even its median is around 80% that is less than all data points of other treatments. It has a high range from 100 to 55%.

If we look at these systems, gap range increases, particularly for Heur+Model2 as load increases. In contrast to it, we observe no general pattern for Z_f. Heur+Model2

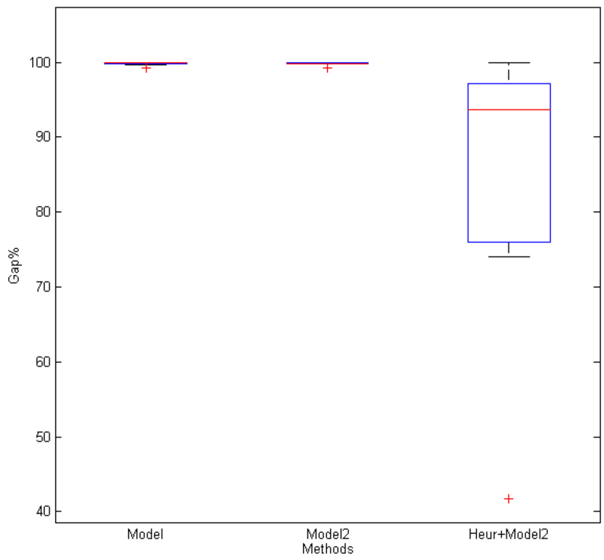

Figure 5.21. Middle Load System Comparison under Gap(%) with 2 mins Time Limit

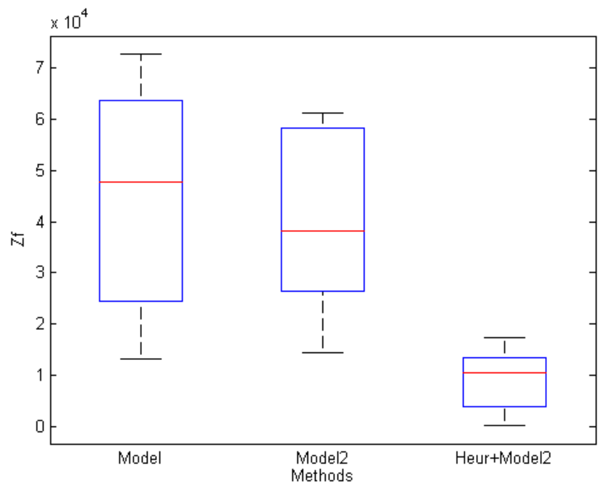

Figure 5.22. High Load System Comparison under Z_f with 2 mins Time Limit

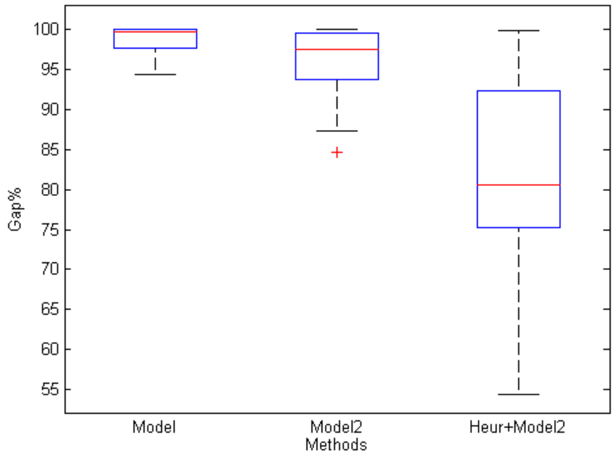

Figure 5.23. High Load System Comparison under Gap(%) with 2 mins Time Limit

results in best procedure in all systems and performance indicators. When we question model improvement impact, we obtain different outcomes. Since we give low time limit, cuts fail to show their effectiveness up to medium size instances. Yet, for high load, cuts and dimensionality reduction techniques prove their efficiency in terms of results. We tabulate average results in Table 5.2. It suggests that Z_f rises as we increase load but gaps do not follow the same pattern. Although Model2 Z_f mean value is less than Model under low load, box plot produces opposite result in terms of medians. So this happen as a result of dispersion of Model2 data which can be seen in Figure 5.18. Finally, we draw a conclusion that gaps are very large even under Heur+Model2, Heur+Model2. Yet, Z_f values of Heur+Model2 are very promising because it diminishes Z_f dramatically. Z_f ratios of Model to Heur+Model2 are 77.60, 66.78 and 4.8 for low, medium and large load systems, respectively. Ratio decreases as load increases but it is almost 5 times better than Z_f of Model even under lowest ratio.

5.9.2. 10 mins Time Limit

Model2 is not better than Model according to Z_f values in Table 5.2 up to medium load. This may happen because Model2 might not show its efficacy under two mins time

Table 5.2. Time comparisons of methods under 2 mins time limit

Load	Methods	Z_f	Gap(%)
Low	Model	3864	99.91
	Model2	3766.7	89.89
	Heur+Model2	49.6	81.1
Medium	Model	10998.2	99.82
	Model2	11698.2	99.83
	Heur+Model2	164.7	85.27
High	Model	45577.1	98.47
	Model2	39354.4	95.21
	Heur+Model2	9493.9	80.21

limit. We need more time to reach a better conclusion about these methods. Hence, we now focus on comparison between Model2 and Model with 10 mins time limit. In terms of gaps, Figure 5.25 is similar to 5.19. For Z_f, Model2's median is significantly lower than Model according to Figure 5.24 whereas it was higher in Figure 5.18. So, we see a meaningful improvement under low load with 10 mins time limit.

We now investigate Z_f in medium load with 10 mins time limit. We show results in Figure 5.26. Median value of Model2 is lower than all values of Model. It was much worse in Figure 5.20 under two mins time limit. Besides, Model2 gap in Figure 5.27 is significantly better than Model because it has wider range than Model whereas it was almost same in Figure 5.19. Hence, we observe a great deal of development in Model2 performance under 10 mins time limit compared to 2 mins.

We treat methods under high load and 10 mins time limit. We show results with reference to Z_f value in Figure 5.28. Model2 median is lower than Model and Model2 is less scattered than Model. Its range is almost one half of Model so it is more reliable. In comparison with Figure 5.22, although median value is same, its box-plot

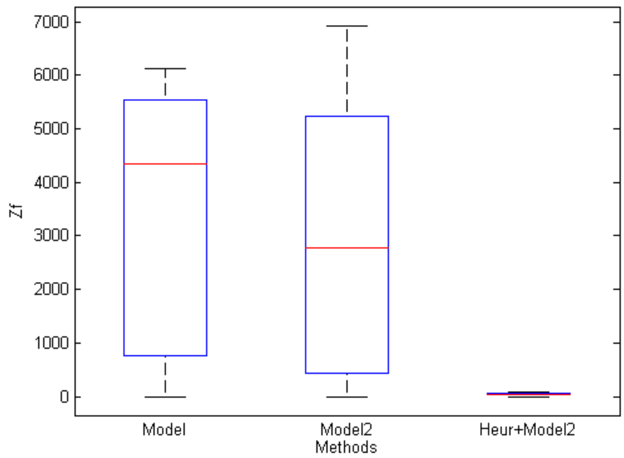

Figure 5.24. Low Load System Comparison under Z_f with 10 mins Time Limit

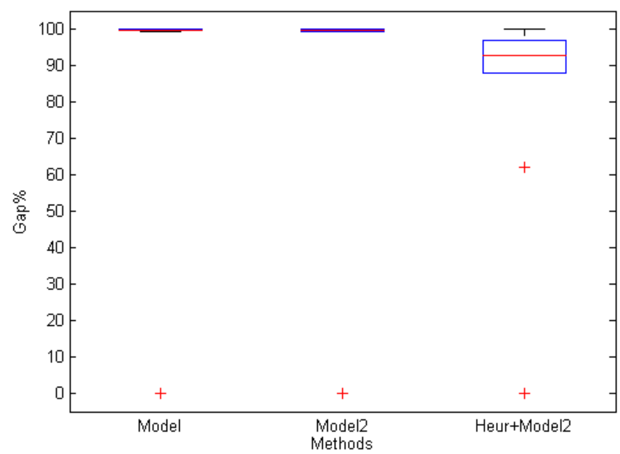

Figure 5.25. Low Load System Comparison under Gap(%) with 10 mins Time Limit

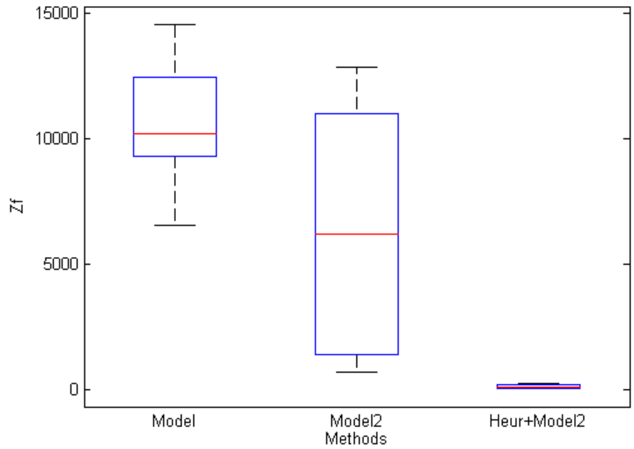

Figure 5.26. Middle Load System Comparison under Z_f with 10 mins Time Limit

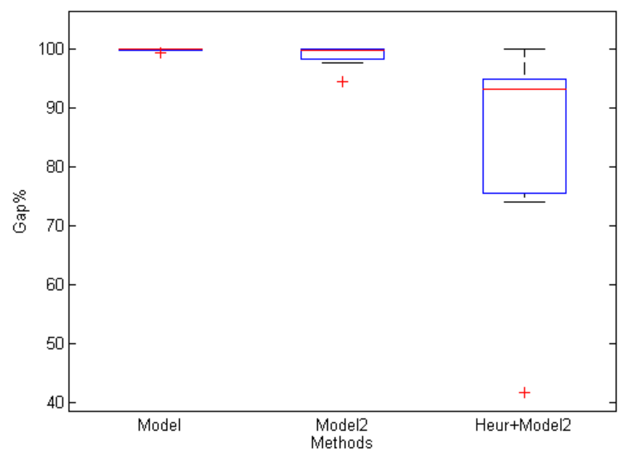

Figure 5.27. Middle Load System Comparison under Gap(%) with 10 mins Time Limit

range is diminished profoundly. Concerning gap, results in Figure 5.29 is similar to Figure 5.23. Yet, we detect a little improvement because although median is almost same, box-plot is more right-tailed. In general, although Model2 outweighs Model with 10 mins time limit under this load, difference between 2 mins and 10 mins time limit regarding performance indicators are slim.

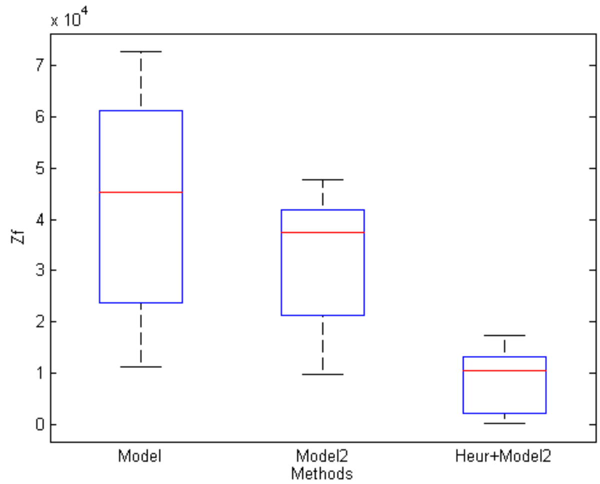

Figure 5.28. High Load System Comparison under Z_f with 10 mins Time Limit

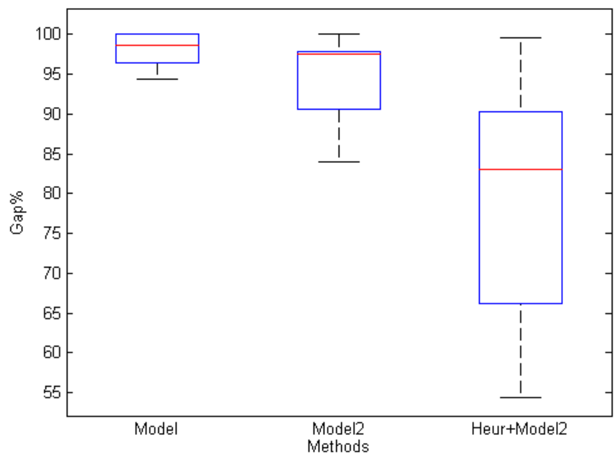

Figure 5.29. High Load System Comparison under Gap(%) with 10 mins Time Limit

We generally infer from box-plots that Model2 performs much better than Model with 10 mins time limit than 2 mins under low and medium loads but difference is limited under high load. It is expected because higher the load, harder it becomes to solve the problem. Furthermore, Heur+Model2 as always show the best performance in all performance measures. We also portray mean outcomes in Table 5.3. It denotes that Model2 reduces Model Z_f with 14.3%, 40.89% and 21.14% in low, medium and large loads, respectively. It was worse than Model in medium load in Table 5.2. Yet, now Model2 shows the best performance diminishing Model Z_f with increased time limit under this load. Besides, it yields less gaps than Model under both time limits. Therefore, we deduce that model improvements diminishes Z_f of the model while ensuring a better solution quality. Nevertheless, increased time limit does not yield better Z_f values for Heur+Model2 under low and high load and barely improves middle load with 7% compared to results in Table 5.2. Since it is the best method and time increment could not manage to lead to significantly better results, we will use Heur+Model2 in comprehensive simulations with two mins time limit.

Table 5.3. Time comparisons of methods under 10 mins time limit

Load	Methods	Z_f	Gap
Low	Model	3475.5	89.87
	Model2	2977.6	89.83
	Heur+Model2	49.6	81.71
Medium	Model	10728.78	99.8
	Model2	6340.89	98.83
	Heur+Model2	152.67	83.6
High	Model	44241.1	97.97
	Model2	34888.7	94.01
	Heur+Model2	9493.9	76.53

5.10. Experiment Plan

We build a discrete-event simulation environment to demonstrate the performance of the solution method under several solution parameters. We employ C++ programming language to build the environment, develop heuristics and solve problems. We call CPLEX 12.7 from C++ environment while solving models. We run simulations on a computer with Intel(R) Xeon(R) CPU(E5620 2.40 Ghz) processor and 12 GB RAM, running a 64-bit Windows Server 2008 R2 Enterprise operating system.

We have four factors that we will use in every setting. These are SLA, m, Π and v. We consider all of them in preventive maintenance and add τ to them in preventive and corrective maintenance. We have Heur+Model2 and two alternative methods to handle stochasticity. Hence, we experiment on three methods and test them under preventive and corrective maintenance. Totally, we aim at running $2^4 + 2^5 3 = 112$ parameter combinations. We show the experiment design in Table 5.4.

Table 5.4. Simulation experimental factors

	Levels		
Factors			
SLA	Case 1	Case 2	
m	2	3	
Π	600	720	
v	0.10	0.20	
τ	2508	10068	
Maintenance	Prev	Prev+Cor	
Method	Heur+Model2	Buffer($\zeta = 10$)	Anticipation

We name each parameter combination as scenario. For every scenario, we build a particular system. Under every scenario, we utilize mersenne-twister random number generator while producing problem parameters. We selected Heur+Model2 as the solution method according to Section 5.9. For each scenario, we run Heur+Model2 dynamically in a rolling horizon fashion. We give two minutes time limit to solve each problem in the horizon and fetch best feasible solution obtained till that time.

While generating job processing times, we set $p = 8$ for preventive jobs, $p_{cor} = 10$ for corrective jobs and variability as $\kappa = 2$ hrs. In a model, we determine coefficients as follows: $\sigma = 4$, $\epsilon = 5$, $\gamma = 10$. Tardiness coefficient in anticipation method is set as $\omega = 4$. Preventive jobs are created according to following parameters: $\beta = 672$, $\varrho = 1.5$, $\theta_1 = 2$. Fleet size is taken as $n = 105$. Given $\phi = 1$, first τ value in table is realized as 2508 whereas $\phi = 0.25$ results in 10068. Finally, in order to generate SLA, we fix these coefficients: $a = 0.95$, $d = 0.10$, $e = 0.05$, $f = 0.05$.

We determine seven KPIs. They are listed with their abbreviations as follows:

(i) Prevearl: Total Preventive Earliness
(ii) Prevtard: Total Preventive Tardiness
(iii) Cortard: Total Corrective Tardiness
(iv) Ymaxviol: Total Y_j^{max} violation
(v) Slaviol: Total SLA violation
(vi) Load: Shop Load %
(vii) Wait: Total Wait Time

We calculate them according to first day of the scheduling period. Results of first five are directly fetched from model. Shop load and total wait times are determined by hand. Shop Load % is an indicator that how much the capacity is used on first day. We calculate it through the following formula: $Load\% = \frac{\sum_{j \in CR} \sum_{t<24} \hat{a_{jt}}}{24m}(100)$ where $\hat{a_{jt}}$ is a realized value of respective variable. Finally, wait time is an indicator that how much time vehicles are stayed idle at the track due to crossings. It is calculated via sum over wait times of each job at the track after maintenance finish.

6. RESULTS AND DISCUSSIONS

6.1. Preventive Maintenance

We use one-way anova tests to test the effect of every factors. To statistically compare different levels of the factors, we make use of two sample t-test with an assumption that they come from population with unequal variances. Each factor level constitutes a sample group from batch means. For instance, batch means with $m = 2$ forms first group whereas $m = 3$ shapes second one. Since this group size is large, they are always distributed normal according to central limit theorem [79]. In this work, we usually employ upper tailed t-test, so it can be stated as follows: Let μ_1 and μ_2 be true population means. Then, null hypothesis is $H_0 : \mu_1 = \mu_2$ is compared with one sided alternative where $H_1 : \mu_1 > \mu_2$. We set $\alpha = 0.05$.

Let $\bar{x_1}$ and $\bar{x_2}$ be the first and second sample means. n_1 and n_2 are sample size of first and second means. S_1 and S_2 be standard deviation of first and second sample, respectively. Then we calculate test statistic TS and degree of freedom, df, as follows:

$$TS = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \tag{6.1}$$

$$df = \frac{(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2})^2}{\frac{(S_1^2/n_1)^2}{n_1 - 1} + \frac{(S_2^2/n_2)^2}{n_2 - 1}} \tag{6.2}$$

According to this case, we reject H_0 if $TS > t_{1-\alpha}$. Instead, we employ p-value in this work. Because t-statistic comparison equals to checking p-value against α [80]. As a definition, it is defined as follows [81]: "p-value is the smallest level of significance where we can still reject H_0". By means of this value, we reject H_0 only if $p < \alpha = 0.05$. Note that lower tailed t-test examines H_1 using $<$ inequality in H_1. We perform all these tests using Matlab R2011a.

6.1.1. Effect of Track Capacity

We tabulate average results in Table 6.1. Change of m does not affect Slaviol because it is zero in all experiments. We employed upper tailed t-test where first sample depends on $m = 2$ and second $m = 3$. Table 6.1 demonstrates significant decrease in Prevtard, Prevearl and Load values and they are validated statistically because their $4.91x10^{-6}, 7.72x10^{-12}, 1.62x10^{-10}$ p-values . We can say that capacity expansion brings about more freedom to schedule jobs. So, it generally leads to improvements in performance measures. According to the table, Wait is increased. When we employ lower tailed t-test, we reject null hypothesis with p = $1.61x10^{-36}$. So, we infer that increase in m boosts Wait as well. It happens as extra crossings exist because of capacity expansion. For Ymaxviol, although m decreases Ymaxviol significantly with p-value 0.0496, it is a very tiny difference with respect to the table.

Table 6.1. KPI values depend on m change

m	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Wait (vehs)	Slaviol (hrs)	Load (%)
2	5.70	11.55	0.011	2.70	0	72.12
3	0.81	6.85	0.003	9.42	0	57.82

6.1.2. Effect of Service Level Agreement

Table 6.2 shows average results. In the table, difference between results seem slim. In order to investigate these deviations significantly, we make use of lower tail t-test where first sample is Case 1 and second is Case 2. For Ymaxviol, Wait and Load, we fail to reject H_0 with p-values 0.8639, 0.3673 and 0.4920, respectively. According to table, we see a decline in Prevearl and Prevtard in contrast to our anticipation. Nevertheless, once we perform an upper tail t-test, they are not found significant with p-values 0.2002 and 0.2098, respectively. We actually expected before experiments that Case 2 impairs KPI values due to its variability. Yet, according to p-values, we fail to reject H_0. Average load is 65% in all scenarios, so it is probably the reason behind this result.

Table 6.2. KPI values depend on SLA change

SLA	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Wait (vehs)	Slaviol (hrs)	Load (%)
Case 1	3.57	9.64	0.01	5.95	0	64.95
Case 2	2.94	8.76	0.005	6.17	0	65.00

6.1.3. Effect of Preventive Interarrival Times

Table 6.3 displays average results according to Π change. We observe a very close outcomes in Ymaxviol and Wait. We utilize upper tailed t-test where first sample depends on $\Pi = 600$ and second $\Pi = 720$. Statistical tests yield that we fail to reject H_0 with Ymaxviol with p=0.2917 and Wait with p=0.2856. Slaviol is zero, so no need to make a test for it. $\Pi = 720$ outweighs $\Pi = 600$ under Prevtard, Prevearl and Load with p-values 0.0029, 0.0027 and $2.65x10^{-6}$, respectively. Since rise in Π means decrease in maintenance job arrival per scheduling period, we guessed some decline in KPI values. Usually results confirm our guesses. Although the change reduces tardiness, Ymaxviol is not influenced considerably. The reason behind this that average Ymaxviol is very low compared to other performance measures according to Table 6.3.

Table 6.3. KPI values depend on Π change

Π	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Wait (vehs)	Slaviol (hrs)	Load (%)
600	4.29	10.68	0.008	6.24	0	70.33
720	2.22	7.72	0.006	5.88	0	59.62

6.1.4. Effect of Variability Factor in Preventive Intervals

The effect of v change is presented in Table 6.4. According to it, there is no much difference except Prevearl. We perform upper tailed t-test where first sample depends on $v = 0.1$ and second $v = 0.2$. According to tests, $v = 0.2$ performs better under

Prevearl with p=0.0225. For Ymaxviol, Prevtard, Wait and Load, we fail to reject H_0 with p-values 0.1361, 0.2585, 0.5738 and 0.2321, respectively. Prevtard is also decreased as a mean but it does not turn out to be significant. Earliness is decreased as estimated because larger allowance decrease risk of a vehicle being out of service. Thus, optimization model drags start time right in time while not influencing Prevtard.

Table 6.4. KPI values depend on v change

v	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Slaviol (vehs)	Wait (hrs)	Load (%)
0.1	4.004	9.55	0.01	6.00	0	65.86
0.2	2.51	8.85	0.005	6.12	0	64.09

6.1.5. Regression Analysis

We append this part to analyze factor effects using multiple linear regression. We disregard their interaction and only focus on factors and regard them as variables. Let we set variables as follows:

x_1 : 0 if $m = 2$, else 1
x_2 : 0 if SLA = Case 1, else 1
x_3 : 0 if $\Pi = 600$, else 1
x_4 : 0 if $v = 0.1$, else 1

We execute linear regression on Minitab. Slaviol is very low so we ignore it. No variable is found significant in the regression under Ymaxviol. Hence, we do not write the equation. Under Prevtard, the equation is as follows: $13.8 - 4.70x_1 - 0.875x_2 - 2.96x_3 - 0.703x_4$. So m and Π are found to be significant variables. Under Prevearl, we obtain following equation: $7.80 - 4.89x_1 - 0.630x_2 - 2.07x_3 - 1.49x_4$. According to it, m is the most significant variable and it is followed by Π. For Wait, regression yields following result: $2.72 + 6.71x_1 + 0.216x_2 - 0.361x_3 + 0.119x_4$. Here only m has a remarkable impact. Lastly for Load, we have following equation: $78.3 - 14.3x_1 + 0.05x_2 - 10.7x_3 - 1.77x_4$. According to it, m is the most profound variable and Π

comes after it. In consequence, we could see that m influences almost all KPI values profoundly. Second Π and third v changes KPIs in the order of decreasing effect.

6.1.6. Marginal Analysis

We have compared $\Pi = 600$ and $\Pi = 720$ systems concerning KPI values so far. In this section, we first analyze interaction effects under these systems then add $\Pi = 480$ as a third level for Π factor. If we take a look at in Table 6.3, there exists no much difference between KPI values of $\Pi = 600$ and $\Pi = 720$. We wonder how decreasing Π from $\Pi = 600$ affects KPIs. Hence, we set $\Pi = 480$ to observe outcomes. Since we increase arrival frequency, we expect rise in KPIs.

We use lower tailed t-test where first sample is $\Pi = 720$ and second one is $\Pi = 480$. We conclude that $\Pi = 480$ increases Ymaxviol, Prevtard, Prevearl and Load remarkably with 0.0002, $6.69x10^{-9}$, $2.79x10^{-7}$, $1.36x10^{-7}$ p-values, respectively. We see no difference in Slaviol and Wait with 0.159 and 0.447 p-values, respectively. We then apply lower tailed t-test where first sample is $\Pi = 600$ and second one is $\Pi = 480$. $\Pi = 480$ raises Ymaxviol, Prevtard, and Prevearl considerably with 0.0002, $1.85x10^{-6}$, 0.038 p-values, respectively. There is no difference in Slaviol, Wait and Load KPIs with 0.159, 0.715, 0.239 p-values, respectively. The results can be seen in Table 6.5. It is interesting to note that once we change $\Pi = 600$ to $\Pi = 480$, mean Load rises from 70.33 to 84.52 but there is no significant difference between these values according to t-test results. Yet, it lifts Prevtard and Ymaxviol too much so system gets beyond control.

Table 6.5. KPI values depend on Π change

Π	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Wait (vehs)	Slaviol (hrs)	Load (%)
480	10.2	51.067	7.32	5.74	0.002	84.52
600	4.29	10.68	0.008	6.24	0	70.33
720	2.22	7.72	0.006	5.88	0	59.62

6.1.7. Multifactor Effect

We inspect interaction of problem parameters and portray results in Figure 6.1. Note that S denotes Case 1 while F corresponds to Case 2. Generally, we see that when $v = 0.1$, Case 1 SLA has a worse impact on Prevtard and Ymaxviol than Case 2 whereas $v = 0.2$ changes this relationship oppositely according to these KPIs. This can be seen clearer once $\Pi = 480$. Regarding Slaviol, only $SLA = F, m = 2, \Pi = 480, v = 0.1$ system has a tiny Slaviol according to Figure 6.2. If we compare loads, we can observe from Figure 6.3 that $\Pi = 480$ raises loads close to 15% and consists of loads close to 100%. Besides, figure and statistical tests suggest that rise of m and Π leads to a decrease in it.

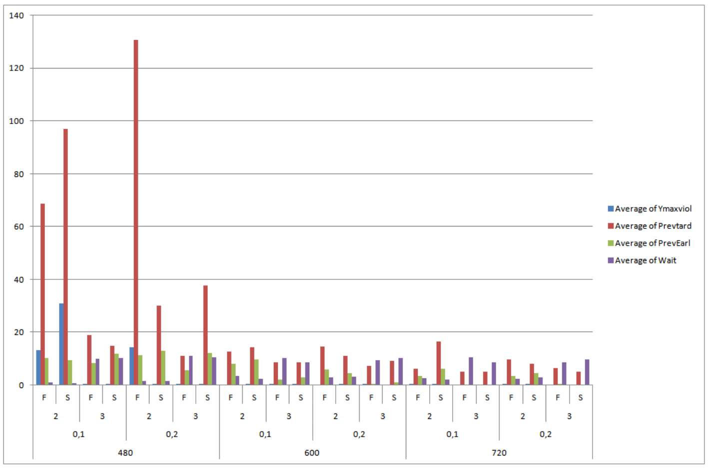

Figure 6.1. Four Performance Indicators under Preventive Maintenance

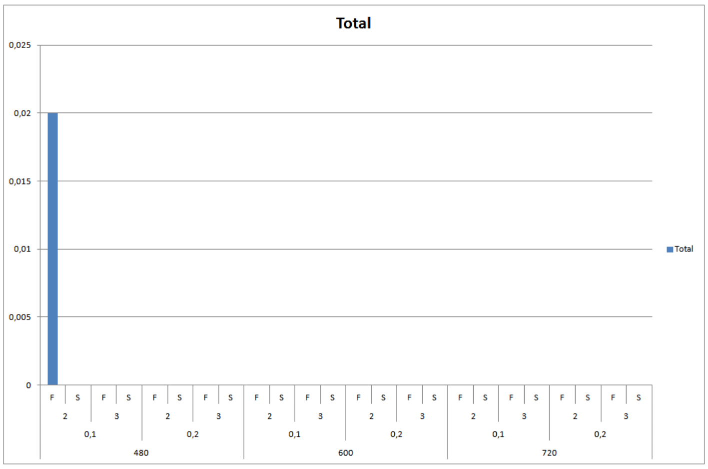

Figure 6.2. Slaviol under Preventive Maintenance

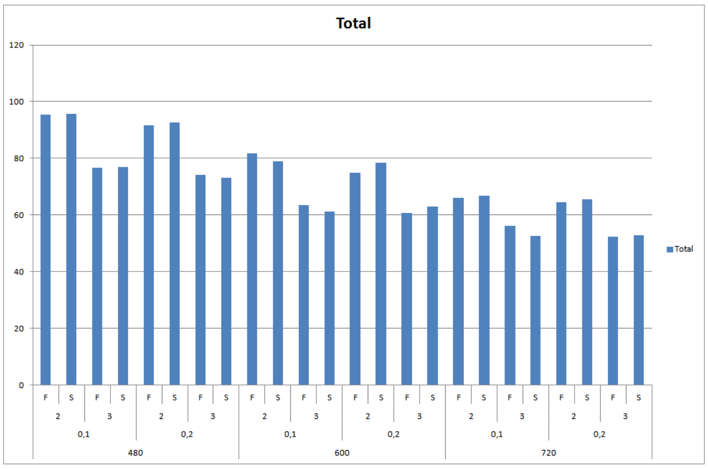

Figure 6.3. Shop Load % under Preventive Maintenance

6.2. Preventive and Corrective Maintenance

We now analyze the system in the existence of vehicle breakdowns. We show the results in Figure 6.4.

6.2.1. Effect of Track Capacity

We show results in Table 6.6. We see decreases in all values but Wait. We employ upper tailed t-test where first sample depends on $m = 2$ and second $m = 3$. We fail to reject H_0 in Slaviol and Cortard with 0.0553 and 0.0623 p-values, respectively. Lower tail t-test for Wait yields that jobs wait more under $m = 3$ with $p = 6.96x10^{-79}$. $m = 3$ beats $m = 2$ under Ymaxviol, Prevtard, Prevearl and Load with 0.003, $1.30x10^{-10}, 1.69x10^{-8}, 1.82x10^{-26}$ p values. It could be stated that we obtain similar results with base preventive case.

Table 6.6. KPI values depend on v change under preventive and corrective maintenance

m	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Wait (hrs)	Slaviol (vehs)	Load (%)
2	2.73	50.18	11.92	10.63	2.44	0.73	83.11
3	0.82	12.49	0.13	9.27	10.07	0	66.06

6.2.2. Effect of Service Level Agreement

We show results in Table 6.7. We see a decline in Prevearl but a considerable growth in Ymaxviol , Prevtard and positive value in Slaviol. We utilize lower tail t-test like we did in preventive maintenance. It turns out that only raise in Ymaxviol is significant with p=0.0482. We fail to reject H_0 in Prevtard with p=0.0734. Yet, we can say that change in SLA causes almost a significant increase in Prevtard. Change in Slaviol is failed statistically with p=0.0553. We reject a significant increase for Cortard, Wait and Load values with 0.1141, 0.3679 and 0.2495 p-values, respectively. Yet, Case 2 decreases Prevearl with p = 0.0048. We come into conclusion that variability in SLA has a bad influence on our system unlike preventive maintenance.

Table 6.7. KPI values depend on SLA change under preventive and corrective maintenance

SLA	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Wait (hrs)	Slaviol (vehs)	Load (%)
Case 1	2.23	26.99	3.19	9.42	6.17	0	73.99
Case 2	1.32	35.68	8.86	10.49	6.34	0.73	75.18

6.2.3. Effect of Preventive Interarrival Times

We display results in Table 6.8. We use upper tailed t-test where first sample depends on $\Pi = 600$ and second $\Pi = 720$. As a result of tests, we fail to reject H_0 in Slaviol, Cortard and Wait with p-values 0.0553, 0.4480 and 0.1509, respectively. $\Pi = 720$ outweighs $\Pi = 600$ under Ymaxviol, Prevtard, Prevearl and Load with p-values 0.0051, 0.0001, 0.0040 and $4.77x10^{-9}$, respectively. Here, increase in Π decreases Ymaxviol significantly which did not occur at preventive maintenance. Besides, Cortard value is not influenced by the change because corrective jobs are always superior than preventives regarding priority.

Table 6.8. KPI values depend on Π change under preventive and corrective maintenance

Π	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Wait (hrs)	Slaviol (vehs)	Load (%)
600	2.24	42.60	10.40	10.01	6.52	0.73	79.50
720	1.32	20.07	1.65	9.89	6.00	0.73	69.66

6.2.4. Effect of Variability Factor in Preventive Intervals

We display results in Table 6.9. We perform upper tailed t-test in which first sample depends on $v = 0.1$ and second $v = 0.2$. According to tests, we could not ascertain any difference between those means in Slaviol, Cortard and Wait with p-values 0.0553, 0.7011 and 0.4227. $v = 0.2$ diminishes Ymaxviol, Prevtard, Prevearl and Load with 0.0004, 0.0040, 0.0033 and 0.0237 p-values, remarkably. This can be observed from Table 6.9. We deduce that all preventive maintenance based performance measures are affected. We explained Prevearl in preventive maintenance. We know that larger interval allowance ensures smooth arrival of preventive jobs to maintenance hangar. So, once interval allowance is diminished to 0.1, Prevtard and Ymaxviol rise. Because due time become tighter that leads to frequent maintenance begin lateness. As a consequence, jobs are tried to be taken to maintenance as early as possible that escalates daily hangar load significantly.

Table 6.9. KPI values depend on v change under preventive and corrective maintenance

v	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Wait (hrs)	Slaviol (vehs)	Load (%)
0.1	2.25	39.27	11.71	9.72	6.31	0.73	76.32
0.2	1.30	23.40	0.34	10.19	6.21	0	72.85

6.2.5. Effect of Mean Time Between Failures

In order to compare KPIs under different cases, we compare τ effect with preventive maintenance. Once we utilize upper tailed t-test with selecting $\tau = 10068$ as first sample and preventive maintenance as second, we find that Prevtard and Load metrics are significantly higher with $1.09x10^{-4}, 8.25x10^{-3}$ p-values, respectively. We fail to reject H_0 in Ymaxviol, Prevearl and Wait with p-values 0.111, 0.952 and 0.436, respectively. Once we use lower tailed t-test, Prevearl in preventive maintenance is found significantly higher with p=0.0476. When we compare $\tau = 2508$ and preventive maintenance, $\tau = 2508$ ascends Ymaxviol, Prevtard and Load measures meaningfully with 0.0002, $1.67x10^{-11}, 8.89x10^{-19}$, respectively. It fails to ascend H_0 in Slaviol and Wait with 0.055 and 0.256 p-values, respectively. Yet, $\tau = 2508$ reduces Prevearl meaningfully with p = $3.41x10^{-7}$ with respect to lower tailed t-test.

We now utilize upper tailed t-test in which first sample depends on $\tau = 2508$ and second $\tau = 10068$. As a result of these tests, we fail to reject H_0 in Slaviol, Prevearl and Wait KPIs with 0.0553, 1 and 0.3060 p-values. Nevertheless, left tailed t-test yields that $\tau = 2508$ lowers Prevearl meaningfully with p=$2.63x10^{-5}$. Furthermore, $\tau = 2508$ escalates Ymaxviol, Prevtard, Cortard and Load considerably with 0.0002, $1.96x10^{-9}, 1.35x10^{-55}, 8.35x10^{-11}$ p-values, respectively.

We tabulate average results in Table 6.10. If we take a distinction between Prev and $\tau = 10068$ into consideration, there is a slim difference under Prevtard and Load even though there exists a significant difference between these factors. However, provided that $\tau = 2508$, KPIs worsened profoundly. Since average load increase from 69.15 to 80.02 on account of τ descend, other KPIs are degraded as well. We see an eccentric ascend in Prevtard and Ymaxviol values with respect to $\tau = 10068$ whereas Prevearl is halved. In general, Prevtard goes up because corrective jobs are superior to preventives in terms of priority so Prevearl plunges as a consequence of it. Cortard is increased significantly as expected because more corrective jobs comes to system.

Table 6.10. Comparison of KPIs under preventive maintenance and preventive and corrective maintenance

	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Wait (hrs)	Slaviol (vehs)	Load (%)
Prev	3.26	9.20	0.01	0	6.06	0	64.97
$\tau = 10068$	2.48	13.78	0.05	3.81	6.13	0	69.15
$\tau = 2508$	1.08	48.89	12.00	16.10	6.38	0.73	80.02

6.2.6. Regression Analysis

As we did in preventive case, we perform a linear regression. We define variables as follows:

x_1 : 0 if $m = 2$, else 1
x_2 : 0 if SLA = Case 1, else 1
x_3 : 0 if $\Pi = 600$, else 1
x_4 : 0 if $\tau = 2508$, else 1
x_5 : 0 if $v = 0.1$, else 1

Under Ymaxviol, we obtain following equation: $25.1 - 11.8x_1 + 5.66x_2 - 8.75x_3 - 11.9x_4 - 11.4x_5$. We can infer that SLA. τ and v are most effective parameters determining Ymaxviol. For Prevtard, we get this result: $82.6 - 37.7x_1 + 8.68x_2 - 22.5x_3 - 35.1x_4 - 15.9x_5$. It means that m, Π and τ are most effective parameters. For Cortard, the equation is like this: $16.1 - 1.36x_1 + 1.07x_2 - 0.116x_3 - 12.3x_4 + 0.467x_5$. Only τ has impact on the KPI. For Prevearl, we obtain that equation: $3.42 - 1.91x_1 - 0.908x_2 - 0.932x_3 + 1.41x_4 - 0.952x_5$. Here, m and τ are the most effective variables that ascertain Prevearl. For Wait, analysis yields the equation: $2.79 + 7.63x_1 + 0.169x_2 - 0.516x_3 - 0.254x_4 - 0.097x_5$. Equation suggest that only m is effective variable that influences Wait. For Load, the analysis brings about following equality: $94.6 - 17.1x_1 + 1.19x_2 - 9.84x_3 - 10.9x_4 - 3.47x_5$. m, Π and τ are most powerful parameters. For all KPIs, we deduce that m and τ are the most crucial parameters.

6.2.7. Multifactor Effect

We show detailed KPI change according to four factors in Figure 6.4. As we observe from picture, parameter values become substantially larger if we change τ to 2508. If we look at KPI results under that value, it seems that $m = 3$ diminishes values profoundly. After that, change in Π and v leads to similar effects.

Most congested system, $\tau = 2508, m = 2, \Pi = 600, v = 0.1$, is loaded close to 100 % according to Figure 6.6. Figure 6.4 shows that SLA type change from Case 1 to Case 2 increases Ymaxviol and Prevtard dramatically. Besides, it is the only system that Slaviol is different from zero which is observed in Figure 6.5. This is interesting, because this change generally has no considerable impact on the system as we observe in different settings of Figure 6.4 and preventive maintenance but here there is an erratic increase in relevant KPIs. Therefore, we infer that Case 2 could ruin KPIs enormously under extreme system load.

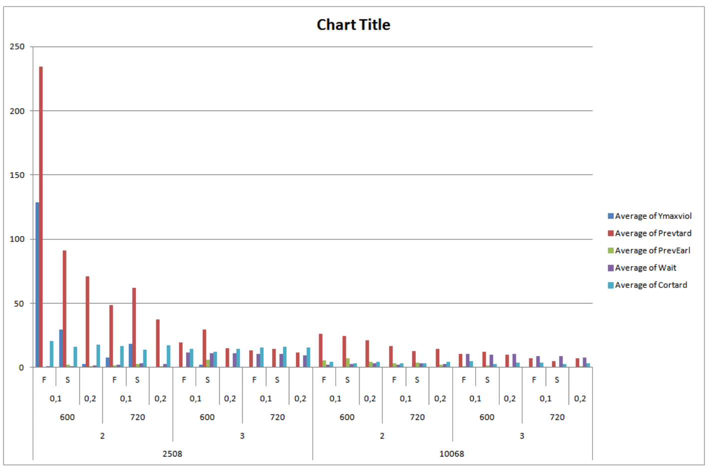

Figure 6.4. Four Performance Indicators under Preventive and Corrective Maintenance

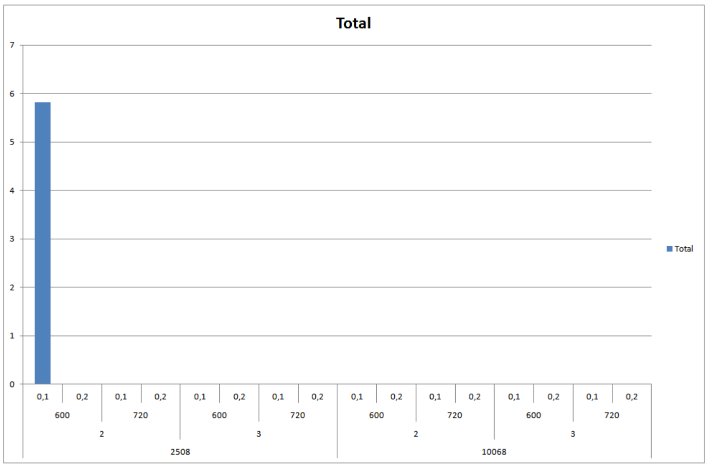

Figure 6.5. Slaviol under Preventive and Corrective Maintenance

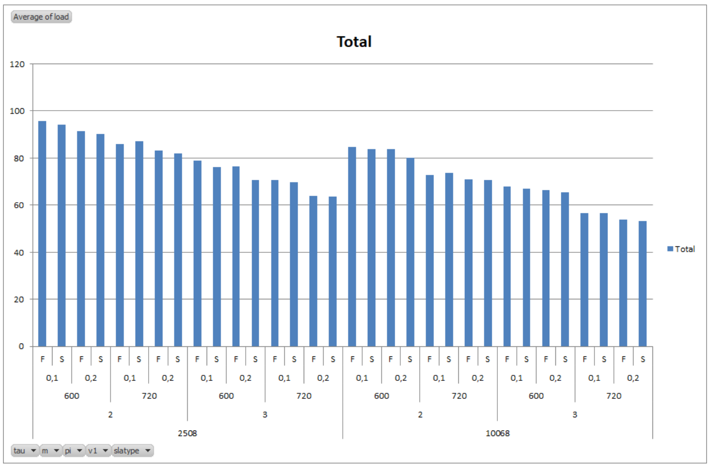

Figure 6.6. Shop Load % under Preventive and Corrective Maintenance

6.2.8. Comparison at High Load

In this section, we compare $\mathit{\Pi} = 480$ and $\tau = 2508$ because they are close to 100% according to Figure 6.9 at their most congested system. Note that $\tau = 1$ points out preventive maintenance. If we take a look at most loaded system $\tau = 2508, m = 2, \mathit{\Pi} = 600, v = 0.1$, $\tau = 2508$ in Figure 6.7, it drastically increases Prevtard and Ymaxviol under $SLA = F$ compared to same system with $\mathit{\Pi} = 480$. Yet, sla type change to S diminishes values considerably in $\tau = 2508$ system whereas it increases these values in same preventive system with $\mathit{\Pi} = 480$. Besides, we observe a big difference in Prevtard between F and S under $\tau = 1, m = 2, \mathit{\Pi} = 480, v = 0.2$ system whereas there is no such a big difference if we alter τ to 2508. So, we deduce that SLA influences two systems differently given they have close loads. Generally, we see Prevearl bars to some extent provided that $\mathit{\Pi} = 480$ but it diminished once we alter parameter to $\tau = 2508$. Finally, we see positive and considerable Slaviol in most loaded system with $SLA = F$ regarding Figure 6.8.

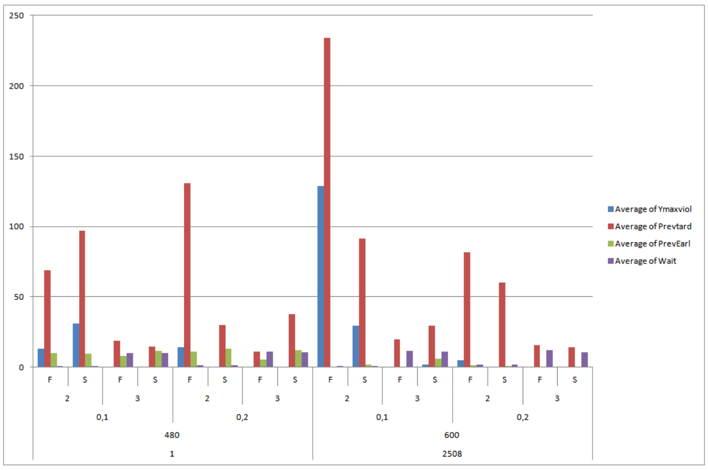

Figure 6.7. Four Performance Indicators under $\mathit{\Pi} = 480$ vs $\tau = 2508$

We now compare these systems statistically. We perform upper tailed t-test in which first sample is $\mathit{\Pi} = 480$ and second one is $\tau = 2508$. We ascertain that Prevearl and Load under $\mathit{\Pi} = 480$ is significantly larger with $1.78x10^{-15}$ and 0.0054 p-values, respectively. We fail to reject H_0 in Slaviol, Ymaxviol, Prevtard and Wait with 0.944, 0.887, 0.399 and 0.833 p-values, respectively. We present results in Table 6.11. If we

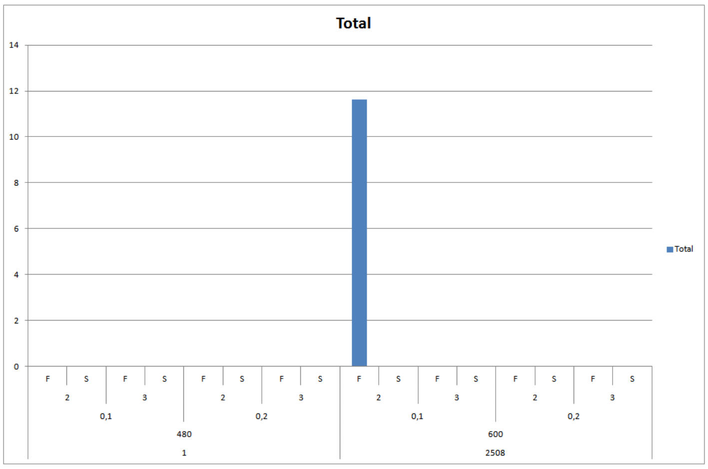

Figure 6.8. Slaviol under $\Pi = 480$ and $\tau = 2508$

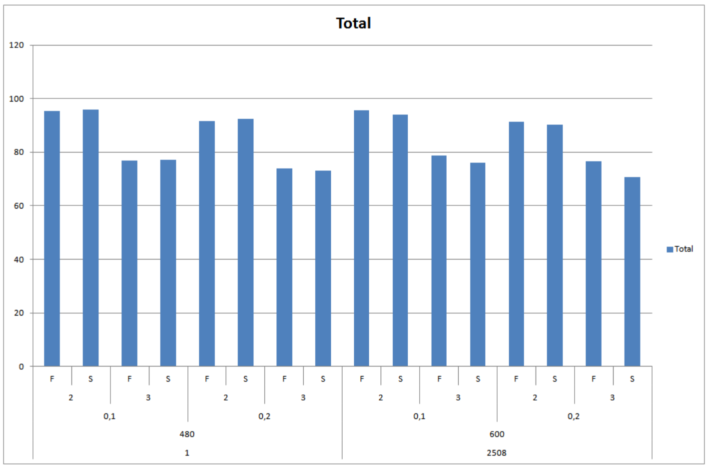

Figure 6.9. Shop Load % under $\Pi = 480$ vs $\tau = 2508$

use a lower tailed t-test, we fail to reject H_0 for Slaviol with p=0.056. It is so close to $\alpha = 0.05$. This results from most congested system that can be observed in Figure 6.8. We have two inferences from these results. First, although Load seems similar with reference to Figure 6.9, it turns out that there exists a significant difference. Second, since Prevearl is diminished despite of less load under $\tau = 2508$, this validates our hypothesis that corrective jobs push preventive ones right in time.

Table 6.11. Comparison of two systems at high load

	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Wait (hrs)	Slaviol (vehs)	Load (%)
$\Pi = 480$	10.2	51.067	7.32	0	5.74	0.002	84.52
$\tau = 2508$	1.08	48.89	12.00	16.10	6.38	0.73	80.02

6.3. Comparison of Solution Methods

Since we develop buffer time and anticipation method to combat stochasticity in the system, we analyze their effects on preventive and corrective system. First, we examine them with reference to pure Heur+Model2 and with respect to each other under every KPI. We name buffer method and anticipation as Method 1 and 2, respectively.

When we compare Method 1 and Heur+Model2 where Heur+Model2 is first sample and 1 is second, we found a significant difference between those values in Prevtard, Load and Wait with 0.0036, 0.0165 and $3.87x10^{-9}$ p-values under upper tailed t-test, respectively. Yet, Method 1 rises Prevearl significantly with p= $9.011x10^{-94}$ according to lower tailed t-test. We unable to find a difference in Slaviol with p=0.06 and Ymaxviol with p=0.109. Second, we consider distinction between Method 2 and 0. As a consequence of two sample t-test, we come into conclusion that Method 2 outweighs 0 in terms of Prevtard and Wait measures with 0.0012 and $3.04x10^{-2}$ p-value, respectively. However, Method 2 boosts Prevearl significantly with p=$1.797x10^{-48}$ according to lower tailed t-test. There is no significant difference between these methods under Slaviol, Ymaxviol, Cortard with 0.164, 0.159 and 0.422 p-values, respectively. Finally, if we compare Method 2 and 1 by means of upper tailed t-test where Method 2 is first sample and 1 is second, we find out that Method 1 diminishes Wait and Load significantly with $2.61x10^{-5}$ and 0.0087, respectively. Yet, it escalates Prevearl profoundly with p = $1.25x10^{-28}$. We could not discern these methods in Slaviol, Ymaxviol, Prevtard and Cortard with 0.051, 0.390, 0.614 and 0.612 p-values, respectively. Average results can be seen in Table 6.12. If we prioritize Prevtard, these methods tackle stochasticity better than Heur+Model2. Among themselves, Method 2 diminishes Prevearl as well but increase Wait. Hence, we left the choice to decision maker.

Table 6.12. Method comparison

Methods	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Wait (hrs)	Slaviol (vehs)	Load (%)
Heur+Model2	1.78	31.34	6.03	9.95	6.26	0.36	74.59
Method 1	18.37	20.84	3.56	10.01	4.46	0.01	71.81
Method 2	9.90	19.84	3.98	9.83	5.63	0.13	74.80

Now, we focus on KPI changes of methods under τ. We define Z which is a weighted sum of MILP model based KPIs according to objective function coefficients of MILP model. First of all, we analyze results when τ = 10068. We perform left tailed t-test where first sample is Heur+Model2 and second one is Method 1. We then employ upper tailed t-test by changing the relationship. According to them, there is no significant difference between methods in Ymaxviol and Cortard in both tests, they are failed to be rejected with 0.126 and 0.841, respectively in left tailed test. Method 1 significantly raises Prevearl and Z value with $1.81x10^{-76}$ and $2.10x10^{-6}$ p-values, respectively. On the other hand, if we employ upper tailed t-test, it diminishes Prevtard, Wait and Load with $2.4x10^{-6}$, $3.06x10^{-7}$ and 0.039 p-values, respectively. We execute left tailed t-test in which first sample is Heur+Model2 and second one is Method 2. Method 2 reduces Prevtard and Wait profoundly with $2.53x10^{-8}$ and 0.031 p-values, respectively. There is no significant difference between treatments in Ymaxviol, Cortard, Load and Z with 0.478, 0.39, 0.74 and 0.89 p-values, respectively. Yet, Method 2 escalates Prevearl with p = $8.89x10^{-34}$ regarding upper tailed t-test. Finally, we compare two proposed methods. We utilize lower tailed t-test in which first sample is Method 1 other one is Method 2. Method 1 is significantly lower than Method

2 in Wait and Load with 0.0003 and 0.006 p-values, respectively. There is no significant difference between them in Ymaxviol, Prevtard and Cortard with 0.877, 0.425, 0.235 p-values, respectively. Yet, Method 2 reduces Prevearl and Z with $3.77x10^{-26}$ and $4.95x10^{-5}$ p-values, respectively.

Second, we increase corrective job arrival frequency to $\tau = 2508$. We compare Heur+Model2 and Method 1 using lower tailed t-test where first sample is Heur+Model2 and second one is Method 1. Method 1 ascends Prevearl significantly with p = $1.02x10^{-38}$. If we utilize upper tailed t-test with same sample order, there is no significant difference between treatments in Slaviol, Ymaxviol, Cortard, Load and Z with 0.06, 0.09, 0.71, 0.079 and 0.14 p-values, respectively. It turns out that Prevtard and Wait falls under Method 1 with p-values 0.033 and 0.0003, respectively. Then, we study difference between Heur+Model2 and Method 2 using lower tailed t-test in which first sample is Method 2 and second one is Heur+Model2. We find out that Method 2 reduces Prevtard with p = 0.013. There is no significant difference between these values in Slaviol, Ymaxviol, Cortard, Wait, Load and Z with 0.164, 0.152, 0.407, 0.182, 0.33, 0.115 p-values, respectively. Nevertheless, Method 2 increases Prevearl with p = $3.61x10^{-21}$ according to upper tailed t-test with same sample order. Finally, we compare proposed methods. We make use of lower tailed t-test where first sample is Method 1 and second one is Method 2. Method 2 escalates Wait with p=0.006. There exist no difference between methods in Slaviol, Ymaxviol, Prevtard, Cortard, Load and Z with 0.05, 0.35, 0.64, 0.78, 0.15, 0.58 p-values, respectively. But, it diminishes Prevearl with p=$7x10^{-12}$ with respect to upper tailed t-test with same sample order.

We display results in Table 6.13. When $\tau = 10068$, Prevtard is decreased in Method 1 and 2 with reference to Heur+Model2. However, they raise Prevearl. So, we need to look at Z values before making a judgment. According to it, Heur+Model2 and Method 2 perform well under $\tau = 10068$. Since these results are supported by statistical analysis as given above, we conclude that Heur+Model2 and Method 2 are best methods. Once τ becomes 2508, Method 1 and 2 significantly reduce Prevtard while increasing Prevearl. It can be confirmed from tests and given table. Although they drop Z value in comparison with Heur+Model2, it is not statistically significant. Hence, we cannot select best method in $\tau = 2508$.

Table 6.13. Model based KPIs and Z values of methods

	Methods	Prevearl (hrs)	Prevtard (hrs)	Ymaxviol (hrs)	Cortard (hrs)	Slaviol (vehs)	Z
$\tau = 10068$	Heur+Model2	2.48	13.78	0.05	3.8	0	35.5
	Method 1	23.89	6.07	0.35	3.51	0	48.9
	Method 2	12.95	6.34	0.05	3.73	0	38.14
$\tau = 2508$	Heur+Model2	1.07	48.89	12	16.1	0.73	185.73
	Method 1	12.85	35.6	6.78	16.52	0.02	158.36
	Method 2	6.85	33.35	7.92	15.93	0.26	154.16

7. CONCLUSION

We investigate maintenance system where there exists a single dead-end track at the maintenance hangar and vehicles fulfill their passenger service according to SLA. The question is how we manage maintenance while satisfying customer service as much as possible. In order to manage this, we build MILP model and create an environment to study how parameters influence several defined KPIs. Model builds a weekly maintenance schedule considering track, SLA and preventive cycle constraints. It minimizes deviation from due time for preventive maintenance and penalizes soft constraint violations. We show that the problem is NP-Hard. Thus, we propose new improvements on the model along with CPLEX parameter fine-tuning and call it as Model2. Finally, we provide Model2 a heuristic as a starting solution and name it as Heur+Model2. We determine three kind of system loads and compare these models under problem instances which are fetched from these systems. We solve these instances under two and ten mins time limits, respectively. Under Z_f performance indicator, Model2 fails to outweigh Model up to middle load but it beats Model under ten mins time limit. So, we conclude that model improvements affect solution quality positively. Yet, this improvement is limited compared to heuristic because Heur+Model2 reduces Z_f dramatically. In terms of solution gaps, Model2 slightly succeeds in diminishing gaps but Heur+Model2 reduces significantly. Yet, it generally reduces gaps to 80%s. We could say that improvement in Z_f is the main cause of it. Since Heur+Model2 outperforms all methods, we select it as a best method. In conclusion, we increase solution quality considerably by improving model and deriving an efficient heuristic. Since Z_f value barely changes under Heur+Model2 when time limit rises, we utilize Heur+Model2 with two mins time limit within comprehensive discrete-event simulations.

We build scenarios for each parameter combinations. Because in practice, tram service providers work under one of them. By means of discrete-event simulations, we offer practitioners alternatives to see effects of factors on KPIs that otherwise they probably do not study due to their primary focus on daily maintenance operations.

We solve problem in rolling horizon manner using Heur+Model2 with two mins limit. Heur+Model2 produces a weekly schedule. We implement a schedule that occurs within first day of the scheduling period and report these statistics. We study statistically that how parameter changes affect KPIs. When we solve pure preventive maintenance problem, regression results suggest that m is the parameter that mostly affects our system. Capacity expansion generally diminishes KPIs except Wait because it brings about extra crossings. It is followed by Π. In general, all parameters but SLA, they have an impact on some KPIs. We hypothesized before experiments that Case 1 generally yields better results than Case 2 regarding KPIs but the effect changes from one system to another. For instance, for particular system, Case 1 lowers KPIs whereas for another, it could raise them so much. So, we fail to find a statistical difference between them. Average load in all experiments is found as 64.97% and it could be regarded as low. Thus, we test additional $\Pi = 480$ parameter level as a marginal analysis. Although it escalates load, it is not found statistically significant. Yet, it considerably raises Prevearl and cause a steep increase in Prevtard and Ymaxviol. Hence, we infer that it is the point where the preventive maintenance system gets out of control.

In real life, we always encounter breakdowns. Hence, this issue needs to be investigated. To do so, we conduct a research in the presence of vehicle breakdowns and name it as preventive and corrective maintenance. As a result of regression analysis, m and τ are determined as the most important parameters with regard to KPIs. Π and v follows them. As a difference, Case 2 causes a statistically significant rise with regard to Case 1 in KPIs, unlike preventive maintenance. The effect is obvious in mostly congested system, $\tau = 2508, m = 2, \Pi = 600, v = 0.1$. Because SLA transition from Case 1 to 2 escalates Ymaxviol and Prevtard dramatically. Hence, we infer that variability in SLA degrades KPIs of the system that works close to full capacity. In order to reduce such abnormal KPIs, we recommend $m = 3$, $\Pi = 720$ or $v = 0.2$ parameter changes in the decreasing order of priority. If we focus on τ effect, the results are almost similar to preventive maintenance when corrective job arrivals are few. Yet, once corrective job frequency increases, that is $\tau = 2508$, KPIs rise considerably, especially tardiness related ones, Prevtard and Ymaxviol. In addition, we carry out a further analysis because both $\Pi = 480$ and $\tau = 2508$ based scenarios lead to close shop loads. We would like to find out an answer to question that whether there is a significant difference between KPIs under these scenarios when they have close loads. As a consequence of t-tests, we observe that $\Pi = 480$ elevates Prevearl significantly. Besides, its load is significantly more than $\tau = 2508$. Hence, we conclude that the solution method tackles preventive jobs by pulling them back in time under $\Pi = 480$ in comparison with $\tau = 2508$.

We develop two methods to ensure a robust schedule under corrective job arrivals. We conjecture before experiments that these treatments decrease Prevtard for the cost of increasing Prevearl. Statistical results confirm our hypothesis. If we look into results elaborately, we investigate performances with regard to τ parameter. We define Z as the amalgam objective function value which is calculated according to objective function coefficients of the MILP model and their related KPI results. When $\tau = 10068$, Heur+Model2 and Method 2 perform better than Method 1 regarding Z value. Once arrival frequency ascends to $\tau = 2508$, it leads to higher load so KPIs are worsened. Although new methods manage to diminish Z value compared to Heur+Model2, it is not found to be statistically significant. So, no method outperforms each other. In general, if the decision maker wants to drop Prevtard to come up with more robust schedule, these methods are better than Heur+Model2. Yet, we cannot decide which one is better because Method 2 diminishes Prevearl but increase Wait compared to Method 1. Therefore, decision maker should select most appropriate method according to own selection criteria.

This thesis has three main deliverables that practitioners could benefit from. First, we introduce Heur+Model2 that maintenance managers could utilize to build weekly schedules. Thanks to model improvements, CPLEX parameter fine-tuning and the heuristic, it yields a good solution under given short time limit. Second, we analyze effects of parameter change on maintenance using discrete-event simulation. We explain most significant parameters under relative KPIs and find out most influential ones by taking into account of all scenarios. Besides, we look at their interaction effects. Thus, maintenance planners could alter their system with the help of these results. Finally, we propose two alternative methods to handle corrective jobs as well as Heur+Model2. We discuss them in terms of robustness and overall performance measures.

As a future study, we can examine scheduling on a free track. Besides, it could be intriguing to extend one single dead-end track to multi dead-end tracks for larger instances.

REFERENCES

1. *Algorithmic Decision Support for Shunt Planning*, Ph.D. Thesis, Erasmus Universiteit Rotterdam, 2006.

2. Beygo, B., *Improvement of Decision Support System for Rail Depot Planning*, Msc thesis, Chalmers University of Technology, 2011.

3. Dekker, R., R. E. Wildeman and F. a. van der Duyn Schouten, "A review of multi-component maintenance models with economic dependence", *Mathematical Methods of Operations Research*, Vol. 45, No. 3, pp. 411–435, 1997.

4. *How to prepare Service Level Agreements*, Tech. rep., Chartered Institute of Purchasing and Supply, 2009.

5. Borndörfer, R., O. Arslan, Z. Elijazyfer, H. Güler, M. Renken, G. Şahin and T. Schlechte, *Line Planning on Path Networks with Application to the Istanbul Metrobus*, Tech. rep., Zuse Institute Berlin, 2016.

6. Haghani, A. and Y. Shafahi, "Bus maintenance systems and maintenance scheduling: Model formulations and solutions", *Transportation Research Part A: Policy and Practice*, Vol. 36, pp. 453–482, 2002.

7. Jing, C., *Assessing The Impact of Capacity of Depots and Vehicle Schedule in Transportation Systems*, Msc thesis, Delft University of Technology, 2011.

8. Giacco, G. L., A. D'Ariano and D. Pacciarelli, "Rolling Stock Rostering Optimization Under Maintenance Constraints", *Journal of Intelligent Transportation Systems*, Vol. 18, No. 1, pp. 95–105, 2014.

9. Andrés, J., L. Cadarso and Á. Marín, "Maintenance scheduling in rolling stock circulations in rapid transit networks", *Transportation Research Procedia*, Vol. 10,

pp. 524–533, 2015.

10. Maróti, G. and L. Kroon, “Maintenance Routing for Train Units: The Transition Model”, *Transportation Science*, Vol. 39, No. 4, pp. 518–525, November 2005.

11. Maróti, G. and L. Kroon, “Maintenance routing for train units: The interchange model”, *Computers and Operations Research*, Vol. 34, No. 4, pp. 1121–1140, 2007.

12. Lai, Y. C., D. C. Fan and K. L. Huang, “Optimizing rolling stock assignment and maintenance plan for passenger railway operations”, *Computers and Industrial Engineering*, Vol. 85, pp. 284–295, 2015.

13. Sriram, C. and A. Haghani, “An optimization model for aircraft maintenance scheduling and re-assignment”, *Transportation Research Part A*, Vol. 37, No. 37, pp. 29–48, 2003.

14. Lai, Y. C., S. W. Wang and K. L. Huang, “Optimized Train-Set Rostering Plan for Taiwan High-Speed Rail”, *IEEE Transactions on Automation Science and Engineering*, Vol. 14, No. 1, pp. 286–298, 2017.

15. Jacobsen, P. M. and D. Pisinger, “Train shunting at a workshop area”, *Flexible Services and Manufacturing Journal*, Vol. 23, No. May, pp. 156–180, 2011.

16. Lin, B. and R. Lin, *An Approach to the High-level Maintenance Planning for EMU Trains Based on Simulated Annealing*, 2017, https://arxiv.org/pdf/1704.02752, accessed at July 2018.

17. Sriskandarajahl, C., A. K. S. Jardine and C. K. Chan, “Maintenance scheduling of rolling stock using a genetic algorithm”, *Journal of Operational Research Society*, Vol. 49, No. 11, pp. 1130–1145, 2014.

18. Vernooij, K., *An Aggregate Planning for Preventive Maintenance of Bogies by NedTrain*, Msc thesis, 2011.

19. Jiravanstit, R. and W. Tharmmaphornphilas, “View of Overhaul Resource Planning for Rolling Stock Using MIP Models.pdf”, *Engineering Journal*, Vol. 21, No. 5, pp. 145–159, 2017.

20. Feo, T. A. and J. F. Bard, “Flight Scheduling and Maintenance Base Planning”, *Management Science*, Vol. 35, No. 12, pp. 1415–1432, 1989.

21. Gopalan, R. and K. T. Talluri, “The Aircraft Maintenance Routing Problem”, *Operations Research*, Vol. 46, No. 2, pp. 260–271, 1998.

22. El Moudani, W. and F. Mora-Camino, “A dynamic approach for aircraft assignment and maintenance scheduling by airlines”, *Journal of Air Transport Management*, Vol. 6, No. 4, pp. 233–237, 2000.

23. Keysan, G., G. L. Nemhauser and M. W. P. Savelsbergh, “Tactical and Operational Planning of Scheduled Maintenance for Per-Seat, On-Demand Air Transportation”, *Transportation Science*, Vol. 44, No. 3, pp. 291–306, 2010.

24. Sarac, A., R. Batta and C. M. Rump, “A branch-and-price approach for operational aircraft maintenance routing”, *European Journal of Operational Research*, Vol. 175, No. 3, pp. 1850–1869, 2006.

25. Basdere, M. and U. Bilge, “Operational aircraft maintenance routing problem with remaining time consideration”, *European Journal of Operational Research*, Vol. 235, No. 1, pp. 315–328, 2014.

26. Al-Thani, N. A., M. Ben Ahmed and M. Haouari, “A model and optimization-based heuristic for the operational aircraft maintenance routing problem”, *Transportation Research Part C: Emerging Technologies*, Vol. 72, pp. 29–44, 2016.

27. Afsar, H., M.-l. Espinouse and B. Penz, “A Two-step Heuristic to Build Flight and Maintenance Planning in a Rolling-horizon”, *2006 International Conference on Service Systems and Service Management*, pp. 1251–1256, October 2006.

28. Boeret, N. J., “Air Canada Saves with Aircraft Maintenance Scheduling”, *Interfaces*, Vol. 7, No. 3, 1977.

29. Yang, Z. and G. Yang, “Optimization of Aircraft Maintenance plan based on Genetic Algorithm”, *Physics Procedia*, Vol. 33, pp. 580–586, 2012.

30. Verhoeff, M., W. Verhagen and R. Curran, “Maximizing Operational Readiness in Military Aviation by Optimizing Flight and Maintenance Planning”, *Transportation Research Procedia*, Vol. 10, pp. 941–950, 2015.

31. Gavranis, A. and G. Kozanidis, “An Exact Solution Algorithm for Maximizing the Fleet Availability of an Aircraft Unit Subject to Flight and Maintenance Requirements”, *European Journal of Operational Research*, Vol. 242, pp. 631–643, 2015.

32. Bradley, P., *Allocating Flight Hours to Army Helicopters*, Msc thesis, United States Navy, 1998.

33. Rosenzweig, V. V., A. Domitrovic and M. Bubic, “Planning of Training Aircraft Flight Hours”, *Croatian Operational Research Review*, Vol. 1, pp. 170–179, 2010.

34. Sgaslik, A., *Planning German Army Helicopter Maintenance and Mission Assignment*, Msc thesis, United States Navy, 1994.

35. Steiner, A., “A heuristic method for aircraft maintenance scheduling under various constraints”, *6th Swiss Transport Research Conference*, p. 28, 2006.

36. Kozanidis, G. and A. Skipis, “Flight and Maintenance Planning of Military Aircraft for Maximum Fleet Availability: A Biobjective Model”, *International Conference on Multiple Criteria Decision Making, Chania, Greece*, pp. 1–11, 2006.

37. Kozanidis, G., A. Gavranis and G. Liberopoulos, “Heuristics for Maximizing Fleet Availibility Subject to Flight & Maintenance Requirements”, `http://citeseerx.`

`ist.psu.edu/viewdoc/download?doi=10.1.1.600.2238&rep=rep1&type=pdf`, accessed at July 2018.

38. Kozanidis, G., A. Gavranis and G. Liberopoulos, "Heuristics for flight and maintenance planning of mission aircraft", *Annals of Operations Research*, Vol. 221, No. 1, pp. 211–238, 2014.

39. Kozanidis, G., A. Gavranis and E. Kostarelou, "Mixed Integer Least Squares Optimization for Flight and Maintenance Planning of Mission Aircraft", *Naval Research Logistics(NRL)*, Vol. 59, No. 3-4, pp. 212–229, 2012.

40. Cho, P., *Optimal Scheduling of Fighter Aircraft Maintenance*, Msc thesis, Massachussets Institute of Technology, 2011.

41. Safaei, N., D. Banjevic and A. K. S. Jardine, "Workforce-constrained maintenance scheduling for military aircraft fleet: a case study", *Annals of Operations Research*, Vol. 186, No. 1, pp. 295–316, May 2011.

42. Bajestani, M. A. and J. C. Beck, "Scheduling an Aircraft Repair Shop", *Proceedings of the Twenty-First International Conference on Automated Planning and Scheduling Scheduling*, pp. 10–17, 2011.

43. Kroon, L. G., R. M. Lentink and a. Schrijver, "Shunting of Passenger Train Units: An Integrated Approach", *Transportation Science*, Vol. 42, No. 4, pp. 436–449, 2008.

44. Huisman, D., L. G. Kroon, R. M. Lentink, M. J. C. M. Vromans, D. Huisman, L. G. Kroon, R. M. Lentink and M. J. C. M. Vromans, "Operations research in passenger railway transportation", *Statistica Neerlandica*, Vol. 59, No. 4, pp. 467–497, 2005.

45. Caprara, A., L. Kroon, M. Monaci, M. Peeters and P. Toth, "Chapter 3 Passenger Railway Optimization", *Handbooks in Operations Research and Management Science*, Vol. 14, pp. 129–187, 2007.

46. Freling, R., R. M. Lentink, L. G. Kroon and D. Huisman, "Shunting of Passenger Train Units in a Railway Station", *Transportation Science*, Vol. 39, No. 2, pp. 261–272, May 2005.

47. Føns, P., *Desicion Support for Depot Planning in the Railway Industry*, Ph.D. Thesis, 2006.

48. Van Den Akker, M., H. Baarsma, J. Hurink, M. Modelski, J. J. Paulus, I. Reijnen, D. Roozemond and J. Schreuder, *Shunting passenger trains: getting ready for departure*, 2008, `http://www.maths-in-industry.org/miis/264/{\%}5Cnpapers3://publication/uuid/C59A4AB8-DECA-4DBB-9DED-F22CB6A94BD6`, accessed at July 2018.

49. Jekkers, D., *Train shunt planning using genetic algorithms*, Msc thesis, Erasmus University Rotterdam, 2009.

50. Gallo, G. and F. D. Miele, "Dispatching Buses in Parking Depots", *Transportation Science*, Vol. 35, No. 3, pp. 322–330, 2001.

51. Winter, T. and U. T. Zimmermann, "Real-time dispatch of trams in storage yards", *Annals of Operations Research*, Vol. 96, pp. 287–315, 2000.

52. Lübbecke, M. E. and U. T. Zimmermann, "Shunting minimal rail car allocation", *Computational Optimization and Applications*, Vol. 31, pp. 295–308, 2005.

53. Cardonha, C. and R. Borndörfer, "A Set Partitioning Approach to Shunting", *Electronic Notes in Discrete Mathematics*, Vol. 35, pp. 359–364, 2009.

54. Borndorfer, R. and C. Cardonha, *A Binary Quadratic Programming Approach to Vehicle Positioning*, Tech. rep., Zuse Institute Berlin, 2010.

55. Blasum, U., M. R. Bussieck, W. Hochsta, C. Moll, H.-h. Scheel and T. Winter, "Scheduling trams in the morning", *Mathematical Methods of Operations Research*,

Vol. 49, pp. 137–148, 1999.

56. Cornelsen, S. and G. Di Stefano, “Track assignment”, *Journal of Discrete Algorithms*, Vol. 5, pp. 250–261, 2007.

57. Hamdouni, M., G. Desaulniers, O. Marcotte, F. Soumis and M. van Putten, “Dispatching Buses in a Depot Using Block Patterns”, *Transportation Science*, Vol. 40, No. 3, pp. 364–377, August 2006.

58. Lentink, R. M., P.-J. Fioole, L. G. Kroon and C. wan’t Woudt, *Applying Operations Research techniques to planning of train shunting*, Tech. rep., 2003.

59. den Hartog, M., *Shunt planning, an integral approach of matching, parking and routing*, Ph.D. Thesis, 2010.

60. PhoenixPublicTransit, *Spare Ratio Overview*, 2012, `https://www.phoenix.gov/publictransitsite/Documents/078815.pdf`, accessed at July 2018.

61. *Program Handbook Oregon Department of Transportation*, Tech. rep., ODOT Public Transit Division, Oregon.

62. *Metrorail Revenue Vehicle Fleet Management Plan*, Tech. rep., Washington Metropolitan Area Transit Authority, 2007.

63. *Transportation and Maintenance Operations Plan - Phase 2 South*, Tech. rep., Florida Department of Transportation District 5, 2015.

64. Pochet, Y. and L. A. Wolsey, *Production Planning by Mixed Integer Programming*, Springer-Verlag New York, 2006.

65. Vieira, G. E., J. W. Herrmann and E. Lin, “Resheduling Manufacturing System : A Framework of strategies, Policies , and Methods”, *Jounal of Scheduling*, Vol. 6, No. 1, pp. 39–62, 2003.

66. Pinedo, M. L., *Scheduling: Theory, Algorithms, and Systems (Third Edition)* , Springer, 2008.

67. Klotz, E. and A. M. N. Newman, “Practical guidelines for solving difficult mixed integer linear programs”, *Surveys in Operations Research and Management Science*, Vol. 18, No. 1, pp. 18–32, 2013.

68. Palmer, R. D., *Maintenance Planning and Scheduling Handbook(Second Edition)*, The McGraw-Hill Companies, 2006.

69. TransportforLondon, *Victoria Line Timetable*, `https://tfl.gov.uk/tube/timetable/victoria/`, accessed at July 2018.

70. BerlinerVerkehrsbetriebe, *Trip Planner Search*, `https://fahrinfo.bvg.de/Fahrinfo/bin/query.bin/en?ld=47.152&protocol=https:&`, accessed at July 2018.

71. Sriskandarajah, A. C., A. K. S. Jardine, C. K. Chan, C. Sriskandarajahl, A. K. S. Jardine and C. K. Chan, “Maintenance Scheduling of Rolling Stock Using a Genetic Algorithm”, *Journal of the Operational Research Society*, Vol. 49, No. 11, pp. 1130–1145, 1998.

72. *Massachusetts Bay Transportation Authority MBTA Light Rail Transit System*, Tech. rep., Massachusetts Bay Transportation Authority MBTA, 2011.

73. Savsar, M., “Effects of maintenance policies on the productivity of flexible manufacturing cells”, *Omega*, Vol. 34, pp. 274–282, 2006.

74. Banks, J., B. L. Nelson, J. S. Carson and D. M. Nicol, *Discrete-Event System Simulation*, Pearson/Prentice Hall, 2005.

75. Kelroy Solutions, *Fleet Maintenance from the Operational Perspective*, Tech. rep., Kelroy Solutions, 2012.

76. *Preventive Maintenance Intervals for Transit Busses: A Synthesis of Transit Practice*, Tech. rep., Transportation Research Board, 2010.

77. Kelton, W. D., R. P. Sadowski and D. T. Sturrock, *Simulation with Arena (Second Edition)*, McGraw-Hill, 2002.

78. MinitabExpressSupport, *Minitab Anderson Darling*, https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/graphs/probability-plot/interpret-the-results/key-results/, accessed at July 2018.

79. RealStatisticsUsingExcel, *Central Limit Theorem*, http://www.real-statistics.com/sampling-distributions/central-limit-theorem/, accessed at July 2018.

80. PASSSampleSizeSoftware, *Two-Sample T-Tests Allowing Unequal Variance*, https://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/PASS/Two-Sample_T-Tests_Allowing_Unequal_Variance-Enter_Difference.pdf, accessed at July 2018.

81. MPHOnlineLearningModules, *p-value Definition*, http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_HypothesisTest-Means-Proportions/BS704_HypothesisTest-Means-Proportions3.html, accessed at July 2018.