

WORKFORCE SHIFT SCHEDULING AT AIRCRAFT LINE MAINTENANCE  
WITH CUMULATIVE DEMAND MODELLING

by

Gökhan Altay

B.S., Industrial Engineering, Yıldız Technical University, 2012

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

Graduate Program in Industrial Engineering  
Boğaziçi University

2018

## ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis supervisor Assoc. Prof. Ali Tamer Ünal for his patience, great support and constructive advices that improved my progress and perspective greatly.

I thank my jury comitee members Assoc Prof. Caner Taşkın and Asst. Prof. Zehra Melis Teksan for their advices and comments at my thesis defence.

I also thank Murat Elhüseyni and Murat Güngör for their advices and support at programming.

Lastly, I thank my parents Tayfun and Nigar for supporting my decisions and for their love and my brother Can for supporting my progress.

## ABSTRACT

# WORKFORCE SHIFT SCHEDULING AT AIRCRAFT LINE MAINTENANCE WITH CUMULATIVE DEMAND MODELLING

This study aims to do overall shift assignments of aircraft line maintenance crew to satisfy line maintenance tasks in time. These tasks can only be satisfied by maintenance crew with necessary qualifications. Therefore, shift assignments are done for tasks that require a specific qualification.

Main objective of this study is to prevent the delay due to not having workforce ready to satisfy maintenance tasks. At our model, we generated desired manpower demand schedule to monitor maintenance crew shift assignments to satisfy maintenance tasks earlier or to keep more worker ready for unplanned demand cases. Moreover, we also consider planned demand schedule coming from maintenance tasks of airplanes that is required to be satisfied in time. We formulated both demand schedules as cumulative in order to assign maintenance workers near desired demand schedule while being able to satisfy planned demand in time.

We used Cplex to solve this model. We noticed that Cplex doesn't perform good for all problem instances. Therefore, to support performance of the Cplex solver, we also offered a simulated annealing algorithm to improve results. This problem is tested for different amounts of airplanes, workforce, planning horizons and demand patterns and results are evaluated.

## ÖZET

# KÜMÜLATİF TALEP MODELLEMESİ İLE UÇAK HAT BAKIMI VARDİYA PLANLAMASI

Bu çalışma, uçak hat bakımı işlerinin zamanında yapılabilmesi için bakım elemanlarının genel vardiya planlamasının yapılmasını amaçlar. Bu işler sadece gerekli sertifikalara sahip bakım elemanları tarafından yapılabilir. Bu yüzden yapılan vardiya atamaları işlerin gerektirdiği sertifikaya göre ayrılmaktadır.

Bu çalışmada ana hedef, bakım ekibinin hazır bulundurulmamasından ötürü uçak hat bakım işlerinin gecikmesinin önlenmesidir. Matematiksel modelimizde, Bakım işlerinin erken yapılması ve planlanmayan bakım talebinin karşılanabilmesi için ulaşmak istediğimiz bir bakım elamanı talep çizelgesi ürettik. Ancak bu çizelgeye ulaşırken, uçaklardaki planlanan bakım işlerinin talep çizelgesinin zamanında karşılanması da gereklidir. Vardiya planlarının, planlanan bakım işlerinin talep çizelgesini zamanında karşılaması ve istenilen talep çizelgesi yakınlarında atanabilmesi için bu talep serilerini kümülatif olarak modelledik.

Bu modeli Cplex ile çözdük. Cplex'in her testte iyi çözüm veremediğini fark ettik. Bu yüzden, daha iyi değerler alabilmek ve Cplex'e destek olması için simulated annealing algoritması önerdik. Bu problemi değişik uçak, işçi, planlama süreleri ve talep düzenleri için çözdük ve sonuçları değerlendirdik.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . . .	iii
ABSTRACT . . . . .	iv
ÖZET . . . . .	v
LIST OF FIGURES . . . . .	vii
LIST OF TABLES . . . . .	viii
LIST OF SYMBOLS . . . . .	ix
LIST OF ACRONYMS/ABBREVIATIONS . . . . .	x
1. INTRODUCTION . . . . .	1
2. LITERATURE REVIEW . . . . .	4
3. PROBLEM DEFINITION AND OPTIMIZATION MODEL . . . . .	8
3.1. Problem Definition . . . . .	8
3.2. Problem Formulation . . . . .	16
3.2.1. Assumptions . . . . .	16
3.2.2. Sets, parameters and variables . . . . .	17
3.3. Model . . . . .	19
4. SIMULATED ANNEALING ALGORITHM . . . . .	22
4.1. Parts of the algorithm . . . . .	23
4.2. Parameters of the algorithm . . . . .	27
5. EXPERIMENTS AND RESULTS . . . . .	30
5.1. Generating Data and Test Assumptions . . . . .	30
5.1.1. Basic parameters . . . . .	31
5.1.2. Generating scheduled cumulative required and desired demand sets with randomization . . . . .	32
5.1.3. Termination rules of problems : . . . . .	35
5.2. Outputs of Solvers . . . . .	36
5.3. Computational Results . . . . .	37
5.3.1. Discussion of results and future works . . . . .	44
6. CONCLUSION . . . . .	46
REFERENCES . . . . .	47

## LIST OF FIGURES

Figure 3.1.	Total assigned worker example for tasks with first qualification. . .	10
Figure 3.2.	Total assigned worker example. . . . .	13
Figure 3.3.	Total assigned worker example for tasks with first qualification. . .	14
Figure 4.1.	Control step . . . . .	27
Figure 4.2.	Simulated annealing algorithm . . . . .	29
Figure 5.1.	Different cases of desired demand with required demand. . . . .	35
Figure 5.2.	Total worker effect to solvers for 1 day 150 planes 9 qualifications.	38
Figure 5.3.	Average objective values of each test instance for 1 day planning horizon. . . . .	41
Figure 5.4.	Total worker effect to solvers for 1 day 150 planes 9 qualifications.	41
Figure 5.5.	Progress graph for 1 day, 300 workers, 100 planes and DDC 2. . .	43
Figure 5.6.	Progress graph for 4 days, 300 workers, 100 planes and DDC 2. . .	44

## LIST OF TABLES

Table 3.1.	Daily plane availability example . . . . .	8
Table 3.2.	Qualification group example . . . . .	9
Table 3.3.	Workforce shift assignment example . . . . .	10
Table 3.4.	Daily aircraft demand identification example. . . . .	11
Table 3.5.	Daily plane demand schedule example . . . . .	12
Table 3.6.	Aircraft daily required and desired demand schedule example . . . . .	15
Table 5.1.	Test parameter sets. . . . .	30
Table 5.2.	Daily plane availability sets for different flights . . . . .	33
Table 5.3.	Daily plane availability sets for short flights . . . . .	34
Table 5.4.	Result table example. . . . .	36
Table 5.5.	Average result table of all problem instances. . . . .	39
Table 5.6.	Result table for planning horizon 1 day. . . . .	42

## LIST OF SYMBOLS

$a_{si}$	Availability matrix, shows that if shift $s$ covers period $i$
$C$	Set of qualifications
$d_{kci}$	Cumulative required demand for aircraft $k$ , until period $i$ , qualification $c$
$e_{ci}$	Desired demand discrepancy for qualification $c$ , until period $i$
$G$	Set of worker groups
$I$	Planning horizon, set of periods
$K$	Set of airplanes
$p$	Acceptance probability
$p_f$	Initial probability value for feasible or infeasible
$o_{kci}$	Cumulative desired demand for aircraft $k$ , until period $i$ , qualification $c$
$S_D$	Daily interval sets for shift starting times
$t_{cg}$	Qualification matrix, shows if worker group $g$ has qualification of $c$
$T$	Temperature value
$w_g$	Worker group sizes
$W$	Total workers
$\alpha^i$	Cooling down procedure example of temperature with iterations
$\beta$	Scaling parameter example for acceptance probability

## LIST OF ACRONYMS/ABBREVIATIONS

DDC	Desired Demand Case
FAA	Federal Aviation Administration
SSRN	Social Science Research Network

## 1. INTRODUCTION

Airline industry is known as an expensive industry and requires punctuality and high quality products and services. Therefore many scheduling studies are made in this area to sustain the quality and leading the expenses.

Crew scheduling is popular at airline scheduling, Crew is second major expense after fuel costs. Crew scheduling studies generally works on scheduling cabin attendants only. However, maintenance crew scheduling has the same importance at airline scheduling as well. In addition to the cost of having qualified maintenance crew, maintenance tasks cause high costs with delay. If major tasks are delayed, it may mean that aircraft is unable to take off in scheduled time.

Maintenance tasks require certification of the maintenance group working on it for specific type of aircraft and work. That adds complexity to manpower planning and some major maintenance types requires different crew than usual. Moreover, maintenance tasks doesn't generally happen for one airport therefore different airports are sometimes included to the maintenance process as well.

There are various kinds of maintenance checks for aircraft depending on their frequency, time and their tasks. There are also unplanned maintenance checks as well. Categorization of these checks vary a bit at literature and at different countries. For literature, [1] shows these categorizations at many papers in detail. Aircraft maintenance types are generally categorized as line maintenance and hangar maintenance. Maintenance checks are frequently done after a certain amount of flying hours regulated by FAA. Some minor maintenance checks are done before plane takes off or after the plane lands or everyday These maintenance works are planned and scheduled periodically for each aircraft. There are also unscheduled maintenance cases as well. Which is included differently to problems.

A,B,C,D checks are major checks. A check is the most frequent check that is done each week includes inspection of landing gear, engines and control surfaces, B check is done after each 300 – 600 hours, and it takes longer time to complete and includes more detailed visual inspections and lubrication. A and B checks can be included as line maintenance. C and D checks take longer time and classified as hangar maintenance. C check takes 1 – 2 week and is done about each year. D check is the most detailed check including changing many parts, painting and detailed inspection which lasts about 3–4 weeks and done in about each 4 years.

Line maintenance also includes smaller maintenance works with more frequency named as Layover maintenance. Layover maintenance includes pre/post-flight checks, transit checks and daily checks. These checks include visual inspection, fluid levels, general security and cleanliness of the flight deck, emergency equipment.

In our problem, we model aggregate shift scheduling of layover maintenance manpower supply with aggregating layover maintenance manpower demand schedule coming from the airplanes. Demand and workforce schedule are separated by qualifications because maintenance tasks require different qualifications at workers. Planned demand schedule, new demand schedule scenarios are introduced to do maintenance tasks earlier and to have more workforce available. Our problem aims to have our maintenance workforce to be assigned to be able to satisfy the demand quickly by trying to assign workforce near these demand scenarios while satisfying planned schedule. Because demand normally occurs before maintenance starts and there is some flexibility at when to satisfy the demand. Moreover unplanned events happen and therefore having more workforce able is advantageous.

We modelled this problem with ILOG Cplex solver. However, solution time of scheduling problems increases exponentially as the problem instances gets bigger. Therefore, it is expected to suffer high solution time for some problem instances. Moreover, solution time also changes with the change of demand and random cases, even if other parameters are the same. Therefore, the solver doesn't always work with same efficiency.

Heuristic algorithms are proposed to get good solution within reasonable time or to support the solvers. In this study, we proposed simulated annealing algorithm which works with Cplex solver. We compare heuristic result with the end result optimization algorithm without the gap for same run time limit. We expect to get better solutions than optimization for some problem instances. Therefore, this algorithm can be used as a supportive tool for optimization.

Remainder of this work goes as follows; In chapter 2, related works at the literature is presented. Chapter 3 defines the problem, shows the scope of the problem and the model. Chapter 4 introduces simulated annealing algorithm we proposed in details. Chapter 5 talks of experimental results and solutions, the way of creating problem instances, computational results, discussions and improvement options. Chapter 6 gives the conclusions.

## 2. LITERATURE REVIEW

Main focus of this work is manpower shift scheduling at aircraft line maintenance. Aircraft maintenance personnel scheduling problems are personnel scheduling problem with qualifications. [2] has a good review including personnel scheduling, workforce scheduling, staff scheduling and staffing papers. Personnel scheduling problems are classified by personnel characteristics and shift definitions, constraints, solution methods and application areas.

Aircraft maintenance scheduling and maintenance manpower planning are mainly considered important because other than the crew and material cost, there is a downtime cost when the maintenance didn't go as planned. [3] has a case study to examine total cost including downtime cost.

Aircraft maintenance problems are affected by flight scheduling, fleet assignment and aircraft routing problems, because time slots of routine maintenance are planned after them or with them. [1] lists these problems and the papers worked on them. [4] has case studies of these problems. Flight scheduling problem shows when the flights arrive and depart according to the demand, manpower and regulations. Fleet assignment problem is solved after flight scheduling problem and it determines which aircraft type is assigned to which flight. Fleet assignment problems also can consider maintenance constraints. Aircraft routing problem is solved after fleet assignment problem which assigns specific tail numbered aircraft to a route and it considers maintenance constraints. There is also maintenance routing problem to do aircraft routing to minimize the maintenance costs and maintenance time and also it ensures feasibility for the maintenance stations and includes the flight hour requirements. [5] is focused on this problem.

These problems are modelled and solved as separate or integrated. [1] shows different integration of these problems among related papers.

[6] integrates fleet assignment, maintenance routing (aircraft routing with maintenance) and crew pairing (crew scheduling) also bender's decomposition is used for decomposition of the complex model. [7] integrates aircraft routing and crew scheduling, gets fleet assignment problem as input and changes some of assignments afterwards. Some papers also add the other problems as constraints with fully integrating. [8] solves fleet assignment problem with adding maintenance and crew scheduling as constraints.

These problems are integrated with maintenance scheduling model as well. [9] has a maintenance scheduling (type A and B checks) problem with original destinations as input and it does aircraft reassignments.

Maintenance scheduling problems are not necessarily provided from in house supply only. [10] models the problem for decision between outsourcing and in house maintenance supply with their respective costs in order to minimize the maintenance cost. Heavy maintenance checks are considered because they are more costly than others.

For airline personnel scheduling problems, airline cabin crew scheduling is more widely studied than maintenance crew scheduling problems. Crew scheduling problems are dependant to flight scheduling, fleet assignment and aircraft routing problems. Crew scheduling problems are divided into crew pairing and crew assignment. [11] has a review about crew scheduling problems, regulations and solution methods. They also come up with solution algorithm. [12] Works on crew assignment part of airline crew scheduling problem.

There are also papers worked on staffing and rostering of airline ground personnel scheduling. Their demand is also dependant to flights. Also ground personnel has a considerable cost to improve and affects ticket prices. It is different from crew scheduling problem because there is no travelling case of the staff. [13] focuses rostering of a ground workforce at an airport. [14] models duty generation, crew scheduling and crew rostering of baggage service section with goal programming models.

Military applications are quite different since flight routines and maintenance patterns are different. [15] maximizes the availability of planes to satisfy the flight routine while scheduling maintenance. It also includes qualifications of workers for maintenance and capacity constraints of the manpower. [16] also maximizes fleet availability at flight and maintenance planning at a military fleet with exact solution algorithm.

In our problem, we aim to fulfill the manpower demand using two ways of assigned tasks. It isn't common to integrate task assignment and shift scheduling in modelling. [17] has formulations for that case with different type of workers and jobs.

There are studies based on line maintenance manpower supply problems relevant to our study. [18] models line maintenance workforce as groups and it minimizes the man hours used for maintenance. It also tested flexible strategies on manpower planning such as flexible working hours, flexible squad size and flexible shift times and it uses Cplex as solver, modelled as mixed integer programming. [19] makes also tests three similar flexible strategies and also it defines certificates on different sets. It also comes up with an algorithm using simpler versions of the complex model and uses Cplex as solver and problems are modelled as mixed integer programming.

There are also papers that also work on rostering part of maintenance manpower planning [20] combined staffing and scheduling at their model by combining shift succession constraints and working hours with manpower demand. It is modelled as non linear mixed integer programming and for solution. It fixes one variable makes it linear and runs the algorithm for all combinations of that variable. [21] includes day off pattern of workers in order to meet manpower demands in a week.

As mentioned at many works, aircraft maintenance tasks requires a skilled workforce for different type of aircraft and jobs. [22] considers the training process of the staff and integrates with the staffing and scheduling problem at [20]. It considers the trade off between advantages of having more skills at the workforce with the training and suffering training expenses and the workforce unavailable during training session.

Other important field is the fairness of task assignment to the skilled workers which intends to prevent over assignments to some workers to increase productivity. [23] proposes a bi objective model minimizes the labour cost similar to some problems at literature and minimizes the deviation from average workload for staff to provide productivity from assigned workers.

Heuristic algorithms are either integrated with or preferred over deterministic solutions at some problems at aircraft maintenance. [24] models aircraft maintenance shift scheduling problem and includes stochastic nature of flight departures and tries to get a more robust solution with adding simulations to the optimization with model enchantment heuristic. [25] proposes a tabu search algorithm to integrated staffing and scheduling problem similar to [20]. It also shows optimization model and compares it with optimization results. [23] also uses a heuristic algorithm with tabu search and compares to Cplex solver for its problem.

### 3. PROBLEM DEFINITION AND OPTIMIZATION MODEL

First section of this chapter defines the scope of the problem and its interesting parts. Second section introduces assumptions, sets, parameters of the problem. Last section includes the optimization model with its constraints and objective.

#### 3.1. Problem Definition

This study focuses on manpower planning at line maintenance. The study is based on fulfilling planned short-term maintenance checks such as pre-flight check, transit check and daily check at the same airport.

These checks involve different tasks such as visual inspections, fluid level checks, engine checks, landing gear checks, equipment checks, cleanliness etc. These tasks can only be fulfilled when an aircraft is able for maintenance. Table 3.1 gives example of daily availability of airplanes. Maintenance tasks can only be assigned at "1" slots. "0" Slots can mean either there is flight assignment or there is other maintenance schedules (like A, B check) or other assignments that prevents layover maintenance at that period.

Table 3.1. Daily plane availability example.

Planes	Periods(hours)																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Plane A	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Plane B	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0

Each maintenance task must be fulfilled by qualified workers. Qualifications of the workers are dependent to plane types and tasks. Workers can have more than one qualification. Qualifications of workers are introduced as groups that shows all skill combinations. Table 3.2 shows some group examples.

These groups have nothing to do with worker group assignments to the shifts as seen at the relevant studies such as [18], [19] and [20]. However, qualification combinations are also considered at [19] in a way. These groups are introduced to classify the workforce by their set of qualifications. Groups are named by the qualifications they have.

Table 3.2. Qualification group example.

Worker Group	Qualifications									Group Size
	1	2	3	4	5	6	7	8	9	
<b>Group 125</b>	1	1	0	0	1	0	0	0	0	6 people
<b>Group 34</b>	0	0	1	1	0	0	0	0	0	8 people
<b>Group 469</b>	0	0	0	1	0	1	0	0	1	7 people
<b>Group 7</b>	0	0	0	0	0	0	1	0	0	10 people

Workforce is assigned to 8-hour shifts to satisfy the demand for each qualification. Worker shifts can start any hour of the day in order to match the flexibility at demand. Table 3.3 gives an example of daily total shift assignments for different qualifications. For instance, 5 people at period 0 and qualification 1 at the table means that there are 5 workers that has qualification 1 assigned to a shift starting from period 0 to period 7.

These workers will be chosen from the worker groups we defined. For instance, from shift assignment example table 3.3, 5 workers at shift of first period for first qualification can only be picked from worker groups that acquires first qualification. If we consider only worker groups shown at table 3.2, These people can only be picked among 6 people of group 125. Moreover, at the same period, worker assignments for second and third qualification cannot be the same people as people assigned to shifts to satisfy tasks with first qualification. But, those people might have the skill to satisfy job at qualification 1. For instance, from qualification table 3.2, 1 remaining person at group 125 can be assigned for a task of qualification 2 at the same period.

Table 3.3. Workforce shift assignment example.

Qualification	Periods(hours)																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	5	2	0	0	0	1	0	0	6	3	1	0	0	0	1	1	0	1	4	0	0	0	0	1
2	5	0	0	2	0	5	0	0	1	1	6	1	0	0	0	0	0	4	2	0	0	0	1	1
3	5	3	0	0	0	0	0	0	0	9	0	0	0	0	0	3	0	0	5	0	0	1	0	0

Since shifts can start any hour of the day, shifts will overlap and total assigned workforce will create a step function for each qualification. Figure 3.1 gives example of amount of assigned workers at periods as a result of shift assignments at table 3.3 for only uses first qualification and shows it as a step chart. This assignment is done to satisfy scheduled demand with first qualification. Amount of workers changes according to the busy and less busy periods at the demand.

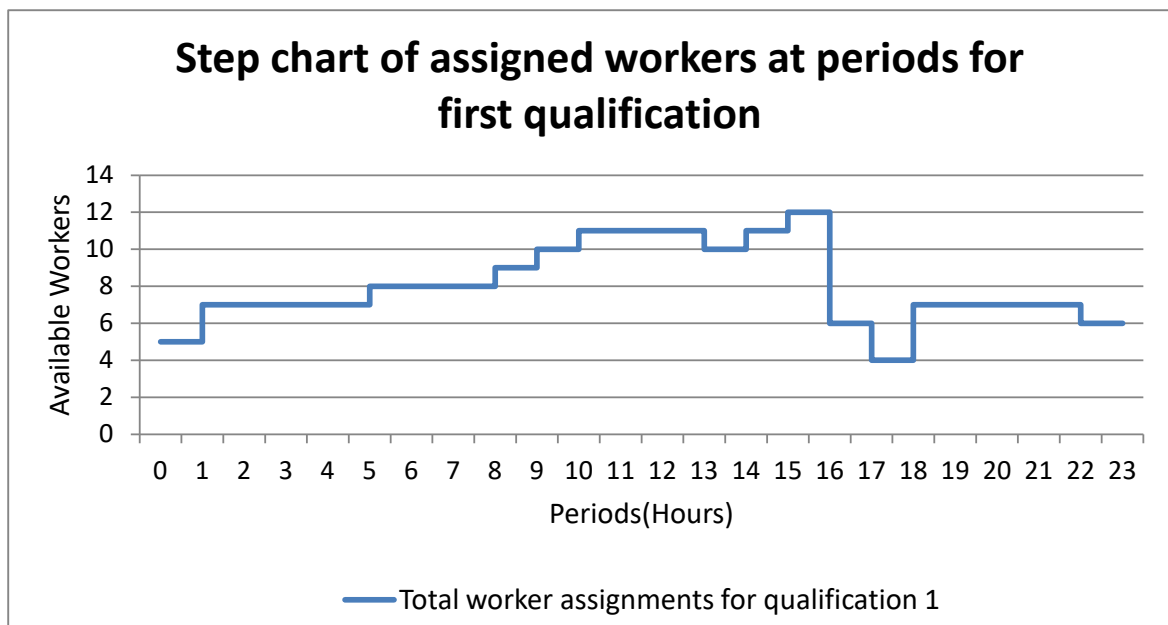


Figure 3.1. Total assigned worker example for tasks with first qualification.

Maintenance tasks sometimes are identified after inspections. These inspections happen periodically and sometimes out of maintenance periods as well. Some tasks are identified at maintenance periods as well, as a result of further inspections at maintenance periods.

These tasks have different urgency. Therefore, some of the tasks must be done at the first next available maintenance slot while some tasks such as cleaning related tasks can wait for later dates. Moreover, all of the tasks are converted into amount of qualified manpower at a period to do worker assignments shift assignments.

Table 3.4 gives example of identification of manpower demand for an aircraft for tasks with different qualifications. Aircraft daily maintenance slots are given at plane availability row of the table. Tasks of Qualification 1 is an urgent task set example which is identified after inspection at maintenance period and it needs to be assigned to same maintenance slot if possible. As seen, inspections can be done out of maintenance slots as well. Demand of Qualification 2 can occur out of maintenance slots but it must be satisfied at maintenance slots. This is a less urgent task example which can be assigned to any time slot of maintenance. Qualification 3 is an example of further checks and inspections at each maintenance slot. Normally the time of the inspection is not related to urgency of the task. These task sets were example.

Table 3.4. Daily aircraft demand identification example.

Tasks and availability	Periods(hours)																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Plane availability	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
Qualification 1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Qualification 2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0
Qualification 3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

When these tasks are scheduled at maintenance slots, it will similar to table 3.5, if no other tasks from before previous periods has been pre-defined. For instance from table 3.5, qualification 3 is done first at maintenance slots as inspections and checks. Then task with qualification 1 is assigned as soon as it is identified since there is empty slot. Tasks with qualification 2 can wait before urgent tasks are done and it is identified later so it can be done at later maintenance slot only or it would be delayed if that slot was busy.

Table 3.5. Daily plane demand schedule example.

<b>Maintenance demand schedule example</b>								
<b>Maintenance Periods</b>	3	4	5	6	7	18	19	20
<b>Workforce Requirement</b>	1	2	0	0	0	1	1	1
<b>Qualification</b>	3	1	0	0	0	3	2	2

At our problem, aircraft demand schedule is called as required demand and this demand must be satisfied at every case at the given period or preferably earlier. While scheduling those tasks urgent tasks can not be scheduled in a flexible way. They must be done as soon as possible, while other tasks can wait.

At similar problems, this demand is assigned to workforce shifts first, then scheduling and rostering are made at second stage. While assigning demand to workforce shifts, workforce groups are used to assign the demand as seen at [18] and [19]. Rostering process also can be integrated with workforce planning as shown at [20], which also uses workforce groups. These problems wish to minimize the manpower costs with providing feasible conditions most importantly task fulfillment.

This problem does workforce shift assignments without considering worker group assignments as mentioned at relevant papers. Because assignments doesn't have to be as groups to handle maintenance tasks at each airplane. Moreover, it creates more flexibility at usage of workers. Either way, this result will be used for further planning to assign the shifts to individuals.

However, this problem doesn't assign workers to the airplanes specifically. Therefore, there will be a problem of handling labor conditions such as availability and working hours. It is also difficult to define daily, weekly and monthly working hour limitations for workers since there is no specific worker assignments this way. But, we can control our total overall shift assignments daily or for longer horizon if needed.

Figure 3.2 shows the total worker shift assignments for all qualifications from the table 3.3. From this figure, 81 workers are assigned to the shifts according to the shift assignments at the end of the day. Therefore, we can control total worker usage this way.

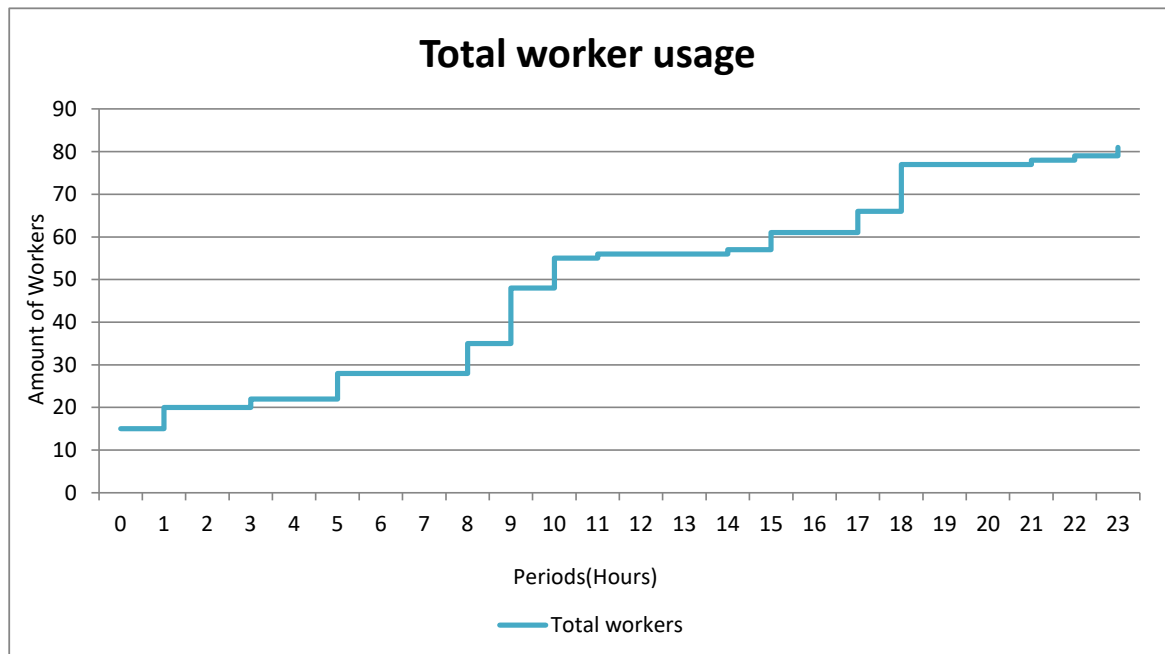


Figure 3.2. Total assigned worker example.

Moreover, we also know how many worker has necessary qualification for a task. Therefore, qualification assignments can be controlled at a horizon as well. Figure 3.2 gives total worker usage from shifts at table 3.3 for first qualification. As observed from figure, 26 workers are assigned for the tasks related to qualification 1 at the end of the day, therefore we can know how many people have qualification 1, therefore it can be used to define workforce limitation for qualification. These conditions will give overall limitations for workers before specific assignments are made by using these assignments as an input.

Moreover, we aggregate the demand of airplanes. Because we don't assign workers to the planes individually or as group as mentioned. We only consider available workforce requirement for demand.

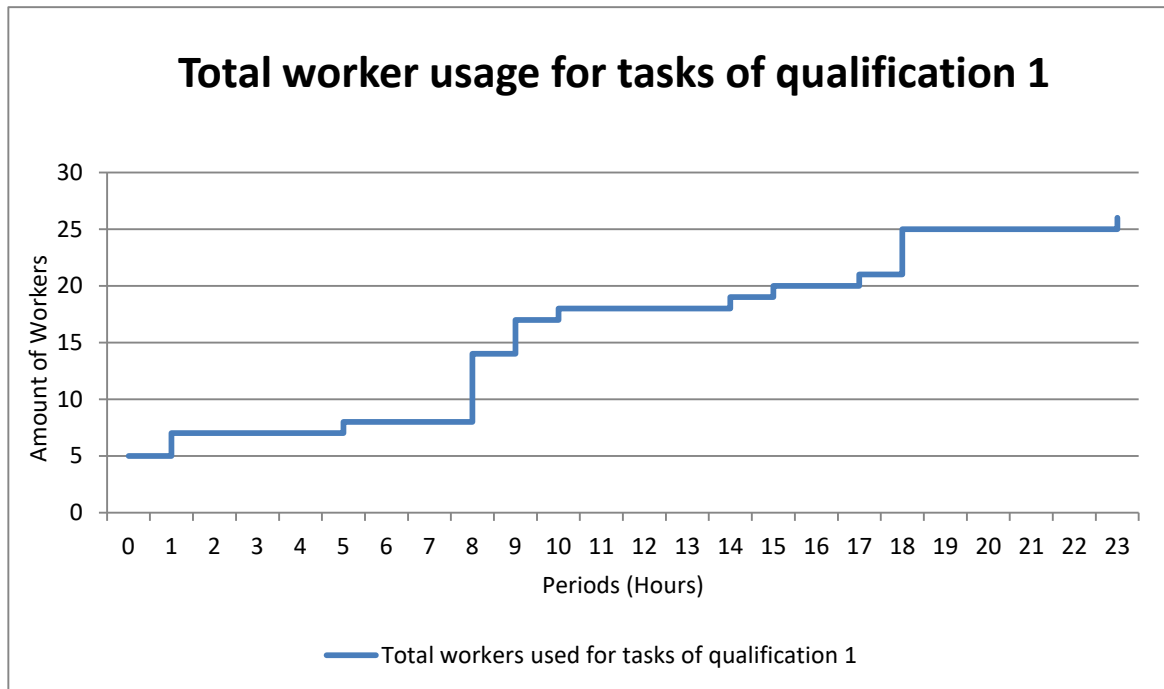


Figure 3.3. Total assigned worker example for tasks with first qualification.

We intend to get total worker shift assignments at the planning horizon.

Our problem is mainly interested in facing as less downtime as possible with using at aircraft due to latency with line maintenance. Therefore, our problem mainly intends to do task fulfillment as good as possible with reasonable time. There will be situations that requires more manpower than necessary. In these cases, normal planning might not able to fulfill the maintenance demand within reasonable time. It could be result of busy planning season, occurrence of unplanned maintenance or result of assignment of some tasks earlier.

Therefore, desired demand schedule is introduced to add these cases to the problem. This demand is not necessarily satisfied fully. But, it is desired to have our workforce able to satisfy this demand as good as possible.

Table 3.6 gives an different example of required and desired demand schedules for an aircraft.

It is similar to table 3.5 and qualifications represent same tasks as shown there. As we can see at 3.6, desired demand might be created to have for more workers available than planned. Urgent tasks and periodical inspections(tasks of qualification 1 and 3 at the example) doesn't change their scheduled period at generating desired demand. Normally, they shouldn't change their scheduled period much as well. Desired sometimes assigns more workers to a tasks to help its completion as seen at 4th period. For less important tasks (tasks of qualification 2), desired demand intends have worker available for them as well and intends to do them earlier.

Table 3.6. Aircraft daily required and desired demand schedule example.

Required and desired demand assignment example for an aircraft												Total Workers
Maintenance Periods	3	4	5	6	7	11	12	13	18	19	20	-
Workforce Required Demand	1	2	0	0	0	1	1	0	1	1	0	7 workers
Qualification for Required Demand	3	1	-	-	-	3	1	-	3	2	-	-
Workforce Desired Demand	1	3	1	1	0	1	1	1	1	0	0	10 workers
Qualification for Desired Demand	3	1	2	2	-	3	1	2	3	-	-	-

Desired demand intends to change the total worker shift assignments around the busy periods or extra demand periods that we wish to add to the system by changing time of satisfying tasks. Therefore, tasks can be assigned to different maintenance periods of aircraft and it also requires more workforce than required demand overall. Therefore, desired demand doesn't necessarily follow the same trend as required demand. Thus, it cannot be solved by simply adding more qualified workers to planned maintenance periods.

In order to assign our total workforce to shifts to satisfy more than required demand and less than desired demand, demands are converted into cumulative. Making desired demand cumulative allows doing shift assignments at the early periods other than the shifts that cover demand period also gives meaning to do closer assignments rather than exact assignments. Desired demand cases are created with intention of having better overall assignments.

They are not generated as exact planning demands although they apply only when planes are available as well.

However, some maintenance tasks needs to be done at the exact planning period. Therefore, required demand is not entirely used as cumulative. For instance, a specific demand that is scheduled at end of planning horizon cannot be satisfied by the shifts at the start of planning horizon. Therefore, condition of only the shifts relevant to a demand period can satisfy that demand is added for required demand. We don't want to break our planned shifts entirely while trying to satisfy desired demand.

With making desired demand cumulative, effect of making worker assignments closer to the demand period is introduced as mentioned. Therefore, closer the assignment is the better it will be. As said, it is generally not possible to do these assignments at exact periods. For instance, if an assignment is done to a shift that covers one period earlier than demand, that wouldn't be comparable to the desired demand schedule if desired demand wasn't cumulative. Now shifts assignments can cover desired demand good overall as well but of course it results over-assignments at early period and which suffers using more manpower than desired demand which is also not intended.

## **3.2. Problem Formulation**

### **3.2.1. Assumptions**

There are some assumptions made to define the scope of problem and make it more solvable:

- We assume we know the task deadlines for any aircraft type. Therefore we know desired and required demands for manpower.
- Model applies for regular maintenance, unexpected events are not considered at required demand. Still, problem intends to have more workforce than necessary while trying to follow different cases desired demand.

- Shifts can start at any hour of the day. Since there will be a fluctuation at both required and desired demand, flexibility is required.
- Shift lengths are the same. For simplicity it is used. We already are able to assign manpower to any hour of the day that matches flexibility at the demand.
- Workers can get multiple skills. However, it is impossible for any worker to satisfy every qualifications.
- Personnel availability is not added to the model. This problem is more interested in creating the shifts and not to assign individuals to the shifts.
- Personnel wages are not added to the model. This problem is more interested in supplying qualified manpower necessary when needed.

### 3.2.2. Sets, parameters and variables

Sets of the problem are defined here:

- $I$ (Planning horizon) : Shows the planning horizon of each problem instance. Planning period symbols  $i, j \in I$  shows each hour of the horizon.
- $S$ (Shift starting times) : Shift starting times  $S$  is same as  $I$  for every problem. Shift starting time symbol  $s \in S$  shows a specific starting time.
- $C$ (Set of qualifications) : Symbol  $c \in C$  used to define qualified usable workforce, qualified demand and the qualified workforce assigned to the shifts.
- $K$ (Set of aircrafts) : This shows the airplanes that will be inspected during the horizon. Symbol  $k \in K$  shows a specific aircraft for maintenance.
- $G$ (Set of worker groups) : Shows set worker groups according to their combination of qualifications.
- $S_D$ (Daily intervals for shift starting times) : At planning horizon, this set shows any daily interval starting from 1st period to 24th period. Next interval will start from 2nd period to 25th period. The last interval reaches the end of horizon.  $s_1, s_2 \in S_D$  shows starting and finishing times of a specific daily interval.

Parameters of the problem are stated below:

- $W$ (Total workers) : It shows the total amount of workers at the system.
- $w_g$ (Worker group size) : It shows the size of each worker group defined. Worker group sizes are generated from the total workers.
- $t_{cg}$ (Qualification matrix) : It is defined to show if a worker group has a specific qualification.

$$t_{cg} = \begin{cases} 1 & \text{worker group } g \text{ has qualification } c \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

- $a_{si}$ (Availability matrix) : It shows whether a shift covers specific period  $i$ . Therefore, it is used for turning workforce demands into shifts. It can be seen at typical scheduling problems. Assigning shifts to periods can also be handled by defining a specific indice set as seen at [18] and [19].

$$a_{si} = \begin{cases} 1 & \text{shift starting at } s \text{ covers } i \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

- $d_{kci}$ (Required cumulative demand) : Shows cumulative minimal manpower demand of aircraft  $k$  with  $c$  qualification from period 0 to period  $i$ .
- $o_{kci}$ (Desired cumulative demand): Shows cumulative desired manpower demand of aircraft  $k$  with  $c$  qualification period 0 to period  $i$ .

Variables of the problem are stated below:

- $x_{cs}$ (Worker shift assignments) : Represents total maintenance workforce with qualification  $c$  assigned to shift  $s$ .
- $e_{ci}$ (Desired demand discrepancy) : It shows absolute difference between total cumulative desired demand for all planes with qualification  $c$ , period  $i$  and assigned workforce with qualification  $c$ , to shifts that cover periods of 0 to  $i$ .

We want to have our assigned manpower close to this demand as possible overall. But it is also unwanted to have more manpower than desired demand as well. Its formulation can be seen at equation 3.3.

$$\mathbf{e}_{ci} = \left| \sum_{k=1}^K \mathbf{o}_{kci} - \sum_{j=1}^i \sum_{s=1}^S \mathbf{x}_{cs} \mathbf{a}_{sj} \right| \quad \text{for each } i \in I, c \in C \quad (3.3)$$

### 3.3. Model

This mathematical model intends to do worker shift assignments close to desired demand as possible while satisfying required demand and worker constraints.

$$\text{Min} \sum_{i=1}^I \sum_{c=1}^C \mathbf{e}_{ci} \quad (3.4)$$

Subject to:

$$\mathbf{e}_{ci} \geq \sum_{k=1}^K \mathbf{o}_{kci} - \sum_{j=1}^i \sum_{s=1}^S \mathbf{x}_{cs} \mathbf{a}_{sj} \quad \text{for each } i \in I, c \in C \quad (3.5)$$

$$\mathbf{e}_{ci} \geq \sum_{j=1}^i \sum_{s=1}^S \mathbf{x}_{cs} \mathbf{a}_{sj} - \sum_{k=1}^K \mathbf{o}_{kci} \quad \text{for each } i \in I, c \in C \quad (3.6)$$

$$\sum_{j=1}^i \sum_{s=1}^i \mathbf{x}_{cs} \mathbf{a}_{sj} \geq \sum_{k=1}^K \mathbf{d}_{kci} \quad \text{for each } i \in I, c \in C \quad (3.7)$$

$$\sum_{s=(i-7)}^i \mathbf{x}_{cs} \mathbf{a}_{si} \geq \sum_{k=1}^K (\mathbf{d}_{kci} - \mathbf{d}_{kc(i-1)}) \quad \text{for each } i \in I - \{1, \dots, 7\}, c \in C \quad (3.8)$$

$$\sum_{s=s_1}^{s_2} \mathbf{x}_{cs} \leq \sum_{g=1}^G \mathbf{w}_g \mathbf{t}_{cg} \quad \text{for each } (s_1, s_2) \in s_D, c \in C \quad (3.9)$$

$$\sum_{c=1}^C \sum_{s=s_1}^{s_2} \mathbf{x}_{cs} \leq \mathbf{W} \quad \text{for each } (s_1, s_2) \in s_D \quad (3.10)$$

$$\mathbf{x}_{cs} \geq 0, \mathbf{x}_{cs} \in \text{Integer} \quad \text{for each } c \in C, s \in S \quad (3.11)$$

$$\mathbf{e}_{ci} \geq 0, \mathbf{e}_{ci} \in \text{Integer} \quad \text{for each } c \in C, i \in I \quad (3.12)$$

Objective function 3.4 minimizes the total desired demand discrepancy variable. Therefore our problem intends to have our worker assignments closer to desired demand overall. Since exact assignment isn't generally possible. Close aggregate assignments are preferred and they suffer less over-workforce assignments.

Constraints 3.5 and 3.6 do the linearization of equation 3.3 and introduces the formula of desired demand discrepancy variable to the model. If the model didn't mind having too much workers assigned, then constraint 3.6 would have zero at right hand side.

There has to be more workforce assignment than required demand for each type of job. Therefore, constraint set of 3.7 and 3.8 define required demand fulfillment.

Constraint 3.7 shows that total required demand has to be fulfilled by total shift assignments of workforce that covers periods until demand period  $i$  for every period and qualification. However, while fulfilling cumulative manpower demand, required demand at specific period can only be fulfilled by the shifts that covers it and availability matrix isn't enough to define this while the required demand is cumulative. Therefore constraint 3.8 is added to the model.

Constraint 3.8 defines that each required demand at period  $i$  can be fulfilled by the shifts related that covers that for every period except starting periods that can be covered before planned horizon as well and for every qualification.

Constraint 3.9 and 3.10 are worker constraints that introduces labour capacities to the model.

Constraint 3.9 defines that assigned workforce for qualification  $c$  must be lower than worker groups that has qualification  $c$  for each day interval. Since, workers can have multiple qualification, there are many workers can satisfy a task with that qualification.

Constraint 3.10 shows that, total assignment of workers cannot be more than total workers at the system for each day interval. This limits assignments to qualifications from the groups for each day in a way, without working on the individual worker schedules.

Constraint 3.11 and 3.12 are boundary constraints that shows non negativity of worker shift assignment and desired demand discrepancy variables and shows that they are integer.

## 4. SIMULATED ANNEALING ALGORITHM

This chapter includes the details of simulated annealing algorithm we proposed. First section explains parts of the algorithm. Second section gives parameters of the algorithm with the pseudo code of the algorithm.

Simulated annealing is a meta-heuristic algorithm that does random search through the solution pool with getting random samples from the solution pool . Algorithm starts with an initial solution. Then goes to a random neighbour solution with a move operator by changing decision variable values. If neighbour solution is better than original, algorithm goes to neighbour solution, otherwise it accepts worse neighbour solutions(including infeasible solutions at our problem) with an acceptance probability to widen the search. This process goes on repeatedly as long as termination condition is satisfied. Originally, termination condition is chosen as number of iterations. After algorithm is terminated, it gives best solution and decision variables found during the search.

Acceptance probability include main parameters of the algorithm. Acceptance probability is based on difference between objective values, a fixed parameter value  $\beta$  to do scaling to generate meaningful probability values, a temperature variable  $T$  supportive to probability calculation and variable  $\alpha$  that shows change of temperature with iteration. Acceptance probability drops as the iterations goes on. This is called as cooling down process of temperature related to change of  $\alpha$ . Example of acceptance probability formula is given at equation 4.1. Negative sign at the formula changes depending on using minimization or maximization problem.

$$AcceptanceProbability = \frac{-(\Delta Objective)}{\beta \alpha^i T_0} \quad (4.1)$$

For our problem, it is helpful to search through worse solutions and infeasible region, while trying to achieve better results. Because, this problem probably would see local maximum often while searching through the solution pool. This also explains why simulated annealing algorithm is preferred over simple greedy heuristic algorithms for this problem.

Moreover, while choosing move operators, we considered that problem often sees infeasibility with move operators that does bigger changes. But, searching through close neighbours also will also make algorithm to progress slowly. As a result, we preferred close neighbour search while deciding moving steps through solution pool. Moreover, we added a function to identify infeasible solutions and process them differently. and they have a smaller probability of accepting them through the search. There are also move operators that intends to make an infeasible solution feasible without losing much of objective value. Move operators change worker shift assignment( $x_{cs}$ ) while moving to neighbour solutions.

We also considered using genetic algorithm as another meta-heuristic algorithm as well. However, genetic algorithm has too many parameters and it would be harder to balance these parameters. Moreover, maintaining feasibility while moving through solution pool is difficult for this problem. Therefore, move operators we will define with genetic algorithm will see infeasibility more often than simulated annealing move operators. If we preferred smaller move operators to avoid it, then it would require long solution time.

#### 4.1. Parts of the algorithm

Simulated annealing algorithm we formulated is based on one initializing function, five different move operators, one control function to define acceptance cases of solutions and penalizing function to identify infeasibility to help control function.

Initializing function: It takes a Cplex solution with 50 percent relative gap limit or the best solution that is found within 5 minutes if that gap cannot be reached in a short time. When relative gap is set to 50, solver gives the last iteration that drops below 50 percent gap. Therefore, it may be a very good solution if last iteration did a big jump. Otherwise, it will be a solution near that gap. MIP Relative gap is difference from best integer objective and lower relaxed bound over best integer solution. Its formula is shown at equation 4.2.

$$relativeGap = \frac{|BestNode - BestInteger|}{e^{-5} + |BestInteger|} \quad (4.2)$$

If problem instance is large and problem cannot find even initial solution with below 95 percent relative gap within 5 minutes, a simple initial solution algorithm is used. It assigns same amount of workers (by using about 60 percent of total workers) to shifts periods of 1, 9, 17,  $\dots$ ,  $I$  with 8 hour difference until the end of planning horizon for each qualification. This can be alternated when demand of tasks with different qualification differ in reality. Then process goes on adding remaining workers by using breaking infeasibility(Add to satisfy demand) move. This process doesn't necessarily add all workers to the system, it stops when demands are satisfied. This makes much better results than simply assigning all workers and it is generally a feasible solution.

Greedy simple heuristic solutions are also tried to generate initial solution. However, this algorithm will work for bigger size problems and calculating objective function repeatedly consumed significant time at the tests we did. If we assigned all workers with breaking infeasibility moves, it also would take longer processing time as well. but, it would take shorter than greedy approach. Either way we intended to give longer time for algorithm's progress.

Add, drop and swap moves are random move methods through the neighbourhood. They tend to go infeasible region while moving through. Picking a totally random case won't improve the solution in a reasonable time. Therefore while picking a move they all check objective value and get the best one among some random tries.

That still may not improve the objective before the move is done but it eliminates worst picks which is less likely to improve at long run.

Drop move: This move takes the recent worker shift assignments( $x_{cs}$ ) and picks a random shift for a random qualification. Then, reduces the workers by one. If it picks an empty cell, it picks again until it gets a non empty cell. This is a small move, but it doesn't go to infeasibility as much and finds small improvements. This process goes on many times and one with best objective of these moves is picked.

Add move: It goes exactly like drop move. It adds a worker to a random shift for a random qualification. It goes on many times to find a okay random pick same as the drop move.

Swap move: It picks takes the recent worker shift assignments and picks the another shift from same qualification then assigns one worker from the other. First picked shift must be non empty else It picks again.

Remaining two move operators are not working randomly, they only act if there is infeasibility and they try to make it feasible in their field. They are separated, because worker and demand limitations lead different kind of solutions. One may make other infeasible as well. Moreover, if both considered in one simple move, many solutions will be skipped.

Drop extra workers move: This move checks how many workers are assigned for each daily interval. If there is over-assignment, it drops the workers according to loose places at demand constraint 3.8 until worker constraint is satisfied. When it finds first loose place it drops a worker from first shift that covers loose period. If it is empty, then it drops from the next one. This can break demand satisfying constraint sometimes but solves worker constraints.

Add To satisfy demand move: There are two demand constraint at real problem that have tendency to be violated as the algorithm goes and it is hard to go through feasible region without explicit definitions of these constraints. Therefore, this move checks if constraint 3.8 is violated for each qualification starting from shift 8, then it assigns the manpower to a shift starting from where it is violated. Constraint 3.8 is generally tighter than constraint 3.7 starting from shift 8. It also checks violation of constraint 3.7 and adds manpower to the violated period shift.

Control function includes accepting rules of neighbour solutions. Penalizing function supports control function by identifying feasibility and penalizing it. These functions are explained in details below:

Penalizing function: This function takes the worker shift assignments and checks if it is feasible for demand and worker constraints, if not by even one case then returns a very high value. It is based on wanting infeasible solutions less while moving through the solution neighbourhood. This is used to identify infeasible solutions. Therefore the case of accepting better infeasible solutions over previous feasible solution will be prevented. Therefore, they will be marked as worse solutions. Still, they will be accepted with a probability as the process goes on.

Control Function: This function includes the solution accepting procedure after move steps. If a solution is improved after a move, accept the decision variable matrix and objective. Also check if the best solution found until this point is improved. If neighbour solution is worse, then accepts solution with a probability changes with the difference from objective value and temperature, which drops as iteration increases. Penalty is added to objective values, therefore, it is less likely to accept an infeasible solution while searching, but it is possible. For an explicit definition, figure 4.1 shows the control step.

```

if NewObjective < OldObjective then
  OldObjective = NewObjective and  $x_{cs}(old) = x_{cs}(new)$ 
  if NewObjective < BestObjective then
    BestObjective = NewObjective and  $x_{cs}(best) = x_{cs}(new)$ 
  end if
else
  if Random(0,1) < p then
    OldObjective = NewObjective and  $x_{cs}(old) = x_{cs}(new)$ 
  end if
end if

```

Figure 4.1. Control step.

## 4.2. Parameters of the algorithm

This section shows the parameters we customized for our algorithm:

$p$ (Acceptance probability): Probability of accepting worse solution depends on change at objective function and temperature as explained. Probability equation we preferred for this problem is shown at equation 4.3. We reduce effect of objective differences in order to do scaling. Probability also changes if the new solution is feasible or not by a  $p_f$  variable.

$$\mathbf{p} = (\mathbf{p}_f - \frac{(\mathit{NewSolution} - \mathit{OldSolution})}{10})\mathbf{T} \quad (4.3)$$

$$\mathbf{p}_f = \begin{cases} 0.8 & \mathit{NewSolution} \text{ is feasible} \\ 0.4 & \text{otherwise} \end{cases} \quad (4.4)$$

$T$ (Temperature) :Initial temperature is taken 1 and it drops very slowly as the iteration increases.

Because, heuristic algorithm will last long and the new solutions are not totally random. Change of temperature can be seen at equation 4.5.

$$\mathbf{T}_{\text{iteration}} = \mathbf{T}_0(0.9999^{\text{iteration}}) \quad (4.5)$$

Termination condition : Algorithm stops after the given time to heuristic algorithm runs out. Given time is the same time Cplex is given after finding initial solution. Given time equation is shown below. Time is checked after each step.

$$\text{givenTime(seconds)} = \text{optimizationFinalTime} - \text{initialSolutionTime} \quad (4.6)$$

Turn: Feasible-maker moves generally makes solution worse and may break the previous move. Therefore it is desired to change solution pool more before making it feasible again. Therefore we added this parameter to the algorithm.

In short, simulated algorithm starts with using initialization function, then identifies if found solution is feasible and goes on. Then, algorithm does drop, add and swap moves in order, since there is no need for a choosing process. After each move, infeasibility is identified and process goes into control function to decide if a move is accepted. These three move is done three times before it goes to infeasibility breaking moves, these moves also have feasibility check and accepting function afterwards because it is not guaranteed that they will generate a good or feasible solution.

These functions go on until termination time is satisfied, termination time is checked after each move to do on time termination. Pseudo code of simulated annealing algorithm is shown at figure 4.2.

```

Initialize(CPLEX)
if relativeGap > 0.95 then
    Initialize(Simple Algorithm)
end if
Penalize, Turn = 0 and iteration = 0
while elapsedTime < givenTime do
    Drop, Penalize, Control and iteration ++
    if elapsedTime > givenTime then
        Break
    end if
    Add, Penalize, Control and iteration ++
    if elapsedTime > givenTime then
        Break
    end if
    Swap, Penalize, Control, Turn ++ and iteration ++
    if elapsedTime > givenTime then
        Break
    end if
    if Turn = 3 then
        Drop Extra Workers, Penalize, Control, Turn = 0 and iteration ++
        if elapsedTime > givenTime then
            Break
        end if
        Add To Satisfy Demand, Penalize, Control and iteration ++
    end if
end while
Return BestObjective and BestSolution

```

Figure 4.2. Simulated annealing algorithm.

## 5. EXPERIMENTS AND RESULTS

In this chapter, first section includes test parameters and their assumptions, which are used as inputs of Cplex and simulated annealing. Second section, describes outputs of the algorithms and evaluates their usage at evaluating the performance. Finally, third part gives computational results and shows how solvers react on different cases of parameters.

### 5.1. Generating Data and Test Assumptions

Table 5.1 gives parameter values of test instances we picked with some test assumptions. These parameters and assumptions are explained in details below at this section.

Table 5.1. Test parameter sets.

<b>Parameters</b>	<b>Values</b>
Planning horizon(days)	1, 4, 7, 14, 18
Set of airplanes(airplanes)	100, 150
Set of worker groups(amount of worker groups)	129
Total workers(workers)	300, 450
Worker group sizes(workers)	Total Workers / Set of worker groups
Set of qualification of tasks(qualifications)	9
Required demand minimum value	0
Required demand maximum value	2
Desired demand case 1 minimum value	2
Desired demand case 1 maximum value	4
Desired demand case 2 minimum value	2
Desired demand case 2 maximum value	4
Amount of replications	3
Optimization and heuristic improvement time(seconds)	600-Initial solution run time
Initial solution run time(seconds)	300
Initial solution relative gap(%)	50

### 5.1.1. Basic parameters

$I$ (Planning horizon): Planning horizon is experimented for 1, 4, 7, 14 and 18 days. Flight schedules of planes and workforce constraints are daily and it is required to see if the assignments are able to satisfy the decision for longer plans. Because aggregated workforce shift assignments are more meaningful for longer horizon planning. This will increase the complexity of the problem.

$K$ (Set of airplanes): Model will be tested for 100 and 150 airplanes to maintenance. Increasing the amount of airplanes is going to make workforce busy. Moreover, it will introduce new plane availability sets for our workforce as we add more flight schedules to the system by each airplane.

$G$ (Set of worker groups): We assumed that workers can get at most 3 qualifications at our data. Because having workers able to do too many tasks is very costly and unlikely to be preferred. Therefore, all qualification combinations of having 1 to 3 among all qualifications, which makes 129 worker groups defined at total that shows workers with different qualification combinations.

$W$ (Total workers): We tested the system for 300 and 450 workers at problem instances. Having more workers makes it easier to follow desired demand closely while being able to satisfy required demand.

$W_g$ (Worker group sizes): Worker group sizes are generated by dividing total workers into worker groups equally and giving remainders to first for worker groups for simplicity. It isn't interesting to change numbers of workers at each groups since it isn't important for results. Therefore, each qualification combination has almost the same amount of workers capable.

$C$ (Set of qualifications): Increasing set of qualifications increases worker groups and reduces the portion of workers that is able to satisfy a specific task from total workers.

However, changing qualification set doesn't change results too much, because that also reduces portion of demand for specific qualification as well. Moreover, changing this parameter doesn't produce comparable results when we tried changing demand generation rules as well. Amount of qualifications is taken as 9 for problem instances.

### **5.1.2. Generating scheduled cumulative required and desired demand sets with randomization**

Although task manpower demand can occur before maintenance slots, we were more interested in using demands as scheduled while doing total worker shift assignments

Therefore, demand sets are assumed to be scheduled at available slots of planes for maintenance. Desired demand and required demand represents different schedules of demand. Desired demand also includes more demand than planned.

While generating these demand schedules, both required and desired demand schedules are generated randomly with using minimum and maximum values. Desired demand has also two different case of generating to create different case of worker assignments.

Moreover, both demand schedules use same plane availability set for each aircraft, therefore, both demands sets are scheduled at same maintenance slots of aircraft.

After these demand schedules are generated, these demand schedules are turned into cumulative to be used on model and solvers.

Plane availability sets: These sets show patterns daily available maintenance slots and maintenance demand(both required and desired) schedules are generated at these time slots. Plane availability sets are assumed to repeat themselves for each day. Thus, every day, available periods(hours) are assumed to be the same. Each airplane follows one set and this depends on flight schedule of the airplane.

Moreover, some airplanes require less maintenance jobs and or have shorter time slots for maintenance. Therefore, some airplanes has shorter flight routes while some has longer flight routes, these cases affects plane availability for maintenance.

While generating problem instances, these sets will start from different periods(hours) because flight schedules are separate at reality and they all have their own daily loop of maintenance order with different starting times.

Table 5.2. Daily plane availability sets for different flights.

Aircraft	Periods after a given starting period																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
<b>1</b>	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
<b>2</b>	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
<b>3</b>	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0
<b>4</b>	1	1	1	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0

Table 5.2 shows instances of plane availability sets for different kinds of flights by their length after a given starting period of a day, While period 0 denotes the starting period. In longer flights, maintenance slots are less separate. For instance, at first case, all of maintenance jobs can be done in one interval each day. Between flights there are minor maintenance tasks needs to be done and there is also a daily maintenance for airplanes which is longer.

There are 15 plane availability sets used. These sets include 3 instances of very long flights, 4 instances of long flights, 4 instances of mid length flights and 4 instances of short flights. Table 5.3 gives plane availability set examples for short distance flight schedules after a given starting period of a day.

Minimum and maximum values: Both required and desired demand are generated with uniform distribution before converted into cumulative.

Table 5.3. Daily plane availability sets for short flights.

Aircraft	Periods after a given starting period																							Workload	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22		23
<b>1</b>	1	1	1	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	8 hours
<b>2</b>	1	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	9 hours
<b>3</b>	1	1	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	6 hours
<b>4</b>	1	1	1	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	10 hours

Manpower demand of an aircraft is scheduled at a period where aircraft is available for maintenance with a value between minimum and maximum amount of workforce. This demand is generated for a task that requires random qualification for that time slot. We didn't define urgency of the tasks is explicitly at tests. We generated a random demand since it is a test instance. In addition, both demand sets and workforce assignment will be aggregated therefore it doesn't have much meaning for test results.

For required demand minimum value is 0 and maximum value is 2. When required demand minimum and maximum values increase, solution pool gets much smaller since it will be harder to satisfy the required demand. We didn't prefer increasing required demand minimum and maximum values for test instances, because it would make desired demand sets closer to required demand which makes both demand treated be almost the same and it causes having problems with workforce assignments.

Cases of desired demand: Model is tested for two cases of desired demand, to do different overall workforce shift assignments:

First case is meant for having manpower able earlier. and it starts normal with its minimum value as 2, maximum value as 4 after a certain period of day, it stops increasing until next day starts, then starts again.

Second case is the simple case with min value 2 and max value 4, it aims just having more workforce than necessary for unknown events and to do jobs earlier.

Figure 5.1 shows an instance of daily cumulative required demand with these two cumulative desired demand cases for one qualification.

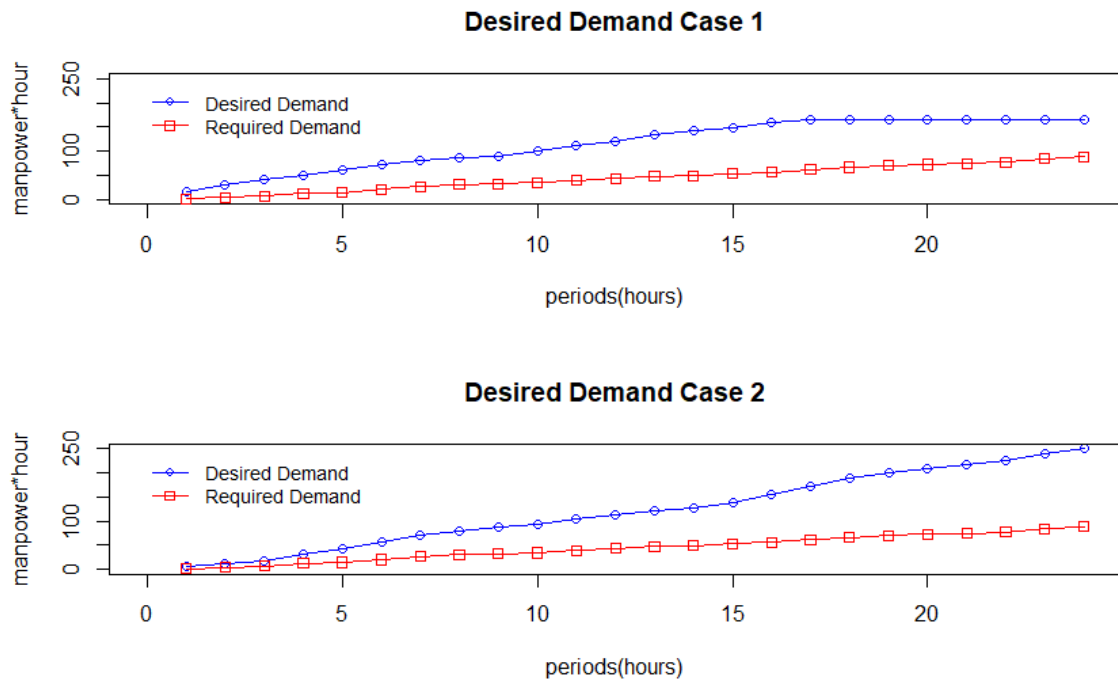


Figure 5.1. Different cases of desired demand with required demand.

### 5.1.3. Termination rules of problems :

Model running time: Initial solution finding process with Cplex has 300 second run time limit if it can't find good enough solution. Afterwards, both Cplex and simulated annealing are given the remaining time to make total time to 600 second. Therefore, Cplex will run 600 second at total, if it can't find optimal solution before and simulated annealing will have same time for improvement.

Model relative gap as initial solution: Solver stops once finding a solution with maximum 0.5 relative gap. Relative gap formula is defined at equation 4.2 After initial solution is found, there is no gap limit as stopping condition.

## 5.2. Outputs of Solvers

After solving each problem instance, we get an output file that includes worker shift assignments, objective values, solver times, relative gaps of optimization solutions and heuristic solution gap. Example of output file can be seen at table 5.4.

Table 5.4. Result table example.

TestCase	I	C	W	K	DDC	R	CplexResult	Ini.Result	Heur.Result	CplexTime	Ini.Time	Heur.Time	RelativeGap	Heur.Gap
5	1	9	300	100	2	2	613	899	630	600.1	0.1	600.5	0	0.03

From table 5.4, first columns are parameters; test case is the number of the simulation case,  $I$  shows planning horizon as amount of days,  $C$  shows set of qualifications,  $K$  shows amount of planes to maintenance, DDC shows desired demand case which is first or second case,  $R$  replication number of same problem instance,

Afterwards, objective values are shown, Cplex result shows final cplex objective value, ini. result shows initial objective value, heur. result shows simulated annealing best objective value.

Next columns show solution run times as seconds, Cplex time is total optimization run time which includes the time to find initial solution. Ini. time shows initial solution run time and heur. time shows simulated annealing algorithm run time.

Next column shows relative gap of final Cplex solution. Relative gap formula was given at equation 4.2 as mentioned. Solution is optimal if relative gap is zero.

Last column shows heuristic gap which is calculated as seen at equation 5.1. This can be negative as well, if heuristic solution has a better result than optimization using same solution time.

$$HeuristicGap = \frac{HeuristicSolution - CplexSolution}{e^{-5} + CplexSolution} \quad (5.1)$$

Each solver's objective value, run time and relative gap shows their performance. Moreover, to evaluate results better and to get generic results, we took average values of each replications of same problem instances. Therefore at average output table, all results are average of three different replications of a problem instance.

Worker shift assignment variable is also found as output of problem well. However, it can't be used to evaluate the solver performance because it is incomparable with other results to understand the performance. This variable is meant to be used as an input to further planning problem problems such as rostering.

### 5.3. Computational Results

ILOG CPLEX 12.5.1 is used as an optimization tool with C# program with visual studio compiler. Simulated annealing code is also written at C#.

With our preferences and assumptions at generating data sets, there are 120 test cases based on 3 replications of 40 different problem instances, including five different planning horizons, two different amount of workers, two different cases of desired demand generation, two different amount of airplanes. Test instances and parameters were shown at table 5.1 in details before.

These problem instances have feasible solution poo because of plane and worker parameter values we picked. For instance, picking very low worker parameter values makes the problem infeasible, since there is a minimal necessary worker amount requirement to satisfy required demand schedule coming from airplanes.

We tested solvers for different worker amounts for 150 planes to maintenance to show the when problem becomes feasible and also to show change of objective values by adding workers. It can be seen at figure 5.2 for 1 day planning horizon. If worker parameter for less than 262 workers, problem starts to be infeasible for all solution pool for some replications. Moreover, it can be observed that objective value drops until a certain point as we increase worker amount.

When it stops improving, it means that we have enough workers to satisfy desired demand schedule as good as possible. For results, simulated annealing algorithm couldn't find a good solution for lowest feasible worker parameter value because, solution pool is too small and therefore it is very difficult to identify next good solution. Moreover, the problem would be feasible by choosing lower worker parameter values, if there were less airplanes to maintenance.

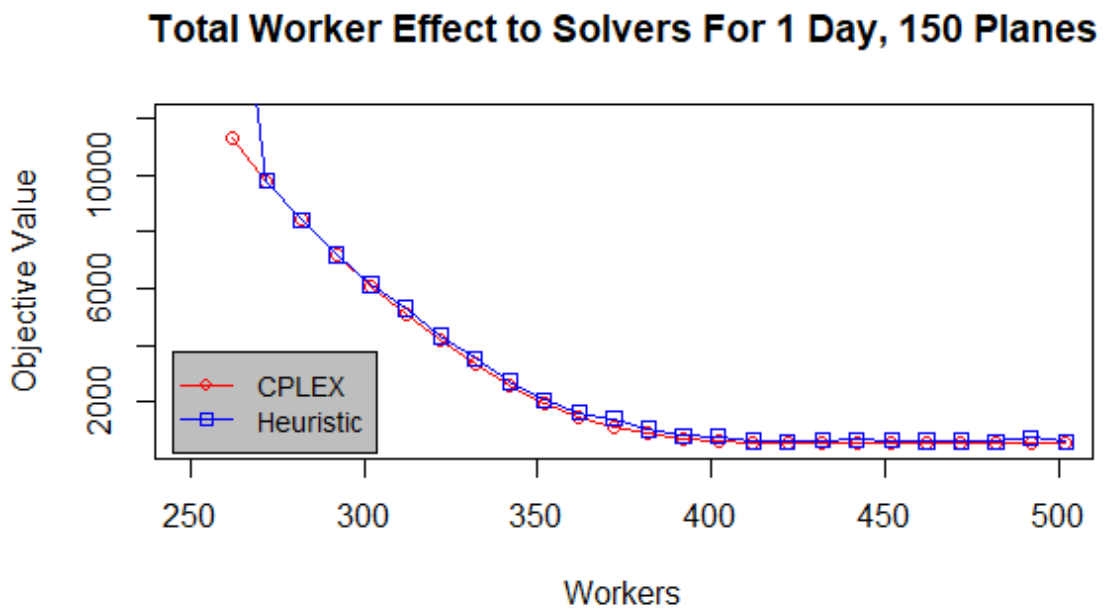


Figure 5.2. Total worker effect to solvers for 1 day 150 planes 9 qualifications.

Average results of test cases can be seen at table 5.5. As we can see, the problem gets much more complex as planning horizon increases. We can see that solvers behave differently for short(1, 4, 7 days) and longer(14, 18 days) planning horizons. Therefore, we generally separated problem instances according to planning horizon while evaluating results. For short planning horizons, solution time can see maximum run time as the planning horizon increases unless solution pool isn't too small. For instance, cases with 300 workers , 150 planes and second desired demand case has a small solution pool. Therefore, solution time is short and Cplex finds optimal.

Table 5.5. Average result table of all problem instances.

TestInstance	I	C	W	K	DDC	CplexResult	Ini.Result	Heur.Result	CplexTime	Ini.Time	Heur.Time	RelativeGap	Heur.Gap
1	1	9	300	100	1	1560	1663.67	1642.67	1.33	0.1	1.27	0	0.05
2	1	9	300	100	2	549.67	776	593	455.77	0	455.93	0	0.08
3	1	9	300	150	1	2704.67	2749.33	2749.33	0.83	0	0.83	0	0.02
4	1	9	300	150	2	5716	5770.67	5770.67	0.43	0.03	0.4	0	0.01
5	1	9	450	100	1	1560	1663.67	1634.33	1.23	0	1.23	0	0.05
6	1	9	450	100	2	549.67	773.67	586.67	417.5	0	417.53	0	0.07
7	1	9	450	150	1	2009.67	2107	2090.33	1.13	0	1.13	0	0.04
8	1	9	450	150	2	619	890.33	727.67	79.17	0	79.23	0	0.18
9	4	9	300	100	1	8700	9656.33	9653.67	463.3	0.8	463.23	0	0.11
10	4	9	300	100	2	2104	3679.67	3561.33	600.07	11.53	588.9	0.05	0.7
11	4	9	300	150	1	15931	16689	16578.33	600.07	2.97	597.77	0	0.05
12	4	9	300	150	2	173718.33	174203.67	173988.33	6.3	1.53	5.43	0	0
13	4	9	450	100	1	8701.33	9506.67	9300.67	436.5	0.67	436.5	0	0.07
14	4	9	450	100	2	2046	3593	2826	600.1	12.17	588.7	0.05	0.39
15	4	9	450	150	1	10227	10771.67	10753	600.03	1.93	598.87	0	0.05
16	4	9	450	150	2	2302.67	3554.67	3183.33	600.1	15.17	585.5	0.05	0.38
17	7	9	300	100	1	15697.33	16099.33	16099.33	600.07	18.5	583.77	0	0.02
18	7	9	300	100	2	3882.33	5594	5525	600.1	85.23	517.03	0.1	0.43
19	7	9	300	150	1	20615.67	22106.67	22106.67	600.37	30.03	572.63	0.01	0.07
20	7	9	300	150	2	487439.33	487453.33	487453.33	14.57	8.8	8.47	0	0
21	7	9	450	100	1	15697.33	16027	16027	600.07	24.83	577.27	0	0.02
22	7	9	450	100	2	3805	5775.67	5712.67	600.13	65.07	537.43	0.08	0.51
23	7	9	450	150	1	16746.33	17544.33	17544.33	600.2	27.77	574.47	0	0.04
24	7	9	450	150	2	4725.33	6774	6669.33	600.27	95.8	506.6	0.13	0.42
25	14	9	300	100	1	27388	28500.33	28500.33	600.1	240.17	370.87	0.04	0.04
26	14	9	300	100	2	205475937	234726923	1586540.33	600.13	306.57	324.43	1	-0.99
27	14	9	300	150	1	60290.67	161432921	1112344.33	600.2	315.17	309.23	0.05	1
28	14	9	300	150	2	2113679.67	2113684.3	2113684.33	105.93	100.97	29.07	0	0
29	14	9	450	100	1	27633.33	259461696	768801	600.2	283.17	346.1	0.04	0.68
30	14	9	450	100	2	356785784	383873427	231251	600.37	303.27	333.73	1	-1
31	14	9	450	150	1	34449.33	120995222	154335	600.2	293.83	325.63	0.04	0.36
32	14	9	450	150	2	369141313	410023255	2477277.67	600.17	304.83	325.63	1	-0.99
33	18	9	300	100	1	46616	47627.67	47626.33	600.2	270.27	373.03	0.02	0.02
34	18	9	300	100	2	449099110	476236092	2469192	600.37	302.93	347.83	1	-1
35	18	9	300	150	1	199433352	280705752	1546861.33	600.37	346.1	320.1	0.81	-0.33
36	18	9	300	150	2	3351741.33	3351744	3351744	125.8	107.7	60.13	0	0
37	18	9	450	100	1	266315025	528371306	1655338	605.3	384.47	292.57	0.35	0
38	18	9	450	100	2	719998710	746903451	320593.33	600.4	306.07	352.57	1	-1
39	18	9	450	150	1	55675.67	455125329	355206.67	600.13	304.1	347.23	0.17	1
40	18	9	450	150	2	612698430	679991698	3964406	600.3	305.77	347.43	1	-0.99

Cplex works well for cases of short planning horizons. Optimal solution(0 relative gap) is found at many cases and highest average relative gap is 13 percent within max run time. Therefore, it is not expected for simulated annealing to beat Cplex solutions here. Moreover, simulated annealing always got initial solutions from Cplex for short planning horizons.

Figure 5.3 shows examples of change of objective values with different workers and airplanes at system for 1 day planning horizon. For desired demand cases, we can see that objective value of first desired demand case is generally higher than the second case for short planning horizons. Because, first desired demand case follows different trend than required demand as seen at figure 5.1. Because, first case of desired demand problem reduces good solution pool and thus, reduces solution time it can be seen from solution times at table 5.5. Moreover, instances with 300 workers, 150 planes are exceptions, where problem instances doesn't have enough workers to do necessary shift assignments as desired. These cases have highest objective values and second desired demand case has higher objective value than first. These observations are also similar for 4 days and 7 days planning horizon as seen at table 5.5.

Simulated annealing improves initial solutions better for second desired demand case for 1 day planning horizon ,if there are enough workers at system to do necessary changes. because, second desired demand case generally gives a larger solution pool to improve the results. However, simulated annealing still can't beat Cplex results for short planning horizons and if we increase planning horizon to 4 days or 7 days, simulated annealing performance drops greatly as seen at figure 5.4 even when there are plenty workers available to do any shift assignments. For many cases, algorithm stays around initial solution, generally because initial solution is already good. However, as planning horizon increases, decision variable gets bigger as well and good improvements will start to require much longer run time.

However, as seen from table 5.5, for 14 days and 18 days planning horizons, Cplex couldn't always find good solutions at given time. Especially at second desired demand case because it generally requires more run time to get good results.

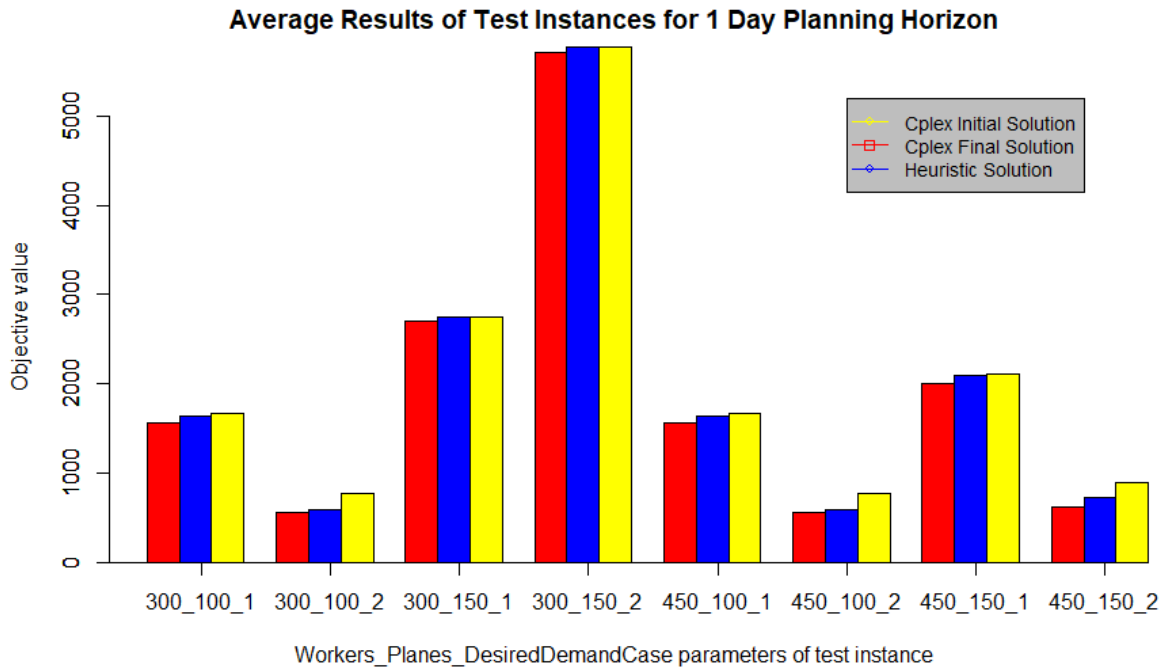


Figure 5.3. Average objective values of each test instance for 1 day planning horizon.

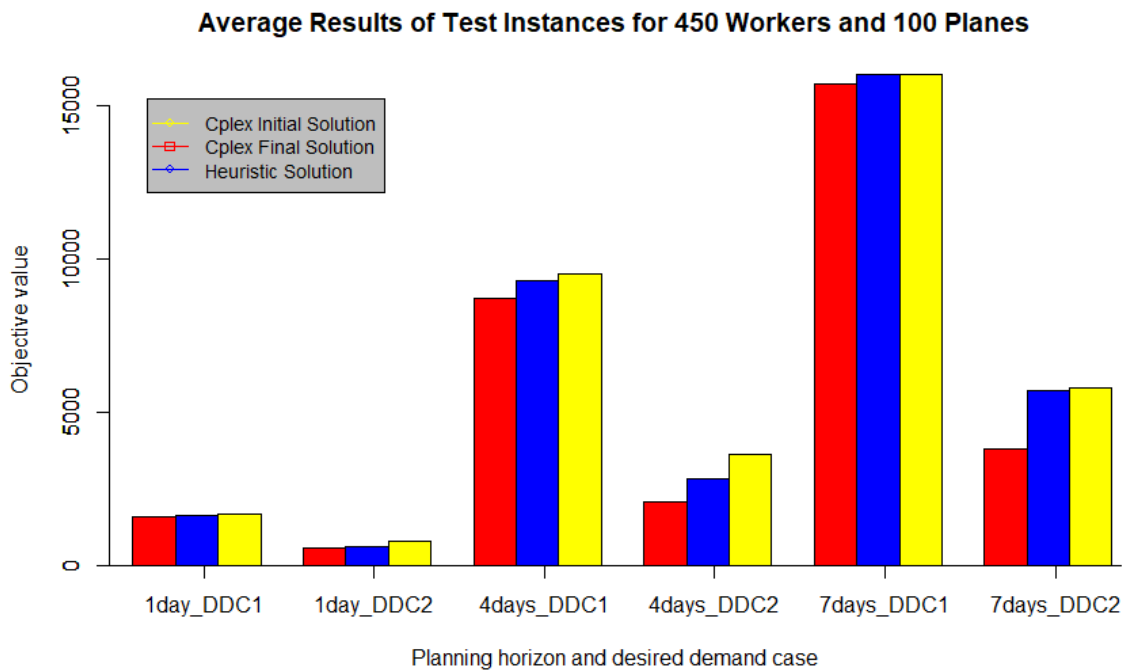


Figure 5.4. Total worker effect to solvers for 1 day 150 planes 9 qualifications.

This also can be seen from high relative gaps of Cplex results at table 5.5. We can see that simulated annealing sometimes doesn't prefer to get initial solution from Cplex. However, after Cplex goes on improving, sometimes it finds a better solution than simulated annealing algorithm at remaining time. It can be seen from instances of 29, 31, 33, 39 from table 5.5. Shortly, for 14, 18 days planning horizons, Cplex finds better solutions if solution pool is small.

If solution pool is larger, then Cplex can not find a good solution and initial solution of simulated annealing is much better than final result of Cplex. It can be seen from the test instances with negative heuristic gap at table 5.5.

To show example of all results of problem instances, results for all replications for 1 day planning horizon are given at table 5.6. As we can see, objective values are quite similar at each replication of same instance. But, solution time and simulated annealing improvement sometimes can change considerably with running same problem instance again.

Table 5.6. Result table for planning horizon 1 day.

TestCase	I	C	W	K	DDC	R	CplexResult	Ini.Result	Heur.Result	CplexTime	Ini.Time	Heur.Time	RelativeGap	Heur.Gap
1	1	9	300	100	1	1	1560	1610	1610	1.4	0.2	1.3	0	0.03
2	1	9	300	100	1	2	1627	1771	1771	1.4	0.1	1.3	0	0.09
3	1	9	300	100	1	3	1493	1610	1547	1.2	0	1.2	0	0.04
4	1	9	300	100	2	1	543	712	620	167.1	0	167.1	0	0.14
5	1	9	300	100	2	2	613	899	630	600.1	0	600.5	0	0.03
6	1	9	300	100	2	3	493	717	529	600.1	0	600.2	0.01	0.07
7	1	9	300	150	1	1	3286	3302	3302	1.1	0	1.1	0	0
8	1	9	300	150	1	2	2934	3020	3020	0.9	0	0.9	0	0.03
9	1	9	300	150	1	3	1894	1926	1926	0.5	0	0.5	0	0.02
10	1	9	300	150	2	1	6293	6337	6337	0.3	0.1	0.2	0	0.01
11	1	9	300	150	2	2	6736	6778	6778	0.2	0	0.2	0	0.01
12	1	9	300	150	2	3	4119	4197	4197	0.8	0	0.8	0	0.02
13	1	9	450	100	1	1	1560	1610	1610	1.5	0	1.5	0	0.03
14	1	9	450	100	1	2	1627	1771	1723	1.2	0	1.2	0	0.06
15	1	9	450	100	1	3	1493	1610	1570	1	0	1	0	0.05
16	1	9	450	100	2	1	543	705	569	52.3	0	52.3	0	0.05
17	1	9	450	100	2	2	613	899	650	600.1	0	600.1	0	0.06
18	1	9	450	100	2	3	493	717	541	600.1	0	600.2	0.01	0.1
19	1	9	450	150	1	1	2108	2181	2181	1	0	1	0	0.03
20	1	9	450	150	1	2	2029	2202	2152	1.2	0	1.2	0	0.06
21	1	9	450	150	1	3	1892	1938	1938	1.2	0	1.2	0	0.02
22	1	9	450	150	2	1	582	817	696	19.7	0	19.7	0	0.2
23	1	9	450	150	2	2	707	1000	816	213.6	0	213.7	0	0.15
24	1	9	450	150	2	3	568	854	671	4.2	0	4.3	0	0.18

For instance, as seen at table 5.6, some test cases might result algorithm to work unnecessarily long. problem can see max run time even for 1 day planning horizon. But, it means solver is proceeding without improvement for a long time. To show this process in details, we recorded how Cplex and simulated annealing algorithms progress over time by taking the objective values around each 10 seconds of run time for some problem instances. Figure 5.5 shows progress of solvers for 14 day horizon, 300 workers, 100 planes to maintenance with desired demand case 2. This problem instance is taken as an example because it consumes more time and has more room for improvement.

Figure 5.5 mainly is an example that cplex doesn't do any improvement at shortest planning horizon after very short time, this is based on process of increasing relaxed lower bound near best integer solution and meanwhile no better solution is found. However, simulated annealing algorithm tries to catch up to Cplex solution as time passes. At other examples for 1 day planning horizon Cplex does work for longer time but stil improvement portion among the used run time is small.

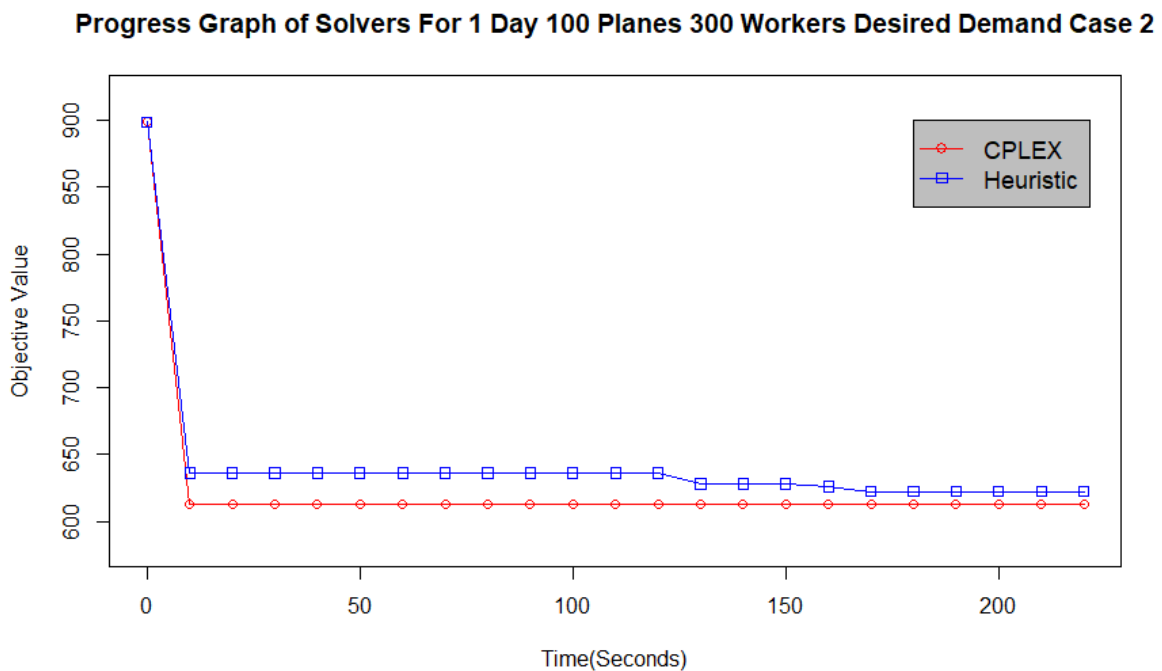


Figure 5.5. Progress graph for 1 day, 300 workers, 100 planes and DDC 2.

If we show this for 4 days planning horizon, figure 5.6 shows that Cplex uses more portion of given time for improvement. Because problem gets larger as planning horizon increases. However, simulated annealing is stopped at a local maximum and couldn't leave it. Probably because it couldn't find the necessary changes to do improvement. However at start simulated annealing started better. If we replicate this case, overall simulated annealing we defined improves solutions worse as planning horizon increases, therefore it will not be too different.

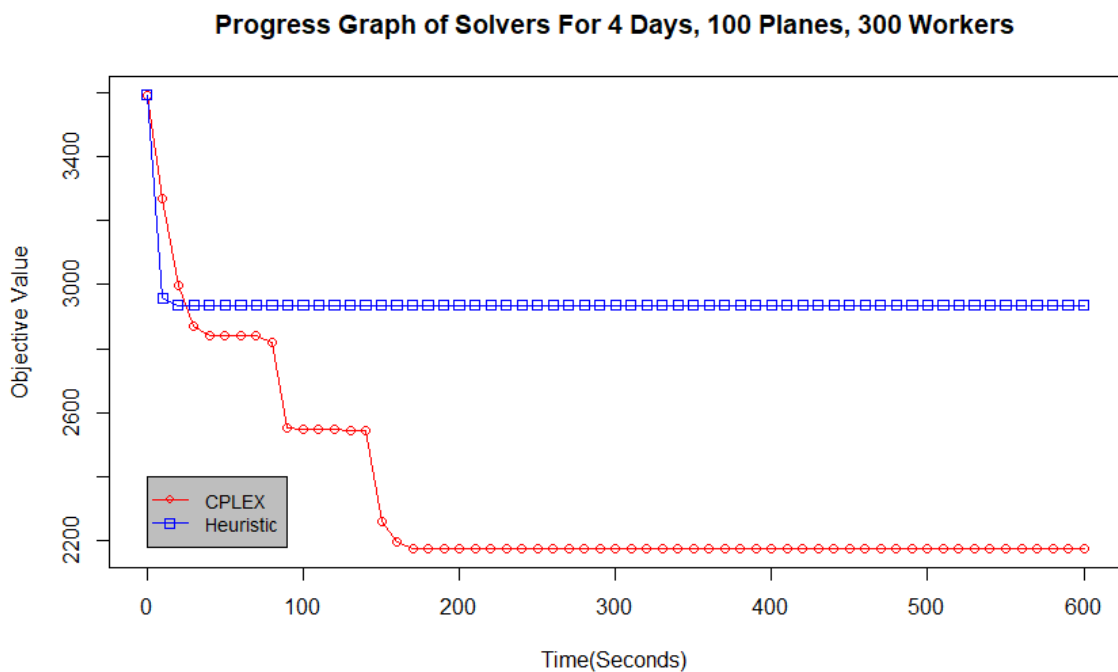


Figure 5.6. Progress graph for 4 days, 300 workers, 100 planes and DDC 2.

### 5.3.1. Discussion of results and future works

Cumulative demand seems to be good for defining a desired demand schedule, in case we want our system to be more able for busy schedule scenarios or unplanned maintenance demands.

Problem can take very long to solve due to change of planning horizon, worker and plane amounts and desired demand case.

Cplex works very smooth for short planning horizons and its improvement doesn't change much after a short time. That also makes it logical to have heuristic take initial solution from Cplex for these problem instances.

For simulated annealing performance, its improvement function works worse as the planning horizon increases. However, its initialization function makes the improvement since Cplex solutions are bad for many cases of longer planning horizons.

Therefore, heuristic improvement functions can be different. However, demand satisfaction constraints makes problem infeasible a lot while searching, therefore it is difficult to come up with a fast and big improvement algorithm. We tested different searching moves for simulated annealing algorithm before.

This problem can be tested for longer run time limit, then Cplex will give better results. However, afterwards problem will trouble with solving instances of even longer planning horizons. Therefore, a heuristic algorithm is helpful as an improvement tool.

From the results we have seen that problem is infeasible for different plane and worker parameter values as a result of demand satisfaction constraints. This should be prevented by introducing delay at maintenance to model with respective penalties.

This study can be used as an input for rostering problems. We will know which qualified workers to assign to which shift. If it is not possible to open shifts all the periods result can be aggregated into few basic shifts.

To specify shift assignments to individuals, assignment variable can be added and it can has its daily availability constraints. Or without using individual assignment variable, we can simply track how many people we used from worker groups with relative constraints.

## 6. CONCLUSION

In this study, we offered a new formulation to do total aircraft maintenance crew shift assignments to satisfy maintenance tasks in a desired schedule by taking account of qualification requirement of the tasks.

In conclusion, this study can be useful to monitor overall workforce shift assignments as by adding different cases of maintenance demand schedule scenarios to keep up with. Although, problem is still bound to keep many worker shifts to satisfy required demand, making a new cumulative demand schedule scenario can help changing maintenance workforce shift assignments in a way to do some maintenance tasks earlier.

For solving this problem, it is observed that Cplex solver behaves good as long as planning horizon is short(7 days or shorter). Therefore, there is no need for heuristic algorithm for these cases. For longer planning horizon cases, given time for Cplex solver can be increased but after increasing problem size further, Cplex will still be unable to solve the problem at a reasonable time.

Simulated annealing was fine as a meta-heuristic because of being able to search from worse solutions and infeasible region. However, move operators were doing very small changes at result and algorithm uses time inefficiently. Therefore, performance of algorithm mainly came from getting a better initial solution. However, it is difficult to build fast move operators that ensures feasibility.

Finally, it seems very difficult to come up with one heuristic algorithm that solves all problem instances efficiently. Because change of some parameters at problem instances entirely changes the solution pool and problem nature.

## REFERENCES

1. Van den Bergh, J., P. De Bruecker, J. Belien and J. Peeters, “Aircraft maintenance operations: state of the art”, *HUB Research Paper*, 2013.
2. Van den Bergh, J., J. Beliën, P. De Bruecker, E. Demeulemeester and L. De Boeck, “Personnel scheduling: A literature review”, *European Journal of Operational Research*, Vol. 226, No. 3, pp. 367–385, 2013.
3. Saltoglu, R., N. Humaira and G. Inalhan, “Scheduled maintenance and downtime cost in aircraft maintenance management”, *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, Vol. 10, No. 3, pp. 602–607, 2016.
4. Bazargan, M., *Airline operations and scheduling*, Routledge, 2016.
5. 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.
6. Papadakos, N., “Integrated airline scheduling”, *Computers & Operations Research*, Vol. 36, No. 1, pp. 176–195, 2009.
7. Mercier, A. and F. Soumis, “An integrated aircraft routing, crew scheduling and flight retiming model”, *Computers & Operations Research*, Vol. 34, No. 8, pp. 2251–2265, 2007.
8. Clarke, L. W., C. A. Hane, E. L. Johnson and G. L. Nemhauser, “Maintenance and crew considerations in fleet assignment”, *Transportation Science*, Vol. 30, No. 3, pp. 249–260, 1996.
9. Sriram, C. and A. Haghani, “An optimization model for aircraft maintenance

- scheduling and re-assignment”, *Transportation Research Part A Policy and Practice*, Vol. 37, No. 1, pp. 29–48, 2003.
10. Bazargan, M., “Airline maintenance strategies—in-house vs. outsourced—an optimization approach”, *Journal of Quality in Maintenance Engineering*, Vol. 22, No. 2, pp. 114–129, 2016.
  11. Kasirzadeh, A., M. Saddoune and F. Soumis, “Airline crew scheduling: models, algorithms, and data sets”, *EURO Journal on Transportation and Logistics*, Vol. 6, No. 2, pp. 111–137, 2017.
  12. Cappanera, P. and G. Gallo, “A multicommodity flow approach to the crew rostering problem”, *Operations Research*, Vol. 52, No. 4, pp. 583–596, 2004.
  13. Dowling, D., M. Krishnamoorthy, H. Mackenzie and D. Sier, “Staff rostering at a large international airport”, *Annals of Operations Research*, Vol. 72, pp. 125–147, 1997.
  14. Chu, S. C., “Generating, scheduling and rostering of shift crew-duties: Applications at the Hong Kong International Airport”, *European Journal of Operational Research*, Vol. 177, No. 3, pp. 1764–1778, 2007.
  15. Safaei, N., D. Banjevic and A. K. Jardine, “Workforce-constrained maintenance scheduling for military aircraft fleet: a case study”, *Annals of Operations Research*, Vol. 186, No. 1, pp. 295–316, 2011.
  16. Gavranis, A. and G. Kozanidis, “An exact solution algorithm for maximizing the fleet availability of an aircraft unit subject to flight and maintenance requirements”, *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 2, 2013.
  17. Telhada, J., “Alternative MIP formulations for an integrated shift scheduling and task assignment problem”, *Discrete Applied Mathematics*, Vol. 164, pp. 328–343,

2014.

18. Yang, T.-H., S. Yan and H.-H. Chen, “An airline maintenance manpower planning model with flexible strategies”, *Journal of Air Transport Management*, Vol. 9, No. 4, pp. 233 – 239, 2003.
19. Yan, S., T.-H. Yang and H.-H. Chen, “Airline short-term maintenance manpower supply planning”, *Transportation Research Part A: Policy and Practice*, Vol. 38, No. 9, pp. 615–642, 2004.
20. Beliën, J., E. Demeulemeester, P. De Bruecker, J. Van den Bergh and B. Cardoen, “Integrated staffing and scheduling for an aircraft line maintenance problem”, *Computers & Operations Research*, Vol. 40, No. 4, pp. 1023–1033, 2013.
21. Alfares, H. K., “Aircraft maintenance workforce schedulingA case study”, *Journal of Quality in Maintenance Engineering*, Vol. 5, No. 2, pp. 78–89, 1999.
22. De Bruecker, P., J. Van den Bergh, J. Belien and E. Demeulemeester, “A two-stage mixed integer programming approach for optimizing the skill mix and training schedules for aircraft maintenance”, *Available at SSRN 2697491*, 2015.
23. Chen, G., W. He, L. C. Leung, T. Lan and Y. Han, “Assigning licenced technicians to maintenance tasks at aircraft maintenance base: a bi-objective approach and a Chinese airline application”, *International Journal of Production Research*, pp. 1–14, 2017.
24. Bruecker, P. D., J. V. den Bergh, J. Beliën and E. Demeulemeester, “A model enhancement heuristic for building robust aircraft maintenance personnel rosters with stochastic constraints”, *European Journal of Operational Research*, Vol. 246, No. 2, pp. 661 – 673, 2015.
25. De Bruecker, P., J. Van den Bergh, J. Belien and E. Demeulemeester, “A Tabu search heuristic for building aircraft maintenance personnel rosters”, *Available at*

*SSRN 2464033*, 2014.