ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

CREW RECOVERY OPTIMIZATION THROUGH DISRUPTION ANALYSIS AND DEEP LEARNING DRIVEN COLUMN GENERATION

Ph. D. THESIS

Ahmet HEREKOĞLU

Department of Industrial Engineering

Industrial Engineering Programme

FEBRUARY 2024

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

CREW RECOVERY OPTIMIZATION THROUGH DISRUPTION ANALYSIS AND DEEP LEARNING DRIVEN COLUMN GENERATION

Ph. D. THESIS

Ahmet HEREKOĞLU
(507132108)

Department of Industrial Engineering

Industrial Engineering Programme

Thesis Advisor: Prof. Dr. Özgür KABAK

FEBRUARY 2024

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

AKSAKLIK ANALİZİ VE DERİN ÖĞRENME TABANLI SÜTUN OLUŞTURMA İLE EKİP KURTARMA OPTİMİZASYONU

DOKTORA TEZİ

Ahmet HEREKOĞLU
(507132108)

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Özgür KABAK

ŞUBAT 2024

Ahmet HEREKOĞLU, a Ph.D. student of ITU Graduate School student ID 507132108 successfully defended the thesis/dissertation entitled "CREW RECOVERY OPTIMIZATION THROUGH DISRUPTION ANALYSIS AND DEEP LEARNING DRIVEN COLUMN GENERATION", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Özgür KABAK** ....
Istanbul Technical University

Jury Members : **Prof. Dr. Başar ÖZTAYŞİ** …..........................
Istanbul Technical University

Assoc. Prof. Bora ÇEKYAY …..........................
Yıldız Technical University

Prof. Dr. Y. İlker TOPÇU …...........................
Istanbul Technical University

Assoc. Prof. Sait GÜL ….......................…..
Bahcesehir University

Date of Submission : 27 December 2023
Date of Defense : 06 February 2024

To my family,

FOREWORD

In a world where global travel has become an integral part of our lives, the aviation industry plays a pivotal role in connecting people, businesses, and cultures. Within this industry, the optimal management of airline crew is a multifaceted challenge that directly impacts operational efficiency, passenger satisfaction, and overall safety. As we strive for more efficient and reliable air travel, the effective recovery of airline crew from disruptions has emerged as a critical area of research and innovation.

It is with great pride and enthusiasm that I present this PhD thesis which represents the culmination of years of rigorous research, exploration, and dedication to addressing one of the most complex and dynamic challenges within the aviation industry.

Traditional methods have long been in use to tackle the problem covered in this thesis, but they come with their limitations. As the aviation landscape evolves, so too must our approaches to crew recovery. This thesis embarks on a journey to explore the integration of cutting-edge technologies, such as deep learning, into the crew recovery optimization process. By leveraging historical data, sophisticated analysis, and the predictive power of artificial intelligence, we seek to revolutionize how we respond to disruptions, ensuring a more agile and responsive recovery.

Our aim is not only to propose innovative solutions but to also establish a comprehensive framework for addressing the multifaceted challenges of crew recovery. This research extends the boundaries of what is possible in this field and, in doing so, aims to bring about a new era of crew recovery optimization.

I extend my deepest gratitude to my advisors, colleagues and mentors who have supported and guided me throughout this journey. Their wisdom, expertise, and encouragement have been invaluable. I sincerely thank the ITU BAP organization for its support.

Finally and most importantly, I would like to express my gratitude to my family for the invaluable support and trust they have given me on this long journey.

February 2024 Ahmet Herekoğlu
(Industrial Engineer)

TABLE OF CONTENTS

Page

FOREWORD .... **ix**
TABLE OF CONTENTS .... **xi**
ABBREVIATIONS .... **xiii**
LIST OF TABLES .... **xv**
LIST OF FIGURES .... **xvii**
SUMMARY .... **xix**
ÖZET .... **xxi**
1. INTRODUCTION **1**
1.1 Artificial Intelligence Is Transforming World 4
1.2 Purpose of Thesis 5
1.3 Hypothesis 7
2. CREW RECOVERY PROBLEM **11**
2.1 Crew Management 12
2.2 Crew Scheduling 13
2.2.1 Crew pairing 16
2.2.2 Crew rostering 17
2.3 Crew Recovery 18
2.3.1 Disruption 21
2.3.2 Disruption management 22
2.3.3 Rules and regulations 24
2.3.4 Recovery actions 25
2.3.5 Optimization approaches of recovery process 30
3. PROPOSED SOLUTION METHODOLOGY **41**
3.1 Disruption Analysis: Learning From Disruptions 42
3.2 AI Model: Predicting Recovery Actions 43
3.3 Optimizer: Optimization of Recovery Actions 44
4. LEARNING FROM DISRUPTIONS **47**
4.1 Text Processing and Graph Clustering 48
4.1.1 Text processing 49
4.1.2 Graph clustering 49
4.1.3 Spectral clustering 52
4.1.3.1 Theory of spectral clustering 52
4.1.3.2 Approximation 54
4.2 A New Spectral Clustering Approximation Proposal 55
4.3 Finding Recovery Actions With New Spectral Method 62
5. PREDICTING RECOVERY ACTIONS **67**
5.1 Understanding Deep Learning 67
5.2 Automatic Machine Learning (AutoML) 69
5.3 Neural Architecture Search 71
5.4 Hyperparameter Optimization 72

5.5 Deep Learning Model For Predicting Recovery Actions 75
6. RECOVERING CREW FLIGHT DUTIES 77
6.1 Deep Learning Integrated Optimization 78
6.2 Column Generation 79
6.3 Proposed Mathematical Model 84
6.3.1 Sets, parameters and variables 84
6.3.2 Model 86
6.3.2.1 Master problem 86
6.3.2.2 Restricted master problem 86
6.3.2.3 Subproblem or pricing problem 86
7. COMPUTATIONAL RESULTS AND DISCUSSION 93
7.1 Technologies and Environment 93
7.2 Architecture 94
7.3 Data 96
7.4 Rules and Assumptions 97
7.4.1 Rules 97
7.4.2 Assumptions 100
7.5 Parameters 100
7.6 Runs & Results 102
7.7 Discussion 106
8. CONCLUSION AND FUTURE RESEARCH 109
REFERENCES 113
CURRICULUM VITAE 127

ABBREVIATIONS

AI	: Artificial Intelligence
ANN	: Artificial Neural Network
AutoML	: Automatic Machine Learning
CAP	: Crew Assignment Problem
CG	: Column Generation
CNN	: Convolutional Neural Network
CPP	: Crew Pairing Problem
CPU	: Central Processing Unit
CRP	: Crew Recovery Problem
CRR	: Crew Rostering Problem
CSP	: Crew Scheduling Problem
DL	: Deep Learning
DSS	: Decision Support Systems
Eurocontrol	: European Air Navigation Safety Organization
GA	: Genetic Algorithm
GB	: Gigabyte
GDPR	: General Data Protection Regulation
GNN	: Graph Neural Networks
GPU	: Graphical Processing Unit
IATA	: International Air Transport Association
IT	: Information Technology
LP	: Linear Programming
MAS	: Multi Agent System
MCDN	: Multi-Commodity Network Design Problem
MILP	: Mixed Integer Linear Programming
MIQCP	: Mixed Integer Quadratic Constrained Programming
MIQP	: Mixed Integer Quadratic Programming
ML	: Machine Learning
MLP	: Multi Layer Perceptron
MP	: Master Problem

NAS	: Neural Architecture Search
NLP	: Natural Language Programming
NP	: Non-Polynomial
OCC	: Operations Control Center
PCA	: Principal Component Analysis
QCP	: Quadratic Constrained Programming
QP	: Quadratic Programming
RBF	: Radial Basis Function
RMP	: Relaxed (Restricted) Master Problem
RNN	: Recurrent Neural Network
SCP	: Set Covering Problem
SP	: Sub Problem
SPP	: Set Partitioning Problem
US	: United States

LIST OF TABLES

Page

Table 2.1: Crew recovery and crew scheduling problem comparison. **21**
Table 2.2: Literature review on crew recovery problem. **31**
Table 4.1: Crew disruption distribution. **63**
Table 5.1: Deep learning model structure. **75**
Table 6.1: Sets in master problem. **85**
Table 6.2: Parameters in master problem. **85**
Table 6.3: Binary variables in master problem. **85**
Table 6.4: Sets in sub problem. **85**
Table 6.5: Parameters in sub problem. **85**
Table 6.6: Binary variables in sub problem. **86**
Table 6.7: Optimization action by recovery action. **90**
Table 7.1: Optimizer global parameters. **101**
Table 7.2: Optimizer runtime parameters. **102**
Table 7.3: Optimizer runtime dataset dimensions. **102**
Table 7.4: Optimization results for 1 day long recovery window. **103**
Table 7.5: Optimization results for 2 days long recovery window. **104**
Table 7.6: Optimization results for 3 days long recovery window. **104**

LIST OF FIGURES

Page

Figure 1.1: Delay Groups. **3**
Figure 1.2: AI penetration levels **4**
Figure 2.1: Overview of commercial airline operation. **14**
Figure 2.2: Overview of crew management. **15**
Figure 2.3: Sequential and integrated crew scheduling problem. **16**
Figure 2.4: A typical pairing. **17**
Figure 2.5: Disruption management. **28**
Figure 2.6: Crew Swapping **29**
Figure 2.7: Multi-commodity flow of crew schedules. **35**
Figure 2.8: Scope of the study. **39**
Figure 3.1: Phases in the study. **41**
Figure 4.1: Mini batch spectral clustering approximation. **57**
Figure 4.2: Initialization. **59**
Figure 4.3: Processing. **59**
Figure 4.4: Eigenvector selection. **60**
Figure 4.5: Clustering. **62**
Figure 5.1: Different deep learning architectures. **68**
Figure 5.2: AutoML optimization. **70**
Figure 5.3: Neural Network architecture and hyperparameter optimization **71**
Figure 5.4: Neural Architecture search techniques. **71**
Figure 5.5: Neural architecture search process. **72**
Figure 5.6: Auto keras modules. **75**
Figure 5.7: Deep learning model architecture. **76**
Figure 6.1: Optimization model flow. **89**
Figure 7.1: Solution architecture. **96**
Figure 7.2: Static and dynamic average optimal value comparison. **105**
Figure 7.3: Static and dynamic average run time comparison. **105**
Figure 7.4: Correlations between hyperparameters and results. **106**

CREW RECOVERY OPTIMIZATION THROUGH DISRUPTION ANALYSIS AND DEEP LEARNING DRIVEN COLUMN GENERATION

SUMMARY

Thanks to globalization, new travel opportunities and economic development have increased the interest in the aviation industry and air transport. The increase in the number of passengers and compliance with the regulations on passenger rights make transformation inevitable for the aviation industry and airlines. Airlines are reorganizing and managing all their resources in line with this transformation. The most critical resources of commercial airlines are the crew and aircraft, which together with passengers are key components of operational efficiency.

Unforeseen events such as bad weather conditions, aircraft malfunctions and crew absence may result in inefficiency in operations and thus in the utilization of aforementioned resources. These events are called disruptions. As disruptions such as delays are the primary and fundamental factor in passenger satisfaction and the airline's financial conditions, aviation companies devote valuable resources to analyzing disruptions and taking necessary actions. Actions known as recovery actions are the ones decided by the airline operations control center to overcome problems in the execution of plans due to disruptions.

The crew recovery problem is a complex optimization problem in the airline industry that involves scheduling and assigning crews to flights while taking into account various constraints such as legal regulations, crew availability, and cost. Several methods have been proposed to solve this problem, including heuristic and metaheuristic algorithms, integer programming, and constraint programming. It is a type of optimization problem that aims to choose the best recovery strategies to overcome crew disruptions. The main goal is to find the minimum cost combination of assignments that solve the problems related with crew plans while considering all constraints, especially flight time limitations.

One popular method for solving the crew recovery problem is the column generation algorithm, which involves generating and adding columns to the problem's LP relaxation until an optimal solution is obtained. Other optimization methods include simulated annealing, genetic algorithms, and ant colony optimization.

However, despite the various optimization methods proposed, the crew recovery problem remains a challenging and computationally expensive task due to the large number of variables and constraints involved. Furthermore, real-world crew recovery problems are dynamic and uncertain, requiring the ability to adapt quickly to unexpected events.

This is where machine learning (ML) can play a crucial role in developing an optimization method with AI support. By leveraging machine learning algorithms, we can learn from past data and experiences to make informed decisions and generate optimized solutions quickly and efficiently. Recovery strategies can be realized by

using actions known as recovery actions. Especially during the preparation of recovery models, more effective strategies are produced by using the actions learned from the past disruptions as inputs in the model.

The main motivation behind using actions as input is that learning-based approaches have the potential to generate more effective algorithms for large-scale and difficult optimization problems by inferring insights from historic datasets. Especially with the learning process, important points that people may miss in the solution process become easily noticeable and increase the success of the recovery process.

In this study, recovery actions provided by a customized deep learning (DL) model are used as input to the proposed optimization model, in which the objective function minimizes the total assignment cost of crew. Crew disruption data including some of the flight disruptions from a large scale airline is analyzed. Based on the analysis of the data, features are generated and recovery actions are extracted. The recovery actions are used as label and supervised learning methodology is used to train a customized deep learning model. Our hypothesis is based on the assumption that deep learning can provide needed insights in order to solve the problem in a shorter time while preserving the optimality at a certain level. The fundamental insight that we derive from the deep learning model is the recovery actions, which will be used while generating new columns. The actions are used in order to configure the columns faster than the classical column generation by either directly modifying the column itself or narrowing down the solution space. The resource or resource groups including crew and aircrafts are filtered according to the information provided by the deep learning. This makes it possible to search for new columns in a narrower solution space, which makes the search time shorter compared to the classical column generation techniques.

The primary goal of the study was to develop a model that balances solution quality and speed. Results indicate that the proposed method outperforms the reference model in terms of runtime while maintaining an acceptable level of optimality. This approach can be valuable for airline companies looking to efficiently address crew recovery challenges. Moreover, it contributes to the existing literature by introducing a new mathematical model and approximation method, demonstrating the potential of deep learning and optimization techniques for addressing complex aviation sector optimization problems and offering valuable insights for future research in the field of crew recovery.

AKSAKLIK ANALİZİ VE DERİN ÖĞRENME TABANLI SÜTUN OLUŞTURMA İLE EKİP KURTARMA OPTİMİZASYONU

ÖZET

Küreselleşme sayesinde yeni seyahat olanakları ve ekonomik gelişme, havacılık endüstrisine ve hava taşımacılığına olan ilgiyi artırmıştır. Yolcu sayısındaki artış ve yolcu haklarına ilişkin düzenlemelere uyum, havacılık sektörü ve havayolları için dönüşümü kaçınılmaz kılmaktadır. Havayolları tüm kaynaklarını söz konusu dönüşüme uyumlu olacak şekilde yeniden düzenlemekte ve yönetmektedir. Havayolları, bu dönemde tüm kaynaklarını dönüşüme uyumlu hale getirmek için kullanmaktadır. Bu, hem içsel süreçlerin hem de müşteri hizmetlerinin daha etkin ve rekabetçi bir şekilde yönetilmesini sağlamayı amaçlamaktadır. Havacılık sektöründeki bu dönüşüm süreci, havayollarının hem operasyonel hem de müşteri odaklı stratejilerini gözden geçirmelerini ve küresel taleplere daha etkin bir şekilde yanıt vermelerini sağlamaktadır. Ticari havayollarının etkin yanıt üretmedeki en kritik kaynakları, yolcularla birlikte operasyonel verimliliğin temel bileşenleri olan ekip ve uçaktır. Bu bağlamda, ekip ve uçakların yönetimi, havayolu şirketlerinin bu değişen dinamiklere uyum sağlamalarında kritik bir rol oynamaktadır.

Öngörülemeyen olaylar, özellikle kötü hava koşulları, uçak arızaları ve ekip yokluğu gibi durumlar, havacılık sektöründe aksaklıklara neden olabilmektedir. Bu aksaklıklar, operasyonel verimliliği ve bu bağlamda kullanılan kritik kaynakları, yani ekip ve uçakları, etkin bir şekilde kullanamama sonucunu ortaya çıkarmaktadır. Aksaklıklar, havacılık şirketleri için hem yolcu memnuniyeti hem de finansal durum açısından kritik öneme sahiptir. Özellikle gecikmeler gibi aksaklıklar, yolcu memnuniyetini olumsuz etkileyerek havayolu şirketinin itibarını zedeleme potansiyeline sahiptir. Ayrıca, operasyonel planlamalarda meydana gelen aksaklıklar, maliyet artışlarına ve kaynak israfına yol açabilir.

Havacılık şirketleri, aksaklıkları etkili bir şekilde yönetmek ve en aza indirmek için önleyici ve iyileştirici tedbirler almak zorundadır. Aksaklıkların analizi, bu olayların nedenlerini anlamak ve gelecekte benzer durumları önceden tahmin etmek adına önemli bir adımdır. Bu analizler, kötü hava koşulları, teknik sorunlar veya personel eksikliği gibi faktörlerin etkilerini değerlendirir ve operasyonel süreçlerin güçlendirilmesi için rehberlik edecek niteliktedir.

Aksaklıkların etkili bir şekilde yönetilmesi, havacılık şirketlerinin operasyonel verimliliğini artırmasına ve müşteri memnuniyetini korumasına yardımcı olur. Bu bağlamda, aksaklıkları analiz etmek ve bu olaylara hızlı ve etkili bir şekilde müdahale etmek, havayolu endüstrisinde başarılı bir şekilde faaliyet göstermek için kritik bir gerekliliktir. Havacılık şirketleri aksaklıkları analiz ederek alınması gereken aksiyonları belirlemek için önemli kaynaklar tahsis etmektedir. Kurtarma aksiyonları olarak bilinen aksiyonlar, aksaklıklar nedeniyle planların gerçekleştirilmesinde meydana gelen sorunların üstesinden gelmek için havayolu operasyon kontrol merkezleri tarafından karar verilen eylemlerdir.

Ekip kurtarma problemi, ekip aksaklıklarının üstesinden gelmek için en iyi kurtarma aksiyonlarını seçmeyi amaçlayan bir tür optimizasyon problemidir. Ana amaç, uçuş süresi sınırlamaları başta olmak üzere tüm kısıtları dikkate alırken sorunlu ekip planlarını kurtaran minimum maliyetli görev kombinasyonunu bulmaktır. Havayolu endüstrisinde yasal düzenlemeler, mürettebat mevcudiyeti ve maliyet gibi çeşitli kısıtlamaları dikkate alarak çizelgeleme ve uçuşlara ekip atamayı içeren karmaşık bir optimizasyon problemidir. Bu sorunu çözmek için metasezgisel algoritmalar, tamsayı programlama ve kısıtlama programlama dahil olmak üzere çeşitli yöntemler önerilmiştir. Ekip aksaklıklarının üstesinden gelmek için en iyi kurtarma stratejilerini seçmeyi amaçlayan bir optimizasyon problemi türü olan ekip kurtarma probleminde temel amaç, uçuş süresi sınırlamaları başta olmak üzere tüm kısıtlamaları göz önünde bulundurarak, ekip planları ile ilgili sorunları çözen görevlerin minimum maliyetli kombinasyonunu bulmaktır.

Ekip kurtarma problemini çözmek için popüler bir yöntem, optimal bir çözüm elde edilene kadar problemin LP gevşemesi ile sütunlar oluşturmayı ve sütunları eklemeyi içeren sütun oluşturma algoritmasıdır. Diğer optimizasyon yöntemleri arasında simüle edilmiş tavlama, genetik algoritmalar ve karınca kolonisi optimizasyonu bulunur.

Bununla birlikte, önerilen çeşitli optimizasyon yöntemlerine rağmen, ekip kurtarma problemi, ilgili çok sayıda değişken ve kısıtlama nedeniyle zorlu ve hesaplama açısından maliyetli olmaya devam etmektedir. Ayrıca, gerçek dünyadaki ekip kurtarma sorunları dinamik ve belirsizdir ve beklenmedik olaylara hızla uyum sağlama becerisi gerektirir.

Yapay zeka destekli bir optimizasyon yöntemi geliştirmede makine öğreniminin önemli rol oynayabileceği alan da burasıdır. Makine öğrenimi algoritmalarından yararlanarak, bilinçli kararlar almak ve hızlı ve verimli bir şekilde optimize edilmiş çözümler üretmek için geçmiş verilerden ve deneyimlerden ihtiyaç duyulan stratejiler öğrenilebilir. Kurtarma stratejileri, kurtarma eylemleri olarak bilinen eylemler kullanılarak gerçekleştirilebilir. Özellikle çözüm modellerinin hazırlanması aşamasında, geçmiş aksaklıklardan öğrenilen aksiyonlar modelde girdi olarak kullanılarak daha etkin çözümler üretilmektedir.

Eylemleri girdi olarak kullanmanın arkasındaki ana motivasyon, öğrenmeye dayalı yaklaşımların, geçmiş veri kümelerinden içgörüler çıkararak büyük ölçekli ve zor optimizasyon sorunları için daha etkili algoritmalar üretme potansiyeline sahip olmasıdır. Özellikle öğrenme süreci ile birlikte kişilerin çözüm sürecinde gözden kaçırabilecekleri önemli noktalar kolayca fark edilir hale gelir ve iyileşme sürecinin başarısını artırır.

Bu çalışmada, amaç fonksiyonunun toplam atama maliyetini minimize ederken özelleştirilmiş bir derin öğrenme modeli tarafından sağlanan kurtarma eylemlerini girdi olarak kullanan bir optimizasyon modeli geliştirilmiştir. Büyük ölçekli bir havayolu şirketinden bazı uçuş aksaklıklarını da içeren ekip aksaklık verileri analiz edilmiştir. Veri analizine sonucunda özellik seti üretilmiş ve kurtarma aksiyonları çıkarılmıştır. Kurtarma aksiyonları etiket olarak kullanılmış ve özelleştirilmiş derin öğrenme modelini eğitmek için denetimli öğrenme yöntemi kullanılmıştır. Bu çalışma kapsamında hipotezimiz, derin öğrenmenin optimalliği belirli bir düzeyde korurken problemi daha kısa sürede çözmek için gerekli içgörüleri sağlayabileceği varsayımına dayanmaktadır. Derin öğrenme modelinden çıkardığımız temel içgörü, yeni sütunlar oluşturulurken kullanılacak kurtarma eylemleridir. Eylemler, doğrudan sütunun kendisini değiştirerek veya çözüm alanını daraltarak klasik sütun oluşturmaya göre sütunları daha hızlı yapılandırmak kullanılır. Ekip ve uçak dahil olmak üzere kaynak veya kaynak grupları, derin öğrenme tarafından sağlanan bilgilere göre filtrelenir. Bu da yeni kolonların daha dar bir çözüm uzayında aranmasını mümkün kılmakta, bu da klasik kolon üretme tekniklerine göre arama süresini kısaltmaktadır.

Çalışmanın temel hedefi, çözüm kalitesi ile hızını dengeleyen bir model geliştirmektir. Sonuçlar, önerilen yöntemin çalışma süresi açısından referans modelini aştığını ve kabul edilebilir bir düzeyde optimaliteyi sürdürdüğünü göstermektedir. Bu yaklaşım, ekip kurtarma sorununu etkili bir şekilde çözmek isteyen havayolu şirketleri için değerli olabilir. Ayrıca, derin öğrenme ve optimizasyon yöntemlerinin karmaşık havacılık sektörü optimizasyon problemlerini çözme potansiyelini göstererek, mevcut literatüre yeni bir matematiksel model ve yaklaşım sunmakta ve ekip kurtarma alanındaki gelecekteki çalışmalara yönelik değerli görüşler sunmaktadır.

1. INTRODUCTION

Globalization has led to a rise in interest in the aviation industry and air transportation, with increased travel opportunities and economic development playing a significant role. Increasing interaction between developing Asian countries, especially China and India, where there are dense populations, and the United States, which is still the world's financial center, increases the need for transportation between the two points day by day. Even if the COVID-19 pandemic introduces significant changes in the aviation sector (Mitsokapas et al, 2021), according to the pre-pandemic trends in air transport, the International Air Transport Association (IATA) reveals that the number of passengers could increase to 8.2 billion in 2037 (IATA, 2018).

The increase in passenger numbers and compliance to regulations on passenger rights make transformation inevitable for airlines. One of the most known forces behind the airline transformation can be seen in a report published by the European Union which plans to improve mobility and design a transportation experience consisting of different transport modes. European Commission aims to have 90% percent of travelers can complete their door-to-door journey within Europe in 4 hours (Jimenez-Serrano and Kazda, 2017).

Airlines should rearrange and manage their all resources to be compliant with the aforementioned transformation. The most critical resources of commercial airlines are listed as crew and aircraft, which are the essential components of operational efficiency in combination with passengers (Kohl et al, 2007). The expected rise in the aviation industry after the COVID-19 pandemic also forces airlines to extend their fleet size and increase their operational capacities. Growth in the volume of operations also increases the total size of airline crew, especially flight crew which can easily be identified by increasing total crew costs (Deveci and Demirel, 2018).

Most airlines face crew costs as the second biggest cost item after fuel costs, and small gains in crew cost margins lead to profound positive advances in operational expenses (Barnhart, 2003). That situation is the primary motivation of airlines for searching robust and cost-effective schedules which can be constrained by the rules and regulations of governments and unions. Due to the multi-stakeholder and constrained nature, crew scheduling is complex and challenging to solve (Schaefer, 2005). On the other hand, in order to solve in an acceptable time frame, certain assumptions are taken into consideration in building crew schedules, such as each flight will be operated as planned (Novianingsih, 2015).

However, on the day of operation, unpredictable events such as bad weather conditions, aircraft failures and crew absence can cause deviations from the operational plan, and the revenue targets behind it (Castro, 2014). These events are called as "Disruption". As of today, the difficulties faced by the aviation sector are still increasing. Disruptions such as flight delays/cancellations and the costs caused by these disruptions are one of the main challenges faced by the industry (Castro, 2014). According to the report published by Eurocontrol, known as the European Air Navigation Safety Organization, delay statistics are getting worse in the long-term performance analysis even if there are minor and local improvements (Eurocontrol, 2019).

As disruptions such as delays are the primary and fundamental factor in passenger satisfaction and the financial situation of the airline, aviation companies are allocating valuable resources for analyzing disruptions and taking necessary actions (Khaksar and Sheikholeslami, 2019). As it can be seen from Figure 1.1, the main primary groups of disruptions such as delays are due to airlines' operations (Eurocontrol, 2019). Therefore, this fact drives aviation companies to look into their way of handling disruptions and understanding them.

It has already been mentioned that two of the most valuable resources of an airline are crew and airplane, which are the fundamental building blocks of the airline operation. Therefore, crew-related problems can be widely seen in disruptions (Novianingsih, 2015). On the other hand, even though traditional approaches can provide solutions to problems faced by airlines; they are rarely able to provide enough insights on realities hiding in the daily operation. Instead of these traditional solutions, it is highly advised to build customized models based on the feedback provided by people responsible for daily operation (Hewitt and Frejinger, 2020). In order to build that customized models, it is very vital to understand the problem itself in great detail.

Addressing the crew recovery optimization problem has been the subject of various conventional methods which have been utilizing heuristics, mixed-integer programming techniques etc., each carrying specific limitations and disadvantages. Traditional approaches, notably, often entail substantial computational time, posing practical limitations. They tend to provide limited adaptability to uncertainties and variations within the problem, which can be less satisfactory in the rapidly changing aviation industry.

However, artificial intelligence (AI) based approaches present a fresh perspective on solving the crew recovery optimization problem. Deep learning and AI techniques offer the capability to analyze vast datasets, recognize patterns, and make rapid decisions, potentially surmounting such intricate challenges. AI can swiftly generate optimized crew schedules, thereby enhancing operational efficiency. AI-based methods can also introduce greater flexibility and adaptability to quickly changing operational conditions, providing substantial advantages. Hence, the integration of AI in addressing the crew recovery optimization problem can empower airline companies to make faster and more effective decisions, significantly improving crew planning processes. This highlights AI's potential in boosting operational efficiency within the aviation industry.

Figure 1.1: Delay groups (Eurocontrol, 2019).

1.1 Artificial Intelligence Is Transforming World

AI is a scientific area consisting of many methods including machine learning covering deep learning and reinforcement learning, machine reasoning covering scheduling and optimization of informations and robotics covering techniques of cyber-physical integration systems (Eurocontrol, 2020).

AI has already been changing the way of doing business, the style of our lives as it gets momentum day by day. As it can be easily guessed that the data produced by human is massive and with the help of the data and increasing computational power, AI will be the key component of every sector for generating effective and efficient economical and social results. It can be seen from Figure 1.2 travel, transport and logistics sectors including aviation are heavily benefiting from AI technics. From efficiency processes to safety critical applications, AI is already leading to a broad range of value propositions to the aviation sector. Since AI minimize the human workforce need and improve the overall capabilities in real world scenarios, AI has tremendous potential for use in numerous areas of the sector (Eurocontrol, 2020).

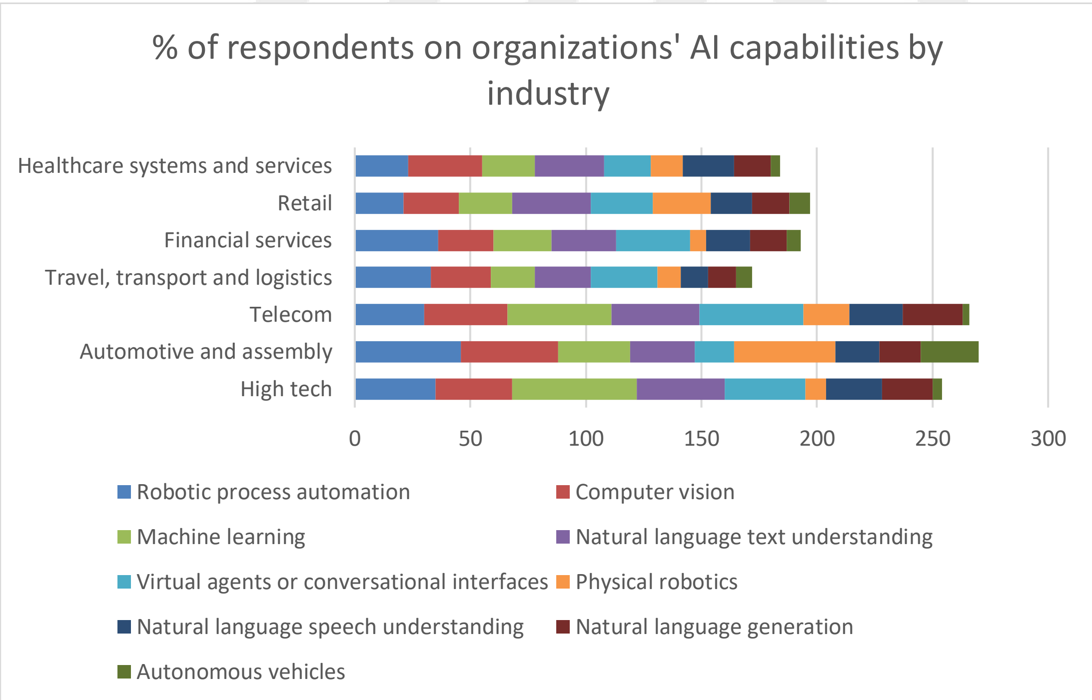

Figure 1.2: AI penetration levels (McKinsey & Company, 2019).

The capabilities of information technology (IT) systems on overcoming the real world problems normally requiring mental work-force of an operator is not only being improved but also changing the world. AI is penetrating into industries fast and widely by the broad usage of open source, cheap, cloud based solutions which also require less data compared to old models (National Security Commission on Artificial Intelligence, 2020).

Applications of AI methods in major optimization problems have been extensively researched including airline optimization problems and crew recovery problems (Wang, 2021; Ramachandran et al, 2022; Muthuram and Frank, 2021; Santos and Gillis, 2019; Lee et al, 2022; Vadlamani et al, 2022). These problems require a complex decision-making process that involves the allocation of resources, scheduling, and routing. In particular, the crew recovery problem has been challenging due to its dynamic nature and the need for timely and effective decision-making.

In this thesis, an optimization model that utilizes machine learning techniques has been developed to solve the crew recovery problem. The model is designed to optimize crew scheduling, reduce delays and costs, and enhance the overall performance of airline operations. Through this research, it is aimed to contribute to the existing body of knowledge on optimization methods and address the existing gap in the field, particularly in crew recovery problem-solving.

To the best of our knowledge, there is no similar work that has been done in this area, which makes our research unique and valuable. We are confident that the results of this research will provide a significant contribution to the field of airline optimization and crew recovery problem-solving.

1.2 Purpose of Thesis

Due to the geographical location of Turkiye, being in the middle of east-west traffic provides a significant advantage in terms of aviation industry. Also, reaching important countries of the world with short travel times (<4 hours) with narrow body aircrafts offers an important operational opportunity. On the other hand, with the openings of new large airports, some of which are planned as one of the largest airports in the world and one of the important hub points for the country's aviation, a significant improvement in the number of total passengers carried, the number of transfer passengers of all airline companies, especially the flag carrier.

This thesis aims to solve crew recovery problem in an intelligent way by examining the characteristics and understanding the inner dynamics of the flight crew disruptions in a large airline operation.

Within the scope of this study, data gathered from a large-scale airline company covering thousands of flights between 2018 and 2019 is analyzed, and common characteristics for flight crew disruption is proposed. By utilizing the data model as a graph, clustering techniques are applied in order to understand features and relationships such as categories hidden in the disruption data. The clustering results reveals the recovery actions performed earlier by the human operators.

Training a new deep learning model with crew disruptions for predicting possible recovery actions and finding the best possible crew composition by an optimizer which utilizes crew recovery actions predicted by the deep learning model is the next step performed.

After all, with the help of the proposed model, findings of deep learning driven optimizer are interpreted for general-purpose utilization in other studies, especially studies focusing on integration of combinatorial optimization and deep learning fields. Because, the necessity to work on a large-scale data set in order to conduct descriptive analysis is one of the obstacles in front of similar studies. Data gathering is the most essential part of the study. Still, airline companies are not very eager to share their operational disruption data in large amounts and without limitations which are the main obstacles in this study's analysis. Because of General Data Protection Regulation (GDPR) and similar regulations, airline companies share only a masked version of their operational disruption data which makes domain knowledge a necessity to interpret the data successfully. This study also contributes by introducing findings of large-scale aviation operation data set since data ownership and sharing is one of the biggest obstacles in the data-driven approaches and machine learning practices in the aviation sector (Mitsokapas et al, 2021).

Even there are many studies on crew recovery in the literature; there are not many published studies including large scale data analysis of flight crew disruptions. As already mentioned, due to the lack of the required large volume of data, no published models on flight crew disruptions are found in the field. As an early work in the data analysis field of airline crew disruptions, this research will provide new opportunities for the upcoming studies. It is thought that this study will provide satisfactory answers because flight crew disruption data of a large airline is used. The data is a large-scale set covering a total flight crew size of about 20,000, 1,500+ flights daily and 300+ aircraft for the pre COVID-19 times. It is also believed that this study will constitute a basis for the following industry practices.

1.3 Hypothesis

Considering the sectoral developments mentioned above, it can be seen that every improvement that will create cost advantage in the aviation sector with low profit margin will have a significant effect. If an effective algorithm / model can be produced with the research activities within the scope of this study, it is thought that it will be beneficial for aviation sector by transforming the study into a software product.

There are not enough studies on the subject of "Flight Crew Operation Recovery Optimization" known as "Crew Recovery Optimization", which are thought to provide the results needed in both domestic and foreign academic literature, especially for operations of a particular scale. Within the scope of this thesis, it is considered that finding a model that can provide satisfactory results in terms of both quality and speed will make a significant contribution to the academic literature and will be an important step towards the development of a product that can be used industrially.

Although satisfactory studies have been carried out for airlines of a small and medium size with the simplified studies conducted in the academic community and the applications in the industry, there is no satisfactory solution or industry practice for airlines with a large size and wide network.

Although certain algorithm controlling methods are used to reduce the time significantly needed to solve large scale problems, sophisticated methods such as column generation based mathematical programs can't be suitable for large-scale problems because the representation of the problem can easily grow and become unmanageable (Gamache et al, 1999). As AI is transforming the interaction of operators with machines as the machines making fast and not explicitly programmed decisions based on large volume of data (National Security Commission on Artificial Intelligence, 2020), we strongly believe that generally AI specifically deep learning methods will provide necessary mechanisms to accelerate sophisticated optimization methods while preserving optimality at an acceptable level as fast as possible.

Since there are not enough studies on integration of deep learning methods to accelerate optimization methods, we strongly believe that the finding of an algorithm that is proposed in the hypothesis and the scope of this thesis and can provide satisfactory results in terms of both quality and speed. It will both make an important contribution to the academic literature and will be an important step towards the development of a product that can be used industrially.

In addition to the model proposal, crew disruption data is also analyzed which is crucial for the effective management of airline operations. Crew disruptions, such as absenteeism, delays, and cancellations, can have a significant impact on the airline's overall performance, causing delays, cancellations, and reduced customer satisfaction. Therefore, it is essential to analyze the crew disruption data to identify the root causes, trends, and patterns that lead to these disruptions. By analyzing crew disruption data, airlines can gain valuable insights into the factors that contribute to disruptions and take proactive measures to mitigate their impact. This analysis can help identify areas that require improvement, such as crew scheduling and training, and provide a basis for developing strategies that enhance crew performance and reduce disruptions.

This thesis stands out from others in the field due to its unique contributions and perspectives. Firstly, it presents a comprehensive analysis of crew disruption data, providing a nuanced understanding of the subject matter. Secondly, it employs an innovative research methodology that combines both optimization and machine learning research methods to provide a more robust and comprehensive analysis as well as solution model on crew recovery problem.

Moreover, this thesis sheds light on a neglected aspect of the research topic by examining the experiences and perspectives of the real world airline problems that are often oversimplified in the existing literature. Additionally, it provides practical insights and recommendations that can be used by decision makers and practitioners, in the field to address the challenges identified in the research.

Finally, this study makes a significant contribution to the crew recovery management field by challenging the dominant theoretical frameworks tailored to small or medium-sized problems and oversimplified business conditions and offering a new theoretical perspective that has the potential to shape future research in the field. Overall, the unique aspects of the thesis offer a valuable and insightful contribution to the field and have the potential to influence literature and practice in the years to come.

In order to generate the value summarized above, this study is organized as follows: Section 2 deeply describes crew recovery problem. Section 3 explains the high level context of the the study conducted in this thesis. Section 4 analyzes disruption and its dynamics and proposes a new approximation method for spectral clustering algorithm to process large amounts of disruption data. In section 5, a deep learning model is introduced for predicting possible recovery actions with the help of automation techniques for machine learning. Afterwards, in section 6, optimization model for solving crew recovery problem is developed which utilizes the results of section 4 and 5. Results covered in this study are shown in Section 7. This section also interprets the results reported by the former phases. Lastly, Section 8 concludes the research and discusses further research opportunities.

2. CREW RECOVERY PROBLEM

Generally airline companies have three main resources they control and huge amount of money is invested in order to schedule and manage them (Ilagan and Sy, 2017). These fundamental resources of an airline are:

- flights
- aircrafts
- crew

Scheduling the resources in airline companies is one of the fundamental processes highly investigated by academic community as well as realized in many industrial practices because of high impact on profitability and competitive advantage (Zhang et al, 2015).

Since generally transportation industry, specifically aviation industry has small operational margins, intelligent way should be found in order to increase efficiency while preserving the quality of services (Huiskamp, 2016). The rise in the aviation industry forces airlines to extend their fleet size and increase their operational capacities. Operational capacity and fleet size increase leads to increased size of airline total crew especially flight crew which can easily be seen in the financial figures of total crew costs (Deveci and Demirel, 2018).

Crew costs are the second biggest cost item after fuel costs and even small improvements in these costs produce serious positive advances in operational costs (Barnhart, 2003). Second to the fuel, flight crew takes on a huge amount of airline expenses and it is a vital task to recover the crew (Zhu et al, 2014).

In order to manage these resources effectively, airline specialists need to identify a schedule with low cost combination of aircraft and crew resources that follows all the safety requirements and has almost no effect on passengers and eliminating the airline's operating complexities (Lettovsky et al, 2000).

As it can been seen from Figure 2.1, the multi objective goals of commercial airline business can be summarized as follows (Ilagan and Sy, 2017):

- Generating maximum revenue from customers
- Minimizing operational costs generated by aircraft and crew resources

In order to generate maximum revenue from passengers, revenue management processes are applied and ticket pricing problems are solved based on the parameters such as destination, cabin class etc. Although pricing figures are generally calculated for seasons covering six or more months, the prices are updated dynamically as the day of flight is being closer.

The overall and simplified version of sequential airline scheduling consists of timetable production, fleet assignment and then crew process including crew pairing and rostering (Ilagan and Sy, 2017).

Aircraft resource planning processes start from timetables based on airline slots and continue with fleet management processes. As the day of operations comes closer, firstly aircraft types with real or artificial tail number are assigned and lastly all flights are assigned with real tail numbers which is basically unique identifier of every aircraft.

Crew resource and overall crew resource management processes can be seen in detailed in Figure 2.2. Those processes are very complex to design and manage since crew contracts are very restricting, airlines are trying to manage daily load of crew and resting in an optimal way while keeping flight security and passenger satisfaction as maximum as possible or dictated by authorities (Ciriani et al, 2013).

2.1 Crew Management

As it can be seen from the Figure 2.2, crew management processes consist of crew scheduling and crew recovery. Crew scheduling is basically generating effective flight sets and assigning crew to that sets and crew recovery is finding the legitimate crew resource for a certain flight if any problem occurs related with flight crew.

Airlines generally prepare crew schedules on a biweekly or monthly basis and are not published to public compared to their flight timetables to the public due to personal data characteristics and privacy (Chen and Chou, 2016; Nissen and Haase, 2006).

2.2 Crew Scheduling

Due to the gradual growth of the airline industry, companies are turning to larger fleets, which, combined with the diversity factor, increases the size of the airline crew planning problem, which is one of the biggest planning problems in the industry (Deveci and Demirel, 2018). Flight networks are getting bigger and crew scheduling turns into complex problems which are difficult to solve (Yan and Tu, 2002). Since the salary costs of crews can exceed hundreds of millions of dollars for airline companies, preparation of airline crew plans is a complex and very important process for airlines (Gamache, 1999).

Literature on crew scheduling problem (CSP) is mainly concentrated on planning part of the crew management process (Abdelghany et al, 2004). Typical CSP is decomposed into two sequential sub problems (Guo, 2005; Nissen and Haase, 2006):

- Crew Pairing Problem (CPP) is the process of finding crew pairings which cover all flights and lead to minimum cost (Guo, 2005).
- Crew Assignment Problem (CAP) which is also known as Crew Rostering Problem (CRR) is assigning pairings to specific crew by taking all ground activities such as days-off, trainings and vacations into consideration (Guo, 2005).

This two step approach is mandatory because for most of the cases it is impossible to solve the scheduling problem in one step due to the computational complexity (Chang, 2012). As it can be seen from Figure 2.3, there is also another type of approach which focuses on integrating two problems and solve in one step.

Although, generally set covering models are used in modeling airline crew scheduling (Novianingsih, 2015), there are also different approaches such as crew scheduling problem are modeled as pure network problem. One example of that kind of model for solving real world problems is the one applied in China Airlines (Yan and Tu, 2002).

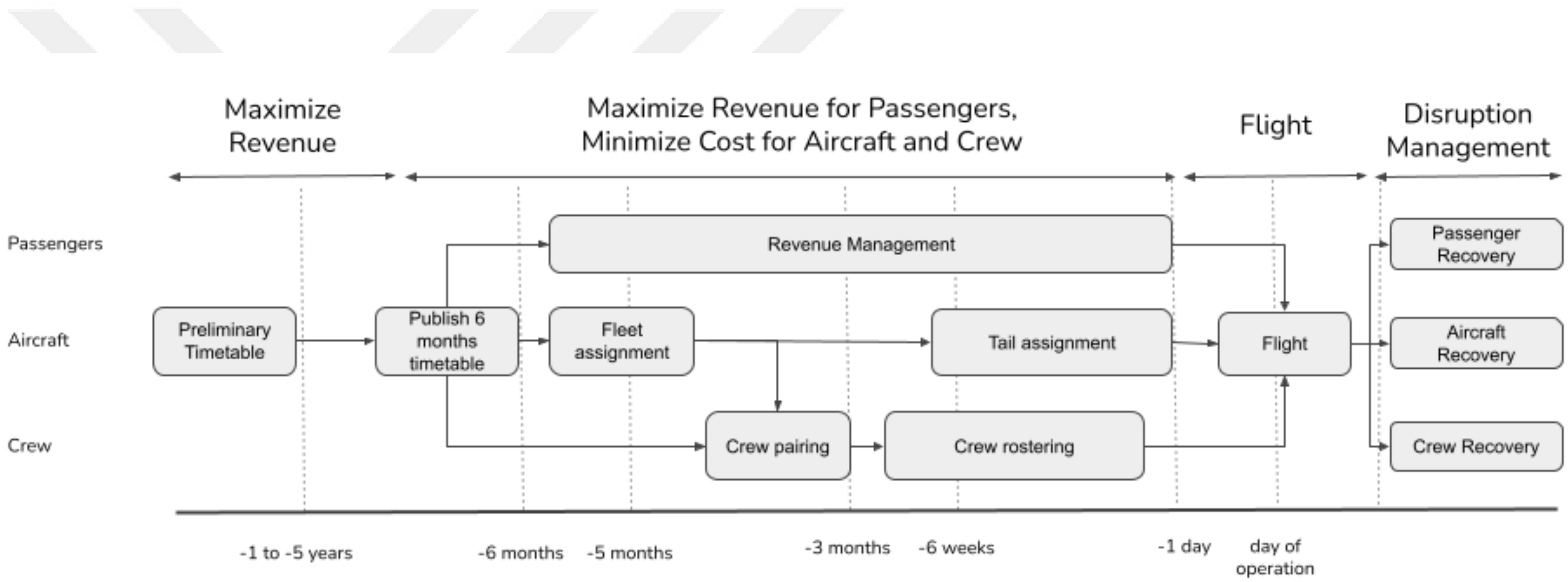

Figure 2.1: Overview of commercial airline operation (Ilagan and Sy, 2017).

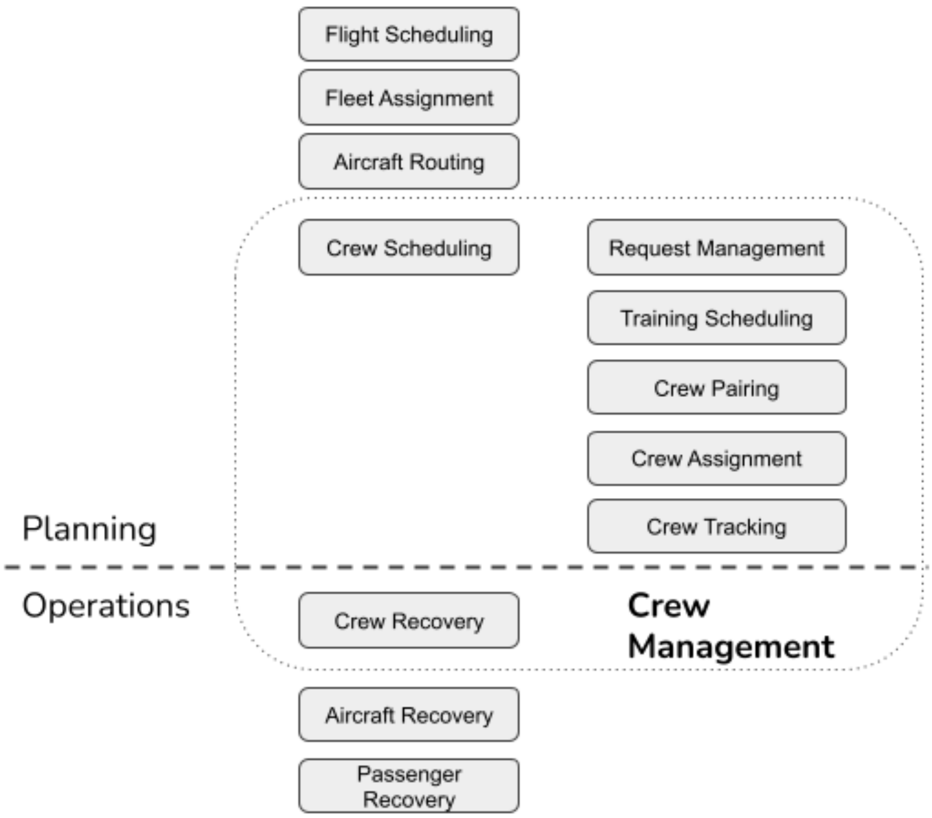

Figure 2.2: Overview of crew management (Guo, 2005).

In this network problem approach, starting, ending arcs and source and sink nodes is used which are implying that every pairing starts and ends at home base. Special kind of arcs are designed to represent different components of the plan such as:

- Duty arcs: representing flights operated by same crew.
- Deadhead arcs: denotes crew which is deadheading.
- Rest arcs: arc connecting two duties and indicating rest time duration them.
- Cyclic arc: structural arc connecting source and sink nodes.

The objective of the network model is to find a flow which leads to minimum cost for crew and the problem is solved with network simplex algorithm (Yan and Tu, 2002).

Finally, the purpose of crew scheduling is to cover all flights in a certain time period by keeping the monthly airline crew duties, which are planned during daily operations when necessary, with minimum cost, and keeping the crew as motivated as possible (Stojković et al, 1998).

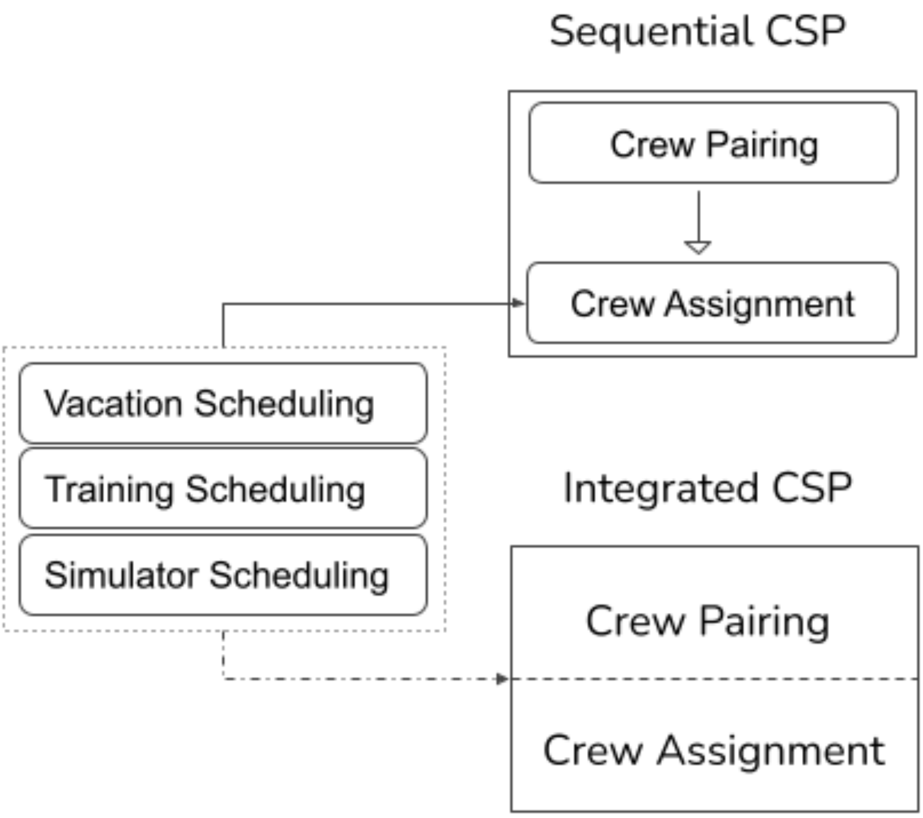

Figure 2.3: Sequential and integrated crew scheduling problem (Guo, 2005).

2.2.1 Crew pairing

As it was already indicated, the crew scheduling process is generally a two-stage process. The primary phase of an airline crew scheduling is crew pairing (Novianingsih, 2015). Crew pairing is the process of finding set of pairings which has mininmum cost and covers all flights in a specific period by taking all regulations into account (Nissen and Haase, 2006). This first step is to create clusters by matching flight segments. Each matching known as pairing is flight segments spread over consecutive days, leaving a home base and returning to the same base in accordance with business rules (Gamache, 1999).

In order to cover the all the flights in the one calendar month timetable with minimum crew cost, sequences flights called as pairings are generated (Medard and Sawhney, 2007). By producing large volume of pairings, selecting the set of pairings costing the most minimal amount and including all flights in the planning period, the pairing problem is solved (Novianingsih, 2015). Pairings are also necessary for calculating crew costs which are also called as budget pairings (Nissen and Haase, 2006).

Pairings start and end at specific airports called as crew base. A pairing duration can cover one day or more and includes duties. Typically they can be as long as five or seven days (Lettovsky et al, 2000). Working periods known as duties are separated from each other by resting periods (Novianingsih, 2015).

A duty is set of flights starting with briefing and ending with debriefing and it is limited by flight duty time limitations (Novianingsih, 2015). Duty generally lasts for one day (Nissen and Haase, 2006) and generally starts one hour before the first flight which is briefing period and ends 15 minutes after the last flight landed which is known as debriefing period (Abdelghany et al, 2004).

In pairing problem, flights are represented with its number, departure airport code, arrival airport code, planned departure and arrival date and times where flight number including carrier code, departure airport and date of departure makes every flight unique (Abdelghany et al, 2004). Every non stop flight is called as flight leg (Nissen and Haase, 2006).

A typical pairing with flight and duty components can be seen in Figure 2.4.

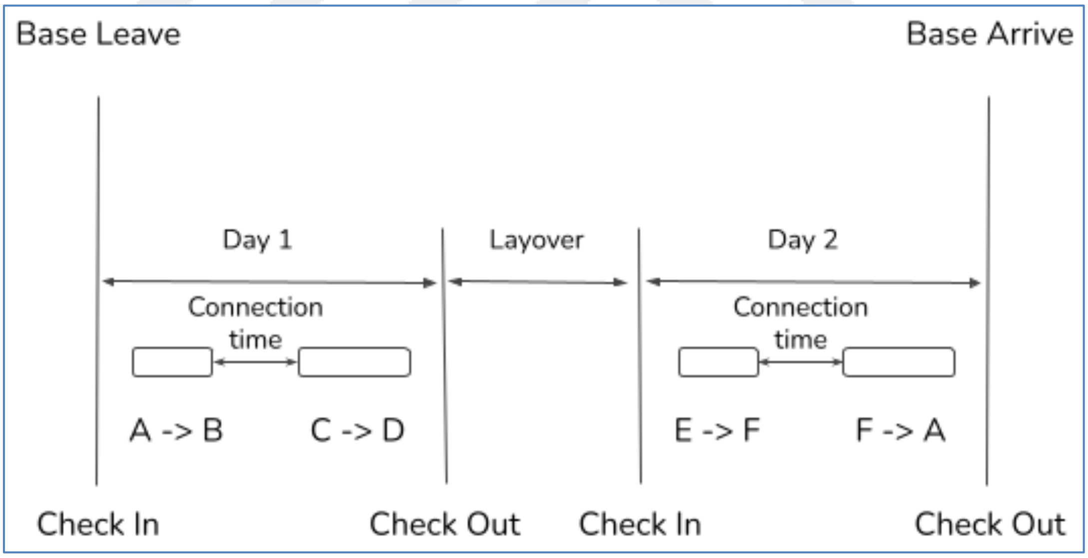

Figure 2.4: A typical pairing.

2.2.2 Crew rostering

The second stage is the crew assignment or crew rostering problem and includes assigning crew and creating personalized monthly plans by taking into account qualifications, rest periods and the pre-planned activities such as training, medical control while minimizing cost and maximizing life quality of flight crew (Gamache et al, 1999; Medard and Sawhney, 2007). Main concern of crew assignment or crew rostering is to provide enough staff to pairings (Nissen and Haase, 2006).

A flight is ready when all of its crew need is met with proper crew resource. Flight crew need is represented as crew need vector which is consisting of numbers of how many crew is needed for a certain position. The need for different responsibility areas such as cabin and cockpit are separated with "//" symbol (Medard and Sawhney, 2007). This vector is known as crew compliment which is combining crew positions in a flight with the number of crew needed for every position (Guo, 2005). While all flight needs should be fulfilled in the solution, every crew should be assigned to flight or ground duties or deadheaded to base (Ramle, 2014).

Three types of crew rostering methods based on the crew requests and preferences are the ones shown below (Guo, 2005; Nissen and Haase, 2006)

- personalized rostering
- bidlines
- preferential bidding

Some airlines, mostly European ones are using a salary system which is based on fixed salaries. On the otherhand, some others mostly from North America, use another system known as pay and credit which means crew earn money as their flight statistics. (Nissen and Haase, 2006). These payment methods are based on the utilized version of crew rostering and changes the cost calculations.

2.3 Crew Recovery

Airlines have there resources which are aircraft, crew and passenger. The planned schedules of airline resources are usually interrupted by the disturbances or unexpected events such as weather conditions, air traffic problems, technical problems and show-off of crew all of which affect the revenue targets (Guo, 2005; Zhang et al, 2015; Castro et al, 2014; Liang et al, 2018). These events are called as "Disruption".

Due to the effectiveness of airline resources, very little buffer is added to the plan in order to absorb the unexpected deviations from original plan. Since the slack is so little, there is the possiblity of penetration of huge impact to the airlines' flight network even if the deviation is small (Nissen and Haase, 2006). Based on figures provided by Amadeus, a global technology provider for the travel industry, deviations from plans cost a total amount of $60 billion per year, which is almost 8% of overall total commercial airline revenue (Jimenez-Serrano and Kazda, 2017). According to the report published by Eurocontrol, delay statistics are getting worse in the long term performance analysis even if there are small and local improvements (Eurocontrol, 2019). In 2016, arrival delays due to disruptions totaled over 59 million minutes, resulting in a cost of US$ 62.55 per minute (Liang et al, 2018). Furthermore, costs incurred due to disruptions in the aviation industry increased from US$ 26.6 billion to US$ 28.2 billion, with an increase of US$ 1.6 billion between 2017 and 2018 (Li et al, 2021). Department of Transportation in United States already put an regulation into action in which $27,500 fine for every passenger if a delay longer than three hours occurs (Petersen et al, 2012).

Even the most appropriate or closest operational plan obtained by using specific optimization models or different techniques is disrupted by the deficiencies caused by internal or external factors. The process of getting a new plan that is revised and minimizes the harmful effects of disruptions while at the same time taking the constraints and goals of the developing environment into account is called recovery management. Recovering passenger problem is passenger recovery, solving aircraft or fleet problem is aircraft recovery and overcoming crew problem is called as crew recovery (de Castro, 2007). The process of recovering the crew operation from the irregularities by minimizing the costs and following the necessary rules is called Crew Recovery (Castro, 2014). The problem solved in this context is called the Crew Recovery Problem (CRP).

CRP is a kind of optimization problem which aims selecting the best recovery actions for overcoming crew disruptions and should be completed under dynamic environment (Novianingsih, 2015; Sato et al, 2009). Since aviation industry has a dynamic and sophisticated characteristics, any deviation from scheduled operations leads to huge negative impacts such as inevitable costs so problems of schedules should be solved by making minimum change as possible (Guo, 2005). The main goal of the CRP is finding minimum cost combination of assignments recovering problematic schedule while flight time limitations and flown flights are preserved (Lettovsky et al, 2000). Crew recovery is also trying to eliminate side effects of the changed rosters which may have impact on flights leading to unwanted effects on passengers (Chen and Chou, 2016).

Since the 90's, airline operation recovery has become one of the major research topics and the problem of crew recovery is the main obstacle in flight operations and investigated by both industrial and academic communities (Hoeben, 2017; Zhu et al, 2014). Operation day delays often result in costly recovery actions, forcing the airline's operational control. With regard to these challenges, robust planning makes it essential to create plans that are less affected by disruptions or provide opportunities for low-cost recovery actions. In this context, different recovery techniques, especially re-optimization and rule-based recovery approaches have also been the subject of literature studies (Kohl et al, 2007).

Although crew recovery has been classified as post-planning processes in practice, there is a clear examples of integrating it into the resource planning sequence of airlines (Ilagan and Sy, 2017). For that purpose, there are some studies aiming predicting disruptions before occurrence and taking necessary precautions (Abdelghany et al, 2004).

Whether the approach is post or pre planning, the most important thing is to find the optimal combination of recovery strategies which leads to minimum disruption cost and overall number of crew used in a timely manner (Abdelghany et al, 2004; Zhang et al, 2017).

Specific studies are also carried out for these unforeseen situations in the literature. For example, deviations caused by the temporary closure of airports cause significant loss of profits and reduced service levels for airlines. In this context, studies of developing a framework to help plans, including crew assignment plans, to address deviations caused by the temporary closure of airports are among the studies observed in the literature (Yan and Lin, 1997).

CRP is a very large problem and classified as NP-Hard problem (Zhang et al, 2015; Novianingsih, 2015). It is also known as crew rescheduling, operational crew scheduling, roster maintenance and tracking (Nissen and Haase, 2006; Medard and Sawhney, 2007).

CRP is a kind of problem different than CSP because of the dynamic nature and obligation to affect minimum number of crew (Lettovsky et al, 2000). The comparison of CSP and CRP is shown in Table 2.1 (Guo, 2005).

Table 2.1: Crew recovery and crew scheduling problem comparison.

Recovery Item	CSP	CRP
Activities	Scheduled flight legs Ground activities Off-day	Scheduled, updated and new flight legs, Ground activities Off-day
Crew	Operating crew	Operating crew Standby / Reserve crew
Time	Weeks before the operation	Day of operations
Duration	Months and beyond	Hours Days
Cost	Assignment cost Transit cost Layover cost	Used standby / reserve crew cost New assignment cost New transit cost New layover cost

2.3.1 Disruption

A situation that forces executors to deviate from the planned state during the operation is generally called a disrupted situation or simply disruption (Clausen et al, 2010). The deviation's root cause is called disruption source (Xu et al, 2016). Accidents, incidents, capacity constraints, diseases, financial problems, geological events, IT systems, security issues, strikes, and weather conditions are the primary and most known possible disruption sources (Jimenez-Serrano and Kazda, 2017). Disruptions are also dependent on the airline resources such as aircraft, crew and passenger. Maintenance, flight delay, a problem with airport infrastructure are some of the sources of disruption, but it can be easily seen that disruptions become connected with events far beyond the originated source (Kohl et al, 2017). This distributed nature of disruption leads to a chaining effect (Ionescu et al, 2016).

The situations mentioned above which lead to disruptions can be summarized as follows (Castro, 2014):

- Problematic Aircraft: An aircraft that could not complete its original plan
- Problematic Flight: A flight that cannot take-off and/or arrive at the scheduled time due to an unexpected event
- Problematic Airport: The airport where the problematic flight takes place
- Problematic Passenger: Passenger who lost one or more flight connections due to a problematic flight
- Problematic Crew: The crew that could not complete the original plan

In this study, we are mainly focused on the intersection subset of the problematic crew and problematic flights or simply flight disruptions which can be defined as disruption originated by a change in flight timetables such as flight cancellation or a 2+ hours delay in the operational window (Jimenez-Serrano and Kazda, 2017). Possible disruptive events such as aircraft type change, cancellation or retiming of a flight is a multi-objective process where passengers' inconvenience should be minimized and airline resources such as crew and aircraft should be utilized effectively (Kohl et al, 2017). Flight disruptions that lead to problematic crew are investigated and analyzed as the focus of this study.

In order to solve a flight crew disruption, schedules should be updated while minimizing costs and obeying crew rules and regulations (Castro, 2014). This process is known as Crew Tracking, which can be defined as staffing flights by changing crew pairs according to disruptions (Kohl et al, 2017). Airlines use two approaches to make their schedules robust for making the crew tracking process easy and fast enough. The first approach is the proactive one and is called robust scheduling, which is building plans with enough buffers. The second one is the reactive type which is simply recovering operations by rebuilding schedules (Vos et al, 2015). By applying robust scheduling, airline operations are safe for small, ordinary and common disruptions. However, in cases like severe weather conditions or the closure of the airport is far beyond the safe area provided by robust schedules. It is also clear that even minor disruptions may lead to chaining effects and make schedules infeasible (Hoeben, 2017).

2.3.2 Disruption management

Disruptions lead to plans that can no longer be implemented during the operation. To recover from an irregular operation, control centers of airlines apply techniques such as aircraft and crew exchanges, redirecting passengers, using reserve or standby resources, canceling or delaying flights until all operational resources are available (Ionescu et al, 2011). Since disruptions on operation day challenge the airline's operational control capacity leading to expensive recovery actions, different recovery models, especially optimization and rule-based recovery approaches, have been the subject of studies in the literature (Ionescu et al, 2011). Approaches are changed when the preferred one does not provide regular performance and speed (Kohl et al, 2017).

Whichever strategy is selected, negative effects of deviations should be minimized by making decisions to get back on track as planned by monitoring the uncertainty (Jimenez-Serrano and Kazda, 2017). Making necessary decisions for replanning and rebuilding plans close to operation day is called as Disruption Management (Kohl et al, 2017; Hoeben, 2017).

Generally, disruption management processes are operated and irregular operations managed by special organizational units of airlines called Operations Control Centers (OCC) which are responsible for returning to pre-disruption plans at minimal cost (Nissen and Haase, 2006; Liang et al, 2018). OCC departments are composed of operators from different administrative units such as aircraft dispatchers, maintenance operators, and crew trackers. These operators start the process by identifying and classifying the problem based on criteria such as the number of passengers impacted, possible time to solve, propagation of the problem. After this first phase, possible solution options are found out and sorted. In order to go back on track with schedules, disruptions in flight timetable is recovered by increasing or decreasing flight speeds, adjusting ground times, changing the aircraft assigned to flight or using spare aircraft. For crew resource, OCC has options to increase working hours, using reserve crew or swapping (Nissen and Haase, 2006). They also updates the routes of passengers, rarely cancel or delay flights (Ionescu and Kliewer, 2011). Finally, the solution which is the most applicable one is applied. This last phase is also known as the recovery phase (Jimenez-Serrano and Kazda, 2017) and the process of finding new plans is called recovery management (Ionescu et al, 2011).

Airline recovery management has become one of the key research topics and decision support tools are developed to assist operators during the irregular operations, and to offer better solutions to operation controllers. Many studies investigate some of the recovery problems, and some studies even explore the integration of recovery problems in a single model. However, due to the long calculation times, such models cannot be used in real-time. Therefore, recent studies require the selection of aircraft or crew, with a time window where the problem must be solved. In this way, it is possible to use models in real-time by shortening the calculation times (Hoeben, 2017). Therefore, operational recovery is divided into three problems which can be solved separately. The main motivation behind this decomposition is based on the fact that a solution without decomposition in practically impossible with the current computational capabilities (Guo, 2005).

2.3.3 Rules and regulations

In order to solve a flight crew disruption, schedules should be updated not only minimizing costs but also obeying crew rules and regulations (Castro et al, 2014). For airline companies, it is a must to take all regulations including governmental, union and company policies into consideration because not complying with regulations of government or union leads to penalties including monetary fines (Abdelghany et al, 2004).

Crew recovery problem is based on the limitations generated by rules and regulations and the most important and restricting one is called as legalities covering qualifications and time obligations (Zhu et al, 2014). Crew composition constraints are realities of real world situations and cover constraints indicating how the crew should be composed like composition of experienced and inexperienced crew (Ramle, 2014) because, the qualifications of the people in the crew assigned to the prepared pairings and the number of people to be assigned to the aircraft must comply with certain rules (Gamache et al, 1999).

Although there are many other constraints mostly airline and regulatory organizations such as governmental institutions and organizations, the most common rules applied to all airlines are listed below (Guo et al, 2005; Yan and Tu, 2002; Abdelghany et al, 2004; Zhang et al, 2015).

- Duty and flight time limits
    - Daily
    - Weekly
    - Monthly
- Rest time limitations
- Consecutive working days limit
- Return to home base rule
- Number of landings
- Maximum allowed delay for a flight.
- Maximum number of deadheads

- Training requirements
- Maximum duration for being away from base

2.3.4 Recovery actions

In order to recover the operations, certain actions should be taken on schedule, crew, fleet and passengers which has a combinatorial characteristics (Petersen et al, 2012). A recovery action is an action decided by an airline operation control center to overcome the problems occurred realization of the schedule because of disruptions (Novianingsih, 2015).

Since, the amount saved by recovery actions has a direct impact on profitability of airlines (Zhang et al, 2017), airlines need computing tools to combat disruptions and develop real-time and efficient solutions (Liang et al, 2018). Even if latest researches are trying to provide airlines with automated and optimized tools for recovering, operators called as crew trackers still recover crew disruptions manually (Abdelghany et al, 2004; Nissen and Haase, 2006). Today's airline companies utilize these operators to manually recover crew schedules by trying to get back to the original schedule as fast as possible even though the recovery process is very complex due to the regulations, rules and different factors (Hoeben, 2017). Scarce resources and limited transportation opportunities are also other dimensions increasing the complexity of disruption management (Vos et al, 2015). Because of the complex nature, analyzing characteristics of disruptions is one of the fundamental building blocks to understand sources of disruptions and find out structural problems which is vital for reducing the probability of occurrence and the effect of disruption (Ionescu et al, 2016). It is also important to formulate more realistic models of problems and make required simplifications or assumptions to find a solution (Vos et al, 2015).

Analysis of operational realization is a very beneficial method for the punctuality of an airline. In this context, data gathered during operations contains valuable information for all kinds of variations and bottlenecks (Goverde, 2005).

That situation is the main motivation of airlines for searching robust and cost-effective schedules which can be constrained by the rules and regulations of governments and unions. Due to the multi-stakeholder and constrained nature, crew scheduling is a complex and hard problem to solve (Schaefer et al, 2005). Finally, in order to have robust and cost-effective operation, having a detailed view on disruption mechanics can be easily classified as a must (Ionescu et al, 2016).

Decision support tools are being developed to assist operational controllers during irregular operations. The aim at this point is to provide better and better solutions to operation controllers (Hoeben, 2017). With a decision support system adaptation, an approach has been adopted where a number of optimization problems will be solved in a way that improves departure times and flights in chronological order. In each problem, the goal is to cover as many flights as possible while minimizing the total system cost resulting from resource assignments and flight delays (Abdelghany et al, 2004).

For example, the results of the CrewSolver decision support system project, which was made to produce the most suitable or most applicable crew recovery solutions globally for Continental Airlines, can be found in the literature. This system has been used in many high-profile incidents, including the December 2000 and March 2001 Norer blizzard, the June 2001 Houston floods and the 9/11 terrorist attacks. It is estimated that Continental Airlines saved approximately US$ 40 million in 2001 with the CrewSolver system only on major disruptions (Yu et al, 2003).

In another system design study, the Multi-Agent System (MAS) paradigm was used at TAP Airlines. This study demonstrates the design and partial implementation of a distributed MAS that represents the current roles in the operations control center. Taking advantage of each operational base having specific resources (both crew and aircraft) and incorporating information on those costs (such as crew salary information and hotel costs), an approach has been adopted to emerge their solution proposals (de Castro, 2007).

Studies on integrated solutions are also frequently encountered in the literature. Integrated recovery plans are generally solved sequentially starting from fleet recovery and continue with crew recovery. In that case, aircraft recovery generates new schedules where crew illegalities due to the aircraft composition are also tried to be minimized. This new schedule is used by crew recovery in order to reassign flights to crew based on the new fleet schedule (Ramle, 2014).

In this context, in a study conducted at Chalmers University of Technology, a main problem that determines whether an aircraft is canceled, delayed or not flown with another aircraft type (for example, with two aircraft representing the aircraft and crew assignment problem), respectively, was examined. With the integrated approach developed, a plan saved with different compensatory scenarios was produced. The results are compared from a sequential approach with the results of the scenario where an appropriate flight schedule was set first and then the crew was assigned to that plan (Ramle, 2014).

Even if there are studies covering complete integrated recovery problem, most of them are just problem formulations due to the not being able to represent real world needs (Petersen et al, 2012). In real world problems, sequential recovery starts with aircraft recovery due to the cost of aircraft resource, then crew recovery and afterwards passenger recovery processes are completed (Zhang et al, 2015). Other interpretation of sequential approach is based on operational priorities where problems arises in the day of operation or close to it should be solved first and others can be solved later. (Abdelghany et al, 2004).

Initial planning process is divided into two steps which are crew pairing and crew rostering due to the its size, crew recovery is usually solved at once because of smaller size compared to the initial one (Nissen and Haase, 2006). However, there are also some studies divide crew recovery into recovery of crew pairing which utilizes local and global pairing recoveries and rebuilding of crew assignments (Ciriani et al, 2013).

The recovery problem types needed to be solved during recovery process are (Abdelghany et al, 2004):

- Connection problem: connection time is less than minimum connection time
- Rest problem: rest time is less than minimum resting time
- Duty problem: Duty time limit is violated.
- Unassignment: No show of crew

The complexity of the problems mentioned above increases with the challenges which are need to be overcome while recovering are listed below (Clausen et al, 2001):

- Data availability
- Short time
- Feasibility
- Organizing people
- Buy in : financial benefit should be large, cost should be small.

One of the most important component of the recovery is recovery window which is a way to narrow solution space, decrease the problem volume and complexity of crew recovery problem. A sliding time window technique is utilized for rearrangement of the disrupted schedules. One of the applications is based on pre-defined time window to narrow the search space of crews can be used for swapping when a connection problem arises (Lettovsky et al, 2000; Maher, 2016). Some studies define recovery window as one day (Petersen et al, 2012).

The overall process for recovery actions can be seen in Figure 2.5.

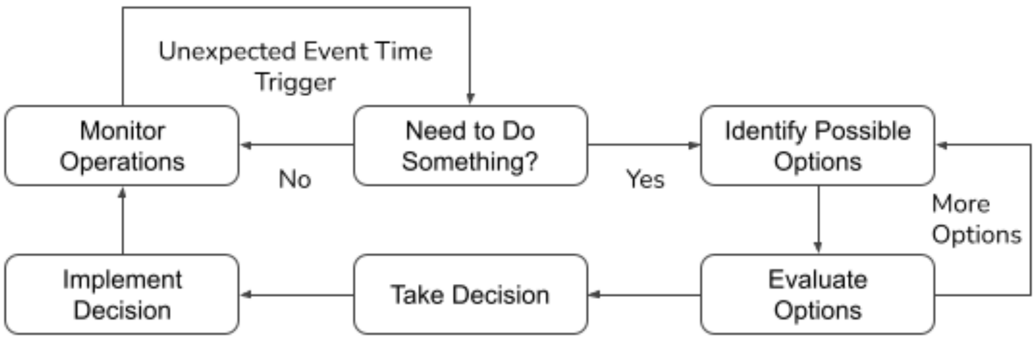

Figure 2.5: Disruption management (de Castro, 2007).

The options to solve the aforementioned recovery problems are the ones listed below (Abdelghany et al, 2004; Zhang et al, 2015):

- Delay flight as much as acceptable
- Cancel flight
- Deadheading
- Use standby or reserve crew
- Swapping crew duties
- Use the crew already disrupted but eligible for other disrupted flights
- Deadheading

Since the related costs are high, utilizing correct recovery strategy has great importance (Maher, 2016).

Delay is generally an option for solving connectivity issues (Novianingsih, 2015).

Deadheading is a costly recovery technique, but it can efficiently fix crew misconnections (Zhu et al, 2014). A deadhead crew may be send to take control of a flight at another airport or to the crew base (Novianingsih, 2015) and does not operate in the flight but fly as a passenger which causes increase in the operational costs because a passenger seat is allocated to crew without a fare while crew allowance is paid (Zhu et al, 2014; Lettovsky et al, 2000).

The process of exchanging flight duties between crew members is called crew swapping (Novianingsih, 2015). Crew swapping is an important topic for disruption management since it provides a contingency option for schedule disruptions (Ilagan and Sy, 2017). An example of swapping crew can be seen in Figure 2.6.

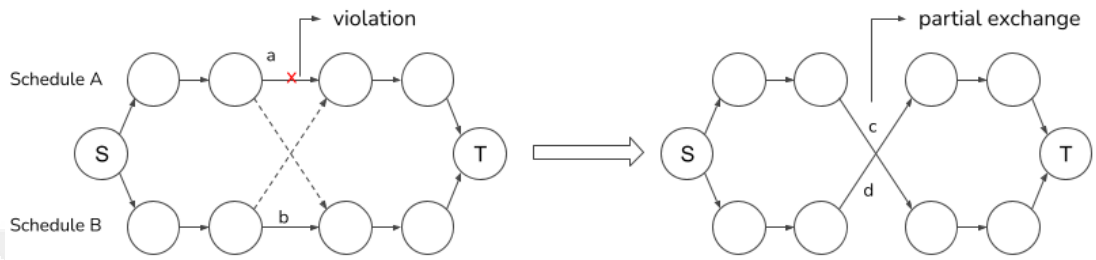

Figure 2.6: Crew swapping (Sato et al, 2009).

If any kind of resource is not used consecutively or is not possible to be exchanged by two flights based on the schedule, these flights are called as resource independent flight. That situation leads to the case that if two flights are on the same pairing, they are not resource independent. A crew is suitable for swapping in a specific flight if the crew (Abdelghany et al, 2004):

- can operate in the same position and fleet type with the one already assigned to that flight.
- should be available at the station before the flight departure time plus minimum connection time.
- after assignment of crew to that specific crew, there should not be any illegality due to the violation of rules and regulations.

Reserve crew which is a costly resource is utilized when other actions do not lead to solution because they get minimum guaranteed hours payment even if no duty is scheduled (Lettovsky et al, 2000).

The overall crew recovery cost covers cost of crew allocation to flight and the cost of deviation from the schedule (Zhu et al, 2014). The detailed structure of crew recovery action cost is generally consisting of the cost items listed below: (Zhang et al, 2017; Le and Sun, 2011; Abdelghany et al, 2004; Zhang et al, 2015)

- Delay cost: cost of delaying flight

- Cancellation cost: cost of cancelling flight
- Assignment cost: cost of assigning a crew to a flight
- Swap cost: cost of changing crew duties
- Reserve cost: cost of using one reserve crew instead of already planned crew
- Deadhead cost: cost of one deadheading crew
- Idle time cost: cost of crew waiting without assignments or for upcoming assignment.
- Destination inconsistency cost: cost occurs if destination changes from original plan.

2.3.5 Optimization approaches of recovery process

Airline industry has benefited from improvements and advanced techniques not only in computational methods but also optimization. However, the advances generally are in the scheduling field (Petersen et al, 2012). Airline crew assignment algorithms, which are widely used in practice, are prepared by assuming that there will be no problems. Deterministic algorithms act with the assumption that all assignments will work as planned (Khaksar and Sheikholeslami, 2019).

Although it is believed that decision support systems (DSS) supported by optimization techniques provide airline operation specialists with cost effective and competitive recovery actions, there is still room for researches in order to be able to cope with large-scale recovery needs (Lettovsky et al, 2000).

Since there are not many studies in the operation phase and due to the need of generating a real time or near real time solution, the complexity of airline operations is aggravated (Petersen et al, 2012). It is also seen that operations research techniques have a very important place in the literature researches for disruption management that are handled during the crew recovery problem (Clausen, 2001). A list of most well-known studies on crew recovery problem covered in this thesis can be seen in Table 2.2.

It is frequently emphasized in the literature that it is necessary to reduce the complexity and size of the problem before applying the recovery approaches. Different approaches such as decomposition and approximation techniques are utilized for recovery problems in order to increase computational efficiency (Maher, 2016).

Table 2.2: Literature review on crew recovery problem.

Study	Objective	Method	Key Findings
Abdelghany et al. (2004)	Develop a decision support system which is able to automate crew recovery process in large-scale airlines.	Mixed Integer Programming	The system is able to proactively solve the problem before their occurrence. The scenario covers 18 crew disruptions for one day.
Chang (2012)	Eliminate rotation generation steps in crew recovery problem to solve the problem with flight legs.	Heuristics, Genetic Algorithm	Model is efficient for solving the crew recovery problem by scheduling the legs directly. Number of disrupted crew is 70 and recovery duration is 18 days in the scenario.
Chen and Chou (2016)	Formulate crew recovery problem as a combinatorial optimization problem with multiple objectives and use genetic algorithm for the solution.	Linear Programming, Heuristics, Genetic Algorithm	The model is successfully able to explore Pareto solutions for all the scenarios. 14 disrupted flights are included in them.
Guo (2005)	Develop a decision support system for real world crew recovery problem based on exact optimization technics and heuristics.	Linear Programming, Heuristics	A dedicated approach and decision support system for crew recovery problem are introduced.
Guo et al. (2005)	Solve crew recovery problem with genetic algorithm.	Heuristics, Genetic Algorithm	An efficient heuristic algorithm is developed. Instances cover 47 to 188 disrupted crew.
Hoeben (2017)	Develop a dynamic optimization model whose objective is the minimization of recovery cost for crew recovery problem.	Linear Programming	A dynamic recovery model is introduced. Scenarios cover 11 to 27 disrupted pairings.
Le and Sun (2011)	Solve crew recovery problem for small and medium sized airlines.	Linear Programming	Model works only for small and medium sized airlines. Instances cover only 1 to 10 crew.
Lettovsky et al. (2000)	Develop a recovery model which restores disrupted crew rosters in almost real-time.	Linear Programming, Branch and Bound	Model shows that medium sized disruptions can be solved in a reasonable time frame. Instances cover 21, 11, 35 disrupted flights.

Table 2.2 (continued): Literature review on crew recovery problem.

Study	Objective	Method	Key Findings
Maher (2016)	Solve integrated airline recovery Problem using column and row generation.	Linear Programming, Column and Row Generation	Model is able to generate near optimal solution in the desired time frame For integrated airline recovery problem. Instances cover 262 to 441 flights with 79 to 182 crew groups respectively.
Nissen et al. (2006)	Develop a duty based network model for crew scheduling.	Network Flow Model, Branch and Price	A fast model is developed which takes labor regulations into consideration. Instances cover single type of disruptions for 1 to 5 flights and airport closure for 60 and 120 minutes.
Novianingsih et al. (2015)	Propose an efficient heuristic model with crew swappings.	Heuristics	An efficient method is modeled for the instance covering 214 flights, 48 pairings and one day horizon.
Petersen et al. (2012)	Present an optimization approach for fully integrated airline crew recovery problem.	Linear Programming, Column Generation	First study to show computational results on fully integrated airline recovery problem with one day horizon.
Potthoff et al. (2010)	Solve crew rescheduling with column generation and lagrangian heuristics.	Linear Programming, Column Generation	Model generates good quality solutions in an acceptable time frame. Instances cover 15 to 59 disrupted duties.
Sato et al. (2009)	Propose a 0-1 integer programming formulation based on a network flow model with heuristic flow modification for crew rescheduling problem.	Integer Programming, Network Flow Model, Heuristics	Model is able to produce efficient solutions in a reasonable time. Instance covers 185 vehicles and disruption duration is 2 hours.
Veelenturf et al. (2016)	Study and develop a real-time crew rescheduling model.	Mixed Integer Programming	The robust model is able to generate good quality solutions in a reasonable time frame. Scenarios cover 15 to 67 disrupted crew.

Table 2.2 (continued): Literature review on crew recovery problem.

Study	Objective	Method	Key Findings
Wei and Song (1997)	Propose a multi-commodity integer network flow model for crew recovery problem.	Network Flow Model, Heuristics	Model is able to generate flexible solutions in terms of partial solutions or multiple solutions. Instances cover 6 airports, 51 flights and 18 pairings.
Zhang et al. (2015)	Provide a solution for integrated aircraft and crew recovery problem.	Network Flow Model, Heuristics	Model consistently produces high quality solutions in a very short time such as under 2 minutes for all instances covering 10 to 20 flights.
Zhang et al. (2017)	Minimize the usage of airline crew while delay costs are considered as chance constraints.	Heuristics, Tabu Search Algorithm	A stable fast algorithm is designed and benchmarked against instances covering 5 aircrafts and 12 flights.
Zhu et al. (2014)	Propose constraint programming model for crew recovery problem.	Constraint Programming	Model generates efficient and feasible solutions for crew recovery problems covering one day.

The main focus of the literature is the restructuring of the planning process, taking into account the operational facts. While crew planning problems at the planning level are big and time-consuming problems and generally in the set covering and set partitioning problems class, the (re) planning time at the operational level is much smaller than the main planning phase (Medard and Sawhney, 2007). Preprocessing techniques should be applied to extract a subset of plans for replanning techniques, with a particular focus on crew recovery. Set partitioning or covering models have linear objectives suitable for relaxation methods such as Lagrangian relaxation. Relaxed solution provides a lower bound for optimality which is used by other optimization method such as branch and bound (Sato et al, 2009). For this purpose, there are studies in which several branching strategies are presented that enable the creation of integer solutions quickly. In these studies, which aim to save the current program by changing the existing program as little as possible by taking advantage of the fact that the planned program is optimal, the results from the sector are also shared from time to time (Lettovsky et al, 2000).

Most studies investigate some part of the recovery problem, even some studies investigate the integration of recovery problems in a single model. However, such models cannot be used in real-time transactions due to their long computation time.

For this reason, recent work requires the selection of a plane or crew with a time window in which the problem must be solved. In this way, by shortening the calculation times, it enables the models to be used in real-time operations, even with many assumptions. These studies lead us to the need for a real-time solution known as the dynamic crew recovery problem in the literature (Hoeben, 2017).

In another study, an integrated recovery problem that integrates crew and aircraft recovery phases was developed and solved using column and row generation method. It has been stated that the application of column and sequential production to the integrated recovery problem shows the improvement in solution runtimes and quality compared to the standard column generation approach and improves solution runtimes by reducing the problem size and thus enabling faster implementation of each linear programming (LP) solution (Maher, 2016).

Problems arise in transportation, manufacturing and telecommunications can also be classified as network design problems (Berger et al, 2000). Most of real world scenarios is represented as multi-commodity flow models (Agarwal, 2002). A multi commodity network flow model is presented where crew is modeled as goods (Le and Sun, 2011). The model is based on the methodology where each trip is modeled as node and a directed link is used for connecting nodes where possible. In that model schedules are modeled as network flow of resources and can be seen in Figure 2.7 (Sato et al, 2009). In network processing problems, flow is between multiple supply nodes and demand nodes where every flow has a predefined level of processing need. Flow consists of many connections from different supply nodes to different demand nodes. In this type of model, three types of constraints are used which are edge capacity, vertex capacity and network processing constraints (Charikar et al, 2018).

Multi-commodity network design problem (MCND) has applications in a wide span of areas due to the its flexibility such as manufacturing, logistics and transportation. As an extension of that kind of models is adding capacity constraint to classic MCND problem. In MCND problems, commodities are used to model communication demands of the node pairs and the objective is to minimize the total cost of network while meeting demand and capacity constraints. Another version of MCND problem is Network loading problem where multiple facilities with specific cost and capacity is added to problem and supply arcs (Frangioni and Gendron, 2009).

Constraint programming techniques are also applied to solve crew recovery problem where the total recovery cost was taken as the main function, and a sequential algorithm is designed to search the solution space by accepting the time and spatial requirements as constraints (Zhu et al, 2014). In another study that can be handled in this context, a mathematical model is proposed to solve the problem by considering the matching rules and time constraints. The results showed that the model can be used in a crew recovery problem for small or medium-sized airlines (Le and Sun, 2011).

Within the scope of multi-objective optimization studies, crew recovery problems were first formulated as combinatorial optimization problems with multiple goals and limitations, and a variant of the genetic algorithm method was used to search for Pareto solutions. In particular, the restrictive relaxation mechanism has been proposed to explore Pareto solutions, replacing previously allocated tasks with flight tasks in case of resource scarcity. The experimental results for the test sample showed the advantages of the proposed approach compared to previous studies in the literature (Chen and Chou, 2016).

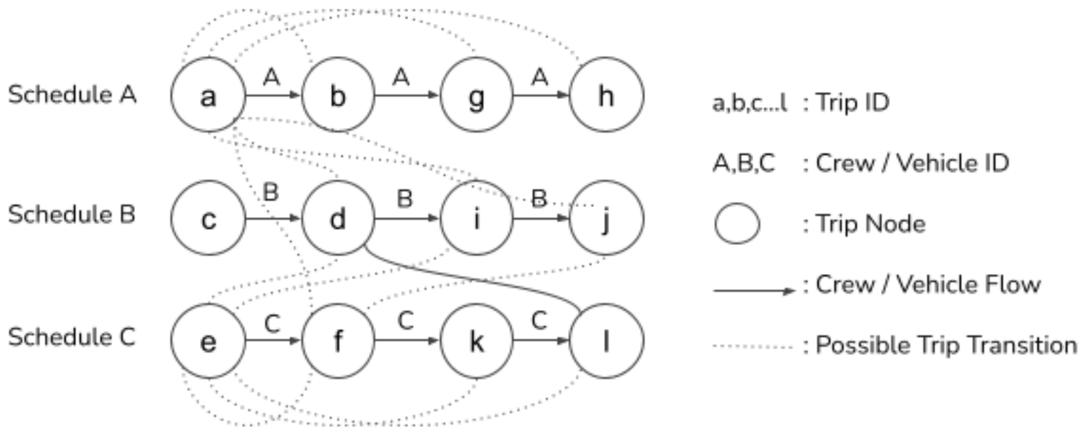

Figure 2.7: Multi-commodity flow of crew schedules (Sato et al, 2009).

In a study examined within the scope of robust optimization, the problem was addressed as a two-stage optimization problem. In the first stage, at the beginning of the disruption, planning was made based on the optimistic scenario (that is, assuming the shortest possible disruption time), taking into account the possibility of another scenario occurring. After determining the number of reorganized crew tasks for recovery, it was ensured that the solution at the first stage was transformed into a suitable program for the scenario. Differently, this approach has been tested on examples of Dutch Railways (NS), which is the main passenger train operator in the

Netherlands rather than an airline (Veelenturf et al, 2016). In another study conducted within the scope of Dutch Railways, an algorithm is presented to reschedule the crew when there is an outage. The algorithm is based on column building techniques with Lagrangian heuristics. Experiments with real-life examples have shown that the method is capable of producing good solutions within a few minutes of computational time (Potthoff et al, 2010). In another study carried out for the rescheduling problem of the railway crew is modeled and solved with the retiming technique. Experiments with real-life disruption data have shown that it is possible to find better solutions with retiming compared to the classical approach (Veelenturf et al, 2012).

In another study conducted in the aviation sector, a new duty period-based formulation was presented for the airline crew recovery procedure tailored to the needs of European airlines. The problem is modeled as resource constraint shortest path problem and solved with dynamic programming techniques. The results showed that the method can provide a solution in a short time after a disruption occurs (Nissen and Haase, 2006).

There are quite a number of heuristic method studies in the literature. Since set partitioning (SPP) or covering (SCP) models can not provide mechanism for identifying differences between new and original schedules (Sato et al, 2009), binary integer programming (IP) techniques supported with heuristics such as local search techniques are utilized on crew recovery problem modeled as network flow in order to easily reflect changes of original and new schedules which is not very possible with set covering or partitioning models (Sato et al, 2009). As a real life application of crew recovery, Continental Airlines model the recovery problem as set covering and supported by roster generating heuristics is applied successfully (Guo et al, 2005).

Heuristics are not only used as a complementary technique such as reducing the number of columns in column generation process (Aydiner, 2019) but also used as standalone solution approach due to the combinatorial nature of the problem where mathematical modeling is not utilized and original pairings are recovered with heuristics methods such as deep-first search (Petersen et al, 2012). In one of these studies, a two-step heuristic algorithm has been proposed for the recovery problem. In the first phase, the aircraft rescue and flight remodeling model was built with the crew. This model is based on the traditional multi-network model for the timetable recovery problem. The purpose of this model also includes minimizing original crew connection disruptions. In the second stage, a flight rescheduling model was created with the principle of integrated crew recovery. Unless improvement is achieved in this model, two stages are repeated. Experimental results have shown that the method can provide better recovery solutions compared to benchmarking algorithms (Zhang et al, 2015).

In another study, a random heuristic interview is proposed to solve real-life crew disruptions. The proposed algorithm aims to balance the workload distribution between different crew members as well as providing quality solutions that meet real life constraints. Experimental tests have shown that the algorithm is capable of generating quality solutions applicable to real-life crew problems in just a few seconds (de Armas et al, 2017).

A genetic algorithm (GA) based approach is also proposed, where disrupted flights are reassigned in an evolutionary process, a special local improvement procedure is applied due to the slow convergence speed achieved by traditional GA. Computational results for disruption scenarios of a mid-size European airline company over various real-life situations are also presented within the scope of the study (Guo et al, 2005).

Some of the networks are solved with neighborhood search where a subset of links are selected as a subnetwork and improved at every step (Agarwal, 2002). Tabu search is also utilized in order to solve the network problems with capacity constraint (Berger et al, 2000). By presenting an integer network flow model and a search algorithm, it has been checked whether it is flexible enough to meet the needs of efficient and practical business to solve realistic sized problems (Wei et al, 1997). Especially within for the cockpit crew, the problem was mathematically formulated as an integer nonlinear multi-network flow model with time windows and additional constraints, and sub-problems with linear time costs and time-limited shortest path problems were solved with a special dynamic programming algorithm (Stojković and Soumis, 2001).

A full-scale optimization is not possible because of the complexity of the schedule volume and the real-time characteristics of the problem. The complexity and the scale of the problem must be narrowed before an optimization strategy is utilized (Lettovsky et al, 2000). Because of that, the recovery problem is usually solved sequentially in a few steps due to the size of the problem space and the need for quick resolution (Liang et al, 2018). Therefore, models should be combined for critical operational problems that arise during the days of operation. For this reason, by studying the feasibility of solving the problem in one step, simple tree search, more advanced column creation and solution techniques based on shortest path algorithms are presented (Medard and Sawhney, 2007).

In general, in order to solve relevant and practical problems, it is seen that it will be very beneficial to hybridize complex AI-based methodologies such as multi-agent systems and reinforcement learning with more classical programming methods to achieve useful goals. With the aforementioned techniques, a solution that can be generalized for many decentralized, distributed and dynamic problems can be presented (Castro, 2014).

It should be stated that including a kind of learning mechanism that learns the previous plans of the crew will increase the quality of the solution. It is also emphasized that expanding the learning mechanism for learning each crew member profile and applying this knowledge to generate future programs would be more efficient for the management of this expensive resource and increase the satisfaction level of each crew member (de Castro, 2007).

Some models are developed with an objective of utilizing original schedules as much as possible (Guo, 2005; Chang, 2012). The use of past flights in machine learning processes focuses on features constraints, and although some restrictions are officially documented, the built-in application is based on many others. In one of the studies, which creates a system where crew assignment constraints are generated from past crew plans with the help of a domain expert, ComCon algorithm is developed to learn the constraints that determine the limits of certain aspects of crew plans was applied. While the algorithm reveals complex multivariate constraints based on a set of user-supplied templates, showing the general nature of the key constraints, the results of an evaluation with crew plans from two commercial airlines showed that the algorithm is able to learn most of the minimum rest constraints (Suraweera et al, 2013).

Finally, key metrics for measuring how well a recovery model performs changes based on the airline scale and type but common metrics in the literature are the one listed below (Zhang et al, 2015):

- total recovery cost
- computational time
- total flight delay

- number of cancelled flights
- number of crew swaps
- total destination inconsistency
- number of deadhead
- number of reserve crew used

The crew recovery problem, detailed in the previous sections, is in the NP-Hard problem class as it requires the best possible solution to be found in a limited time frame. Although satisfactory studies have been made for airlines of certain sizes as it can be seen in Table 2.2, there is no satisfactory solution or industry practice for airlines with a certain size and wide network presented in the following sections within the studies published in the academic community or the applications found in the industry.

It is thought that the study will contribute to the academic literature for large-scale operations as well as its industrial dimension. In particular, it is considered that the thesis will be an inclusive and complementary thesis with its academic and industrial dimensions, by providing the industry dimension that is missing in such studies. The scope of the study in terms of the phases of crew management discussed in the former sections can be seen in Figure 2.8.

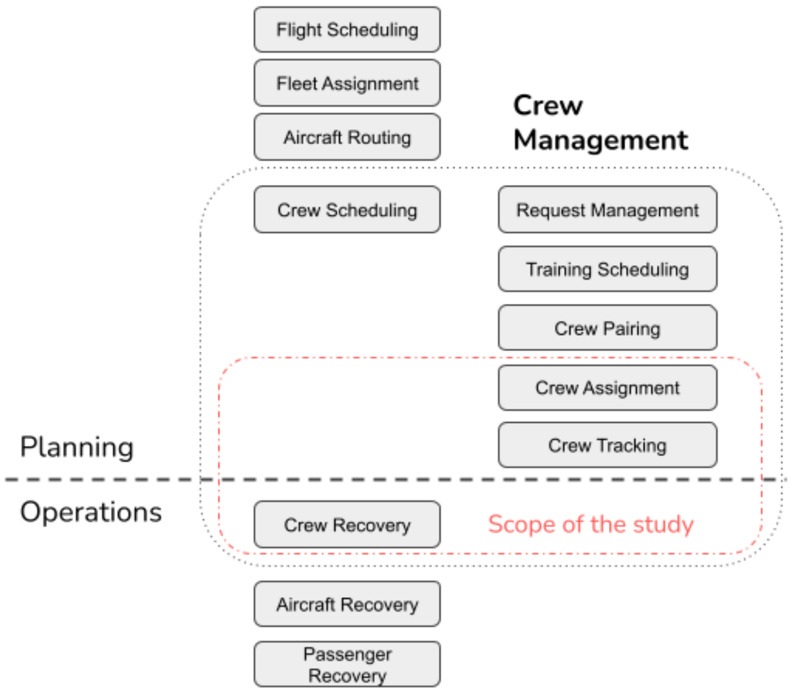

Figure 2.8: Scope of the study.

Crew recovery is a crucial component of the overall airline operations, and its efficient management is closely interconnected with various other processes, including passenger recovery, flight recovery, crew pairing, crew rostering, and crew tracking. An effective crew recovery system ensures that crew disruptions are managed promptly and efficiently, which, in turn, contributes to overall operational efficiency, passenger satisfaction, and safe flight operations. The integration of these processes is essential for a well-coordinated and resilient airline operation in the face of unforeseen events and disruptions.

In our thesis, we aim to enhance this integration and resilience through the implementation of advanced AI-based techniques, addressing crew recovery optimization, and thereby, elevating the overall efficiency and adaptability of airline operations.

3. PROPOSED SOLUTION METHODOLOGY

In order to make the aforementioned contribution in the former section, new methods are introduced to transfer the experiences of people who are in the aviation sector, who work in the crew operation and who make manual interventions during the operation into the digital environment.

Alternative mathematical models and decomposition methods are used to accelerate the aforementioned methods. Thus, it will be ensured that the solution space is searched with an intelligent strategy so that results quickly give the best or near-best solutions.

The new mathematical model proposed in this study is supported by artificial intelligence model built by deep learning techniques. In particular, it is believed that modeling past recovery actions with deep learning techniques will make the model faster and more effective.

This study, basically, consists of there stages which can be seen in Figure 3.1 and explained in the sub sections below.

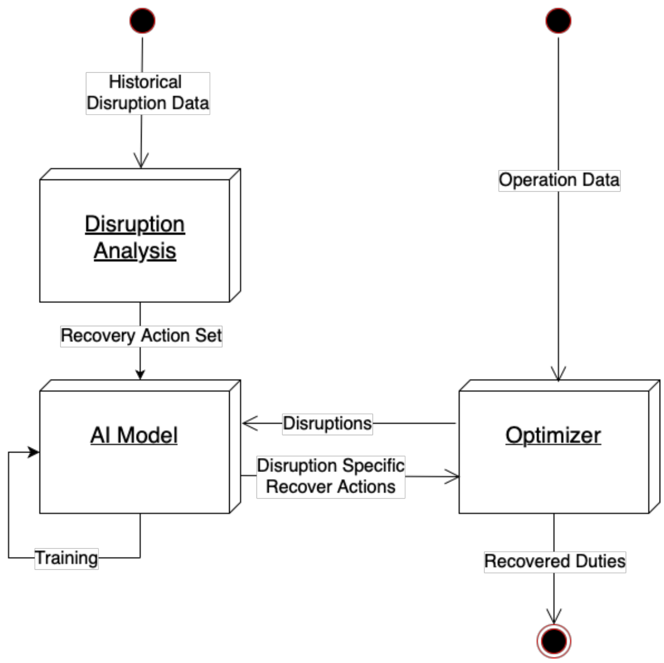

Figure 3.1: Phases in the study.

The Figure 3.1 illustrates the key components of a crew recovery optimization system. It begins with historical disruption data analysis, where the system identifies recovery actions based on past disruptions. These recovery actions serve as labels and are utilized to train a supervised AI model, enabling it to learn and predict suitable responses to future disruptions. After the model has been trained, it is deployed within the system. In parallel, real-time disruption data and operational data are integrated into a column generation-based optimization process. The optimizer leverages the AI model by sending disruption information and receiving recovery action predictions. Subsequently, the optimizer engages in optimization, generating efficient recovered crew duty assignments. This integrated approach combines historical data analysis with AI-driven predictions and optimization techniques to enhance crew recovery processes, resulting in more effective and adaptive solutions.

3.1 Disruption Analysis: Learning From Disruptions

The data analysis is the fundamental effort of studies from different disciplines such as engineering to management and aims generating knowledge in order to make decisions (Famili et al, 1997). In this phase, the disruption data used in this study has been analyzed and recovery actions which will be the base for the rest of the study is found out.

In data analysis, data preprocessing techniques or combinations of them are used to build models (Mishra et al, 2020). Even if data processing process is challenging and requires time to be completed, it is a must in order to improve the quality of data (Frye et al, 2021). There are many different steps in data processing such as feature selection, removing noisy data, detecting outliers and replacing missing values (Slišković et al, 2009). We have divided the process into 4 steps which are

- Data Cleaning
  - Outlier Analysis
  - Missing Data Analysis
- Standardization
- Data Reduction
- Data Transformation

First of all, the data which is obtained from airline operational systems, is cleaned. At this stage, the outlier observations are found out and removed from the data set. Also, missing values are analyzed and taken care of with appropriate methods such as filling with mean.

In the next stage, the features that make up the data are analyzed, and the minimum feature set that could represent data is determined. This step is called as feature selection (Subasi, 2020). Then, the new data set is modeled with these features. The reduced features are then subjected to the standardization process, and the data is standardized by applying various transformations.

Data reduction process is also carried out in order to reduce the large scale volume by reducing both the number of observations and the variables. The data is reduced to a scale that can be processed more easily. Finally, data transformations are performed and categorical variables are encoded with appropriate encoding methods.

Completion of data preprocessing is followed by clustering of data in order to make the processing of large data set's labeling with recovery actions possible. Natural language programming (NLP) techniques are applied to convert text data into numerical vectorized format.

The vectors generated from text data is clustered with a new spectral clustering approximation algorithm. Lastly, the clusters generated by the aforementioned algorithm is investigated by human experts and final recovery actions matching with clusters are decided.

3.2 AI Model: Predicting Recovery Actions

Machine learning studies generally consists of two phases known as training and inference. The training phase is the stage where a machine learning model learns how to the do its designed job or gain capability from an existing large scale data set whereas the inference which is also known as prediction is applying the gained capability on a new data set which has not been used by the model (Copeland, 2016).

During inference stage, machine learning algorithms deal with different types of tasks which can be seen below (Sarker, 2021):

- Classification: Predicting a label for given data

- Regression: Predicting a continuous value for a variable which depends of other predictor variables
- Clustering: Findings groupings in a large data set
- Dimensionality Reduction: Reducing dimensions of high dimensional data
- Rule Mining: Finding if-then style relationships in a data set
- Reinforcement Learning: Learning by experiments in an interactive environment

In this study, a classification problem is solved with machine learning techniques. As it is indicated above, classification problem predicts a label for given data. In mathematical terms, classification is a function converting inputs to output also known as label, category or target with different mapping types seen below (Sarker, 2021):

- Binary Classification
- Multi Class Classification
- Multi Label Classification

The classification problem type in the scope of this study is in the multi class classification category. Multi class classification is the classification problem type with three or more categories (Brownlee, 2020).

In this phase, a multi class classification model that infers the recovery actions is developed. A model that predicts what kind of recovery action should be taken specific to each flight and disruption. The model is firstly trained with the training data set selected from the data processed in the analysis phase. After expected success is reached with deep learning model, it is used for inference.

3.3 Optimizer: Optimization of Recovery Actions

When disruption events occur in the airline industry, it's not enough to merely identify recovery actions; these actions need to be seamlessly integrated into the existing crew schedules. This integration is a complex task because it must respect various operational constraints, such as legal rest requirements, crew availability, and flight schedule continuity. Recovery actions serve as guidelines for what needs to be done, but how these actions are executed while maintaining overall schedule integrity is a challenging puzzle.

The heart of the matter lies in solving an optimization problem. It involves reshuffling and adapting crew schedules to align with the recovery actions, all while adhering to the rules and regulations governing the aviation industry. This optimization problem takes into account various parameters, constraints, and preferences. It seeks to find the most efficient and effective way to adjust the schedules so that disrupted services can be resumed, passengers can be accommodated, and safety and regulatory requirements can be met.

In the final stages of the study, a mathematical model is developed to address this optimization problem. This model acts as a critical bridge between the recovery actions derived earlier and the crew duty schedules. It effectively translates the strategic recovery actions into tactical adjustments to the schedules, ensuring that crew members are assigned tasks that fit the new operational reality while still respecting their working hours, rest periods, and contractual agreements.

The aim of this mathematical model is to deliver optimal solutions, taking all these factors into consideration. It's a systematic approach to optimizing the crew schedules that would be nearly impossible to achieve manually or with traditional methods. The result is a practical and feasible set of schedules that enable the airline to efficiently recover from disruptions while maintaining regulatory compliance and passenger service quality.

In summary, the process of updating crew duty schedules in the context of crew recovery optimization is a sophisticated undertaking that requires the careful coordination of recovery actions with a mathematical model to ensure operational efficiency, regulatory compliance, and passenger satisfaction. This is the pivotal stage that bridges the gap between high-level strategies and on-the-ground crew scheduling, ultimately enhancing the resilience and effectiveness of airline operations in the face of unexpected disruptions.

4. LEARNING FROM DISRUPTIONS[1]

During the operation, the planned programs are often revised due to disruptions caused by severe weather conditions, technical problems and diseases of the crew. The increasing focus on cost reduction in major airlines has focused attention on developing new and cost-effective methods to address airway disruptions (Kohl et al, 2007).

As disruptions have direct impact on customer statisfaction and financial situation of the airline, aviation companies are allocating valuable resources for analyzing disruptions and taking necessary actions (Abdelghany et al, 2004; Khaksar and Sheikholeslami, 2019). As an example, delays can be investigated. If a delay occurs in a flight, it will decrease the scheduled connection time between delayed flight and the next one. In the case where actual connection time is less than minimum connection time, it will not be possible to complete the flight duty with the the scheduled crew (Novianingsih, 2015).

Since scarce resources and limited transportation opportunities are also other dimensions increasing the complexity of disruptions (Vos et al, 2015), analyzing characteristics of disruptions is one of the fundamental building blocks to understand sources of disruptions and find out structural problems which is vital for reducing the probability of occurrence and the effect of disruption (Ionescu et al, 2016). It is also important to formulate more realistic models of problems and make required simplifications or assumptions to find a solution (Vos et al, 2015).

Analysis of operational realization is a very beneficial method for the punctuality of an airline. In this context, data gathered during operations contains valuable information for all kinds of variations and bottlenecks (Goverde, 2005).

[1] This chapter is based on the following publication: Herekoğlu, A., Kabak, Ö. (2023) Spectral Clustering Approximation For Large Scale Crew Disruption Data Of An Airline Company For Intelligent Crew Recovery. Journal of Soft Computing and Decision Analytics, 1(1), 139-160. DOI: https://doi.org/10.31181/jscda11202315

Even traditional analysis approaches can provide solutions to problems faced by airlines, they are rarely able to provide enough insights on realities hiding in the daily operation. Instead of these traditional solutions, it is highly advised to built customized models based on the feedback provided by people responsible for daily operation (Hewitt and Frejinger, 2020). In order to have robust and cost-effective operation, having a detailed view on disruption mechanics can be easily classified as a must (Ionescu et al, 2016).

4.1 Text Processing and Graph Clustering

During the flight operations especially recovery operations, extensive amount of text data has been produced and saved into responsible systems. These data sometimes come from sensors and other types of machines and sometimes produced by human operators. Among available information, one of the most important ones is text data provided by crew trackers, which include details of the disruption and clues for recovery actions. This valuable piece of information is essential in order to be used for future disruptions. Utilizing machine learning techniques by converting text data into vector space is critical for extracting data patterns (Cheng et al, 2020).

We conduct graph analysis on text data about flight crew disruption pro- vided by crew trackers. Graph analysis is one of the key and efficient tools for investigating generic features, characteristics and connectivity of complex networks such as air traffic, which are very dynamic by nature (Dunn and Wilkinson, 2016). Since dis- turbances in transportation networks have huge effects due to the connected and dependent nature of these networks, finding relationships between different network attributes is crucial for understanding the data (Sohouenou, 2020). Since graph is an adequate representation of complex data structures, applying clustering techniques on graphs reveals information hidden in them (Vathy-Fogarassy and Abonyi, 2013).

In order to understand patterns and relationships among chunks of texts, data is processed based on the text processing techniques. Since we study- ing large amount of texts, we investigate needed information about the topic in the text processing subsection. We model the text data as graph and applied graph clustering techniques are summarized in the graph clustering subsection. In order to understand the relationships, especially neighborhood between chunks, similarity measures are discussed. Since similarity measures are very important for building the correct adjacency and affinity matrices, right similarity measure should be selected.

4.1.1 Text processing

Text processing techniques such as summarization, text mining, text classification or text clustering are active and wide research areas especially with the development in Natural Language Processing applications. This area aims to convert the large set of documents into manageable chunks without sacrificing essential features and information in the original documents by modeling the documents set as graphs (Alami et al, 2021). It also includes methods for extracting information from texts that are not easily found (Zhang et al, 2019).

The common text preprocessing steps are (Alami et al, 2021; Janani and Vijayarani, 2019):

- Punctuation and non - letter removal
- Stop words removal
- Tokenization
- Stemming

4.1.2 Graph clustering

A graph is defined as pair of sets consisting of vertices and edges (Schaeffer, 2007). Graphs are denoted as

$$G = (V,E) \tag{4.1}$$

where V is the set of vertices and E is the set of edges.

$$n=V$$

is called the order of graph and it is simply the number of vertices. As v and t are the endpoints of edges, if v,t pair is unordered then it is undirected graph (digraph) otherwise directed graph (Schaeffer, 2007).

The path is sequences of vertices with an edge connecting vertices consecutively. If every pair of vertices has at least one path connecting each other, this graph is known as complete graph (Nascimento and de Carvalho, 2011). On the other hand, if there exists at least one path connecting every vertex combination, this graph is called connected graph (Luisa, 1995).

$$E = m$$

is known as the size of graph (Schaeffer, 2007). In a weighted graph, there exists a function that assigns a weight on each edge. The weights of edges are used to identify similarities inside a graph. Although different types of functions are used for deciding on the similarity of vertices in a graph such as sigmoid function, the definition of similarity and selection of the similarity measure is highly dependent on the domain of research (Schaeffer, 2007; Nascimento and de Carvalho, 2011).

In this study, a weighted undirected completed graph will be investigated.

Clustering is a set of exploratory data analysis techniques that have been utilized in many research fields such as social sciences, psychology or biology. The primary motivation behind the clustering techniques is to find similar behavior in empirical data by labeling data points as similar if they are in the same group and dissimilar otherwise (von Luxburg, 2007). Formally, identifying the underly- ing structures in the heterogeneous data by dividing data based on similarity measures into formerly unknown groups is called clustering (Schaeffer, 2007). The groups found by running clustering techniques are known as clusters. The key idea behind the clustering algorithms is having maximum similarity inside clusters and minimum similarity between clusters (Beauchemin, 2015).

Clustering techniques can be divided into two groups as follows (Nascimento and de Carvalho, 2011):

- hierarchical: creates tree structures by utilizing aggregation or division.
- partitioning: creates non-hierarchical structure by dividing and moving data between clusters

Since the exact description of what transforms a collection of items into cluster is not clear (Steinbach et al, 2004), it is not a common expectation to have clusters with strong similar concepts (Steinbach et al, 2004).

Graph clustering is one of the famous research topics of today as the usage of graphs is increasing day by day as can be seen in social networks, supply chains and electronic commerce. It can be defined as finding similar nodes in a graph (Almeida et al, 2011). Finding vertex partitions in a graph is graph clustering which is a pattern recognition problem (Tautenhain and Nascimento, 2020). Graph clustering is based on the idea of having many within-cluster edges and fewer between clusters (Schaeffer, 2007).

Although graph clustering is a highly sophisticated research area, there is no universally accepted definition of a good cluster in the literature (Schaeffer, 2007).

Optimal solutions to graph clustering problems are NP-hard and require high-level of computational power. As the graphs' size increases, manuel anal- ysis of graphs becomes impossible (Almeida et al, 2011). Spectral clustering methods are being utilized due to the efficient relaxation capacity in order to cluster the graphs without manuel intervention, (Beauchemin, 2015).

There is an increasing interest in spectral clustering methods(Nascimento and de Carvalho, 2011). These methods have shown high performance while partitioning graphs and are easy to implement by utilizing linear algebra software packages (von Luxburg, 2007), (Beauchemin, 2015). On the other hand, their application to large-scale problems has not been common yet (Nascimento and de Carvalho, 2011).

The success of any clustering algorithm is the correct definition and calcu- lation of neighborhood in the feature items (France et al, 2012). The documents to be clustered are compared with each other according to the similarity measures. The most common distances used for similarity measures for text processing are cosine and euclidean distances (Janani and Vijayarani, 2019).

Cosine Distance: One of the most common distance measures when text documents are modeled as vectors. Cosine distance between vectors calculated based on the angle between document vectors. The cosine distance of vectors v1 and v2 is calculated below (Janani and Vijayarani, 2019; Steinbach et al, 2004).

$$\mathrm{cosine}(v_1, v_2) = v_1 * v_2 \, / \, |v_1||v_2| \qquad (4.2)$$

where * is the vector dot product, and is the vector length (Assent, 2012).

Euclidean Distance: General distance measure based on euclidean space for all kinds of data analytics problems with multi-dimensional data. The Euclidean distance is calculated as the formula given below. The Euclidean distance of vectors d1 and d2 is:

$$\mathrm{euclidean}(d_1, d_2) = |d_1 - d_2|^2 \qquad (4.3)$$

Some studies suggest fractional distances for calculating similarities (Aggarwal et al, 2001). To sum up, in high dimensional space, the selection of distance calculation method is not clear and is mostly based on heuristic approach (Aggarwal et al, 2001).

4.1.3 Spectral clustering

Spectral clustering is becoming very popular and applied to different problems because of its easy and effective mechanism, making it a common clustering technique (Afzalan and Jazizadeh, 2019; Tremblay and Loukas, 2020). It is a clustering algorithm that utilizes spectral features of graph laplacian by finding connected components, leading to clusters of data points where similar data points are grouped based on the eigenvectors of the similarity matrix (Alami et al, 2021; Correa and Lindstrom, 2012).

4.1.3.1 Theory of spectral clustering

Spectral clustering takes eigenvectors related to eigenvalues of normalized Laplacian or other modifications of the adjacency matrix representing the graph structure into consideration while partitioning the graph (Schaeffer, 2007). Even if there are some variants of well-known clustering algorithm K-means to over- come specific issues, k-means has some certain drawbacks such as dependency on initialization method and the risk of stuck at local optimum (Alami et al, 2021). Also, high dimensionality is not very friendly with K-Means. It leads to poor results even if dimensionality reduction techniques such as Principal Component Analysis (PCA) improves results, which is also not effective against complex data structures (Duan et al, 2021).

The ease of implementation and outperforming performance by using stan- dard linear algebra methods are the main advantages of spectral clustering (Duan et al, 2021). It is also able to work with different geometries (Yan et al, 2009). Because of that spectral clustering is used where K-means can not able to work (El-Bhissy et al, 2014). On the other hand, heavy need on sorting and decomposition makes the spectral algo- rithm less performant for large data sets with high dimensions, which are also known as the ”curse of dimensionality” and makes it impossible to distinguish neighborhood in a meaningful way (Andrews and Fox, 2007; Assent, 2012; Aggarwal et al, 2001; Tremblay and Loukas, 2020; Duan et al, 2021; Wang et al, 2009; Wu et al, 2009). As of our best knowledge, there is no single definition for at what limit high dimensionality starts (Assent, 2012).

Spectral clustering generally consists of three steps which are (Tremblay and Loukas, 2020):

- Preparing graph

- Spectral embedding
- Clustering

One of the ways for modeling text documents is using graph structures. Even if more significant graphs in size lead to complex clustering processes, graph structures are good at representing characteristics hidden in the datasets (Liu et al, 2013). In order to utilize spectral clustering methods on graphs, similarity graphs that are based on the pairwise similarities are utilized. There are three common similarity graphs in the spectral clustering literature which are (Janani and Vijayarani, 2019) :

- e - Neighborhood Graph: nodes of which pairwise distances are smaller than e are connected.
- k - Nearest Neighbor Graph: nodes are connected with the nearest k neighbors.
- Fully connected Graph: all nodes are connected.

The connection between nodes of a graph is represented by an adjacency matrix known as affinity matrix (Andrews and Fox, 2007). In order to eliminate correlations while preparing the adjacency matrix with the complex data structures in high dimensional space, kernel functions are applied (Talebi et al, 2020). However, the choice of the kernel is still a debating item in the literature (Andrews and Fox, 2007). In other words, the benefit of applying a kernel is projecting features into high dimensions (Assent, 2012). The radial basis function (RBF) kernel is a way to define a normalized Laplacian matrix from a feature set distinguished based on Euclidean similarity (Talebi et al, 2020).

Eigengap heuristic is a way to determine the number of clusters while applying spectral clustering (Talebi et al, 2020). It tries to find the largest eigengap value when eigenvalues are ordered (Talebi et al, 2020). Since eigengap heuristic is mostly able to process well-defined features but not complex features with different scales, different versions of it have been developed to cope with real-world problems (Afzalan and Jazizadeh, 2019). Even if local scaling is an efficient way to build models by taking every data points' measurement scales, it requires more computational power than global scaling since it introduces extra steps while generating affinity matrix (Afzalan and Jazizadeh, 2019). Besides, it also introduces new parameters such as the number of neighbors, which is a vital element for reaching good results. Even if it is usually set as 7, this parameter is also domain dependent (Correa and Lindstrom, 2012).

Although there are different spectral clustering algorithms in the literature by various researchers (von Luxburg, 2007), the spectral clustering algorithms are categorized into three groups (Verma and Meila, 2003):

- Recursive Spectral: Recursively divides data into two groups based on a single eigenvector.
- Multiway Spectral: Multiple eigenvectors are taken into account in order to generate multiple partitions.
- Non-spectral: Simpler group techniques other than the former two groups.

As it is stated in the literature, the application of spectral clustering on large data sets and large graphs leads to complexity and high resource consumption(Yan et al, 2009; Wang et al, 2009). In order to overcome the problem occurred when spectral clustering is applied to big data approximation methods are introduced.

4.1.3.2 Approximation

It can be easily found out that a large data set with millions of rows with double precision requires more than 80 gigabyte (GB) of memory even if a small subset is used (Li et al, 2011). This need is basically because of the heavy calculations made while sorting the affinity matrix and decomposing the Laplacian matrix. The complexity of building an affinity matrix is $O(n^2p)$ whereas decomposing the eigenvalues and eigenvectors requires $O(n^3)$ time (Liu et al, 2013; Cao et al, 2013), where n is the number of data items. Even some methods such as Nystr¨om Method try to reduce the need for computational resources for spectral clustering on the data set; it still requires high memory capacity (Yan et al, 2009). Spectral clustering methods still are not scalable enough to handle big data (Li et al, 2011).

In order to overcome the problems between spectral clustering and big data, numerous approximation methods have been introduced with acceptable accuracy and lighter computational cost (Wang et al, 2009; Wand and Dong, 2012). In order to reduce the size of the graph and computational resource need, nodes are randomly selected from the graph, clustered according to distance with the selected nodes and eigen decomposition is applied on the centroids of these clusters (Liu et al, 2013). Another way is to approximate by selecting good representatives of data points via sampling (Tremblay and Loukas, 2020; Yan et al, 2009). Local information is also used in order develop a spectral clustering approximation where general clustering assumptions of inner cluster similarity and closeness of nearby data items are taken into consideration (Cao et al, 2013). As the data size gets bigger, by using constrained normalized cuts and graph construction techniques, scalable constrained spectral clustering method is developed in order to eliminate the computational burden of classical spectral clustering algorithm (Liu et al, 2017).

Unfortunately, approximations methods come with drawbacks such as limited performance improvement, taking assumptions into account while running (Tremblay and Loukas, 2020) and need of tuning parameters before processing which is not applicable for many real world applications (Alshammari and Takatsuka, 2019). On the other hand, the quality metrics of our approximation algorithms can be seen below as it can be found in the literature (Tremblay and Loukas, 2020):

- Less computational power compared to the regular version of the algorithm
- Acceptable solution quality

The literature on approximations reveals that current approximation methods are run on small or medium datasets and do not take the computational speed into consideration while introducing additional complexity (Alguliyev et al, 2021). In the next section, a new approximation technique which both requires less computational power and provides acceptable solution quality is proposed while being able to process large volume of data.

4.2 A New Spectral Clustering Approximation Proposal

In this section, we presented a new approximation approach for spectral clustering which requires less computational power and is fast enough to provide acceptable solution quality, as we already mentioned.

The proposed method is a self-tuning approximation method for spectral clustering algorithm which firstly finds out hyperparameters and then runs for providing acceptable solutions after processing large amounts of text data. Benchmark results based on a common text data set are also provided in this section.

As it can be seen from Algorithm 1 in Figure 4.1, the approximation algorithm takes a set of text data or simply documents as input and outputs clustered documents. Documents are read first, then preprocessed with cleaning such as eliminating empty rows etc. Afterwards, text data is converted to lower case, all non-letter characters and stop words are removed. The processed data set is converted to word embeddings, also known as word or documents vectors and then they are normalized so that all values are between −1 and 1.

Word embedding is a relatively new technique that provides good performance on NLP models by solving common problems such as scalability. It is used to convert documents into vectors by utilizing pre-trained model based on neural network architecture trained on large scale text data (Zhang et al, 2020), (Kumhar et al, 2021), (Ruas et al, 2020). In this study, Facebook's FastText library is used in order to convert documents into vectors by utilizing word embeddings (FastText, 2021).

Initialization is the step where the algorithm applies search for best parameter values in a predefined space. This step produces all kinds of metrics in order to decide ideal values for affinity matrix calculations and the number of clusters. A user-defined number of iterations are run on normalized vectors and aforementioned parameters are found. After defining parameters, the actual clustering run is applied to the data set and results are appended to the document set as a new column.

The flow of the initialization phase is seen from Algorithm 2 in Figure 4.2; the initialization procedure tries different sigma values, which is a parameter for calculating affinity matrix for predefined times. In all these iterations, samples from the main data set is selected and spectral clustering is applied. The result of the clustering process is evaluated with performance metrics and these metrics are provided to find best values for the parameters needed for the main run of the algorithm.

It is clear that for real world problems, the ground truth is not available at the algorithm's runtime. Because of this fact, internal indices are utilized for correctly align the algorithm. There are many internal indices in the literature; three of them which are widely used are listed below (Wang et al, 2009):

- Silhouette index
- Davies-Bouldin index
- Calinski-Harabasz index

The internal indices can be used to tune the algorithm to provide high performance as a post-processing step. For example; Davies-Boulding and Silhouette indexes have certain applications for this purpose (Duan et al, 2021; Talebi et al, 2020).

In our initialization step, we use these three internal indices to find out which clusters are better than the others. Since these indices have different scales we normalize them with min-max normalization and have values on the same scale. The normalized values are summed to get a new feature which is called as ”Score”:

Algorithm 1 Mini Batch Spectral Clustering Approximation

Input: Set of Documents
Output: Clustered Documents
Require: *documents*, *batch_size*, *number_of_iterations*, *sigma_values*

```
data = read_data(file_name)
data = clean_data(data)
data = lower_text(data)
data = remove_non_letter_characters(data)
data = remove_stopwords(data)
vectors = vectorize_data(data)
vectors = normalize_vectors(vectors)
scores = Initialization(vectors, batch_size, experiment_iterations, sigma_values)
ideal_sigma = get_ideal_sigma(scores)
upper_limit, lower_limit = get_limits(scores)
data = Run(vectors, batch_size, ideal_sigma, upper_limit, lower_limit)
data = load_data(documents)
data = clean_data(data)
data = lower_text(data)
data = remove_non_letter_characters(data)
data = remove_stopwords(data)
vectors = vectorize_data(data)
vectors = normalize_vectors(vectors)
scores = Initialization(vectors, batch_size, number_of_iterations, sigma_values)
ideal_sigma = get_ideal_sigma(scores)
upper_limit, lower_limit = get_limits(scores)
clustered_data = Run(vectors, batch_size, ideal_sigma, upper_limit, lower_limit)
return clustered_data
```

Figure 4.1: Mini batch spectral clustering approximation.

Score = Silhouette Score+Davies Bouldin Score+Calinski Harabasz Score

In order to find out what is the computational cost and gain related with every cluster we get from the algorithm, two new metrics are calculated as below.

Cost Per Cluster = Iteration Run Time/Number Of Clusters

Gain Per Cluster = Score/Number Of Clusters

After these calculations, we have enough information on what every new cluster provides how much improvement to our scores.

$$Improvement\ Per\ Cluster = Gain\ Per\ Cluster - Cost\ Per\ Cluster$$

These calculated metrics are provided to calculate values of ideal sigma, upper limit and lower limit. Those last two, upper and lower limits, are used to limit the number of clusters in a limited space and stabilize eigengap heuristics which becomes destabilized very easily. The ideal sigma is the value where the algorithm provides the best score. The upper limit is the maximum of number of clusters in scores where sigma is equal to the ideal sigma and the lower limit is the minimum one.

After calculating the parameters, the approximation algorithm is applied to the whole data set to find the real clusters which can be seen in Algorithm 3 in Figure 4.3. This procedure takes all vectorized texts and divides them into chunks based on the batch size provided by the user. Every batch is clustered separately with spectral clustering and all clusters are added to data as a label. After all items in the data set are processed, centroids of clusters are calculated. These centroids are clustered with the K-Means algorithm rather than the whole data. The clusters provided by K-Means are broadcasted to the data set in order to have the final clusters.

There are two important subprocedure in the algorithm which are the selection of eigenvectors to be clustered and clustering the eigen vectors. The first subprocedure which is responsible part for calculating spectral cluster- ing parameters can be seen in Algorithm 4 in Figure 4.4. This step is a classical spectral clustering implementation with normalized Laplacian matrix but includes a stabilization process by bounding the eigengap heuristic with parametric bounds.

Algorithm 2 Initialization

Input: Document Vectors
Output: Parameters

Require: *vectors*, *sigma_values*, *batch_size*, *maximum_iteration_numbers*

```
number_of_clusters = 0
upper_limit = batch_size
lower_limit = 2
scores = [ ]
mini_batch_data = get_random_data(data)
for sigma in sigma_values do
    for iteration in range(maximum_iteration_numbers) do
        number_of_clusters, normalized_n_eigen_vectors = Eigen_Selection(vectors, sigma, stabilized = False, lower_limit,upper_limit)
        clusters = Cluster(number_of_clusters, normalized_n_eigen_vectors)
        run_time = calculate_run_time()
        silhouette_score = get_silhouette_score(mini_batch_data, clusters)
        calinski_harabasz_score = get_calinski_harabasz_score(mini_batch_data, clusters)
        davies_bouldin_score = get_davies_bouldin_score(mini_batch_data, clusters)
        improvement_per_cluster = calculate_improvement_per_cluster(silhouette_score, calinski_harabasz_score, davies_bouldin_score)
        scores.add(sigma, number_of_clusters, run_time, silhouette_score, calinski_harabasz_score, davies_bouldin_score, improvement_per_cluster)
    end for
end for
return scores
```

Figure 4.2: Initialization.

Algorithm 3 Processing

Input: Document Vectors
Output: Clustered Documents

Require: *data*, *batch_size*, *ideal_sigma*, *upper_limit*, *lower_limit*

```
total_item_count = number_of_items_in_data
total_processed_item_count = 0
while total_processed_item_count < total_item_count do
    mini_batch_data = get_random_data(batch_size)
    stabilised = False
    if number_of_clusters is same for n iterations then
        stabilised = True
    end if
    number_of_clusters, normalized_n_eigen_vectors = Eigen_Selection(vectors, sigma, stabilized, lower_limit, upper_limit)
    if number_of_clusters is in range(lower_limit, upper_limit) then
        lower_limit, upper_limit) = set_limits(number_of_clusters)
    end if
    clusters = Cluster(number_of_clusters, normalized_n_eigen_vectors)
    total_processed_item_count += batch_size
    update_data_with_clusters(mini_batch_data, clusters)
end while
centroids = calculate_centroids_by_clusters(data)
clusters = Cluster(number_of_clusters, centroids)
update_data_with_clusters(data, clusters)
return data
```

Figure 4.3: Processing.

Algorithm 4 Eigenvector Selection

Input: Document Vectors

Output: Number of Clusters, Selected Eigenvectors

Require: *vectors*, *sigma*, *stabilized*, *lower_limit*, *upper_limit*

```
distances = get_euclidean_distances(vectors)
affinity_matrix = get_affinity_matrix(distances, sigma)
degree_matrix = get_degree_matrix(affinity_matrix)
laplacian_matrix = get_laplacian_matrix(affinity_matrix, degree_matrix)
eigen_values, eigen_vectors = get_eigens(laplacian_matrix)
if stabilized then
    number_of_clusters = get_eigen_gap(eigen_values)
end if
if number_of_clusters ≤ lower_limit then
    number_of_clusters = lower_limit
end if
if number_of_clusters ≥ upper_limit then
    number_of_clusters = upper_limit
end if
largest_n_eigen_vectors = eigen_vectors[number_of_clusters]
transposed_n_eigen_vectors = transpose_eigen_vectors(eigen_vectors)
normalized_n_eigen_vectors = normalize_eigen_vectors(transposed_eigen_vectors)
return number_of_clusters, normalized_n_eigen_vectors
```

Figure 4.4: Eigenvector selection.

The distances between vectors are calculated as Euclidean distances which are briefly discussed. After calculating distances, they are used to build an adjacency matrix. We prefer to use a version of the adjacency matrix which is known as the weighted adjacency matrix, used in different studies such as analyzing high order of networks (Benson et al, 2016). It is defined as:

$$A = [a_{ij}] \tag{4.4}$$

where A is n × n matrix and

$$A = \begin{cases} \text{wij the weight of edge if there is one between vertices vi and vj} \\ 0, \text{otherwise} \end{cases}$$

In our study, we used distance as weights in the adjacency matrix. This decision is computationally efficient but has a significant drawback. Distance is not an indicator of similarity but dissimilarity. The distance-based measurement is converted to similarity and the normalized adjacency matrix is converted to an affinity matrix. Even if the adjacency matrix and affinity matrix are used interchangeably in literature (Boughanem et al, 2009; Dana, 2018; Zhang, 2020), we use distance-based adjacency as a dissimilarity indicator and affinity as a similarity indicator.

Even if there are many sophisticated and complex similarity functions in literature, Gaussian Similarity Function (Gopal, 2018) converts the adjacency matrix to an affinity matrix. The Gaussian function, defined below as kernel function, is commonly used since it is one of the best kernel applications to deal with fully connected graphs with complex statistical features (Duan et al, 2021). The Gaussian Kernel formula is (Tremblay and Loukas, 2020; Wang et al, 2009; Ng et al, 2002):

$$\exp(-|p_i - p_j|^2/2\sigma^2) \tag{4.5}$$

Since Gaussian requires iterations in order to find out the ideal parameters (Favati et al, 2020) deciding on these parameters is a human-driven process (Ng et al, 2002), tirals based on different parameters values are already conducted in Initialization phase.

Affinity matrix is used to calculate the degree matrix(d_i), an important component for calculating Laplacian matrix of the graph represented with affinity matrix (von Luxburg, 2007).

$$d_i = \sum_j^n a_{ij} \tag{4.6}$$

By using affinity and degree matrices, Laplacian matrix (L) is calculated which is used to find eigenvalues and eigenvectors (Ng et al, 2002).

$$L = D^{1/2} A D^{1/2} \tag{4.7}$$

If the algorithm is not stabilized yet, we utilize the eigengap heuristics to find out number of clusters. After getting the number of clusters, we update our upper or lower bounds. After that update step, eigenvalues are calculated in ascending order. Based on the values, the largest eigenvectors as many as the number of clusters are selected. The eigenvectors are firstly transposed and then normalized. The procedure sends back the number of clusters and normalized eigenvectors to main flow.

The second important subprocedure is the clustering step which is utilized after calculating desired number of clusters and normalized eigenvectors. The flow of the subprocedure can be seen in Algorithm 5 in Figure 4.5. This step divides the eigen vectors into the given number of clusters.

Algorithm 5 Clustering

Input: Document Vectors

Output: Number of Clusters, Selected Eigenvectors

Require: $number\_of\_clusters$, $normalized\_n\_eigen\_vectors$

```
clusters = apply_kmeans_clustering(normalized_n_eigen_vectors, number_of_clusters)
return clusters
```

Figure 4.5: Clustering.

4.3 Finding Recovery Actions With New Spectral Method

Data structures in the transportation industry can be easily modeled as graphs, networks of nodes. Some studies in literature aim to understand structures in networks such as the one covering air transportation network (Guimera et al, 2005).

Generally, data sets being analyzed are not suitable for building graph at the beginning. Data transformation methods are applied to convert data set into a building graph is possible in order to manage computationally demand- ing graph processes (Schaeffer, 2007). A weighted undirected complete graph is prepared which is connecting 25,000 nodes in sample data with more than 400,000,000 edges.

In the analysis within the scope of the study, crew operation data of a large-scale European airline is used. This data covers more than 60 millions entry related with crew disruptions between 2012 and 2019. There are different disruption types in the data set such as flight, ground activity etc. As it is already indicated, flight disruptions are selected from this data set for analysis purposes.

A total of 22,592,770 crew disruptions related to flight duties, covering 2018 and 2019 have been selected due to the low data quality before 2018. After all data cleaning process, 3,250,000 document records are selected as mature enough for analysis.

The most frequent disruptions can be seen in Table 4.1.

It is impossible to process the volume of data we have mentioned before with the reference spectral clustering implementations such as the one in the SKLearn library (Scikit-learn, 2021) due to computational capacity limitations. We have applied the proposed methodology to the data set and successfully cluster the data. The initialization process is run 30 times for every sigma values in the sigma values set which includes values of [0.001, 0.01, 0.1, 1, 2, 5, 7, 10, 20, 50, 70, 100, 150, 200].

Table 4.1: Crew disruption distribution.

Disruption	Count	Frequency (%)
Late for Duty Check-in	4.008.976	17.78
Flight Delay	2.970.943	13.18
Min Connection Time in Domestic Stations	1.721.046	7.63
Open Time for Flight Attendant	1.459.628	6.48
Duty End Time Change	1.302.066	5.78
Duty Start Time Change	1.013.114	4.49
Flight Cancellation	587.931	2.61
Aircraft Change in International Station	587.031	2.60
Open Time for First Officer	586.872	2.60
Open Time for Cabin Chef	521.717	2.31
Others	699.818	34.53

The initialization scores can be seen in Figure 4.6. The initialization scores are analyzed and ideal sigma score is 0.1, upper limit as 92 and lower limit as 47. After the algorithm is run on the data set, it converges to the lower limit and finds 47 clusters. The total processing time is 2 hours 10 minutes for 3,250,000 documents.

Our findings summarized in this section will provide some insights to researchers dealing with the problem of crew recovery. Even if the data is gathered from only one airline both the volume of data and the characteristics of the airline making it possible to generalize these findings to most airlines. The utilization of narrow and wide-body aircrafts, large network of destinations and high number of crew make it possible to cover many daily problems most airlines face today.

The 47 clusters in 3,250,000 documents are investigated with airline experts in order to find the recovery actions or some insights for recoveries. The close enough clusters are concatenated if possible and this human in loop process leads to 23 recovery actions listed below.

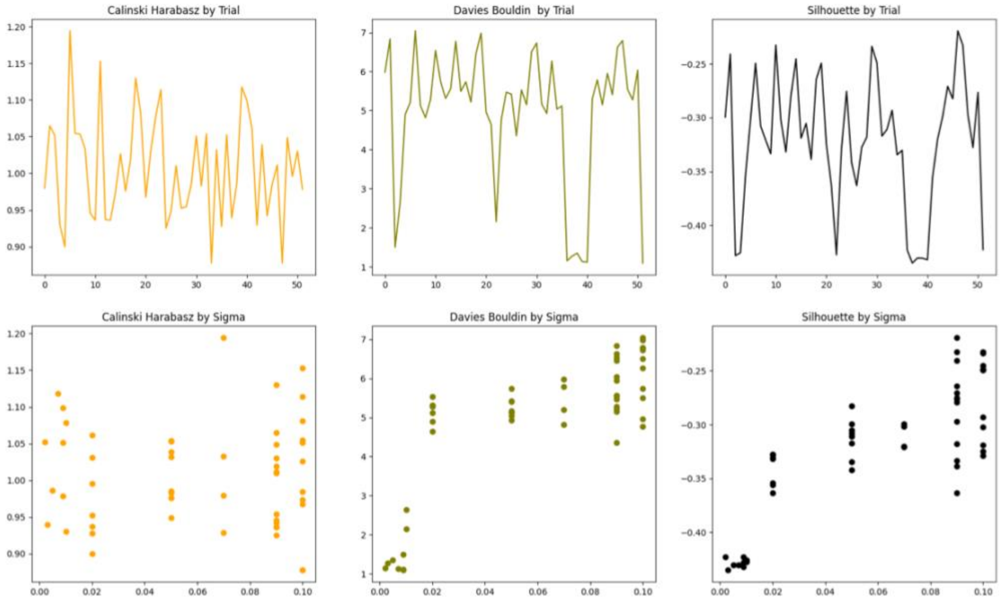

Figure 4.6: Initialization scores.

- ASSIGN CABIN CREW: Assign a new cabin crew
- ASSIGN COCKPIT CREW: Assign a new cockpit crew
- CANCEL FLIGHT: Cancel the flight disruption that occurs
- DEADHEAD: Transport crew as passengers between two stations
- DEASSIGN CABIN CREW: Deassign one assigned cabin crew
- DEASSIGN COCKPIT CREW: Deassign one assigned cabin crew
- DELAY FLIGHT: Delay the flight disruption occurs for a certain amount of time
- SWAP AIRCRAFT: Change the aircraft
- SWAP AIRCRAFT WITH AIRCRAFT HAS AVAILABLE SEAT: Change the aircraft with available seats
- SWAP AIRCRAFT WITH AIRCRAFT HAS RESTBUNK OR REST-
- SEAT: Change the aircraft with rest area for crew
- SWAP CREW WITH CREW AT SPECIFIC AIRPORT: Change assignment of crew with another one who is currently at a specific station
- SWAP CREW WITH CREW HAS CATEGORY QUALIFICATION: Change assignment of crew with another one who has CAT qualification
- SWAP CREW WITH CREW HAS HEALTH QUALIFICATION: Change
- assignment of crew with another one who has health qualification
- SWAP CREW WITH CREW HAS LANDING QUALIFICATION: Change assignment of crew with another one who has already landed at destination airport before

- SWAP CREW WITH CREW HAS LANGUAGE QUALIFICATION: Change assignment of crew with another one who has language qualification
- SWAP CREW WITH CREW HAS LINE CHECK QUALIFICATION: Change assignment of crew with another one who has passed in flight controls
- SWAP CREW WITH CREW HAS PASSPORT OR VISA: Change assignment of crew with another one who has passport/visa for the destination country
- SWAP CREW WITH CREW HAS SPECIAL AIRPORT QUALIFICATION: Change assignment of crew with another one who has specific qualification for the destination station
- SWAP CREW WITH CREW HAS TECHNICAL QUALIFICATION: Change assignment of crew with another one who has legitimate qualification for flying with a specific aircraft
- SWAP CREW WITH CREW HAS VALID FLIGHT CREW DOCUMENTS: Change assignment of crew with another one who has necessary crew documents for the destination station
- SWAP CREW WITH CREW NOT FLOWN TO AIRPORT IN THE MONTH: Change assignment of crew with another one who has not flown to the destination station in current month
- SWAP CREW WITH EXPERIENCED CREW:Change assignment of crew with another experienced one
- SWAP CREW WITH STANDBY CREW: Change assignment of crew with another one who is on airport / home standby duty

By using these actions, recovery strategies can be determined or solutions to be used during recovery can be modeled more accurately. In particular, during the preparation of recovery optimization models, more effective models will be produced by using the actions determined by this study as input to the model. For example, providing the listed actions to the optimization model as a constraint will result in the production of optimization models that produce faster solutions.

Lastly, more details about business rules, flight numbers or departure air- ports are not revealed because of data sharing regulations and use even if data is masked.

5. PREDICTING RECOVERY ACTIONS

The study of intelligent agents that imitate humans in learning and problem solving methods is known as AI (Khallaf and Khallaf, 2021).

ML is classified as a subcategory of the field of AI. Machine learning, which focuses on the realization of learning without providing clear and unambiguous computer programming routines; makes use of supervised, unsupervised, semi-supervised or reinforcement learning methods (Khallaf and Khallaf, 2021).

Machine learning can be studied basically in two sub-fields. These areas are known as shallow learning and deep learning areas. With shallow learning and the use of constraints, algorithms that do not contain many layers achieve results by using linear or nonlinear features. This category includes Logistic Regression, Support Vector Machines, etc. (Dong et al, 2021). Shallow models developed to solve problems quickly and efficiently are insufficient in complex problems encountered in the modern world such as natural language processing and image processing (Neu et al, 2021; Dong et al, 2021).

With the increasing amount of computational power and memory, the area called deep learning has emerged, which consists of multi-layered neural networks and performs much more successfully than traditional ML approaches based on shallow learning (Watson et al, 2020; Neu et al, 2021). Deep learning has been named “deep” due to the amount of layers it contains, and it has created a fundamental change in approach in the field of machine learning, enabling features and patterns to be learned from past experiences and used in calculations (Watson et al, 2020; Khallaf and Khallaf, 2021).

5.1 Understanding Deep Learning

Deep learning, with its innovative approach to modeling fields such as, network design, estimation and resource allocation, has opened new pages in the studies of many problems, especially image recognition and time series problems (Jiang, 2021). In particular, it has gained popularity in academia and industry with its significant contributions to automating time-consuming manual processes (Khallaf and Khallaf, 2021).

Although DL models contain three types of layers: input, hidden, and output, they are quite successful in complex real-world problems, as the representations used for the properties to be learned can have complexity beyond the human mind. (Khallaf and Khallaf, 2021; Neu et al, 2021). In fact, there are many hidden layers between input and output in deep learning models, which includes a large number of neural classifier units consisting of activation functions and linear regression (Dong et al, 2021).

Deep learning models are being updated and developed day by day thanks to big data and developing technology (Dong et al, 2021). These models are characterized by architectures in which they undergo mathematical transformation adjusted using various learning and optimization algorithms as the data progress from the given input layers to the output layers (Watson et al, 2020). The most common deep learning architecture types can be seen in Figure 5.1.

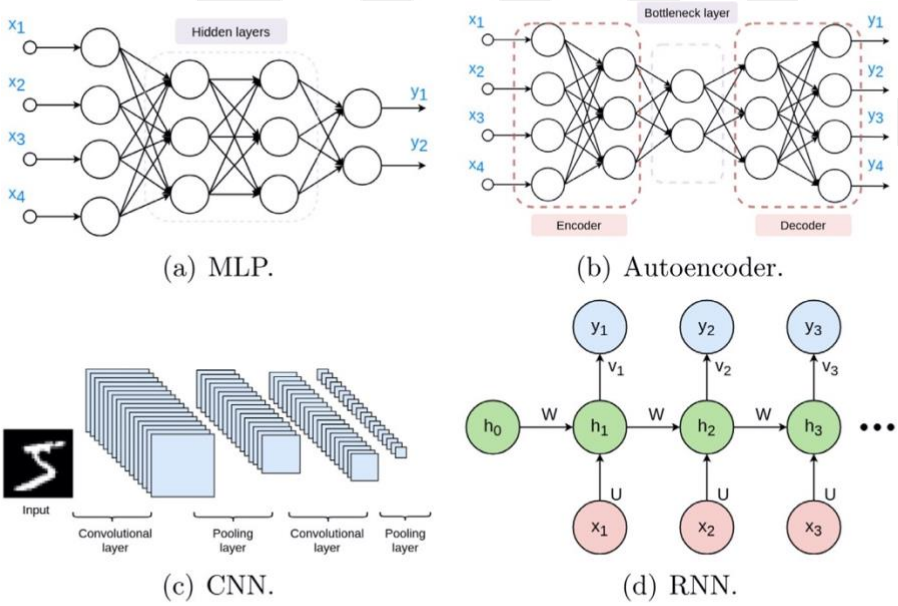

Figure 5.1: Different deep learning architectures (Bochie et al, 2021).

Different deep learning architectures have been applied in many different fields, especially in areas such as image classification, object detection and natural language processing, to solve AI problems (He et al, 2021). In this context, new approaches have been proposed in the field of machine learning for selecting the most appropriate algorithm and automating the determination of hyper parameter values of these algorithms. It is aimed to provide researchers with less expertise in the field of machine learning to benefit from techniques by minimizing human intervention in the processes with the techniques built within the framework of these approaches and approaches (Elshawi et al, 2019). Particularly for unstructured data, a paradigm shift is taking place, from feature engineering to specifying network components and architectures with deep learning methods using manual techniques that involve laborious processes (Wistuba et al, 2019).

Artificial neural network (ANN) design is a process that is difficult for beginners to model for their own needs, largely based on the knowledge and experience of researchers (Ren et al, 2020). Therefore, searching for suitable architectures for relevant models is a time-consuming, laborious and error-prone task in itself (Wistuba et al, 2019; Elsken et al, 2019). For this reason, a new approach to end-to-end automation of all machine learning processes, namely automatic machine learning (AutoML) technique, has emerged in order to reduce the costs incurred during the development of machine learning models, especially deep learning (He et al, 2021).

5.2 Automatic Machine Learning (AutoML)

Generally machine learning specifically deep learning projects require human resources who have deep knowledge on the domain and machine learning techniques. To deal with such a bottleneck of the resource scarcity, the technique of automating machine learning pipelines or tasks is developed which is known as AutoML (Krauss et al, 2020).

AutoML can be defined as a combination of process automation and machine learning in order to create a sequential learning model within a limited budget. This sequential model includes the processes starting from data preparation, model generation and evaluation, testing the model and finally deploying it to the live environment as it can be seen in Figure 5.2 (He et al, 2021).

AutoML is gaining huge attention among the experts as well as non-experts and the literature is expanding with new studies which are focusing on hyperparameter optimization, neural architecture search, pipeline optimization and feature engineering (Waring et al, 2020), most common ones are hyperparameter optimization and neural architecture search (Nikitin et al, 2021). Although, different approaches have been proposed to guide the AutoML studies (Cohen-Shapira and Rokach, 2021), AutoML approaches generally consist of a problem class to solve such as regression or classification, data to process in different formats such as image or text, pipeline to execute, model types to use in design phase and optimization method for model automation (Nikitin et al, 2021).

In general, a simple two-layer artificial neural network, whose number of parameters exceeds the number of data it performs in the learning process, can complete the learning processes with high performance (Zhang et al, 2016). However, different techniques are used to determine high performance model architectures specific to complex problems. These techniques solve an optimization problem that involves determining the machine learning algorithm that offers the most optimal performance and adjusting the hyperparameters of the algorithm, using metrics such as accuracy (Elshawi et al, 2019). Within the scope of these solutions, it can be seen that AutoML applications give better results than human experts modeling traditional machine learning models for different problem areas, especially in model selection and hyperparameter optimization (Soares et al, 2019; Patel et al, 2021).

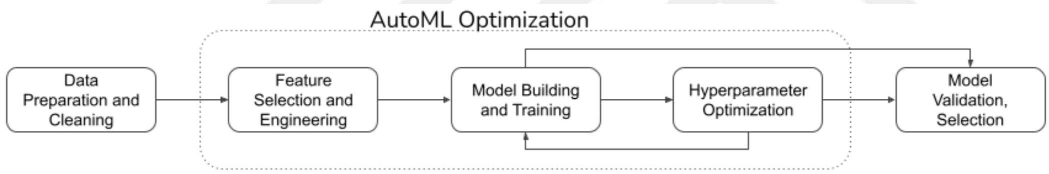

Figure 5.2: AutoML optimization (Waring et al, 2019).

Model production can be briefly discussed in two different stages within the framework of the AutoML approach. The first of these is to determine the search space in which the best model will be searched. In this space framework, artificial neural network architectures are used as well as traditional methods. The second stage is the optimization stage. The optimization of the hyper-parameters used in training, such as learning rate and batch size, and the parameters related to the architecture of artificial neural networks, such as the number of layers, are investigated at this stage (He et al, 2021). Overall process is summarized in Figure 5.3.

Determining the search space is an important component of neural network architecture research in terms of time and solution quality (Wistuba et al, 2019). Deep learning techniques have transformed the traditional approach of manually specifying features into automatic determination of designs of neural architecture (Ren et al, 2020). Neural architecture search (NAS) technique has been developed for automatic modeling of deep neural networks (Jin et al, 2019). In this context, it is aimed to find a robust and well-performing architecture with minimal human intervention by performing certain basic operations in a predefined search space (He et al, 2021; Ren et al, 2020).

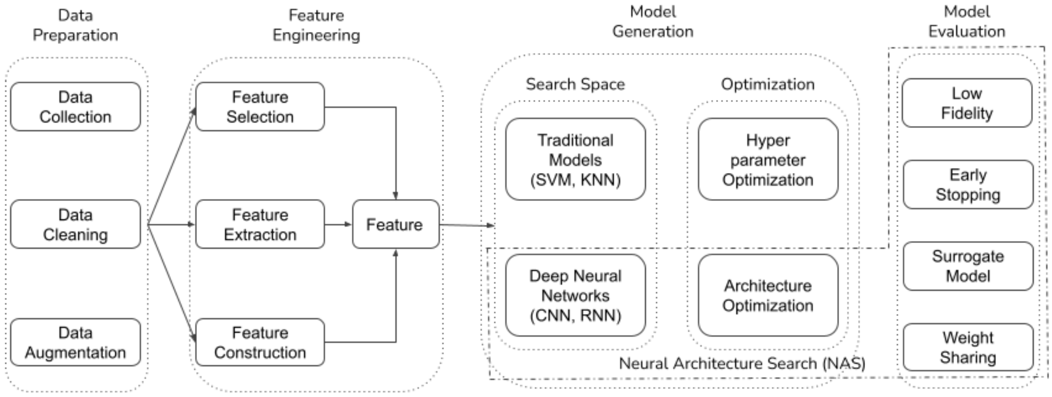

Figure 5.3: Neural Network architecture and hyperparameter optimization (He et al, 2021).

5.3 Neural Architecture Search

Neural Architecture Search (NAS) is one of the automated machine learning techniques. This technique, whose main purpose is to design the appropriate model architecture, is divided into categories such as random search, reinforcement learning, gradient-based methods, evolutionary methods and Bayesian optimization as it can be seen in Figure 5.4 (Elshawi et al, 2019; Xue et al, 2022).

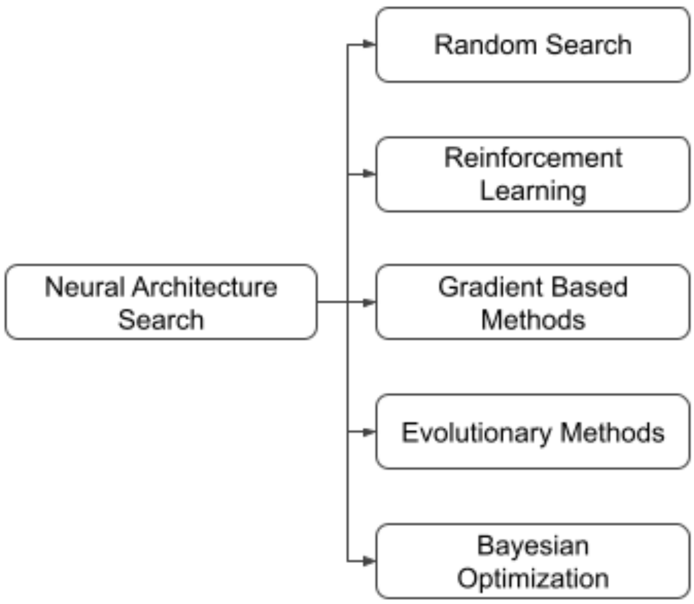

Figure 5.4: Neural Architecture search techniques (Elshawi et al, 2019).

During the neural architecture search, changes are made to the architecture of the reference models while preserving their functionality. This, that is, changing the architecture while maintaining the functionality is called network morphism (Jin et al, 2019).

Overall NAS process can be seen in Figure 5.5.

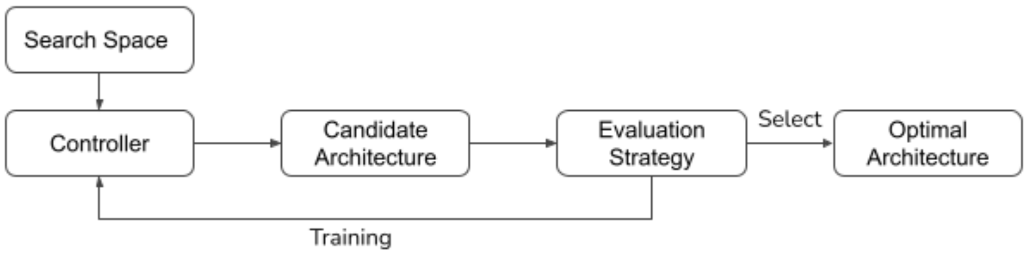

Figure 5.5: Neural architecture search process (Ren et al, 2020).

Current NAS research efforts can be summarized as follows (Ren et al, 2020):

- Modular search: It is a search method in which neural architectures are made from a set of different modules.
- Continuous search: It is a search method in which artificial neural network parameters are made continuously with gradient search.
- Neural architecture recycling: It is a search method that aims to achieve high performance by taking successful architectures that have proven themselves as a reference and making changes on these architectures.
- Under-training: A form of search that aims to determine the relative ranking of models through the sharing of common components of candidate architectures, using less resources.

5.4 Hyperparameter Optimization

There are two types of parameters in machine learning models. The first of these are hyperparameters that somehow control the behavior of the algorithm, usually quite specific to that algorithm, whose values the model designer must determine before starting training. The second one is the parameters optimized in the training of the model (Waring et al, 2019).

The most basic task of AutoML, which includes techniques such as hyper parameter optimization, feature engineering, architectural alternative search techniques, is to automatically determine hyperparameters in order to obtain the best possible, superior performance. The values of the hyperparameters directly affect the performance of the model. Therefore, determining the hyper parameter values is one of the most important steps in model development (Waring et al, 2019).

Mesh search is a computationally expensive method that tries to find all possible combinations for hyper parameter values. In cases where the number of hyperparameters increases, it suffers from the handicaps of multidimensionality. On the other hand, as a different alternative, it is the random search alternative, where the combinations of hyper parameter values are randomly sampled and searched within this sample, which can be easily run in parallel (Elshawi et al, 2019).

Random search is a search method that does not promise optimum value and does not require computational power to find the best values(He et al, 2021).

In general, automation tools used in the field of machine learning can be classified into three main categories: centralized, distributed and cloud-based (Elshawi et al, 2019).

- Central Tools
  - Auto Weka
  - Auto Mecha
  - Auto Sklearn
  - TPOT
  - ML-Plan
  - SmartML
  - AutoStacker
  - AlphaD3M
  - OBOE
  - PMF
  - VDS
- Distributed Tools
  - MLBase
  - TransmogrifAI
  - Rafiki

- Cloud Based
    - Google AutoML
    - Azure AutoML
    - Amazon Sage Maker

In addition to these, there are specialized tools especially for neural architecture search (Elshawi et al, 2019).

- Auto Keras
- Auto Net
- Neural Network Intelligence
- Devol

As suggested in (Jin et al, 2019), a neural architecture search approach including Bayesian optimization was developed as an open source AutoML system and made available to the community. The system, called Auto Keras, has been developed to be easily used by people who are not experts in programming and data science, and to make use of the central processing unit (CPU) and graphical processing unit (GPU) easily. The architecture of this system can be seen below (Jin et al, 2019).

In Auto-Keras' system, while the search algorithms work on the CPU, model training is performed by the GPU. In order for the processes to be parallel, different neural networks are trained in separate processes with the training data. While a graphic-based structure is kept in permanent temporary memory for network morphism operations, each developed model is stored in the storage layer (Jin et al, 2019).

In summary, when the system user starts searching for the best neural architecture for any data set, he starts the search process using the API. In the search phase, a new architecture is created using Bayesian optimization, CPU. Graphics module is used to transform the created architecture into a real neural network in temporary memory. The new architecture is copied to the GPU for Model Trainer to train with the dataset. The trained model is saved in the Model Store. The performance of the model is given as feedback to the previous stages to be used in the next stages (Jin et al, 2019).

Inner dynamics of Auto-Keras system is shown in Figure 5.6.

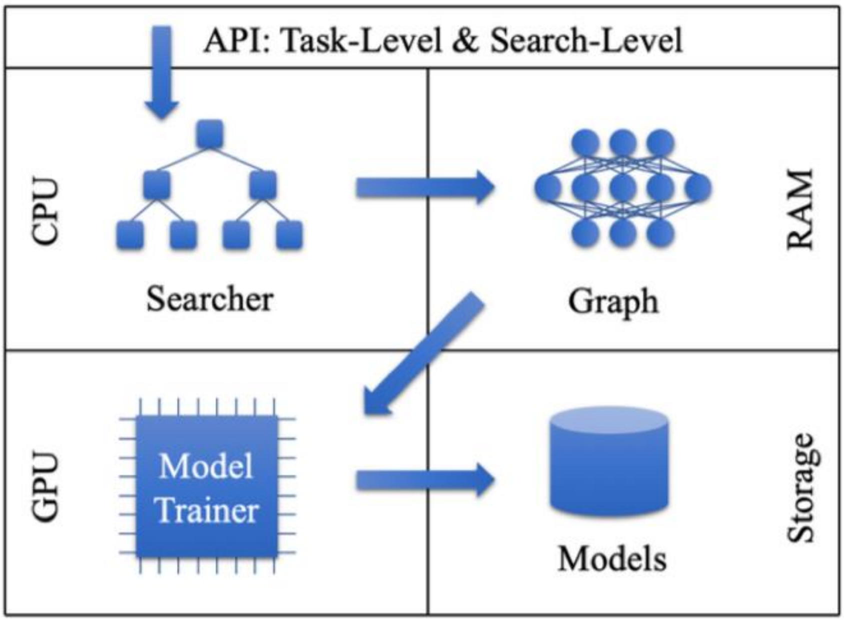

Figure 5.6: Auto keras modules (Jin et al, 2019).

5.5 Deep Learning Model For Predicting Recovery Actions

The information about the structure of the deep learning model remodeled with AutoML approach and AutoKeras within the scope of our study is shown in Table 5.1. The model has totally 343,336 parameters and 343,319 are trainable and 17 of them are non-trainable parameters.

The architecture diagram of the automatically created deep learning model is also found in Figure 5.7.

Table 5.1: Deep learning model structure.

Layer (type)	Output Shape	Param #
Input Layer	(None, 8)	0
Multi Category Encoding Layer	(None, 8)	0
Normalization Layer	(None, 8)	17
Dense Layer	(None, 1024)	9216
Relu Layer	(None, 1024)	0
Dense Layer	(None, 256)	262400
Relu Layer	(None, 256)	0
Dense Layer	(None, 256)	65792
Relu Layer	(None, 256)	0
Dense Layer	(None, 23)	5911
Classification Head Layer	(None, 23)	0

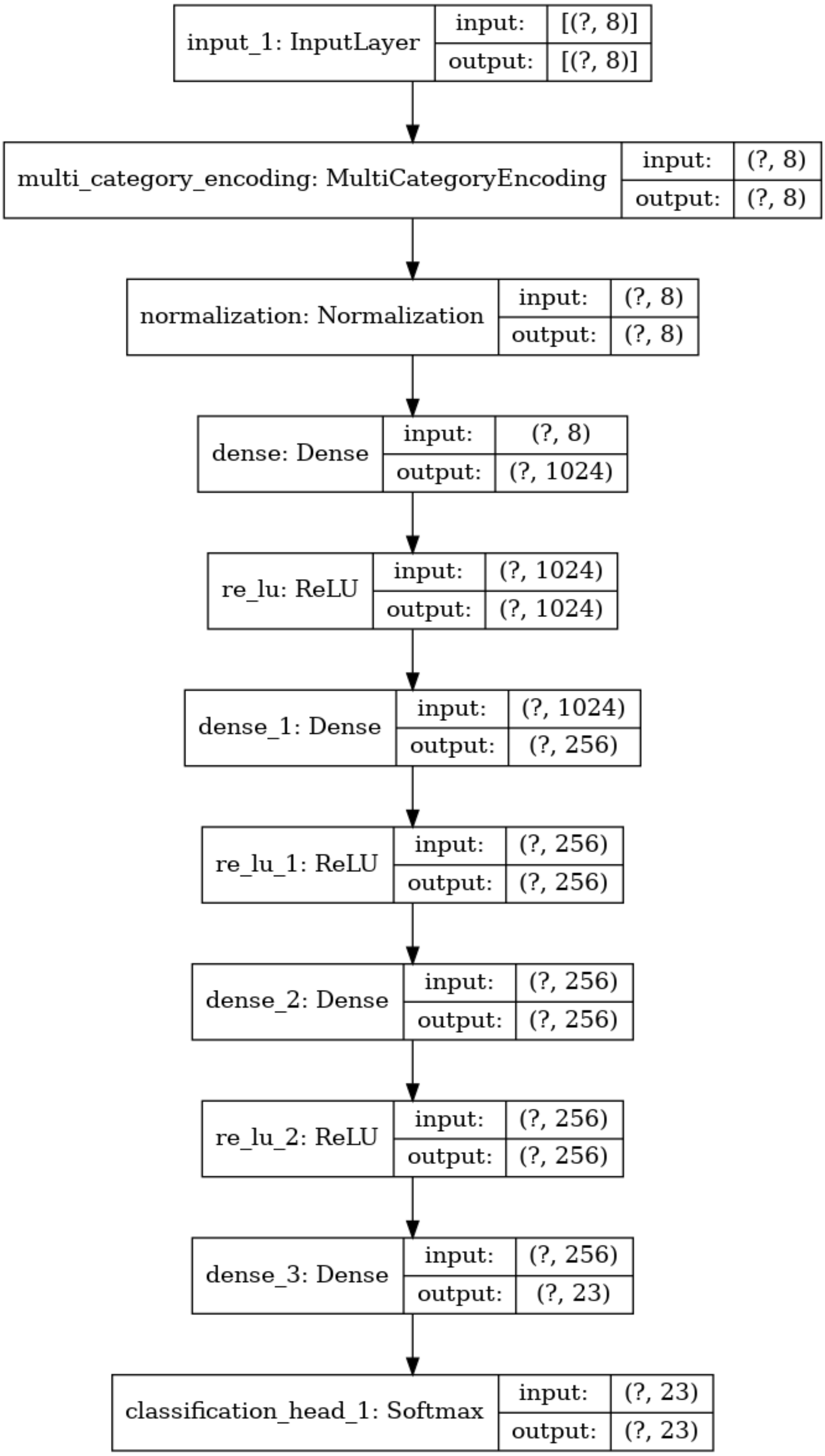

Figure 5.7: Deep learning model architecture.

6. RECOVERING CREW FLIGHT DUTIES[2]

Crew Recovery Problem, has a multi-objective characteristic that is one of the reasons which make the problem NP-hard (Chen and Chou, 2016). Therefore, CRP is an important and sophisticated topic specifically in combinatorial optimization and generally in operational research. It is classified as NP-hard and it requires massive computational power (Zhang et al, 2017). Since CRP search space includes large amount of combinations of crew and flights covering a multi-day time period under dictation of certain constraints, it is not possible to utilize an inclusive search for crew recovery. Because of that, most of the studies in the literature are not mature enough to solve the real world problems covering multiday schedules covering many constraints while seeking for multi objective (Chen and Chou, 2016). Even if, many approaches are investigated in the literature of crew recovery are different in nature, they have common characteristics especially related to simplification of real world problems which makes them unsuitable for direct usage in the real world (Ramle, 2014). One of the simplifications is dropping aircraft qualifications where all crew can operate in all type of aircrafts. (Zhang et al, 2017)

Due to the complex nature of the problem, generally two types of solution methods are investigated in the literature (Zhang et al, 2017):

- Column generation and branching techniques
- Heuristics

To overcome the complex nature of crew problems in the day of operation, column generation and underlying shortest path algorithm based pricing techniques are suggested A search scheme is used to locate the problematic crews and to create a compact storage for column generation in order to minimize the run time of the recovery algorithm (Novianingsih, 2015).

---

[2] This chapter is based on the following publication: Herekoğlu, A., Kabak, Ö. (2023) Crew recovery optimization with deep learning and column generation for sustainable airline operation management. Annals of Operations Research, 1-29. DOI: https://doi.org/10.1007/s10479-023-05738-z

Although, optimality is provided by exact methods, due to the increasing calculational time most of the optimization problems is not solved with optimality (Huiskamp, 2016). Because finding an acceptable recovery solution in short amount of time is very challenging. Therefore, solution quality is neglected to have a solution in acceptable time frame (Zhang et al, 2015).

In order to increase the solution quality, information generated by the solution in the former iterations is used before adding constraints to the model (Ramle, 2014).

One way of utilizing column generation in recovery is using a heuristic approach based on illegalities in the input provides information about crew assignments from subproblem to master one (Ramle, 2014).

6.1 Deep Learning Integrated Optimization

Operations research techniques are widely applied to a significant part of the problems that arise in computational sciences, such as electronics, transportation, management, retail and manufacturing. Recent developments in machine learning in general and deep learning in particular have increased the interest in algorithms in this field for NP-hard problems. Recent studies have benefited from learning algorithms in solving classical NP-hard problems such as traveling salesman and backpack (Li et al, 2018; Gasse et al, 2019; Bengio et al, 2021).

The main motivation behind these techniques is that learning-based approaches have the potential to generate more effective algorithms for NP-hard problems by learning from large data sets. Especially, with the learning process, useful patterns and points that can be missed by human designers while designing algorithms become easily noticeable, increasing the success in these studies. This paves the way for powerful and easily generalizable algorithms in terms of performance (Li et al, 2018).

Especially in the field of IP, tree search algorithms such as branch and bound are among the most common techniques used in solving combinatorial optimization problems. These techniques recursively partition the search space to find the solution. There are examples in the literature for the combination of optimization algorithms and machine learning by making use of machine learning techniques for searching space effectively (Balcan et al, 2018; Gasse et al, 2019; Gupta et al, 2020; Lodi and Zarpellon, 2017; Nair et al, 2020).

On the other hand, machine learning techniques are also used in cases where polynomial time heuristic algorithms are used for problems such as clustering or vehicle routing modeled as graphs. It is seen that different optimization techniques, especially heuristic methods supported by reinforcement learning developed in this context, give consistent results among many different problems (Dai et al, 2017; Nazari et al, 2018; Mazyavkina et al, 2021).

As it can be seen in (Li et al, 2018), this kind of studies are seen as steps towards optimizers to be produced for NP-hard problems by blending classical optimization methods and learning techniques (Li et al, 2018).

6.2 Column Generation

Current parings, current flight schedule, crew rosters, reserve crew list and crew information including limits are used as an input in the crew recovery optimization model (Abdelghany et al, 2004). It is not a good idea to solve the recovery problem by processing flights in chronological order because it makes reaching global optimum impossible. (Abdelghany et al, 2004). Number of possible solutions generated with enumeration can easily reach to huge numbers even if network is comparably small (Villa and Hoffman, 2006). When disruption occurs time is the most critical factor. In order to minimize the time needed for solving problem, there are some problem specific and effective solution methods which are based on classical methods which are able to solve large scale optimization problems (Vitale et al, 2021).

Generally, instead of solving the big models at once, decomposition methods such as Dantzig-Wolfe decomposition is utilized and every crew schedules is represented as columns in column generation (Nissen and Haase, 2006) which is used to overcome the problem of huge numbers of variables in formulation (Lübbecke and Desrosiers, 2005). Dantzig-Wolfe decomposition is a mathematical technique used for solving large-scale optimization problems by dividing them into smaller subproblems. In this method, a large optimization problem is decomposed into smaller subproblems, each of which can be solved more efficiently. The subproblems are then solved iteratively and the solutions are combined to obtain an optimal solution for the original problem.

The Dantzig-Wolfe decomposition approach is particularly useful when dealing with problems that have a large number of variables, constraints, or both. Instead of solving the entire problem at once, the Dantzig-Wolfe decomposition approach allows for the optimization problem to be divided into smaller, more manageable subproblems that can be solved more efficiently.

The approach involves representing the optimization problem in a specific way, where the original problem is expressed as a set of linear constraints, and the objective function is represented as a linear combination of a set of subproblems. The subproblems are formulated in such a way that they correspond to the individual parts of the original problem.

Once the problem has been decomposed, the Dantzig-Wolfe decomposition approach utilizes a technique known as column generation. In this technique, each subproblem is solved individually, and the optimal solutions are used to generate new columns in the original problem's constraint matrix. These new columns represent new potential solutions to the original problem, and the process is repeated iteratively until an optimal solution is found.

Column generation is a method mainly based on the works Dantzig and Wolfe (DW) and Gilmore and Gomory and is one of the best applicable method for large scale linear problems with high number of variables and suitable for relaxation (Muter et al, 2013). It was first Dantzig and Wolfe who developed the idea of reformulating the LP as a column wise structure and then applied to cutting stock problem by Gilmore and Gomory (Lübbecke and Desrosiers, 2005). The technique was first proposed by Dantzig and Wolfe in the 1950s, who used it to solve the transportation problem. They realized that in a transportation problem, it is not necessary to explicitly list all possible routes between sources and destinations, but instead only the active routes need to be considered. This concept of generating columns on demand, based on the active constraints, formed the basis of column generation.

In column generation, the original problem is formulated with a limited number of columns, and the remaining columns are generated dynamically through an iterative process. The approach is particularly useful for problems with a large number of variables, where it is not feasible to generate all possible columns at the outset due to computational limitations.

Since most of the variables are zero at the optimality in large scale LP problems, starting the LP with a small number of columns and then adding necessary columns is the main motivation behind the column generation technique (Muter et al, 2013). The iterative process begins by solving a reduced version of the original problem, known as the pricing problem. The pricing problem identifies a new column that would be useful in the original problem and generates this column. This new column is then added to the original problem and the process is repeated iteratively until an optimal solution is reached.

The advantage of column generation is that it can significantly reduce the computational time required to solve large-scale optimization problems. By generating columns on demand, the number of variables in the problem can be kept relatively small, making it more computationally tractable. With the help of column generation, only columns which provides improved objective value are added to the solution. Column generation technique consists of two types of problems which are known as: Master and Sub or Pricing problems.

The master problem is the original optimization problem that needs to be solved. It contains a limited set of columns or variables, and its solution gives a lower bound on the optimal objective value. The master problem is typically formulated as a linear program (LP) and is solved using a standard LP solver. In other words, the role of master problem (MP) as a the linear relaxation of the problem in column generation is being a starting point for the solution method and it is never solved completely because of the large size (Gamache et al, 1999; Liu et al, 2010).

The procedure for generating columns are called as subproblem or pricing (Villa and Hoffman, 2006). The subproblem (SP), or pricing problem, is used to generate new columns that can improve the solution of the master problem. The subproblem is utilized for controlling the existence of new columns (Gamache et al, 1999). It is a smaller optimization problem that is solved iteratively during the column generation process. The subproblem is typically formulated as a linear program and its objective is to identify a new column that improves the solution of the master problem. The subproblem is solved using specialized algorithms that are designed to solve it efficiently.. The column selection is based on the values provided by pricing problem (Huiskamp, 2016). The pricing problem is used to find out best composition which leads to better LP relaxation of master problem by using the dual variables. The LP duality theory indicates that a solution gets better if there are non-basic variables with negative reduced costs (Liu et al, 2010).

The column generation process involves solving the subproblem and adding any new columns that improve the solution to the master problem. This process continues until the solution of the master problem is optimal. The addition of new columns to the master problem is done dynamically, which allows the column generation process to focus on the most promising columns and reduce the computational effort required to solve the problem.

Let's consider a linear programming model shown below as master model:

$$\min \sum_{i \in I} c_i x_i$$

$$\text{subject to} \sum_{i \in I} a_i x_i \geq b, \tag{6.1}$$

$$x_i \geq 0, \qquad i \in I$$

While solving this mathematical model, a non-basic variable to be able to enter the basis is searched. Let's call $\pi \geq 0$ vector of dual values. So in the sub or pricing problem, negative reduced cost is looked for with an objective function seen below:

$$\min c_i - \pi^T a_i \tag{6.2}$$

As it is indicated in the literature, since enumerating all possible reduced costs is computationally impossible, a subset of master problem called Relaxed or Restricted Master Problem (RMP) is solved and columns with negative reduced costs are added to that problem. The RMP is a simplified version of the master problem in column generation. It is formed by removing some of the constraints of the original master problem and relaxing some of the integrality constraints of the decision variables. By doing this, the RMP becomes a linear programming problem with a smaller number of variables and constraints, which can be solved more efficiently.

Afterwards, the RMP is reoptimized (Lübbecke and Desrosiers, 2005). In column generation, the RMP is solved to obtain an initial feasible solution for the problem. The RMP may contain additional artificial variables that are used to make the initial solution feasible. Therefore, column generation starts with an initial RMP, mostly with artificial decision variables which are used to make the initial solution feasible. The reduced costs of variables which are not covered in RMP are evaluated and suitable variables are selected with the most negative reduced cost. Then, RMP is reoptimized with new variables in every iteration (Alvelos et al, 2013). The column with most negative reduced cost calculated by using dual values is selected as the candidate column and the iterations continues until no candidate column is found with negative reduced costs (Huiskamp, 2016; Vitale et al, 2021). However, there are other derivatives of pricing problems which produces more columns with negative reduced costs (Borgwardt and Patterson, 2022).

The reduced cost of a variable is the amount by which the objective function would decrease if the variable were increased by one unit. Negative reduced costs indicate that increasing the value of the variable would improve the objective function, and thus adding the variable to the RMP would improve the solution.

There are several techniques to generate new columns, such as Branch-and-Price, Gomory Cuts, and Benders Decomposition, which can produce more columns with negative reduced costs than standard pricing problems. These techniques can improve the efficiency and effectiveness of the column generation algorithm.

When the sub or pricing problem is solved, two possibilities can arise. First, if the sub or pricing problem is infeasible, then it means that there are no new columns that can be added to the RMP that would improve the solution. In this case, all variables not covered by the RMP are set to 0, and the master problem is solved to obtain an optimal solution (Demassey et al, 2005) .

Second, if the sub or pricing problem is feasible, then it means that there are new columns that can be added to the RMP to improve the solution. The columns represented by variables with negative reduced cost are then added to the RMP to improve the solution, and the RMP is reoptimized. The process of generating new columns and reoptimizing the RMP continues until no more columns with negative reduced cost can be found (Demassey et al, 2005).

The feasibility of the sub or pricing problem is critical to the success of the column generation algorithm. If the sub or pricing problem is infeasible, then the algorithm terminates, and the solution obtained from the RMP is optimal. On the other hand, if the sub or pricing problem is feasible, then the algorithm continues to generate new columns until no more columns with negative reduced cost can be found (Demassey et al, 2005).

Column generation is flexible for finding the best way to decide on pricing problem while providing solutions for routing and scheduling models as a decomposition technique (Vitale et al, 2021). With decomposition technique, a stronger lower bound for the problem is found very fast compared to the ones provided by LP relaxation (Frangioni and Gendron, 2013). The lower bound is a value that represents the minimum possible value that the objective function can take for any feasible solution. It is important to find a strong lower bound as close as possible to the optimal solution to reduce the gap between the lower bound and the optimal point, which is called the optimality gap. The gap between lower bound and optimal point is wanted to be as small as possible. The quality of the lower bound is very important for applying bounding methods such as branch and bound (Villa and Hoffman, 2006). Correct usage of information of dual model is very helpful of efficient convergence (Medard and Sawhney, 2007). Bounding methods such as branch and bound use the lower bound as a guide to search for the optimal solution, so having a tight lower bound is essential for an efficient and effective solution.

The dual model provides useful information about the problem and can help to improve the convergence of the column generation algorithm. By using the dual information, one can obtain better candidate columns and improve the RMP solution. Dual information can also be used to detect infeasible subproblems, as well as to determine the optimality of the solution.

6.3 Proposed Mathematical Model

Proposed model aims to solve the set partitioning problem as the main problem by making use of the column generation technique in the mathematical model. In this problem, the aim is to assign as many resources as it needs to each flight, and to convert the invalid crew assignments after the disruption into a legitimate crew assignment set.

In the sub-problem of this mathematical model, actions predicted by the learning model form the column formation strategies and minimizing the assignment cost is the objective.

6.3.1 Sets, parameters and variables

The sets, parameters and binary variables that are used in the proposed master problem are shown in Table 6.1, Table 6.2 and Table 6.3.

Table 6.1: Sets in master problem.

$\boldsymbol{G}$	**Set of all pairings including disrupted pairings**
$\boldsymbol{P}$	Set of all real patterns
$\boldsymbol{P_g}$	Set of real patterns covering pairing g $\in G$
$\boldsymbol{S}$	Set of all synthetic patterns
$\boldsymbol{S_g}$	Set of synthetic patterns covering pairing g $\in G$

Table 6.2: Parameters in master problem.

$\boldsymbol{c_p}$	**Cost of pattern $\boldsymbol{p \in P}$**
$\boldsymbol{c_s}$	Cost of synthetic pattern s $\in S$
$\boldsymbol{l_p}$	Equals 1 if pattern $p \in P$ is legitimate, 0 otherwise

Table 6.3: Binary variables in master problem.

$\boldsymbol{x_p}$	Equals *1* if pattern $p \in P$ is used, *0* otherwise
$\boldsymbol{x_s}$	Equals *1* if synthetic pattern $s \in S$ is used, *0* otherwise

The sets, parameters and variables that are used in the mathematical model of sub problem as in Table 6.4, Table 6.5 and Table 6.6.

Table 6.4: Sets in sub problem.

$\boldsymbol{R}$	Set of all resources, union of aircraft and crew resources
$\boldsymbol{R_c}$	Set of resource type captain
$\boldsymbol{R_o}$	Set of resource type first officer
$\boldsymbol{R_a}$	Set of resource type cabin chief
$\boldsymbol{R_s}$	Set of resource type flight attendant
$\boldsymbol{R_{ac}}$	Set of resource type aircraft
$\boldsymbol{G}$	Set of all pairings including disrupted pairings

Table 6.5: Parameters in sub problem.

$\boldsymbol{c_{pr}^a}$	**Assignment cost of resource r $\in \boldsymbol{R}$ to pattern $\boldsymbol{p \in P}$**
$\boldsymbol{c_{pr}^s}$	Swapping cost of resource r $\in R$ to pattern $p \in P$
$\boldsymbol{c_{pr}^d}$	Deadheading cost of resource r $\in R$ to pattern $p \in P$
$\boldsymbol{c_{pr}^c}$	Cancelling cost of assignment of resource r $\in R$ to pattern $p \in P$
$\boldsymbol{c_{pr}^l}$	Delaying cost of assignment of resource r $\in R$ to pattern $p \in P$
$\boldsymbol{c_{pr}^p}$	Penalty cost of assignment r $\in R$ to pattern $p \in P$ if soft rules are broken
$\boldsymbol{n_{gc}}$	Resource type captain need for pairing g $\in G$
$\boldsymbol{n_{go}}$	Resource type first officer need for pairing g $\in G$
$\boldsymbol{n_{ga}}$	Resource type cabin chief need for pairing g $\in G$
$\boldsymbol{n_{gs}}$	Resource type flight attendant need for pairing g $\in G$
$\boldsymbol{\pi_g}$	Dual variables of restricted master problem covering the pairing g $\in G$ in the solution by only one pattern
$\boldsymbol{l_{pr}}$	Equals 1 if utilization of assigning resource r $\in R$ to pattern $p \in P$ is legitimate, 0 otherwise
$\boldsymbol{p_{pr}}$	Penalty or soft cost of assigning resource r $\in R$ to pattern $p \in P$

Table 6.6: Binary variables in sub problem.

x_{pr}	Equals *1* if resource r ∈ R is assigned to pattern $p \in P$, *0* otherwise

Each pattern for the flights which are not started is represented with binary variable x and index p. Each resource is symbolized with index r.

6.3.2 Model

The components of the model is explained below and the optimization flow can be seen in Figure 6.1.

6.3.2.1 Master problem

$$min \sum_{p \in P} c_p x_p + \sum_{s \in S} c_s x_s$$

$$s.t.$$

$$\sum_{p \in P_g} l_p x_p + \sum_{s \in S_g} x_s = 1, \forall g \in G \qquad (6.3)$$

$$x_p, x_s \in \{0, 1\} \qquad (6.4)$$

6.3.2.2 Restricted master problem

A subset of patterns is considered in the restricted master problem (RMP). The column generation needs information about values for the dual variables, therefore the requirement of integrality from RMP is removed.

$$min \sum_{p \in P} c_p x_p + \sum_{s \in S} c_s x_s$$

$$s.t.$$

$$\sum_{p \in P_g} l_p x_p + \sum_{s \in S_g} x_s = 1, \forall g \in G \qquad (\pi_g) \qquad (6.5)$$

$$x_p, \ x_s \geq 0 \qquad (6.6)$$

6.3.2.3 Subproblem or pricing problem

The subproblem will be solved for every pairing , $g \in G$.

$$c_p - (\pi_g) \qquad (6.7)$$

One of the key points in this model is the cost calculation which is done in the subproblem phase. The costs are:

- Assignment costs
- Swapping costs
- Deadheading costs
- Cancellation costs
- Delaying costs
- Penalties

$$c_p = \sum_{r \in R} c_{pr}^a + c_{pr}^s + c_{pr}^d + c_{pr}^c + c_{pr}^l + c_{pr}^p$$

$$min \sum_{r \in R} (c_{pr}^a + c_{pr}^s + c_{pr}^d + c_{pr}^c + c_{pr}^l + c_{pr}^p) x_{pr} - (\pi_g)$$

$$s.t.$$

$$\sum_{r \in R_c} l_{pr}\, x_{pr} = n_{gc} \tag{6.7}$$

$$\sum_{r \in R_f} l_{pr}\, x_{pr} = n_{go} \tag{6.8}$$

$$\sum_{r \in R_a} l_{pr}\, x_{pr} = n_{ga} \tag{6.9}$$

$$\sum_{r \in R_s} l_{pr}\, x_{pr} = n_{gs} \tag{6.10}$$

$$\sum_{r \in R_{ac}} l_{pr}\, x_{pr} = 1 \tag{6.11}$$

$$\sum_{r \in R} p_{pr}\, x_{pr} = 0 \tag{6.12}$$

$$x_{pr} \in \{0, 1\} \tag{6.13}$$

Our objective function is minimizing total assignment cost. In the restricted master problem the only difference is that for a subset of patterns, cost is being calculated. With constraint (6.3) or relaxed version of it with (6.5) we are sure that all pairings are covered with one legal pattern (synthetic patterns are assumed as legal). Constraint (6.4) and relaxed version of it (6.6) is about decision variables.

From constraint (6.7) to (6.12), (6.12) included is used for assigning correct number of legal resources to patterns for every resource group. Constraint (6.13) is a binary integer constraint.

The flow of the optimization model can be seen in Figure 6.1.

On the other hand, we will use the information provided by the deep learner as follows. Before the details, the learning algorithm is providing the list of recovery actions as seen below. All actions will be used while generating new columns can be seen in Table 6.7.

The listed actions are used in order to configure the columns faster than the classical column generation by either directly modifying the column itself or narrowing down the solution space. The resource or resource groups are filtered according to the information provided by the deep learning makes it possible to search for new columns in a narrower solution space which makes the search time smaller compared to the classical column generation techniques.

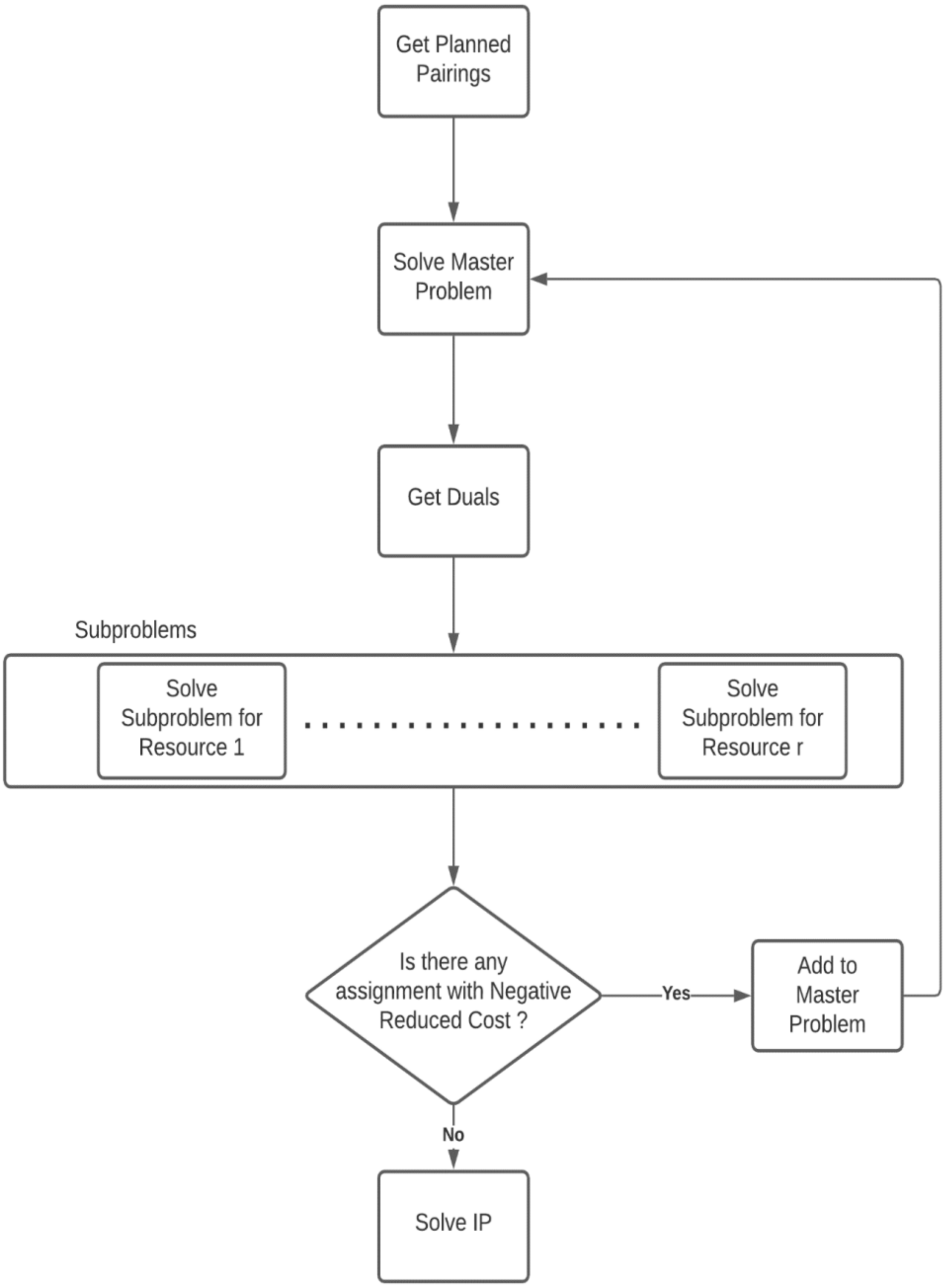

Figure 6.1: Optimization model flow.

Table 6.7: Optimization action by recovery action.

Action	Optimization Action
ASSIGN CABIN CREW	Preserve rest of the current pattern composition and solve for only cabin crew for assigning a new one.
ASSIGN COCKPIT CREW	Preserve rest of the current pattern composition and solve for only cockpit crew for assigning a new one.
CANCEL FLIGHT	Eliminate resource need for this flight and remove including pairing for regeneration of pairings.
DEADHEAD	Preserve current composition, do not assign any duty for specific crew till returning back to base and be ready for duty.
DEASSIGN CABIN CREW	Preserve rest of the current pattern composition and solve for only cabin crew for removing a current one.
DEASSIGN COCKPIT CREW	Preserve rest of the current pattern composition and solve for only cockpit crew for removing a current one.
DELAY FLIGHT	Use the pattern composition as it is just update the departure time and check for legality and solve if necessary.
SWAP AIRCRAFT	Preserve rest of the current pattern composition and solve for only aircraft and use the resource set except current one.
SWAP AIRCRAFT WITH AIRCRAFT HAS AVAILABLE SEAT	Preserve rest of the current pattern composition and solve for only aircraft and use the resource set with available seat except current one.
SWAP AIRCRAFT WITH AIRCRAFT HAS RESTBUNK OR RESTSEAT	Preserve rest of the current pattern composition and solve for only aircraft and use the resource set with rest bunt / seat except current one.
SWAP CREW WITH CREW AT SPECIFIC AIRPORT	Use crew set with crew who are at a certain station at the time of recovery in the solution steps.
SWAP CREW WITH CREW HAS CATEGORY QUALIFICATION	Use crew set with crew who have certain category qualification in the solution steps.

Table 6.7 (continued): Optimization action by recovery action.

Action	Optimization Action
SWAP CREW WITH CREW HAS HEALTH QUALIFICATION	Use crew set with crew who have certain health qualification in the solution steps.
SWAP CREW WITH CREW HAS LANDING QUALIFICATION	Use crew set with crew who have certain landing qualification in the solution steps.
SWAP CREW WITH CREW HAS LANGUAGE QUALIFICATION	Use crew set with crew who have certain language qualification in the solution steps.
SWAP CREW WITH CREW HAS LINE CHECK QUALIFICATION	Use crew set with crew who have certain line check qualification in the solution steps.
SWAP CREW WITH CREW HAS PASSPORT OR VISA	Use crew set with crew who have certain passport qualification in the solution steps.
SWAP CREW WITH CREW HAS SPECIAL AIRPORT QUALIFICATION	Use crew set with crew who have certain special airport qualification in the solution steps.
SWAP CREW WITH CREW HAS TECHNICAL QUALIFICATION	Use crew set with crew who have certain technical qualification in the solution steps.
SWAP CREW WITH CREW HAS VALID FLIGHT CREW DOCUMENTS	Use crew set with crew who have certain documents with valid dates in the solution steps.
SWAP CREW WITH CREW NOT FLOWN TO AIRPORT IN THE MONTH	Use crew set with crew who have not flown to airport in a month in the solution steps.
SWAP CREW WITH EXPERIENCED CREW	Use crew set with crew who have certain working experience level in the solution steps.
SWAP CREW WITH STANDBY CREW	Use crew set with crew who have duty in the standby / reserve duty set in the solution steps.

7. COMPUTATIONAL RESULTS AND DISCUSSION

As it has been already indicated, the main motivation behind this thesis is providing a new way of generating columns by using real world experiences.

Within the framework of the approach summarized, the mathematical model giving details above has been prepared and studies have been started to adapt it with Python language.

7.1 Technologies and Environment

All of the development including deep learning and optimizer is done with Python programming language with version 3. Python is a widely used programming language in academia and industry. It allows rapid development of simple but effective prototypes, thanks to data structures that are in line with the object-oriented programming paradigm. Not only prototypes, but also industrial applications with high success rates can be developed. Python can be used on many different platforms as a dynamically typed interpreted language (Python).

The deep learning model is developed with TensorFlow and Keras frameworks. TensorFlow is an end-to-end open source platform for machine learning applications development. In particular, it allows machine learning researchers to use the latest technologies in this field and to easily distribute the applications they develop. It has powerful plugins, flexible libraries and community resources that enable this. It has a big and active ecosystem (TensorFlow). Keras is a framework that provides a simpler API layer for TensorFlow which makes TensorFlow simpler and more productive. Keras provides a deep learning interface for models and optimizers while TensorFlow provides an infrastructure for differentiable programming using gradients and tensors (Keras).

The proposed optimizer uses a solver developed for solving large mathematical models Within the scope of this thesis, Gurobi Solver , one of many different solvers on the market, is selected. Gurobi is a powerful mathematical programming solver for linear, quadratic or mixed integer programming problems. After the users introduce the problems to the system through mathematical models, it is ensured that the best solution is found among many possibilities. Gurobi Optimizer is able to solve different types of convex and non-convex problems. The main ones are shown below (Gurobi).

- Linear programming (LP)
- Quadratic programming (QP)
- Mixed integer linear programming (MILP)
- Quadratic constrained programming (QCP)
- Mixed integer quadratic programming (MIQP)
- Mixed integer quadratic constrained programming (MIQCP)

The optimizer also uses parallelization techniques in order to reach the expected performance values and produce results in a shorter time. Although parallelization is a difficult subject for many reasons, Ray library, which is an advanced system that makes this work very easy in Python, has been used. Developed for distributed systems in Python, Ray provides easy distribution and scaling of all kinds of operations, from deep learning to the presentation of models in a live environment. Ray is an open source project with very rich plugins and integration possibilities with different systems.

The experiments of the optimizer consisting of technologies aforementioned above are run on a virtual server on Microsoft Azure environment with 8 CPUs and 32 GB memory running Ubuntu 20.04 operating system.

7.2 Architecture

The proposed solution is implemented by following object oriented programming paradigm with a modular architecture which can be seen in Figure 7.1. The critical modules that makes the solution available is the ones listed below.

- Data Processor: Data processor module is responsible for loading all the needed input data from the provided data files. The input includes information about, flights, pairings, current assignments, time limits, flight requirements, flight needs and crew qualifications.

- Disruption Generator: In order simulate the real world disruption this module is utilized where fixed initial data set is disrupted with the synthetic disruptions based on the rules and regulations.
- Cost Calculator: The cost of every assignment combination and resource is calculated by the cost calculator based on the cost parameters. The cost combination contains the items listed in the mathematical model.
- Legality Checker: Two different legitimacy checks take place within the scope of the model. One is the resource legitimacy check, which checks the suitability of assigning each source to a pairing, and the other is the pattern check, which performs an overall check of all selected patterns. The reason for the existence of the aforementioned second control, performs controls such as composition controls, because some controls are not meaningful during the individual resource control, but when the sources come together, the control becomes meaningful. All these are carried out within the framework of the rules that will be explained in the following sections. In summary, this module is the part that checks whether the resource assignments and compositions made based on the rules are legitimate or not.
- Inferencer: It is a deep learning model that predicts the recovery actions can be used for resolving the disruptions. As mentioned, it is a model that predicts what kind of recovery action should be taken specific to each flight and disruption by transforming the disruption into previously determined input features. The prediction is done by the deep learning model which was developed specifically for this job with AutoML techniques.
- Solver: This module works with the mathematical models specified in the previous sections by taking previously learned recovery actions and calculated costs into consideration and utilizing the column generation technique. Thanks to the solver, it aims to reach the solution by solving the sub-problems created as many as the number of pairs that disruptions occur. One of the main differences of this module is that unlike the rest of the system, it works in a sequential manner with a serial logic. While the other parts of the application benefit from parallelism to a significant extent, this part has been developed in this way because it has to prevent memory problems due to the use of common resources by sub-problems.

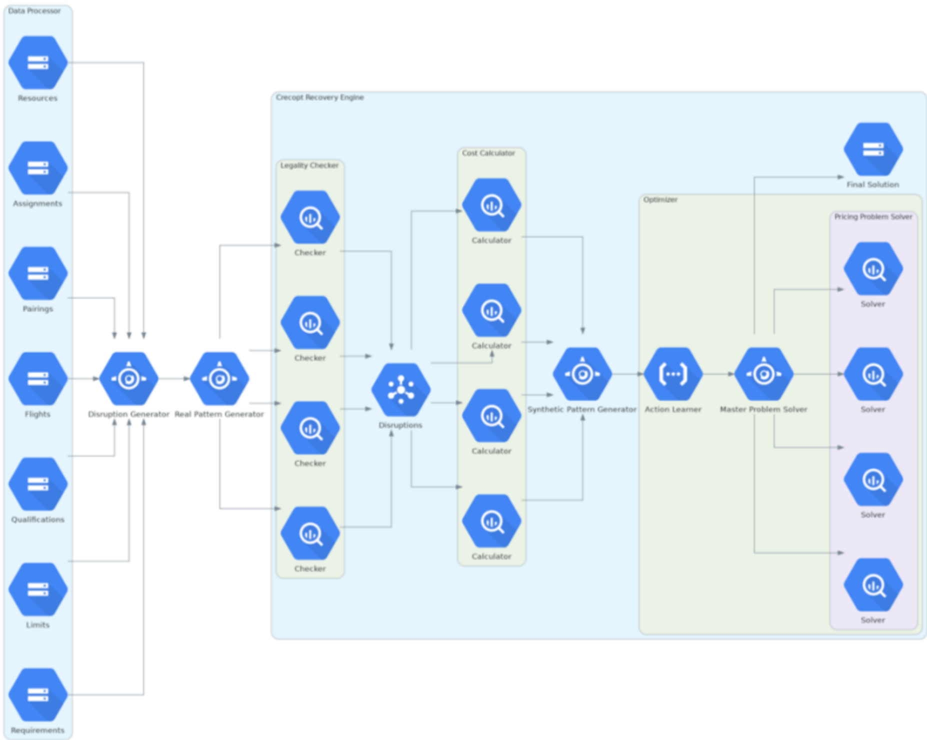

Figure 7.1: Solution architecture.

7.3 Data

As can be seen in the literature, hypothetical data is utilized in studies in order to reflect real world scenarios (Ilagan and Sy, 2017; Aranjuelo et al, 2021). During the benchmark runs performed within the scope of this thesis, synthetic data derived from real data of a large-scale airline is used and computational results are reported by using synthetic data.

Due to GDPR and similar regulations, it was not possible to use the masked data directly. As a result of processing the masked data with reverse engineering methods, a usable data set was obtained. The data set obtained within the scope of these studies is a subset of the received data set, not the whole. The main reason for this is that the relations between the data were broken during the preparation of the masked data. In particular, the most common problems were the dates which were not encoded with a consistent conversion and the fields used as primary keys. Due to all these reasons, it is not possible to use the whole data set and the parts generated by reverse engineering. The sub-dataset is consisting of approximately 500 daily flights of this large-scale airline where as the whole set has more than 1500 daily flights.

In general, the quality of the data used in model development and testing has a direct impact on the performance of the models. Therefore, generating synthetic data is a very important and important issue. On the other hand, if the effort required to produce synthetic data is not spent, more effort will be required to compensate for the negative performance that occurs (Aranjuelo et al, 2021).

7.4 Rules and Assumptions

Three types of checks are carried out within the scope of legality checks. The first of these is the qualification control, the second is the limit control and the third is the composition control.

Within the scope of qualification control, it is checked whether the resources assigned to specific pairing have the qualifications required for the the pairing. These checks include aircraft type qualification, language proficiency and capability of flying to critical airports. While all stations require English language knowledge, Russian, German, Spanish and Chinese language competencies are also checked according to the geography of station.

Limit control consists of daily, 7-days, 30-days, 90-days and 365-days forward and backward controls based on the date of the assignment. Within the scope of these controls, both block time and duty time are reviewed with sliding window approach. The relevant threshold values are derived specifically for the prototype from the values decided by the authorities.

Within the scope of composition control, it is aimed to check whether the resource as many as the need required by the pairing and need vector is legitimately fulfilled.

Legality control works with a completely parallel logic and uses all the resources of the infrastructure in which the system is deployed.

7.4.1 Rules

The rules handled within the scope of legality control and the explanations regarding these rules are listed below.

- Blocktime_Daily: The rule that controls daily block times.
- Blocktime_Monthly: The rule that controls monthly (30-day) block times.
- Blocktime_Yearly: The rule that controls yearly (365-day) block times.

- Blocktime_Weekly: The rule that controls weekly (7-day) block times.
- Blocktime_Quarterly: The rule that controls quarterly (90-day) block times.
- Dutytime _Daily: The rule that controls daily duty times.
- Dutytime _Monthly: The rule that controls monthly (30-day) duty times.
- Dutytime _Yearly: The rule that controls yearly (365-day) duty times.
- Dutytime _Weekly: The rule that controls weekly (7-day) duty times.
- Dutytime _Quarterly: The rule that controls quarterly (90-day) duty times.
- Acqual_320: The rule that controls qualification for Airbus A320 aircraft.
- Acqual_330: The rule that controls qualification for Airbus A330 aircraft.
- Acqual_340: The rule that controls qualification for Airbus A340 aircraft.
- Acqual_738: The rule that controls qualification for Boeing 738 aircraft.
- Acqual_777: The rule that controls qualification for Boeing 777 aircraft.
- Acqual_787: The rule that controls qualification for Boeing 787 aircraft.
- CAT3_320: The rule that controls qualification for CAT III instrument landing system for Airbus A320 aircraft.
- CAT3_330: The rule that controls qualification for CAT III instrument landing system for Airbus A330 aircraft.
- CAT3_340: The rule that controls qualification for CAT III instrument landing system for Airbus A340 aircraft.
- CAT3_738: The rule that controls qualification for CAT III instrument landing system for Boeing 738 aircraft.
- CAT3_777: The rule that controls qualification for CAT III instrument landing system for Boeing 777 aircraft.
- CAT3_787: The rule that controls qualification for CAT III instrument landing system for Boeing 787 aircraft.
- CAT3B_320: The rule that controls qualification for CAT III B instrument landing system for Airbus A320 aircraft.
- CAT3B_330: The rule that controls qualification for CAT III B instrument landing system for Airbus A 330 aircraft.
- CAT3B_340: The rule that controls qualification for CAT III B instrument landing system for Airbus A340 aircraft.
- CAT3B_738: The rule that controls qualification for CAT III B instrument landing system for Boeing 738 aircraft.

- CAT3B_777: The rule that controls qualification for CAT III B instrument landing system for Boeing 777 aircraft.
- CAT3B_787: The rule that controls qualification for CAT III B instrument landing system for Boeing 787 aircraft.
- Language_ENG: The rule that controls English language qualification.
- Language_DEU: The rule that controls German language qualification.
- Language_ESP: The rule that controls Spanish language qualification.
- Language_RUS: The rule that controls Russian language qualification.
- Language_CHA: The rule that controls Chinese language qualification.
- Language_FRA: The rule that controls French language qualification.
- C_Over_Resourced: The rule that controls whether more captains are assigned than needed.
- P_Over_Resourced: The rule that controls whether more first officers are assigned than needed.
- A_Over_Resourced: The rule that controls whether more cabin chiefs are assigned than needed.
- S_Over_Resourced: The rule that controls whether more flight attendants are assigned than needed.
- AC_Over_Resourced: The rule that controls whether more aircrafts are assigned than needed.
- C_Under_Resourced: The rule that controls whether less captains are assigned than needed.
- P_Under_Resourced: The rule that controls whether less first officers are assigned than needed.
- A_Under_Resourced: The rule that controls whether less cabin chiefs are assigned than needed.
- S_Under_Resourced: The rule that controls whether less flight attendants are assigned than needed.
- AC_Under_Resourced: The rule that controls whether less aircrafts are assigned than needed.

7.4.2 Assumptions

Generally, in order to reach a solution in an acceptable time frame, certain assumptions are taken into consideration in building crew schedules such as each flight will be operated as planned (Novianingsih, 2015). The prototype prepared in the context of the proposed optimization model is also based on some assumptions. One of these assumptions is that all certifications are valid for all team members. In real life, some certificates are only used for cockpit members, while some certificates are only used for the cabin. Cockpit or cabin side certificates also vary according to the classes in the crew vector. Within the scope of the prototype, it was assumed that the entire crew should have all the certificates required for that flight. In other words, certifications such as ILS certifications, which cabin crews do not have to or cannot obtain in real life, are also defined for cabin crew within the scope of this prototype.

On the other hand, another assumption is valid for standby teams. These teams are positioned as home standby, as is practiced in some airlines, as in the airline whose data is used within the scope of this study. Within the framework of this assumption, since the relevant teams were waiting at their homes during the Standby duty, the duty and block times were also accepted as zero. Therefore, the related duty periods in limit calculations have been developed to start after the flight assignment.

7.5 Parameters

Two different parameter groups were used in the test runs of the optimization model. These are the general parameters that control the operation of the optimization, which remain constant throughout all the runs, and the runtime parameters whose values are changed between runs.

Optimization general parameters, which are fixed for all runs, include flight and duty time limit values, fixed and varying resource costs on a flight basis, the cost of synthetic patterns. The parameters that take the values see in the Table 7.1.

The runtime parameters changed for every optimization run are the parameters including the number of flights to be disrupted by the simulator and the length of the recovery window. The sets of runtime parameters for the runs are shown in the Table 7.1 below.

Table 7.1: Optimizer global parameters.

Category	Group	Parameter	Value
Thresholds	Crew Dutytime	Daily	18 hours
		Weekly	60 hours
		Monthly	190 hours
		Quarterly	500 hours
		Yearly	1500 hours
	Crew Blocktime	Daily	16 hours
		Weekly	50 hours
		Monthly	100 hours
		Quarterly	250 hours
		Yearly	900 hours
	Aircraft Dutytime	Daily	22 hours
	Aircraft Blocktime	Daily	20 hours
Cost	Salary	Captain	10 monetary unit
		First Officer	5 monetary unit
		Cabin Chief	2 monetary unit
		Flight Attendant	1 monetary unit
Cost	Allowances	Captain	0.1 monetary unit
		First Officer	0.05 monetary unit
		Cabin Chief	0.02 monetary unit
		Flight Attendant	0.01 monetary unit
Penalties	Pattern	Synthetic Penalty	100,000,000 monetary unit
		Cancellation	1000 monetary unit
		Delaying	500 monetary unit
	Resource	Unavailable Resource Penalty	100 monetary unit

Table 7.2: Optimizer runtime parameters.

Parameter	Values
Number of Disrupted Flights	1 – 5 – 10 – 25 – 50 flights
Length of Recovery Window	1 – 2 – 3 days

7.6 Runs & Results

The test data used in the benchmark runs is filtered from the dataset used for training of deep learning model. The data covers subset of flights from June 2018 and the all pairings are created with a simple heuristic for the benchmark runs because the pairing information provided by the airline is masked and can not be regenerated with the reengineering process. The benchmark data produced from the masked real world data covers six different fleets and almost 13.500 crew members and more than 400 aircrafts.

Every run includes a simulation step where disruptions are simulated based on the number of disrupted flights parameter. According to the number of disrupted pairings synthetic patterns are created in order to be able to initialize the column generation with an initial feasible solution. Since the synthetic patterns are very costly (big-M) compared to the other costs modeled in the optimizer, they are removed from the solution as the new improving solutions are produced by the subproblem model. The column generation procedure is using the solution provided by the former iteration. If the iterations are not improving enough, the run is stopped and the current optimal value with run time is reported.

Table 7.3 summarizes the dimensions of the dataset used based on the recovery window. The dimensions include the number of duty assignments. Furthermore, the recovery window starts form the beginning of the June 2018 and covers the days as long as the recovery window provided. That means recovery window consists of 1st, 2nd and 3rd days of June 2018.

Table 7.3: Optimizer runtime dataset dimensions.

Length of Recovery Window	# of Assignments
1 Day	7257
2 Days	9317
3 Days	11025

In order to compare the results, another version of the optimizer is developed which does not use recovery actions. It utilizes the mathematical model and its results is used as a baseline for comparison. The version which does not use the actions provided by the deep learning model is called as “Static Optimizer” and the one which utilizes deep learning technics is called as “Dynamic Optimizer”.

The optimization model has been run for 10 times for every combinations of the parameters shown in Table 7.2. As it can be seen, total number of runs are 150. The optimal value scores and run times are reported at the end of every run.

The disruption generator generates disruptions for different number of flights for every run of the static and dynamic optimizers that means these systems are solving the same number of disrupted flights but working on different disruption sets in every run.

The results of the runs are listed in the Table 7.4 for 1 day recovery window, Table 7.5 for 2 days recovery window and Table 7.6 for three days recovery window.

Table 7.4: Optimization results for 1 day long recovery window.

# of Disrupted Flights	**Optimizer Type**	**Average Optimal Value**	**Average Run Time**
1	Static	5402.82	65.32
	Dynamic	5501.04	63.32
5	Static	5884.77	428.53
	Dynamic	6760.80	368.43
10	Static	6855.30	966.57
	Dynamic	7535.56	731.68
25	Static	10619.30	2501.04
	Dynamic	10342.94	1851.04
50	Static	16430.49	4904.57
	Dynamic	17792.31	3763.25

Table 7.5: Optimization results for 2 days long recovery window.

# of Disrupted Flights	Optimizer Type	Average Optimal Value	Average Run Time
1	Static	5402.63	75.97
	Dynamic	5597.43	62.91
5	Static	6466.34	460.23
	Dynamic	5787.65	385.87
10	Static	7241.09	1014.88
	Dynamic	7150.06	758.98
25	Static	10035.87	2501.74
	Dynamic	11026.01	1963.78
50	Static	15361.98	5064.82
	Dynamic	18466.77	4045.18

Table 7.6: Optimization results for 3 days long recovery window.

# of Disrupted Flights	Optimizer Type	Average Optimal Value	Average Run Time
1	Static	5595.32	69.74
	Dynamic	5403.84	66.27
5	Static	5789.99	469.83
	Dynamic	6718.90	371.18
10	Static	7145.78	1084.69
	Dynamic	6761.19	817.55
25	Static	9753.53	2611.37
	Dynamic	11745.12	2081.10
50	Static	15744.21	5102.58
	Dynamic	16102.42	4111.08

As it has been already mentioned in the hypothesis of the study, a fast but acceptably optimality preserving optimizer is investigated in the scope of this research. We can easily see that our model called as Dynamic Optimizer is far better than the reference model known as Static Optimizer in terms of run time.

Although, it is expected that reference model will have better optimality since it is the exact solution for our problem, there are some results where Dynamic Optimizer is better than the Static Optimizer. This is mainly due to the fact that both optimizers are working on different disruption data sets for same parameter combinations. Sometimes, the disruption generator may generate a disruption set which is costly compared to other one. And if this is the case for the Static Optimizer than the result maybe costlier than the Dynamic Optimizer. However, run time results clearly show the fast solution capability of Dynamic Optimizer.

The optimal value comparison is shown in Figure 7.2 and the run time comparison is shown in Figure 7.3. As we further analyze the results, we found out that number of disrupted flights have stronger relationship with average optimal and run time values compared the relationship with recovery window. The correlation of hyperparameters and results are shown in Figure 7.2.

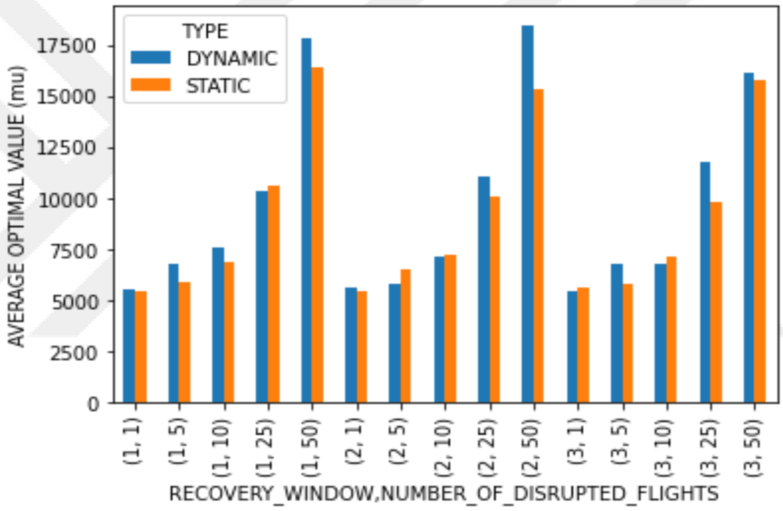

Figure 7.2: Static and dynamic average optimal value comparison.

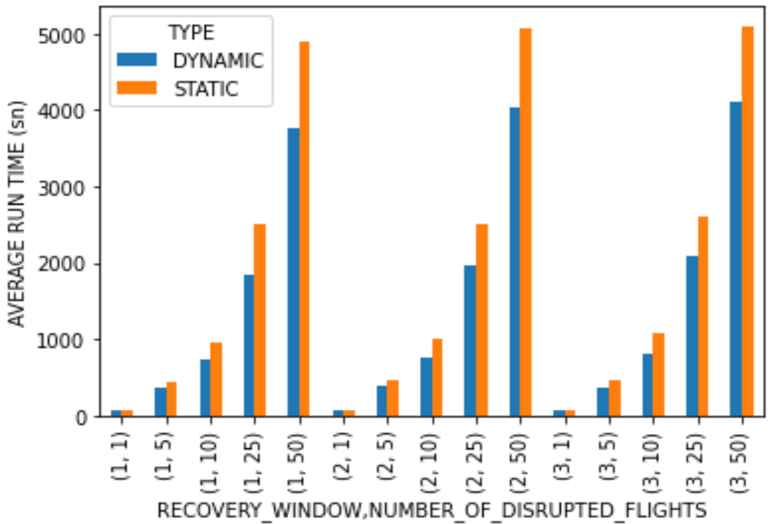

Figure 7.3: Static and dynamic average run time comparison.

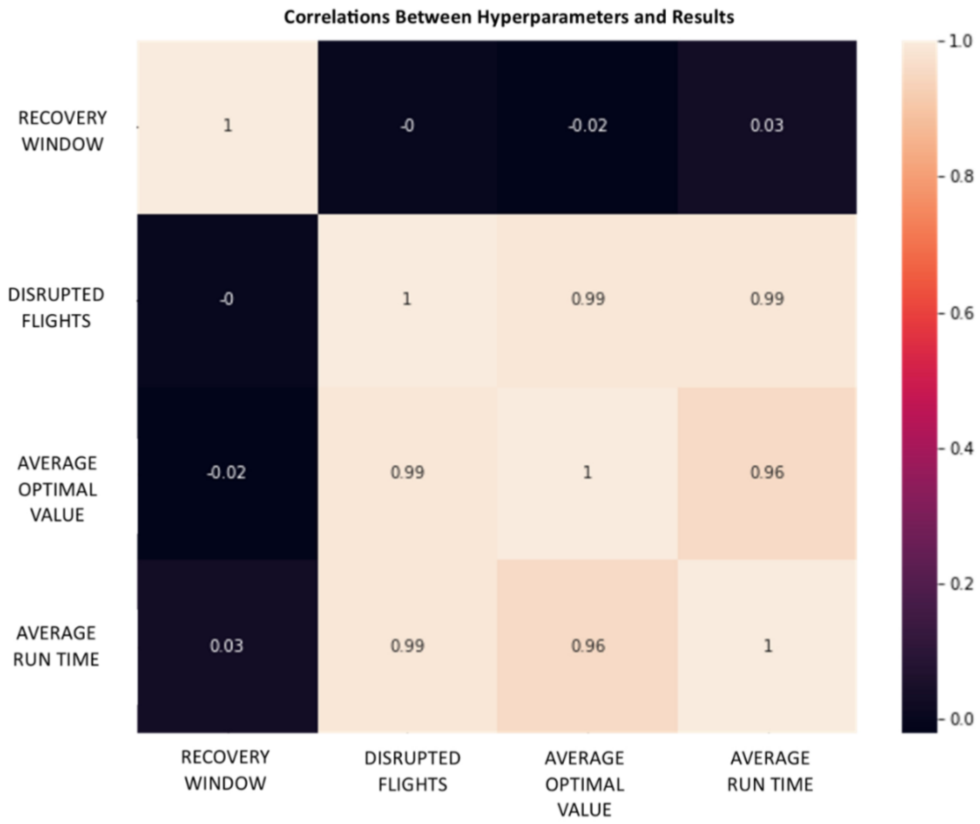

Figure 7.4: Correlations between hyperparameters and results.

In order to test the results, we have set a null hypothesis H_0 which simply indicates the optimization model has not benefited from the learning mechanism and alternative hypothesis H_1 which tells that the optimizer has benefited from the learning mechanism in terms of speed. Since calculated p-value is 0.000960 and smaller than significance (0.05), we reject the null hypothesis. We can say that the optimizer hasbenefited from the learner in terms of runtime.

7.7 Discussion

In this study, we aimed to investigate a new model for the solution of crew recovery problem by utilizing machine learning and combinatorial optimization. Overall, the findings of the study provide valuable insights for a fast optimization model which can be used for real world problems. However, there were several difficulties encountered during the research process and some missing parts that need to be addressed.

One of the main difficulties we faced was the limited availability of data for building machine learning model. Due to the data protection regulation, masked data set is provided. Masking data can often result in a biased sample, which can lead to inaccurate conclusions. This occurs because masking data typically involves removing certain relationships from the data set, which can result in an unrepresentative sample.

Masking data can also limit the generalizability of the findings. This is because the masked data set may not accurately represent the population of interest, which can make it difficult to generalize the results to the broader population (Ajayi and Adebiyi, 2014). It can limit the ability to explore complex relationships between variables. This is because masking data often involves removing observations with missing data, which can result in a loss of information that may be important for exploring complex relationships (Groenwold and Dekkers, 2020).

In terms of missing parts, there were some aspects of the study that were not fully explored. For example, we did not investigate all the rules which is more than 400. In order to simplify the scope of the problem and reduce the needed computational power, only selected rules are taken into consideration.

In conclusion, while this study provides valuable insights for a fast crew recovery solution which can be applied to the real world problem, there were several difficulties encountered during the research process and some missing parts that need to be addressed in future research. By addressing these limitations, we can not only gain a more comprehensive understanding of the effects of hybridization of machine learning and optimization models, but also improve the generalizability of the findings.

8. CONCLUSION AND FUTURE RESEARCH

In this study, crew recovery problem is investigated and a new type of solution method based on deep learning and column generation technique is proposed. Crew recovery problem is a complex problem to solve due to the characteristic which requires to be solved in a narrow timeframe in order to be applicable for operation. For this purpose, many solution approaches have been proposed in the literature which are mainly focusing on decreasing the solution time of the problem.

Crew disruption data such as crew absence, aircraft technical malfunctions of a large scale European airline is analyzed and used in this study. From disruption data, recovery actions are generated and crew recovery problem is solved with an optimization model based on recovery actions. The proposed crew recovery optimization model has an intelligently reduced solution space leading to effective solutions.

The initial hypothesis of the study as it has already been indicated in the former sections is finding a model that can provide satisfactory results in terms of both quality and speed. As it can be seen from the results, we have been satisfied with the optimal value and speed of our model which is far better than the reference model in term of run time while preserving the optimality at an acceptable level.

The proposed solution method can be used by airline companies to solve crew recovery problem effectively and efficiently. This study has contributed to the existing literature by providing a new mathematical model and approximation method. The integration of deep learning and optimization methods can be utilized for other complex optimization problems in the aviation sector. This study has also provided valuable insights for future studies in the area of crew recovery problem.

Although, the proven results are the main outcomes of the study, there are three more outcomes which are very important for the literature especially for the upcoming studies. These outcomes are:

- A new approximation method for spectral clustering

- New mathematical model for crew recovery problem
- Deep learning guided optimization method

It is vital for every machine learning study to have proper and high quality data set. The main drawback of the text based data sets is labeling which are very hard and requires sophisticated methods for labelling (Zhang et al, 2021). We have proposed a new spectral clustering approximation for high dimensional data structure such as texts in order to find the clusters easily and with low computational cost. The clusters will be base for labeling purposes. Without the approximation, it is not possible to process large amount of data with acceptable CPU and memory resources (Yıldırım and Özdoğan, 2011; Suribabu et al, 2023).

Due to the complexity of the recovery problems, especially crew recovery problem, it is not possible to find many mathematical models for them. We have developed a new mathematical model for the problem which is the second important outcome of the study. Since the model utilizes decomposition techniques, it is not only efficient for small sized problems but also the large ones.

Integration of deep learning and optimization methods is gaining momentum in the academic sphere and some practical applications are also seen in the industry for certain problems. We are one of the early adopters of this approach in the aviation space and apply this approach to the one of the most sophisticated problems the sector is facing today. The study has highlighted the importance of reducing the solution space for effective optimization of crew recovery problem. The study has also shown that the integration of deep learning and optimization methods can provide significant benefits in terms of search space and optimality. The development of a new mathematical model has shown the importance of innovative and creative approaches to solving complex optimization problems. The information provided by the inference mechanism helps optimizer to search the space intelligently and faster with an acceptable optimality loss. Since the crew recovery problem looks for faster and applicable results rather than optimal ones, this approach will provide many benefits to the solution approaches focusing on the problem.

This study has some limitations, such as the use of data from a single airline and the focus on crew recovery problem only. Even if the source of data is single airline, the characteristics that are derived from the data is representing most of the cases an airline may face today. Due to the wide range of aircraft types, large flight network and high number of flight crew, this study will serve as a basis for further prediction and recovery solutions.

We believe that more computational power will lead to new insights about the data set but the methodology proposed in this study will be a basis for further researches.

It can be easily seen from the latest studies in the literature, new methods introduce opportunities to utilize new techniques to solve crew recovery problem. Especially, since graphical neural networks, which also benefit from reinforcement learning techniques, exhibit strong learning and generalization capabilities in capturing information that is hidden in network topologies which is not easy to extract (Jiang, 2021), we strongly believe that it will be one of the mainstream focus area for further studies.

Future studies can also build upon this study by utilizing larger and more diverse datasets to test the proposed solution method. Other optimization problems in the aviation sector can be explored using the integration of deep learning and optimization methods. Studies can be conducted to compare the proposed solution method with other existing methods in terms of solution quality and efficiency.

Finally, addressing crew recovery problems in the airline industry not only enhances operational efficiency as it has been discussed in this thesis but also contributes significantly to sustainability. Crew recovery, which involves managing unforeseen events plays a pivotal role in maintaining the reliability of flight schedules. By developing effective solutions for crew recovery, airlines can minimize disruptions, reduce delays, and optimize crew utilization. This, in turn, leads to a more sustainable operation by improving on-time performance and decreasing fuel consumption associated with delays and inefficiencies. Additionally, efficient crew recovery practices contribute to enhanced employee satisfaction and well-being, fostering a positive work environment within the airline industry. Consequently, by prioritizing solutions for crew recovery problems, airlines can establish a more sustainable and resilient framework, ensuring both operational success and environmental responsibility which is another path for future studies.

REFERENCES

Abdelghany, A., Ekollu, G., Narasimhan, R. and Abdelghany, K. (2004). A proactive crew recovery decision support tool for commercial airlines during irregular operations. *Annals of Operations Research*, *127(1-4)*, 309-331.

Afzalan, M. and Jazizadeh, F. (2019). An automated spectral clustering for multi-scale data. *Neurocomputing*, *347*, 94–108.

Agarwal, Y.K. (2002). Design of capacitated multicommodity networks with multiple facilities. *Operations Research*, *50(2)*, 333-344.

Aggarwal, C.C., Hinneburg, A. and Keim, D.A. (2001). On the surprising behavior of distance metrics in high dimensional space. *Proceedings of ICDT 2001: International Conference on Database Theory*, 420–434, London, UK, January 4–6, 2001.

Ajayi, O. O. and Adebiyi, T. O. (2014). Application of Data Masking in Achieving Information Privacy. *IOSR Journal of Engineering*, *4(2)*, 13-21.

Alami, N., Meknassi, M., En-nahnahi, N., El Adlouni, Y. and Ammor, O. (2021) Unsupervised neural networks for automatic arabic text summarization using document clustering and topic modeling. *Expert Systems with Applications*, *172*, 114652.

Alguliyev, R.M., Aliguliyev, R.M. and Sukhostat, L.V. (2021) Parallel batch k-means for big data clustering. *Computers & Industrial Engineering*, *152*, 107023.

Almeida, H., Guedes, D., Meira, W. and Zaki, M.J. (2011). Is there a best quality metric for graph clusters? *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part I*, 44–59, Athens, Greece, September 5 - 9, 2011.

Alshammari, M. and Takatsuka, M. (2019). Approximate spectral clustering with eigenvector selection and self-tuned k. *Pattern Recognition Letters*, *122*, 31-37.

Alvelos, F., Sousa, A. D. and Santos, D. (2013). Combining column generation and metaheuristics. *In Hybrid metaheuristics*. Springer: Berlin, Heidelberg.

Andrews, N.O. and Fox, E.A. (2007). Recent developments in document clustering. Retrieved Jan 16, 2021 from https://vtechworks.lib.vt.edu/bitstream/handle/10919/19473/docclust.pdf

Aranjuelo, N., García, S., Loyo, E., Unzueta, L. and Otaegui, O. (2021). Key strategies for synthetic data generation for training intelligent systems based on people detection from omnidirectional cameras. *Computers & Electrical Engineering*, *92*, 107105.

Assent, I. (2012). Clustering high dimensional data. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2(4)*, 340–350.

Aydiner, O.S. (2019). *Optimizing airline operations under uncertainty* (Doctoral dissertation). Retrieved from http://repository.bilkent.edu.tr/bitstream/handle/11693/52081/10260120_OzgeSafakAydıner.pdf

Balcan, M.F., Dick, T., Sandholm, T. and Vitercik, E. (2018). Learning to branch. *Proceedings of the 35th International Conference on Machine Learning* (Vol *80*, pp344-353) Stockholm, Sweden, July 10-15, 2018.

Barnhart, C., Belobaba, P. and Odoni, A.R. (2003). Applications of operations research in the air transport industry. *Transportation Science*, *37(4)*, 368–391.

Beauchemin, M. (2015). On affinity matrix normalization for graph cuts and spectral clustering. *Pattern Recognition Letters*, *68*, 90–96.

Bengio, Y., Lodi, A. and Prouvost, A. (2020). Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, *290(2)*, 405-421.

Benson, A.R., Gleich, D.F. and Leskovec, J. (2016). Higher-order organization of complex networks. *Science*, *353*, 163–166.

Berger, D., Gendron, B., Potvin, J.Y., Raghavan, S. and Soriano, P. (2000). Tabu search for a network loading problem with multiple facilities. *Journal of Heuristics*, *6(2)*, 253-267.

Bochie, K., Gilbert, M. S., Gantert, L., Barbosa, M. S., Medeiros, D. S. and Campista, M. E. M. (2021). A survey on deep learning for challenged networks: Applications and trends. *Journal of Network and Computer Applications*, *194*, 103213.

Borgwardt, S. and Patterson, S. (2022). A column generation approach to the discrete barycenter problem. *Discrete Optimization*, *43*, 100674.

Boughanem, M., Berrut, C., Mothe, J. and Soule-Dupuy, C. (2009). Advances in information retrieval. *ECIR 2009: Proceedings of 31th European Conference on IR Research*, Toulouse, France, April 6-9, 2009.

Brownlee, J. (2020), *4 Types of Classification Tasks in Machine Learning*. Retrieved April 18, 2022, from https://machinelearningmastery.com/types-of-classification-in-machine-learning/

Cao, J., Chen, P., Dai, Q. and Ling, W. K. (2014). Local information-based fast approximate spectral clustering. *Pattern Recognition Letters*, *38*, 63-69.

Castro, A.J.M., Rocha, A.P. and Oliveira, E. (2014**).** A new approach for disruption management in airline operations control. *Studies in Computational Intelligence, 562*.

Chang, S.C. (2012). A duty-based approach in solving the aircrew recovery problem. *Journal of Air Transport Management*, *19*, 16-20.

Charikar, M., Naamad, Y., Rexford, J. and Zou, X.K. (2018). Multi-commodity flow with in-network processing. *Proceedings of the International Symposium on Algorithmic Aspects of Cloud Computing*. Helsinki, Finland, August 20–21, 2018.

Chen, C.H. and Chou, J.H. (2016). Multiobjective optimization of airline crew roster recovery problems under disruption conditions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47(1)*, 133-144.

Cheng, M.Y., Kusoemo, D. and Gosno, R.A. (2020). Text mining-based construction site accident classification using hybrid supervised machine learning. *Automation in Construction*, *118*, 103265.

Ciriani, T.A., Fasano, G., Gliozzi, S. and Tadei, R. (2013). Operations research in space and air (Vol. *79*). Retrieved from https://link.springer.com/book/

Clausen, J., Larsen, J., Larsen, A. and Hansen, J. (2001). Disruption management-operations research between planning and execution. *OR/MS Today*, *28(5)*, 40-43.

Clausen, J., Larsen, A., Larsen, J. and Rezanova, N.J. (2010). Disruption management in the airline industry - Concepts, models and methods. *Computers & Operations Research*, *37(5)*, 809-821.

Cohen-Shapira, N. and Rokach, L. (2021). Automatic selection of clustering algorithms using supervised graph embedding. *Information Sciences*, *577*, 824-851.

Copeland, M. (2016). *What's the Difference Between Deep Learning Training and Inference?* Retrieved April 18, 2022, from https://blogs.nvidia.com/blog/2016/08/22/difference-deep-learning-training-inference-ai/

Correa, C.D. and Lindstrom, P.: Locally-scaled spectral clustering using empty region graphs. KDD12: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1330–1338, Beijing, China, August 12 - 16, 2012.

Dai, H., Khalil, E.B., Zhang, Y., Dilkina, B. and Song, L. (2017). Learning combinatorial optimization algorithms over graphs. Retrieved July 30, 2021, from https://arxiv.org/pdf/1704.01665.pdf

Dana, K.J. (2018). Computational Texture and Patterns: From Textons to Deep Learning. Williston: Morgan & Claypool.

de Armas, J., Cadarso, L., Juan, A.A. and Faulin, J. (2017). A multi-start randomized heuristic for real-life crew rostering problems in airlines with work-balancing goals. *Annals of Operations Research*, *258(2)*, 825-848.

de Castro, A.J.M. (2007). *Designing a multi-agent system for monitoring and operations recovery for an airline operations control centre* (Master's thesis). Retrieved from https://www.researchgate.net/publication/37649681_Designing_a_Multi-Agent_System_for_Monitoring_and_Operations_Recovery_for_an_Airline_Operations_Control_Centre

Deveci, M. and Demirel, N.C. (2018). A survey of the literature on airline crew scheduling. *Engineering Applications of Artificial Intelligence, 74*, 54–69.

Demassey, S., Pesant, G. and Rousseau, L. M. (2005). Constraint programming based column generation for employee timetabling. *In International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming,* 140-154, Springer: Berlin, Heidelberg.

Dong, S., Wang, P. and Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, 100379.

Duan, L., Ma, S., Aggarwal, C. and Sathe, S. (2021). Improving spectral clustering with deep embedding, cluster estimation and metric learning. *Knowledge and Information Systems*, *63(3)*, 675–694.

Dunn, S. and Wilkinson, S.M. (2016). Increasing the resilience of air traffic networks using a network graph theory approach. *Transportation Research Part E: Logistics and Transportation Review, 90*, 39–50.

El-Bhissy, K., El-Faleet, F. and Ashour, W.M. (2014). Clustering using optimized Gaussian kernel function. *International Journal of Artificial Intelligence and Application for Smart Devices*, *2*, 41-56.

Elshawi, R., Maher, M. and Sakr, S. (2019). Automated machine learning: State-of-the-art and open challenges. Retrieved July 30, 2021, from https://arxiv.org/pdf/1906.02287.pdf

Elsken, T., Metzen, J.H. and Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, *20(55)*, 1-21.

Eurocontrol. (2019). All-Causes Delay and Cancellations to Air Transport in Europe-2019. Retrieved July 30, 2021, from https://www.eurocontrol.int/publication/all-causes-delay-and-cancellations-air-transport-europe-2019

Eurocontrol. (2020). FLY AI Report. Retrieved Jan 16, 2021 from https://www.eurocontrol.int/publication/fly-ai-report

Famili, A., Shen, W. M., Weber, R. and Simoudis, E. (1997). Data preprocessing and intelligent data analysis. *Intelligent data analysis*, *1(1)*, 3-23.

FastText. (2021). Library for efficient text classification and representation learning. [Software] Retrieved July 30, 2021. Available from https://fasttext.cc

Favati, P., Lotti, G., Menchi, O. and Romani, F. (2020). Construction of the similarity matrix for the spectral clustering method: Numerical experiments. *Journal of Computational and Applied Mathematics*, *375*, 112795.

France, S.L., Carroll, J.D. and Xiong, H. (2012). Distance metrics for high dimensional nearest neighborhood recovery: Compression and normalization. *Information Sciences*, *184(1)*, 92–110.

Frangioni, A. and Gendron, B. (2009). 0–1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, *157(6)*, 1229-1241.

Frangioni, A. and Gendron, B. (2013). A stabilized structured Dantzig–Wolfe decomposition method. *Mathematical Programming*, *140(1)*, 45-76.

Frye, M., Mohren, J. and Schmitt, R. H. (2021). Benchmarking of Data Preprocessing Methods for Machine Learning-Applications in Production. *Procedia CIRP*, *104*, 50-55.

Gamache, M., Soumis, F., Marquis, G. and Desrosiers, J. (1999). A column generation approach for large-scale aircrew rostering problems. *Operations Research*, *47(2)*, 247-263.

Gasse, M., Chételat, D., Ferroni, N., Charlin, L. and Lodi, A. (2019). Exact combinatorial optimization with graph convolutional neural networks. Retrieved July 30, 2021, from https://arxiv.org/pdf/1906.01629.pdf

Gopal, M. (2018). Applied Machine Learning. Retrieved from http://books.google.com/books

Goverde, R.M.P. (2005). *Punctuality of railway operations and timetable stability analysis* (Doctoral dissertation). Retrieved from https://repository.tudelft.nl/islandora/object/uuid:a40ae4f1-1732-4bf3-bbf5-fdb8dfd635e7/datastream/OBJ/download

Groenwold, R. H. and Dekkers, O. M. (2020). Missing data: the impact of what is not there. *European journal of endocrinology*, *183(4)*, E7-E9.

Guimera, R., Mossa, S., Turtschi, A. and Amaral, L.A.N. (2005). The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, *102*, 7794–7799.

Guo, Y. (2005). *Decision support systems for airline crew recovery* (Doctoral dissertation). Retrieved from http://deposit.ddb.de/cgi-bin/dokserv?idn=976568284&dok_var=d1&dok_ext=pdf&filename=976568284.pdf

Guo, Y., Suhl, L. and Thiel, M.P. (2005). Solving the airline crew recovery problem by a genetic algorithm with local improvement. *Operational Research*, *5(2)*, 241.

Gurobi. The Fastest Solver. Retrieved February 4, 2022, from https://www.gurobi.com

Gupta, P., Gasse, M., Khalil, E.B., Kumar, M.P., Lodi, A. and Bengio, Y. (2020). Hybrid models for learning to branch. Retrieved July 30, 2021, from https://arxiv.org/pdf/2006.15212.pdf

He, X., Zhao, K. and Chu, X. (2021). AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, *212*, 106622.

Hewitt, M. and Frejinger, E. (2020). Data-driven optimization model customization. *European Journal of Operational Research*, *287*, 438–451.

Hoeben, N.J.M. (2017) *Dynamic Crew Pairing Recovery* (Master's thesis). Retrieved from https://repository.tudelft.nl/islandora/object/uuid:b7077ebe-6404-425b-beff-14d6cf5bb0a0?collection=education

Huiskamp, K. (2016). *Using Column Generation for the Bus Line Planning Problem* (Master's thesis). Retrieved from https://thesis.eur.nl/pub/37903/

IATA, IATA Forecast Predicts 8.2 billion Air Travelers in 2037**.** (2018). Retrieved July 30, 2021, from https://www.iata.org/pressroom/pr/Pages/2018-10-24-02.aspx

Ilagan, I.F. and Sy, C. (2017). Robustness through possible crew swaps in airline operations. Proceedings of *2017 IEEE International Conference on Industrial Engineering and Engineering Management*, 11-15, Singapore, December 10-13, 2017.

Ionescu, L., Gwiggner, C. and Kliewer, N. (2016). Data analysis of delays in airline networks. *Business & Information Systems Engineering*, *58*, 119–133.

Ionescu, L. and Kliewer, N. (2011). Increasing flexibility of airline crew schedules. *Procedia-Social and Behavioral Sciences*, *20*, 1019-1028.

Ionescu, L., Kliewer, N. and Schramme, T. (2011) A comparison of recovery strategies for crew and aircraft schedules. *Operations Research Proceedings*, *2010*, 269–274.

Janani, R. and Vijayarani, S. (2019). Text document clustering using spectral clustering algorithm with particle swarm optimization. *Expert Systems with Applications*, *134*, 192–200.

Jiang, W. (2021). Graph-based Deep Learning for Communication Networks: A Survey. Retrieved February 2, 2022, from https://arxiv.org/pdf/2106.02533.pdf

Jimenez Serrano, F.J. and Kazda, A. (2017). Airline disruption management: yesterday, today and tomorrow. *Transportation Research Procedia, 28*, 3–10.

Jin, H., Song, Q. and Hu, X. (2019). Auto-keras: An efficient neural architecture search system. *KDD19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1946-1956), Anchorage, USA, August 4 - 8, 2019.

Keras. Keras is an API designed for human beings, not machines.. Retrieved February 4, 2022, from https://keras.io

Khaksar, H. and Sheikholeslami, A. (2019). Airline delay prediction by machine learning algorithms. *Transactions on Civil Engineering (A)*, *26(5)*, 2689–2702.

Khallaf, R. and Khallaf, M. (2021). Classification and analysis of deep learning applications in construction: A systematic literature review. *Automation in Construction*, *129*, 103760.

Kohl, N., Larsen, J. A. and Larsen, A., R., S., T. (2007)**.** Airline disruption management perspectives, experiences and outlook. *Journal of Air Transport Management, 13*, 149–162.

Krauss, J., Pacheco, B. M., Zang, H. M. and Schmitt, R. H. (2020). Automated machine learning for predictive quality in production. *Procedia CIRP*, *93*, 443-448.

Kumhar, S.H., Kirmani, M.M., Sheetlani, J. and Hassan, M. (2021). Word embedding generation for urdu language using word2vec model. *Materials Today: Proceedings*.

Le, M. and Sun, L. (2011). Optimal airline crew recovery considering flight time constraint and paring rule. *Proceedings of 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE),* 118-121, Changchun, China, December 16-18, 2011.

Lee, J., Lee, K., and Moon, I. (2022). A reinforcement learning approach for multi-fleet aircraft recovery under airline disruption. *Applied Soft Computing*, *129*, 109556.

Lettovský, L., Johnson, E.L. and Nemhauser, G.L. (2000). Airline crew recovery. *Transportation Science*, *34(4)*, 337-348.

Li, Z., Chen, Q. and Koltun, V. (2018). Combinatorial optimization with graph convolutional networks and guided tree search. Retrieved July 30, 2021, from https://arxiv.org/pdf/1810.10659.pdf

Li, Y., Clarke, J.P. and Dey, S.S. (2021). Using submodularity within column generation to solve the flight-to-gate assignment problem. *Transportation Research Part C: Emerging Technologies*, *129*, 103-217.

Li, M., Lian, X.C., Kwok, J.T. and Lu, B.L. (2011). Time and space efficient spectral clustering via column sampling. *Proceedings of CVPR 2011: Conference on Computer Vision and Pattern Recognition (CVPR)*, 2297–2304, Colorado Springs, USA, June 20 - 25, 2011.

Liang, Z., Xiao, F., Qian, X., Zhou, L., Jin, X., Lu, X. and Karichery, S. (2018). A column generation-based heuristic for aircraft recovery problem with airport capacity constraints and maintenance flexibility. *Transportation Research Part B: Methodological*, *113*, 70-90.

Liu M, Haghani A and Toobaie S. (2010) Genetic Algorithm–Based Column Generation Approach to Passenger Rail Crew Scheduling. *Transportation Research Record*. *2159(1)*, 36-43.

Liu, J., Wang, C., Danilevsky, M. and Han, J. (2013). Large-scale spectral clustering on graphs. *Proceedings of Twenty-Third International Joint Conference on Artificial Intelligence*, 1486–1492, Beijing, China, August 3 - 9, 2013.

Liu, W., Ye, M., Wei, J. and Hu, X. (2017). Compressed constrained spectral clustering framework for large-scale data sets. *Knowledge-Based Systems*, *135*, 77-88.

Lodi, A. and Zarpellon, G. (2017). On learning and branching: a survey. Top, *25(2)*, 207-236.

Luisa, B. (1995). Handbook of Combinatorics Volume 1. Retrieved from http://books.google.com/books

Lübbecke, M.E. and Desrosiers, J. (2005). Selected topics in column generation. *Operations research*, *53(6)*, 1007-1023.

Maher, S.J. (2016). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, *50(1)*, 216-239.

Mazyavkina, N., Sviridov, S., Ivanov, S. and Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, *134***,** 105400.

McKinsey & Company, Global AI Survey AI proves its worth but few scale impact. (2019). Retrieved Mar 4, 2021 from https://www.mckinsey.com/featured-insights/artificial-intelligence/global-ai-survey-ai-proves-its-worth-but-few-scale-impact.

Medard, C.P. and Sawhney, N. (2007). Airline crew scheduling from planning to operations. *European Journal of Operational Research*, *183(3)*, 1013-1027.

Mishra, P., Biancolillo, A., Roger, J. M., Marini, F. and Rutledge, D. N. (2020). New data preprocessing trends based on ensemble of multiple preprocessing techniques. *TrAC Trends in Analytical Chemistry*, *132*, 116045.

Mitsokapas, E., Schafer, B., Harris, R.J. and Beck, C. (2021). Statistical characterization of airplane delays. *Scientific Reports, 11*, 1–11.

Muter, I., Birbil, Ş.İ. and Bülbül, K. (2013). Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. *Mathematical Programming*, *142(1)*, 47-82.

Muthuram, N., and Frank, F. C. (2021). Optimization of machining parameters using artificial Intelligence techniques. *Materials Today: Proceedings*, *46*, 8097-8102.

Nascimento, M.C.V. and de Carvalho, A.C.P.L.F. (2011). Spectral methods for graph clustering – a survey. *European Journal of Operational Research, 211*, 221–231.

National Security Commission on Artificial Intelligence, National Security Commission on Artificial Intelligence Final Report. (2021). Retrieved Mar 4, 2021 from https://www.nscai.gov/wp-content/uploads/2021/03/Full-Report-Digital-1.pdf

Nair, V., Bartunov, S., Gimeno, F., von Glehn, I., Lichocki, P., Lobov, I., O'Donoghue, B., Sonnerat, N., Tjandraatmadja, C., Wang, P. and Addanki, R. (2020). Solving Mixed Integer Programs Using Neural Networks. Retrieved July 30, 2021, from https://arxiv.org/pdf/2012.13349

Nazari, M., Oroojlooy, A., Snyder, L.V. and Takáč, M. (2018). Reinforcement learning for solving the vehicle routing problem. Retrieved July 30, 2021, from https://arxiv.org/pdf/1802.04240.pdf

Neu, D. A., Lahann, J. and Fettke, P. (2021). A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artificial Intelligence Review*, 1-27.

Ng, A., Jordan, M. and Weiss, Y. (2002) On spectral clustering: analysis and an algorithm. *Advances in Neural Information Processing Systems*, *14*, 849–856.

Nikitin, N. O., Vychuzhanin, P., Sarafanov, M., Polonskaia, I. S., Revin, I., Barabanova, I. V., Maximov, G., Kalyuzhnaya, A., V. and Boukhanovsky, A. (2022). Automated evolutionary approach for the design of composite machine learning pipelines. *Future Generation Computer Systems*, *127*, 109-125.

Nissen, R. and Haase, K. (2006). Duty-period-based network model for crew rescheduling in European airlines. *Journal of Scheduling*, *9(3)*, 255-278.

Novianingsih, K., Hadianti, R. and Uttunggadewa, S., E., S. (2015). A solution method for airline crew recovery problems. *International Journal of Applied Mathematics and Statistic*s, *53(4)*, 137–149.

Patel, S. A., Patel, S. P., Adhyaru, Y. B. K., Maheshwari, S., Kumar, P. and Soni, M. (2021). Developing smart devices with automated Machine learning Approach: A review. *Materials Today: Proceedings.*

Petersen, J.D., Sölveling, G., Clarke, J.P., Johnson, E.L. and Shebalov, S. (2012). An optimization approach to airline integrated recovery. *Transportation Science*, *46(4)*, 482-500.

Potthoff, D., Huisman, D. and Desaulniers, G. (2010). Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, *44(4)*, 493-505.

Python. The official home of the Python Programming Language. Retrieved February 4, 2022, from https://www.python.org

Ramachandran, K. K., Mary, A. A. S., Hawladar, S., Asokk, D., Bhaskar, B. and Pitroda, J. R. (2022). Machine learning and role of artificial intelligence in optimizing work performance and employee behavior. *Materials Today: Proceedings*, *51*, 2327-2331.

Ramle, A. (2014). *Integrated Recovery of Schedules for Crew and Fleet in Airline Operations—A Mathematical and Computational Approach* (Master's thesis) Retrieved from https://www.chalmers.se/en/departments/math/research/research-groups/optimization/OptimizationMasterTheses/MScThesis-AgnesRamle-final.pdf

Ray. Scaling Python made simple, for any workload. Retrieved February 4, 2022, from https://www.ray.io

Ren, P., Xiao, Y., Chang, X., Huang, P.Y., Li, Z., Chen, X. and Wang, X. (2020). A comprehensive survey of neural architecture search: Challenges and solutions. Retrieved July 30, 2021, from https://arxiv.org/pdf/2006.02903.pdf

Ruas, T., Ferreira, C.H.P., Grosky, W., de Franca, F.O. and de Medeiros, D.M.R. (2020) Enhanced word embeddings using multi-semantic representation through lexical chains. *Information Sciences*, *532*, 16–32.

Santos, B. F., and Gillis, M. M. (2019). Optimizing the prices for airline flight passes. *Transportation Research Procedia*, *37*, 266-273.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN Computer Science*, *2(3)*, 1-21.

Sato, T., Sakikawa, S., Morita, T., Ueki, N. and Murata, T. (2009). Crew and vehicle rescheduling based on a network flow model and its application to a railway train operation. *IAENG International Journal of Applied Mathematics*, *39(3)*, 142-150.

Schaeffer, S.E. (2007). Graph clustering. *Computer Science Review*, *1(1)*, 27–64.

Schaefer, A.J., Johnson, A.J. E. J. and Kleywegt, L., N.G. (2005). Airline crew scheduling under uncertainty. *Transportation Science, 39*, 340–348.

Scikit-learn. (2021). scikit-learn Machine Learning in Python. [Software] Retrieved July 30, 2021. Available from https://scikit-learn.org/

Slišković, D., Grbić, R. and Nyarko, E. K. (2009). Data preprocessing in data based process modeling. *IFAC Proceedings Volumes*, *42(19)*, 559-564.

Soares, E.F.D.S., Campos, C.A.V. and de Lucena, S.C. (2019). Online travel mode detection method using automated machine learning and feature engineering. *Future Generation Computer Systems*, *101*, 1201-1212.

Sohouenou, P.Y.R., Christidis, P., Christodoulou, A., Neves, L.A.C. and Presti, D.L. (2020). Using a random road graph model to understand road networks robustness to link failures. *International Journal of Critical Infrastructure Protection*, *29*, 100353.

Steinbach, M., Ertoz, L. and Kumar, V. (2004). The challenges of clustering high dimensional data. *New Directions in Statistical Physics*, 273–309.

Stojković, M. and Soumis, F. (2001). An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science*, *47(9)*, 1173-1310.

Stojković, M., Soumis, F. and Desrosiers, J. (1998). The operational airline crew scheduling problem. *Transportation Science*, *32(3)*, 232-245.

Subasi, A. (2020). Practical Machine Learning for Data Analysis Using Python. Academic Press.

Suraweera, P., Webb, G.I., Evans, I. and Wallace, M. (2013). Learning crew scheduling constraints from historical schedules. *Transportation Research Part C: Emerging Technologies*, *26*, 214-232.

Suribabu, D. D. D., Sarma, T. H. and Reddy, B. E. (2023). Approximating Clustering for Memory Management and request processing. Retrieved October 28, 2023, from https://arxiv.org/pdf/2308.11008v1.pdf

Talebi, H., Peeters, L., Mueller, U., Tolosana-Delgado, R. and van den Boogaart, K. (2020). Towards geostatistical learning for the geosciences: A case study in improving the spatial awareness of spectral clustering. *Mathematical Geosciences*, *52*, 1035–1048.

Tautenhain, C.P.S. and Nascimento, M.C.V. (2020). An ensemble based on a bi-objective evolutionary spectral algorithm for graph clustering. *Expert Systems with Applications*, *141*, 112911.

TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. Retrieved February 4, 2022, from https://www.tensorflow.org

Tremblay, N. and Loukas, A. (2020). Approximating spectral clustering via sampling: a review. *Sampling Techniques for Supervised or Unsupervised Tasks* (pp.129–183). Retrieved from https://arxiv.org/pdf/1901.10204.pdf

Vadlamani, S. L., Shafiq, M. O. and Baysal, O. (2022). Using machine learning to analyze and predict entry patterns of low-cost airlines: A study of Southwest Airlines. *Machine Learning with Applications*, *10*, 100410.

Vathy-Fogarassy, A. and Abonyi, J. (2013). Graph-Based Clustering and Data Visualization Algorithms. Retrieved from https://link.springer.com/book/

Veelenturf, L.P., Potthoff, D., Huisman, D., Kroon, L.G., Maróti, G. and Wagelmans, A.P. (2016). A quasi-robust optimization approach for crew rescheduling. *Transportation Science*, *50(1)*, 204-215.

Veelenturf, L.P., Potthoff, D., Huisman, D. and Kroon, L.G. (2012). Railway crew rescheduling with retiming. *Transportation Research Part C: Emerging Technologies*, *20(1)*, 95-110.

Verma, D. and Meila, M. (2003). A comparison of spectral clustering algorithms. Retrieved Mar 4, 2021 from https://sites.stat.washington.edu/spectral/papers/UW-CSE-03-05-01.pdf

Villa, C. and Hoffman, K. (2006). A column-generation and branch-and-cut approach to the bandwidth-packing problem. *Journal of research of the National Institute of Standards and Technology*, *111(2)*, 161.

Vitale, I., Broz, D. and Dondo, R. (2021). Optimizing log transportation in the Argentinean forest industry by column generation. *Forest Policy and Economics*, *128*, 102483.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*, 395–416.

Vos, H.W.M., Santos, B.F. and Omondi, T. (2015). Aircraft schedule recovery problem, a dynamic modeling framework for daily operations. *Transportation Research Procedia, 10*, 931–940.

Wang, Q. (2021). Knowledge-based approach for dimensionality reduction solving repetitive combinatorial optimization problems. *Expert Systems with Applications*, *184*, 115502.

Wang, L. and Dong, M. (2012). Multi-level low-rank approximation-based spectral clustering for image segmentation. *Pattern Recognition Letters*, *33(16)*, 2206-2215.

Wang, L., Leckie, C., Ramamohanarao, K. and Bezdek, J. (2009). Approximate spectral clustering. *Proceedings of 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 134–146. Bangkok, Thailand, April 27-30, 2009.

Wang, K., Wang, B. and Peng, L. (2009). Cvap: validation for cluster analyses. *Data Science Journal*, *8*, 88-93.

Waring, J., Lindvall, C. and Umeton, R. (2020). Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine*, *104*, 101822.

Watson, C., Cooper, N., Palacio, D. N., Moran, K. and Poshyvanyk, D. (2020). A Systematic Literature Review on the Use of Deep Learning in Software Engineering Research. Retrieved February 2, 2022, from https://arxiv.org/pdf/2009.06520.pdf

Wei, G., Yu, G. and Song, M. (1997). Optimization model and algorithm for crew management during airline irregular operations. *Journal of Combinatorial Optimization*, *1(3)*, 305-321.

Wistuba, M., Rawat, A. and Pedapati, T. (2019). A survey on neural architecture search. Retrieved July 30, 2021, from https://arxiv.org/pdf/1905.01392.pdf

Wu, S., Feng, X. and Zhou, W. (2014). Spectral clustering of high-dimensional data exploiting sparse representation vectors. *Neurocomputing*, *135*, 229–239.

Xu, P., Corman, F. and Peng, Q. (2016). Analyzing railway disruptions and their impact on delayed traffic in chinese high-speed railway. *IFAC-PapersOnLine, 49(3)*, 84–89.

Xue, C., Hu, M., Huang, X. and Li, C. G. (2022). Automated search space and search strategy selection for AutoML. *Pattern Recognition*, *124*, 108474.

Yan, D., Huang, L. and Jordan, M.I. (2009). Fast approximate spectral clustering. KDD09: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 907–916. Paris, France, June, 28 – July,1 2009.

Yan, S. and Lin, C.G. (1997). Airline scheduling for the temporary closure of airports. *Transportation Science*, *31(1)*, 72-82.

Yan, S. and Tu, Y.P. (2002). A network model for airline cabin crew scheduling. *European Journal of Operational Research*, *140(3)*, 531-540.

Yıldırım, A. A. and Özdoğan, C. (2011). Parallel wavelet-based clustering algorithm on GPUs using CUDA. *Procedia Computer Science*, *3*, 396-400.

Yu, G., Argüello, M., Song, G., McCowan, S.M. and White, A. (2003). A new era for crew recovery at continental airlines. *Interfaces*, *33(1)*, 5-22.

Zhang, C., Bengio, S., Hardt, M., Recht, B. and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. Retrieved July 30, 2021, from https://arxiv.org/pdf/1611.03530.pdf

Zhang, D., Lau, H.H. and Yu, C. (2015). A two-stage heuristic algorithm for the integrated aircraft and crew schedule recovery problems. *Computers & Industrial Engineering*, *87*, 436-453.

Zhang, H., Du, Y., Sun, J. and Li, Q. (2020) Improving interpretability of word embeddings by generating definition and usage. *Expert Systems with Applications*, *160*, 113633.

Zhang, F., Fleyeh, H., Wang, X. and Lu, M. (2019). Construction site accident analysis using text mining and natural language processing techniques. *Automation in Construction*, *99*, 238–248.

Zhang, S., Jafari, O. and Nagarkar, P. (2021). A survey on machine learning techniques for auto labeling of video, audio, and text data. Retrieved October 28, 2023, from https://arxiv.org/pdf/2109.03784.pdf

Zhang, Q., Ma, Y., Yang, Z. and Chen, Z. (2017). The civil aviation crew recovery time-space network model based on a tabu search algorithm. *In Chinese Intelligent Systems Conference.* Singapore: Springer.

Zhang, X. (2020). A Matrix Algebra Approach to Artificial Intelligence. Retrieved from https://link.springer.com/book/

Zhu, B., Cao, X.L., Wang, Y. and Gao, Q. (2014). Constraint Programming Method for Crew Schedule Recovery. *In Applied Mechanics and Materials*. *496*. 1788-1791.

CURRICULUM VITAE

Name Surname : Ahmet HEREKOĞLU

EDUCATION :

- **B.Sc. :** 2005 – Istanbul Technical University, Faculty of Management, Industrial Engineering
- **M.Sc. :** 2012, Marmara University, Faculty of Engineering, Industrial Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2007 – 2010 : Istanbul Metropolitan Municipality
- 2010 – 2020 : Turkish Airlines
- 2020 – 2021 : Valensas
- 2021 – 2022 : Turk Telekom
- 2022 – …… : Avsos

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Spectral clustering approximation for large scale crew disruption data of an airline company for intelligent crew recovery
- Crew recovery optimization with deep learning and column generation for sustainable airline operation management