İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BÜYÜK ÖLÇEKLİ HAVAYOLU EKİP EŞLEME PROBLEMLERİNİN ÇÖZÜMÜ İÇİN BİR KOLON TÜRETME STRATEJİSİ

DOKTORA TEZİ

Bahadır ZEREN

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

OCAK 2017

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BÜYÜK ÖLÇEKLİ HAVAYOLU EKİP EŞLEME PROBLEMLERİNİN ÇÖZÜMÜ İÇİN BİR KOLON TÜRETME STRATEJİSİ

DOKTORA TEZİ

Bahadır ZEREN
(511072112)

Uçak ve Uzay Mühendisliği Anabilim Dalı

Uçak ve Uzay Mühendisliği Programı

Tez Danışmanı: Prof. Dr. İbrahim ÖZKOL

OCAK 2017

İTÜ, Fen Bilimleri Enstitüsü'nün 511072112 numaralı Doktora Öğrencisi Bahadır ZEREN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "BÜYÜK ÖLÇEKLİ HAVAYOLU EKİP EŞLEME PROBLEMLERİNİN ÇÖZÜMÜ İÇİN BİR KOLON TÜRETME STRATEJİSİ" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. İbrahim ÖZKOL** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Prof. Dr. Ahmet Duran ŞAHİN** İstanbul Teknik Üniversitesi	.....
	Prof. Dr. Abdurrahman HACIOĞLU Hava Harp Okulu	.....
	Prof. Dr. Gültekin ÇETİNER Marmara Üniversitesi	.....
	Doç. Dr. Nihan ÇETİN DEMİREL Yıldız Teknik Üniversitesi	.....

Teslim Tarihi : 15 Aralık 2016
Savunma Tarihi : 25 Ocak 2017

Aileme, eşime ve Zeyneb'ime,

ÖNSÖZ

Bu tez çalışmasında havayolu sektöründeki planlama süreçlerinden en önemlilerinden biri olan ekip eşleme optimizasyonu konusunda araştırma ve geliştirme çalışmaları yapılmıştır. Türk Hava Yolları'nın da desteği ile ticari uygulamalar ile yarışacak seviyede kaliteli sonuç üretebilen, detaylı bir algoritma geliştirilmiştir.

Tez çalışmam süresince kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren danışman hocam sayın Prof. Dr. İbrahim ÖZKOL'a ve manevi desteklerini her an yanımda hissettiğim sevgili aileme teşekkürlerimi sunuyorum.

Ayrıca bu tez çalışması süresince bize desteklerini esirgemeyen Türk Hava Yolları Genel Müdürü sayın Dr. Temel KOTİL'e ve Gn. Md. Yrd. sayın Dr. Ahmet BOLAT'a, "Ekip Planlama" ve "Kurumsal Gelişim ve Bilgi Teknolojileri" Başkanlıklarına teşekkürü bir borç bilirim.

Ekim 2016

Bahadır Zeren
Uçak Yüksek Müh.

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **vii**
İÇİNDEKİLER ..... **ix**
KISALTMALAR ..... **xi**
ÇİZELGE LİSTESİ ..... **xiii**
ŞEKİL LİSTESİ ..... **xv**
ÖZET ..... **xvii**
SUMMARY ..... **xix**
1. GİRİŞ **1**
1.1 İlgili Çalışmalar 2
1.2 Temel Kavramlar, Kurallar ve Tanımlar 5
1.2.1 Görev/Uçuş görevi (Duty/Flight duty) 5
1.2.2 Dinlenme süresi (Rest period) 6
1.2.3 Ekip eşlemesi (Crew pairing) 6
1.2.4 Pas uçuş 6
1.2.5 Uzun menzil uçuşu (ER flight) 7
1.3 Problem Tanımı 7
1.4 Ana (Master) Model 8
2. PROBLEMİN GENETİK ALGORİTMA İLE ÇÖZÜMÜ **11**
2.1 Genetik Algoritmalar 11
2.2 Ekip Eşleme Problemi İçin Geliştirilen GA Konfigürasyonu 13
2.2.1 Kromozom tasarımı 13
2.2.1.1 Kolon temelli gösterim (Column-based representation) 14
2.2.1.2 Satır temelli gösterim (Row-based representation) 15
2.2.2 Seçim (Selection) 15
2.2.3 Çaprazlama (Crossover) operatörü 16
2.2.4 Mutasyon (Mutation) operatörü 16
2.2.5 Uygunluk (Feasibility) operatörü 17
2.2.6 Kromozom iyileştirme (Improvement) operatörü 18
2.2.7 Populasyon yenileme (Population replacement) 19
2.2.8 Deneysel sonuçlar ve değerlendirme 19
2.3 Paralel Genetik Algoritma Konfigürasyonu 20
2.3.1 PGA modelleri 21
2.3.1.1 Global populasyon modeli (Master-Slave) 21
2.3.1.2 Ada modeli (Island/Coarse-Grained) 22
2.3.1.3 Hücresel model (Cellular/Fine-Grained) 22
2.3.2 Paralel genetik algoritma uygulaması 23
2.3.3 PGA test sonuçları ve değerlendirme 24
2.3.3.1 320 filosu PGA sonuçları 25
2.3.3.2 737 filosu PGA sonuçları 26
3. ARKAPLAN **29**

3.1 Doğrusal Programlama (Linear Programming) 29
3.1.1 Doğrusal eşitlikler (Linear equations) 29
3.1.2 Karar değişkenleri (Decision variables) 30
3.1.3 Amaç fonksiyonu (Objective function) 30
3.1.4 Kısıtlamalar (Constraints) 30
3.1.5 Negatiflik önleyici kısıtlamalar (Non-negativity constraints) 31
3.1.6 Genelleştirilmiş kanonik doğrusal programlama formu 31
3.1.7 Doğrusal programlama problemlerindeki temel özellikler 31
3.2 Doğrusal Programlama Çıktıları 32
3.2.1 İndirgenmiş maliyet (Reduced-cost) 32
3.2.2 Arttıran/eksilten yapay değişken (Slack/Surplus variable) 33
3.2.3 Gölge fiyat (Shadow-price/Dual variable) 33
3.3 Dualite (Duality) 34
3.4 Doğrusal Programlama Çözüm Yöntemleri 35
3.5 Kolon Türetme (Column Generation) 37
3.6 Stabilizasyon 39
3.7 Dal-sınır Algoritması (Branch&Bound Algorithm) 40
4. YENİ KOLON TÜRETME STRATEJİSİ 43
4.1 Sistemin Ayağa Kaldırılması 43
4.1.1 Görev üretme 43
4.1.2 Görev-uçuş yatı bağlantısı ağı (Duty-flight overnight connection network) 44
4.1.3 İlk ekip eşlemelerinin üretimi (Initial crew pairing generation) 46
4.2 Kısmi Çözüm Üretme Yaklaşımı 47
4.2.1 Ücretlendirme probleminin sezgisel çözümü 49
4.2.2 Ücretlendirme probleminin kesin çözümü 51
4.2.3 Bağlantı sabitleme (Connection fixing) sezgiseli 55
4.3 Çözüm Stratejisine Genel Bakış 57
5. DENEYSEL SONUÇLAR VE TAVSİYELER 59
6. SONUÇ VE VİZYON 67
KAYNAKLAR 71
ÖZGEÇMİŞ 75

KISALTMALAR

GA **:** Genetik Algoritmalar
PGA **:** Paralel Genetik Algoritmalar
LP **:** Lineer Program/Programlama
DP **:** Doğrusal Program/Programlama
TP **:** Tamsayı Programı/Programlama
DPG **:** Doğrusal Program Gevşetmesi

ÇİZELGE LİSTESİ

Sayfa

Çizelge 1.1 : Örnek uçuş görevi **5**
Çizelge 1.2 : Normal uçuş görev süreleri **5**
Çizelge 1.3 : Uzun menzil uçuları azami uçuş görev süreleri **6**
Çizelge 1.4 : Asgari dinlenme süreleri **6**
Çizelge 1.5 : Örnek ekip eşlemesi **6**
Çizelge 2.1 : PGA verisetleri **24**
Çizelge 2.2 : 320 filosu PGA sonuçları **25**
Çizelge 2.3 : 737 filosu PGA sonuçları **26**
Çizelge 5.1 : Test verisi **59**
Çizelge 5.2 : Paremetre setleri **60**
Çizelge 5.3 : Parametre kümesi 1 için kalite kriterleri ve maliyet raporu **61**
Çizelge 5.4 : Parametre kümesi 2 için kalite kriterleri ve maliyet raporu **62**
Çizelge 5.5 : Parametre kümesi 3 için kalite kriterleri ve maliyet raporu **63**
Çizelge 5.6 : Kalite kriterleri yüzdelik değişimleri özet karşılaştırma raporu **64**

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : Temel genetik algoritma döngüsü **13**
Şekil 2.2 : Kolon temelli gösterim **14**
Şekil 2.3 : Satır temelli gösterim **15**
Şekil 2.4 : Füzyon çaprazlama operatörü **16**
Şekil 2.5 : Uygunluk operatörü **18**
Şekil 2.6 : İterasyon ve maliyet değişimi karşılaştırması **20**
Şekil 2.7 : Global populasyon modeli (Master-Slave) **21**
Şekil 2.8 : Ada modeli (Island/Coarse-Grained) **22**
Şekil 2.9 : Hücresel model (Cellular/Fine-Grained) **23**
Şekil 2.10 : Farklı CPU sayıları ile 320 çözümü performans karşılaştırması **25**
Şekil 2.11 : Farklı CPU sayıları ile 737 çözümü performans karşılaştırması **26**
Şekil 3.1 : Örnek dal-sınır düğümü. **41**
Şekil 4.1 : Görev oluşturma örnek kodu **43**
Şekil 4.2 : Görev-uçuş bağlantısı ağında bulunan düğüm ve bağlantılar. **45**
Şekil 4.3 : Tam-bağlı görev ağı örneği **46**
Şekil 4.4 : Görev-uçuş yatı bağlantısı ağı örneği **46**
Şekil 4.5 : İterasyonlar arası ekip eşleme karşılaştırması **48**
Şekil 4.6 : Kısmi çözüm üretimi yaklaşımı örnek kodu **49**
Şekil 4.7 : Ücretlendirme probleminin sezgisel çözümü örnek kodu **50**
Şekil 4.8 : Kesin ücretlendirme modelinin çıktısı ile ekip eşleme üretimi **52**
Şekil 4.9 : Örnek ekip eşlemesi **53**
Şekil 4.10 : Optimizasyon süresince ondalıklı değişken sayısının değişimi **56**
Şekil 4.11 : Önerilen çözüm stratejinin genel şeması **58**
Şekil 5.1 : Toplam üstün gelinen koşum sayısı yüzdesi **65**

BÜYÜK ÖLÇEKLİ HAVAYOLU EKİP EŞLEME PROBLEMLERİNİN ÇÖZÜMÜ İÇİN BİR KOLON TÜRETME STRATEJİSİ

ÖZET

Bir ekip eşlemesi (crew pairing); aynı ikamet merkezinde (domicile) başlayan ve biten, birbirini takip eden uçuşların oluşturduğu dizidir. Ekip eşleme planlaması ise havayolu ekip planlama sürecinin en önemli maliyet belirleyici aşamasıdır. Dolayısıyla ekip eşleme optimizasyonu, operasyonel ekip maliyetlerini minimize eden ve ekiplerin verimli kullanımını maksimize eden oldukça önemli bir süreçtir.

Ekip eşlemesi üretimi sürecinde gözönünde bulundurulması ve uyulması gereken, resmi yönetmelikler veya şirket prosedürleri kaynaklı birçok kısıtlama ve kural bulunmaktadır. Optimizasyonun temel amacı, bu kural ve kısıtlamaları gözönünde bulundurarak, havayolu tarifesinde bulunan tüm uçuşları kapsayan düşük maliyetli ve sivil havacılık kurallarına uygun ekip eşleme kümeleri üretmektir.

Bu araştırma ve geliştirme çalışmasında, ekip eşleme optimizasyonu ile ilgili var olan çalışmalar incelenmiş ve öncelikle ekip eşleme problemini çözmek üzere bir genetik algoritma yaklaşımı geliştirilmiştir. Genetik algoritmalar, yöneylem alanındaki bir çok farklı optimizasyon probleminin çözümünde kullanılan güçlü bir metasezgisel (metaheuristics) tekniktir. Çalışmanın bu aşamasında, genetik algoritmanın performansını arttırmak üzere yeni operatörler/sezgisel metodlar geliştirilmiş ve küçük ve orta ölçekli filolar için başarılı sonuçlar elde edilmiştir. Fakat büyük ölçekli filolar için, geliştirilen genetik algoritmanın paralelleştirilmiş versiyonu bile matematiksel yöntemlerin ürettiği sonuçlara tercih edilebilir kalitede çözüm üretememiştir. Elde edilen bu bulguları takiben, çalışma, özellikle dinamik kolon türetme teknikleri (dynamic column generation) kullanan matematiksel yaklaşımların üzerinde yoğunlaştırılmıştır. Ve bu bağlamda bilimsel literatüre katkı niteliğinde olan; yeni bir kolon türetme stratejisi, bir ücretlendirme ağı (pricing network) tasarımı ve ekip eşlemesi eliminasyonu için kullanılan bir sezgiseli (heuristics) geliştirilmiştir. Önerilen stratejide, ana problem küme-kapsama (set-covering) problemi, ücretlendirme alt problemi (pricing sub problem) ise en kısa yol (shortest-path) problemi olarak modellenmiş; sezgisel ve kesin (exact) algoritmaların birarada kullanımı ile görev-uçuş yatı bağlantısı çizgesi (duty-flight overnight connection graph) üzerinde verimli bir şekilde çözülmüştür. Önerilen strateji, Türk Hava Yolları'ndan elde edilen gerçek dünya verileri ile test edilmiş ve bu çalışmada sunulan test sonuçlarından da anlaşılacağı üzere Türk Hava Yolları'nda hali hazırda kullanılan sistem ile karşılaştırıldığında yüksek kalitede ve son derece rekabetçi çözümler üretebilme yeteneğine sahip oluduğu görülmüştür. Aynı zamanda önerilen çözüm yaklaşımının, amaç fonksiyonundaki (objective/fitness function) ceza katsayılarına olan hassasiyeti, daha az pas (deadhead) uçuş üretmesi, daha hafif donanımlar ve tek kanallı (single-threaded) bir yazılım ile neredeyse aynı sürelerde sonuç üretebilme gibi avantajları olduğu da görülmektedir.

A COLUMN GENERATION STRATEGY FOR LARGE SCALE AIRLINE CREW PAIRING PROBLEMS

SUMMARY

A crew pairing is a sequence of flight legs beginning and ending at the same crew domicile hub. Crew pairing planning is the primary cost-determining phase in airline crew scheduling. Therefore, optimizing crew pairings of an airline timetable is an extremely important process which helps to minimize operational crew costs and to maximize crew utilization.

There are various restrictions imposed by regulations or company policies that must be considered and satisfied in crew pairing generation process. Keeping these restrictions and regulations in mind, the main goal of the optimization is the generation of low cost sets of valid crew pairings which cover all flights in airline's timetable. And the ultimate aim of airline crew scheduling is assignment of crew members to flights so as to satisfy all crew need of all flights in airline's timetable. There are some important key performance indicators (KPI) of outputs of crew scheduling processes like total man day, number of overnights, deadhead time, ground time etc. It is always desired to reduce these KPI values as much as possible. By doing this company can increase the crew utilization and basically keep crew on air as much as possible and avoid losing time because of other reasons except flights.

There are two main sub processes of strategic crew scheduling. First one is crew pairing generation that constitutes the main topic of this study. In crew pairing generation process, flight sequences that constitute flight duties and then crew pairings are generated so as to cover all flights in airline's timetable. Optimization methods play very important role in crew pairing generation process because of critical cost factors that must be minimized. Crew costs are the second biggest cost after fuel costs for airline companies. This situation makes crew scheduling and especially crew pairing optimization significantly important operation that must be carried out carefully. Even small amount of percentages can cost millions of dollars. The second sub process is crew rostering process. In this phase, crew assignment to all crew pairings generated in previous crew pairing generation phase is done. Crew rostering has less impact on total crew costs compared to crew pairing generation process. It mainly affects workload balance between crew members and aims to increase fairness of the plan for crew members. Some US companies does not carry out rostering process. Instead, crew assignment is done by a seniority based bidding process and then a market style system that provides possibility of trip trading. Eventually the importance of optimum crew scheduling cannot be overstated as all airlines operate in a very complex environment. In addition, with increasing competitiveness in the marketplace, airline companies are in a position to better manage their expenses using effective flight and crew scheduling techniques.

In literature, airline crew pairing problem is mostly modeled as set-covering or set-partitioning problems. These models perfectly fit almost all optimization modeling needs that arise in airline crew scheduling. Both of them are combinatorial and are proven to be NP-complete. In literature, airline crew pairing problem is mostly modeled as set-covering or set-partitioning problems. These models perfectly fit almost all optimization modeling needs that arise in airline crew scheduling. Both of them are combinatorial and are proven to be NP-complete. There are two main approaches to solve these kind of problems: Heuristic methods and mathematical (LP based) methods. The most widely used heuristic method in this field is genetic algorithms which is basically a search heuristic that mimics the process of natural selection. There are two main column generation strategies used in GA approaches. The first strategy is offline column generation which is based on generating huge amount of columns (crew pairings) just before optimization phase. Then optimization phase takes place and the best subset of these pre-generated columns is selected. The other strategy is based on generating columns during optimization phase. In this strategy columns are generated from choromosomes (represent solutions in GA context) using special heuristic. Nevertheless metaheuristics can give high quality solutions for only small sized problems.

On the mathematical side, different column generation strategies exist as is seen in heuristic approaches. Some earlier studies which rely on mathematical methods implemented partial enumeration for large scale problems. Thus they increased performance by reducing problem size. The second strategy which is called “column generation” or “dynamic column generation” is the most widely used and studied technic in last two decades. It provides a very efficient way to generate only necessary columns to improve objective value further. Column generation method divides whole problem into two parts: Master problem and sub (pricing) problem. In this study master problem is modeled as set-covering problem and the dual variable information (shadow price) of each row (flight leg) is calculated by solving it. The pricing sub problem is modeled as shortest-path problem and new columns (crew pairings) are generated by finding columns with least negative reduced cost using dual information obtained from master problem.

Even though set-covering and set-partitioning problems require integer solutions at the end, the core mechanism of column generation relies on LP. Calculation of dual variables and reduced costs etc. require LP solutions. There are a few strategies that are used to obtain integer solutions for such problems. A very largely used approach is based on using some special heuristic which eliminate or select pairings based on their linear variable values and reduced costs. Another approach is based on using branch and bound algorithm to get integer solution at the end of column generation phase. But this implementation fails for large sized problems if there is no special performance increasing heuristic. The most popular strategy for last two decades is the branch and price approach which is based on generating columns on a branch and bound tree. Branch and price is an exact method that guarantees finding global optima but requires employment of special heuristic for performance improvements with a little loss in solution quality. But even with special heuristics, it is a relatively more time consuming technic according to some test results published. It also requires parallel programming tech-nics which are more sophisticated and relatively difficult to implement.

For this research study, already existing works related to crew pairing optimization are examined and first a genetic algorithm approach is developed to solve crew

pairing problems. Genetic algorithms which is a powerful metaheuristics that is used in wide variety of optimization problems in the area of operations research. During this phase of the study, additional operators/heruristics are developed to improve genetic algorithms's performance and it produced succesful results for small and middle sized fleets. A parallel implementation of the improved genetic algorithm approach was developed to be able to solve large sized instances of crew pairing problem. This parallel implementation was developed based on island parallel genetic algorithm model which is efficient in working both on tcp-ip networks or on a massively parallel high-tech servers. After comprehensive tests and experiments it is seen that for large sized fleets, even a parrallel implementation of the improved genetic algorithm approach has not been able to generate preferable solutions compared to mathematical approachs. After this point of time, the study has been shifted to work on mathematical approachs, especially ones that use dynamic column generation technics and a new column generation strategy, a pricing network design and a pairing elimination heuristics are developed as a contribution to the previous studies. In the proposed strategy, the main problem is modeled and solved as a set-covering problem and the pricing sub problem is modeled as a shortest-path problem which is efficiently solved over a duty-flight overnight connection graph by the combined usage of heuristic and exact algorithms. The proposed strategy has been tested with real world data obtained from Turkish Airlines and it is seen that it is capable of generating very competitive solutions compared to current practices in Turkish Airlines. It is also observed that there are various advantages of the proposed solution approach such as sensitivity to penalty coefficients, generating less deadheads, very close solution times with a single threaded software and light weight hardware.

1. GİRİŞ

Ekip Planlama sürecinin nihai amacı, havayolu tarifesindeki tüm uçuşların ekip ihtiyacını karşılayacak şekilde uçucu ekiplerin bu uçuşlara atanmasıdır. Bu süreçteki proseslerin çıktılarının değerlendirilmesi için kullanılan; toplam adam gün, yatı sayısı, pas süresi, yer süresi vb. gibi önemli maliyet kriterleri bulunmaktadır. Daima bu kriter değerlerinin mümkün olduğu kadar aşağı düşürülmesi arzu edilir. Bu sayede şirket, uçucu ekiplerin uçuş dışındaki aktiviteler ile zaman kaybetmesini engelleyerek mümkün olduğu kadar havada görev yapmalarını ve görev sürelerinin verimli kullanımını sağlar.

Stratejik ekip planlama, iki adet temel alt süreçten oluşmaktadır. Bunlardan ilki, bu çalışmanın da ana konusunu oluşturan ekip eşlemesi oluşturma (crew pairing generation) aşamasıdır. Ekip eşlemesi oluşturma aşamasında; havayolu tarifesindeki tüm uçuşları kapsayan, uçuş görevlerini (flight duties) ve ardından da ekip eşlemelerini oluşturan uçuş dizileri üretilir. Optimizasyon metodları, ekip eşlemesi oluşturma aşamasında, minimize edilmesi istenen ve yukarıda bahsedilen kritik maliyet göstergeleri sebebi ile çok önemli bir rol oynamaktadır. Ekip maliyetleri havayolu firmaları için yakıt maliyetlerinden sonraki en büyük ikinci maliyet kalemidir (Andersson ve diğ, 1997). Bu durum, ekip planlamayı ve özellikle de ekip eşleme optimizasyonu konusunu dikkatli bir şekilde icraa edilmesi gereken oldukça önemli bir operasyon haline getirmektedir. Öyleki, optimizasyon çıktılarındaki küçük yüzdelik farklar bile milyonlarca dolar ek maliyet anlamına gelebilmektedir.

İkinci temel alt süreç ise ekip atama (crew rostering) aşamasıdır. Bu aşamada, bir önceki aşamada üretilen tüm ekip eşlemeleri için gerektiği kadar uçucu ekip ataması yapılır. Ekip atama sürecinin toplam ekip maliyetlerine olan etkisi, ekip eşleme sürecinin etkisine göre daha düşüktür. Atama süreci daha çok uçucu ekip arasında dengeli iş dağılımının yapılmasını amaçlar. Bu sayede uçucu ekipler nazarında adil uçuş planları üretilmesini sağlar. Amerika meşeili bazı firmalar, bahsedildiği gibi tek aşamalı ekip atama süreçleri gerçekleştirmezler. Bunun yerine, aylık ekip planları (bidlines) üretilir ve kıdem temelli bir tercih mekanizması ile atama süreçleri uçucu

ekiplerin tercihleri doğrultusunda yönetilir. Tercih aşamasından sonra ek bir sistem ile uçucu ekiplere sözkonusu aylık planlar için değiş tokuş imkanı sunulur.

Sonuç olarak, oldukça karmaşık süreçler içerisinde yürütülen havacılık operasyonları içindeki ekip planlama süreçlerinin önemi son derece büyüktür. Buna ek olarak sektördeki git gide artan rekabetçi koşullar, havayolu firmalarını maliyetlerini verimli uçuş ve ekip planlama teknikleri ile yönetmesini zorunlu kılmaktadır (Zeren ve Özkol, 2012).

Aynı zamanda stratejik ekip planlama yazılımlarının dünya çapında Jeppesen, Adopt gibi sadece bir iki yabancı firma tarafından geliştiriliyor olması ve Türk Hava Yolları gibi büyük filolara sahip firmaların ihtiyaçlarını karşılayan yerli bir optimizasyon çözümünün olmaması bu çalışmanın temel motivasyon unsurlarındandır. Elde edilen başarılı sonuçlar da benzeri bir sistemin yerli insan kaynağı ve bilgi birikimi ile rahatlıkla üretilebileceğini göstermektedir.

1.1 İlgili Çalışmalar

Havayolu ekip eşleme problemi literatürde daha çok küme-kapsama (set-covering) veya küme-ayırma (set-partitioning) problemi olarak modellenmektedir. Bu modeller havayolu ekip planlama alanında ortaya çıkan modelleme ihtiyaçlarının hemen hemen tümüne uymaktadır. Her ikisinin de kombinatoryel (combinatorial) ve NP-Tam (Np-Complete) oldukları kanıtlanmıştır (Garey ve Johnson, 1979).

Bu tip problemleri çözmek için kullanılan temel iki yaplaşım vardır: 1- Metasezgisel (metaheuristics) metodlar. 2- Mathematical (DP tabanlı) metodlar. Bu alanda kullanımı en yaygın olan metaheuristic teknik, temelde doğadaki seçim sürecini taklit eden bir arama/tarama sezgiseli (heuristics) olan genetik algoritmalardır. Genetik algoritmalarda iki temel kolon türetme stratejisi kullanılmaktadır. İlk strateji devre dışı (offline) kolon türetmedir. Bu strateji, optimizasyon aşamasından önce büyük miktarda kolon (ekip eşlemesi) türetmeye dayanmaktadır. Ardından optimizasyon aşaması başlatılır ve üretilen bu büyük miktardaki kolonların amaç fonksiyonunda (objective/fitness function) en iyi değeri veren alt kümesi bulunur (Beasley ve Chu, 1996; Kornilakis ve Stamatopoulos, 2002; Zeren ve Ozkol, 2012). Diğer strateji kolonların optimizasyon aşamasında üretilmesine dayanmaktadır. Bu stratejide,

kolonlar özel sezgiseller kullanılarak kromozomlardan (GA bağlamında çözümleri temsil eder) üretilir (Aickelin, 2002; Ozdemir & Mohan, 2001).

Zeren ve Ozkol (2012) çalışmasında, Beasley ve Chu (1996) çalışmasından referans alınan GA modeli, ek operatör ve sezgiseller ile güçlendirilmiş ve bu iyileştirilmiş GA'nın aylık 710 uçuşu olan bir filo için global eniyiye (optimum) göre % 0.04 sapma ile sonuç üretebildiği gösterilmiştir. Fakat daha sonra elde edilen tecrübeler, GA'nın sadece küçük filolar için yüksek kaliteli sonuçlar üretebildiğini, uçuş sayısı arttıkça çözüm kalitesinin düştüğünü ve GA'nın özellikle pas (deadhead) süresini verimli bir şekilde düşürmekte zorlandığını göstermiştir. Takip eden süreçte geliştirilen GA; Luque ve Alba (2011), Harik ve diğ. (1999), Cant´u-Paz (2007), Tongchim ve Chongstitvatana (2000) çalışmaları referans alınarak, ada (island (coarse-grained)) modeli ile parallelleştirilmiştir. Fakat paralel uygulamanın da aylık ortalama 7000 uçuşa sahip olan bir filo için global eniyiye %7 oranında yaklaşabildiği tespit edilmiştir.

Matematiksel tarafta da metasezgisellerde olduğu gibi farklı kolon türetme stratejileri mevcuttur. Matematiksel metodlara dayanan daha önceki dönemde yapılan çalışmaların bazılarında büyük problemler için kısmi kolon türetme uygulanmıştır. Böylece problemin ölçeği küçültülerek performans artışı sağlanmıştır (Baker ve diğ, 1979, Klabjan ve diğ, 2001). Kolon türetme veya dinamik kolon türetme (dynamic column generation) olarak adlandırılan ikinci strateji ise 90 lardan sonra üzerinde en çok çalışılan ve kullanılan tekniktir. Bu teknik, sadece amaç fonksiyonunu daha iyiye götürecek kolonların üretilmesini sağlayan verimli bir yol sağlar. Kolon türetme tekniği tüm problemi iki ana parçaya ayırır. Ana (master) problem ve alt/ücretlendirme (sub/pricing) problemi. Bu çalışmada ana problem küme-kapsama problemi olarak modellenmiştir ve her satırın (uçuş) dual değişken bilgisi (gölge fiyat/shadow price) ana problemin çözümü ile hesaplanmaktadır. Ücretlendirme alt problemi ise en kısa yol problemi olarak modellenmiştir ve yeni kolonlar (ekip eşlemeleri) en düşük negatif indirgenmiş maliyet (reduced cost) değerini veren kolonların, ana problemden elde edilen dual değişken bilgisinin kullanılarak bulunması ile üretilir. Kolon türetme ile ilgili daha detaylı bilgi için Desrosiers ve Lübbecke (2005), Lübbecke (2011), Lübbecke ve Desrosiers (2005), Vanderbeck (2005) referanslarına başvurulabilir.

Küme-kapsama ve küme-ayırma problemleri en nihayetinde tamsayı çözümlere ihtiyaç duysa da kolon türetme tekniğinin temel mekanizmaları DP ya dayanmaktadır. Dual değişken, indirgenmiş maliyet hesaplamaları gibi gereksinimler sebebi ile algoritma çalıştırıldığı sürece, DP çözümlerine ihtiyaç duymaktadır. Bu nedenle DP dan hareketle tamsayı çözüm elde edebilmek için çeşitli stratejiler geliştirilmiştir. Bu yaklaşımlardan ilkinde, Anbil ve diğ. (1992) çalışmasında olduğu gibi, ekip eşlemelerinin doğrusal değişken ile indirgenmiş maliyet değerleri gözönüne alınarak seçilmelerine veya elimine edilmelerine dayanan bazı özel sezgisel algoritmalar kullanılmıştır. Andersson ve diğ. (1998), Gustafsson (1999), Medard ve Sawhney (2007), Wedelin (1995) çalışmalarında ise bir çeşit yuvarlama (rounding) sezgiseli geliştirilerek ana problemi Lagrangian gevşetmesi (relaxation) ve subgradient tekniğini kullanarak çözen bir sistem üzerinde uygulanmıştır.

Bir başka strateji de DP iterasyonlarının en sonunda tamsayı çözüm elde etmek için dal-sınır (branch and bound) algoritması kullanılmasına dayanmaktadır. Keza bu uygulama da büyük ölçekli problemlerde, eğer performans arttırıcı özel bir sezgisel yaklaşım kullanılmıyor ise başarısız olmaktadır.

90 lardan sonra en çok kullanılan stratejilerden birisi de dal-sınır ağacında kolon türetmeye dayanan dal-ücret (branch and price) tekniğidir (Barnhart ve diğ, 1998). Dal-ücret tekniği, global eniyinin bulunduğunu garanti eden fakat çözüm kalitesinde önemsiz seviyelerde feragat edilerek, performansı önemli ölçüde arttıracak özel sezgisellere ihtiyaç duyan kesin (exact) bir yaklaşımdır. Dal-ücret yaklaşımının sezgisel bir uygulaması için Vance ve diğ. (1997) çalışması referans alınabilir. Daha güncel bir çalışma ve test sonuçları için ise Desaulniers ve Soumis (2010) çalışması incelenebilir. Test sonuçlarından da görüleceği üzere aylık 7000 civarı uçuşu bulunan bir filonun çözümü için dal-ücret yaklaşımı ortalama 10 saat gibi bir süreye ihtiyaç duymaktadır. Yaklaşım aynı zamanda daha karmaşık ve uygulanması göreceli olarak daha zor olan paralel programlama tekniklerinin kullanımına ihtiyaç duymaktadır.

Bu çalışmada, sezgisel ve kesin kolon türetme metodlarının, görev-uçuş yatı bağlantısı çizgesi (duty-flight overnight connection graph) üzerinde birlikte kullanılarak çözüldüğü bir kolon türetme stratejisi önerilmiştir. Önerilen strateji aynı zamanda çözüm performansını arttıran, bağlantı sabitleme (connection fixing) sezgiseli olarak adlandırılan özel bir görev ve eşleme eliminasyon sezgiseli kullanmaktadır. Test sonuçları kısmında görüleceği üzere, önerilen stratejinin, Türk

Hava Yolları'nda kullanılan halihazırdaki pratikler ile karşılaştırıldığında yüksek kalitede, rekabetçi çözümler üretebilme kabiliyetine sahip olduğu görülmektedir.

1.2 Temel Kavramlar, Kurallar ve Tanımlar

1.2.1 Görev/Uçuş görevi (Duty/Flight duty)

Bir veya daha fazla uçuştan oluşan ve bitiminde uçuş ekibinin dinlenme periyodunun başladığı zaman dilimine görev/uçuş görevi denir. Örnek bir uçuş görevi Çizelge 1.1'deki gibidir.

Çizelge 1.1 : Örnek uçuş görevi.

Görev Başlangıcı	Görev Bitişi	Kalkış Saati	İniş Saati	Kalkış Yeri	Varış Yeri
25.03.2007 06:20		25.03.2015 07:35	25.03.2015 10:45	IST	ESB
		25.03.2015 11:45	25.03.2015 14:45	ESB	IST
		25.03.2015 16:00	25.03.2015 17:00	IST	ADB
	25.03.2007 19:30	25.03.2015 18:00	25.03.2015 19:00	ADB	IST

Uçuş görevleri farklı ekip sayıları ile temel olarak üç şekilde uygulanır.

- Normal uçuş ekibi ile
- İlave uçuş ekibi ile
- İlave tam uçuş ekibi ile

Uçuş görevinin normal uçuş ekibi ile uygulanması durumunda, görevin uygulandığı uçak filosunun ihtiyaç duyduğu kadar kokpit ve kabin ekibi kullanılır. İlave uçuş ekibi ve ilave tam uçuş ekibi uygulamaları ise, görev süresinin uzatılmasının gerektiği durumlarda uygulanabilmektedir. İlave ekip, ekstradan bir kaptan ve bir kabin memurunu, ilave tam ekip ise ekstradan tam bir normal ekibi içermektedir.

Uzun menzil uçuş görevlerinde ise uçuş görev süresi, normal ekip ile uygulansa dahi uzun olabilmekte fakat dinlenme süreleri buna göre farklı uygulanmaktadır.

Uygulanabilecek azami uçuş görev süreleri, SHGM Sivil havacılık Talimatı (2014) de Çizelge 1.2 ve Çizelge 1.3'deki gibi verilmiştir.

Çizelge 1.2 : Normal uçuş görev süreleri.

Görev başlangıç saati	1–4 iniş	5 iniş
05.00 – 14.00	14 saat	13 saat
14.01 – 17.00	13 saat	12 saat
17.01 – 04.59	12 saat	11 saat

Çizelge 1.3 : Uzun menzil uçuları azami uçuş görev süreleri.

Görev tipi	Uçuş Görev Süresi
Normal Uçuş Ekibi	14 saat
İlave Ekip ile	16 saat
İlave Tam Ekip ile	18 saat

1.2.2 Dinlenme süresi (Rest period)

Dinlenme süresi, bir uçucu ekibin uçuş görevinin bitiminde başlayan ve her türlü görevden muhaf tutulduğu süredir. Asgari dinlenme süreleri SHGM Sivil havacılık Talimatı (2014) de Çizelge 1.4'deki gibi verilmiştir.

Çizelge 1.4 : Asgari dinlenme süreleri.

Bir önceki uçuş görev süresi	Asgari dinlenme süresi
6 Saate kadar	8 saat
11 Saate kadar(Dahil)	10 saat
11 Saatten daha fazla	12 saat
12–14 Saat veya ZD farkı 3 saatten fazla	14 saat
Uzun Menzil Uçuşları	2 yerel gece / 36 saat

1.2.3 Ekip eşlemesi (Crew pairing)

Ekip eşlemesi, bir veya birden fazla uçuş görevinden oluşan ve belirli bir ikamet merkezinden başlayıp tekrar aynı ikamet merkezinde biten uçuş/uçuş görevi dizisidir. Örnek bir ekip eşlemesi Çizelge 1.5'teki gibidir.

Çizelge 1.5 : Örnek ekip eşlemesi.

Uçuş Görev No	Görev Başlangıcı	Görev Bitişi	Kalkış Saati	İniş Saati	Kalkış Yeri	Varış Yeri
1	25.03.2015 10:00		25.03.2015 11:15	25.03.2015 13:20	IST	ESB
			25.03.2015 14:20	25.03.2015 16:25	ESB	IST
		25.03.2015 23:05	25.03.2015 17:25	25.03.2015 22:35	IST	ALA
Dinl.						
2	26.03.2015 19:45	27.03.2015 03:25	26.03.2015 20:45	27.03.2015 02:55	ALA	IST
Dinl.						

Ekip eşlemesi oluşturma işlemi, ekip planlama sürecinin ilk ve en önemli aşamasıdır. Planlamanın başarısı ve maliyetlerin minimize edilebilmesi, tamamen bu aşamada kullanılan algoritmanın optimum ekip eşleme kümesi oluşturabilme yeteneğine bağlıdır.

1.2.4 Pas uçuş

Bir uçucu ekip eğer görevli değil de yolcu olarak uçuş yapıyor ise buna pas uçuş denmektedir. Pas uçuşlar ile özellikle, ekibin başka bir uçuşu icraa etmek için

sözkonusu uçuşun başlama noktasına veya bitiş noktasına gitmesi gerektiği durumlarda karşılaşılır.

Pas uçuşlar, yolcu taşıma kapasitesini düşüren ve ekibin verimli kullanımını azaltan bir unsurdur. Bundan dolayı planlama sürecinde daima pas uçuş sayısının minimize edilmesi istenir.

1.2.5 Uzun menzil uçuşu (ER flight)

Uzun menzil uçuşu, 4 saatten daha fazla zaman dilimi geçilen bir uçuş görevini veya tek inişli uçuş için 8 saat veya daha fazla süren uçuşu, iki inişli uçuşlarda ise uçuşlardan en az birisinin süresi 6 (dahil) saati geçmesi ve bu iki uçuşun sürelerinin toplamının 8,5 (dahil) saati geçmesi durumunda ortaya çıkan uçuş görevini ifade eder (SHGM Sivil havacılık Talimatı, 2014).

1.3 Problem Tanımı

Ekip eşleme problemindeki temel amaç, uçuşlara atanacak olan ekip elemanlarının maliyetlerinin minimize edilmesidir. Her uçucu ekip elemanı bir ikamet merkezinde oturmakta ve atanacağı ekip eşlemelerinin de bu ikamet merkezinden başlayıp tekrar aynı merkezde bitmesi gerekmektedir.

Ekip eşlemelerinin yönetmeliklere uygun (legal) olabilmesi için birtakım kısıtlamalar göz önüne alınmalıdır. Bu kısıtlamalardan bazıları zamansal ve konumsal kısıtlamalar iken diğerleri ise Sivil Havacılık Yönetmeliğinden (SHGM Sivil havacılık Talimatı, 2014) ve/veya havayolu şirketinin uyguladığı özel uygulamalardan gelmektedir.

Zamansal kısıtlamalar

Ekip eşlemesindeki bir uçuş bacağının kalkış saati ondan önceki uçuş bacağının bitiş saatinden sonra yer almak zorundadır. Aynı zamanda önceki uçuş bacağının bitişi ile sonraki uçuş bacağının kalkış saati arasında yönetmelikle sınırlanan bir bağlantı süresi (connection time) olmak zorundadır.

Konumsal kısıtlamalar

Ekip eşlemesinde arka arkaya gerçekleştirilen iki uçuş bacağının ilkinin varış yeri ile, ondan sonraki uçuş bacağının kalkış yeri aynı olmak zorundadır. Aynı zamanda ekip

eşlemesinin ilk uçuş bacağının kalkış yeri ile son uçuş bacağının iniş yeri de aynı ikamet merkezi olmalıdır.

Yönetmelik kaynaklı kısıtlamalar

Bu kurallar temelde, minimum ve maksimum bağlantı süresi, maksimum uçuş süresi, maksimum uçuş görev süresi, minimum dinlenme süresi, maksimum ekip eşleme süresi gibi limitleri belirlerler.

1.4 Ana (Master) Model

Literatürde havayolu eşleme problemi küme-kapsama veya küme-ayırma modelleri ile modellenmektedir. Bu çalışmada, sözkonusu iki modellerden küme-kapsama modeli kullanılmıştır. Bunun temel sebebi ekip eşleme problemi ile küme-kapsama modeli arasında çok net bir analoji bulunmasıdır. Küme-kapsama modeli, ekip eşleme probleminin modellenmesinde gereken her türlü temsil ihtiyacını karşılayabilmektedir.

Analojik olarak küme-kapsama modelindeki her satır havayolu tarifesindeki bir uçuşa ve her kolon ise üretilmiş bir ekip eşlemesine tekabül etmektedir. Her uçuş en az bir ekip eşlemesi tarafından kapsanmalıdır. Birden fazla kez kapsanan uçuşlar ise pas (deadhead) uçuşları temsil etmektedirler.

F havayolu tarifesinde bulunan uçuşların kümesi ve P üretilen ekip eşlemelerinin kümesi olduğu varsayılırsa, sözkonusu ekip eşleme problemi için küme-kapsama modeli aşağıdaki gibi formüle edilir (Barnhart ve diğ, 2003):

$$Min \sum_{p \in P} c_p \, x_p \tag{1.1}$$

$$\sum_{p:i \in p} x_p \geq 1 \qquad i \in F \tag{1.2}$$

$$x_p \in \{0, 1\} \qquad p \in P \tag{1.3}$$

Denklem 1.1'de, c_p, P kümesinde bulunan tüm ekip eşlemelerinin maliyet değerlerini, x_p, herhangi bir ekip eşlemesinin çözüm kümesinde olup olmadığını gösteren karar değişkenlerini gösterir. Böylece Denklem 1.1 çözümün toplam maliyetini vermektedir. Denklem 1.2, F kümesindeki tüm uçuşların kapsanmasını garanti eden eşitsizlik kısıtlamasıdır. Denklem 1.2'nin herhangi bir uçuş için 1'den

daha büyük bir değer vermesi, sözkonusu uçuşun pas uçuş olduğunu göstermektedir. Denklem 1.3 ise küme-kapsama problemindeki karar değişkenlerinin 0 ile 1 arasında değer alabileceğini gösteren standart kısıtlamaları temsil eder. Türk Hava Yolları'nda sadece bir adet ikamet merkezi (İstanbul) bulunmaktadır. Fakat birden fazla ikamet merkezi (hub) olması durumunda, her ikamet merkezinin iş gücü kapasitesi oranında iş yükü dağılımı yapılması gerekeceğinden amaç fonksiyonuna ek terimler eklemek gerekecektir.

2. PROBLEMİN GENETİK ALGORİTMA İLE ÇÖZÜMÜ

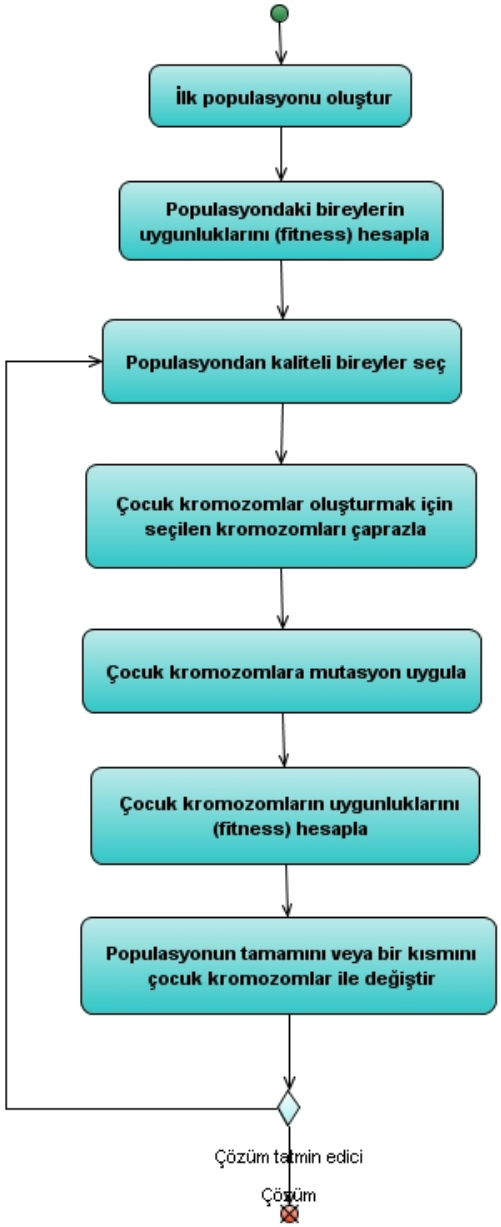
Metasezgiseller (metaheuristics), literatürde ekip eşleme probleminin çözümü konusunda en çok çalışılan başlıklardan birisidir. Tabu araması, parçacık sürü optimizasyonu (particle-swarm optimization), karınca kolonisi optimizasyonu (ant-colony optimization), genetik algoritmalar en çok üzerinde çalışılan tekniklerdendir. Çözüm tekniklerinin değerlendirilmesi sürecinde daha yaygın kullanıldığı için ekip eşleme probleminin öncelikle genetik algoritmalar kullanılarak çözülmesi konusunda geliştirilen yöntemler incelenmiştir. Bu çerçevede, küme-kapsama problemini kolon tabanlı gösterim ile çözen Beasley ve Chu (1996), ekip eşleme problemini kolon tabanlı gösterim ile çözen Kornilakis ve Stamatopoulos (2002), ekip eşleme problemini özgün bir kolon tabanlı kromozom tasarımı ile çözen Park ve Ryu (2006), küme-kapsama problemini satır tabanlı gösterim ile alt problemlere parçalayarak (divide and conquer) çözen bir strateji geliştiren Hernandez (1996), küme-kapsama problemini satır tabanlı gösterim ve ek sezgiseller kullanarak çözen Aickelin (2002) yine ekip eşleme problemini bir uçuş çizgesi (flight graph) üzerinde modelleyerek çözen Özdemir ve Mohan (2001) gibi yayınlar üzerinde çalışılmıştır. Sözkonusu referanslardan yola çıkılarak farklı kromozom modelleri ve GA konfigurasyonları üzerinde kod geliştirmeleri yapılmıştır. Bu bölümde detayları anlatılan nihai GA konfigurasyonu, ek sezgiseller ile güçlendirilmiş ve test sonuçları Zeren ve Ozkol (2012) çalışmasında sunulmuştur.

2.1 Genetik Algoritmalar

Genetik algoritmalar, bir çok farklı optimizasyon problemine uyarlanabilen, olasılıklara dayanarak (probabilistic) çalışan, "zeki" arama-tarama algoritmaları olarak tanımlanabilirler (Beasley ve Chu, 1996). Teorik temelleri Holland tarafından ortaya atılmıştır (Holland, 1975). Genetik algoritmalar, doğadaki biyolojik organizmaların, geçirdikleri evrimsel süreçlerden esinlenilerek geliştirilmiştir. Evrimsel süreç ilerledikçe, populasyon doğal seleksiyon prensibine göre evrimleşerek gelişmektedir. Populasyon içinde, diğerlerine nazaran, çevrelerine daha

iyi adapte olmuş olan bireyler daha yüksek yaşama ve çoğalma şansına sahiptirler ve daha sağlıksız bireyler elenirler. Bu şekilde evrimsel süreç boyunca sağlıklı olan bireylerin genleri yani taşıdıkları bilgi nesiller boyunca bir sonraki nesile aktarılır. Sağlıklı bireylerin gen kombinasyonları, kendilerinden daha sağlıklı yeni bireyler oluşmasını sağlayabilmekte ve bu sayede evrimsel sürec boyunca populasyonda daha sağlıklı bireyler oluşturulabilmektedir.

Genetik algoritmalar, rastgele üretilen ilk populasyon ve ondan sonra gelen yeni nesiller üzerinde genetik operatörleri uygulayarak, bu evrimsel optimizasyon sürecini simüle eder. Populasyondaki her birey, problem için mümkün bir çözümü ifade etmektedir ve probleme özgü bir şekilde kodlanarak kromozomlarla temsil edilirler. Her kromozomun yani bireyin çözüm kalitesi (fitness), kullanılan amaç fonksiyonu ile hesaplanır. Yüksek kalitede olan bireyler, genlerindeki bilgiyi, diğer yüksek kalitedeki bireyler ile çaprazlama (crossover) operatörü vasıtası ile yeni nesillere aktarırlar. Bu şekilde, ebeveynlerin genleri değişik kombinasyonlar ile çocuk nesillere aktarılır ve sürekli yeni çözümler üretilir. Lokal maksimum ve minimuma yakalanmamak için her nesilde, kromozom üzerindeki bazı genlerin değiştirilmesi yolu ile belirli bir oranda mutasyon (mutation) operatörü uygulanır. Ve en nihayetinde yeni nesil ya tamamen eski nesil ile yer değiştirilir (generational approach), ya da eski nesildeki kalitesiz olan bireyler ile yeni nesildeki kaliteli bireyler yer değiştirilir (steady-state approach). Bu süreç tatmin edici kalitede bir çözüm bulunana kadar tekrarlanır. Temel genetik algoritma yaşam döngüsü Şekil 2.1’deki gibidir.



Şekil 2.1 : Temel genetik algoritma döngüsü.

2.2 Ekip Eşleme Problemi İçin Geliştirilen GA Konfigürasyonu

Yapılan çalışmalar ve detaylı testler sonunda, Zeren ve Ozkol (2012) çalışmasında sunulan, ekip eşleme probleminin çözümünde en iyi performansı veren GA konfigürasyonu bu bölümde detaylandırılmaktadır.

2.2.1 Kromozom tasarımı

Genetik algoritmalarda küme kapsama probleminin çözümü için temelde iki çeşit kromozom tasarımı önerilmiştir. Bunlar, kolon temelli gösterim (column-based

representation) (Beasley ve Chu, 1996) ve satır temelli gösterimdir (row-based representation) (Hernandez, 1996).

2.2.1.1 Kolon temelli gösterim (Column-based representation)

Kolon temelli gösterim, yalın ve düz bir gösterim şeklidir ve bu çalışmada da kolon temelli gösterim esas alınmıştır. Üretilen ekip eşlemesi sayısının p olduğu düşünülürse, kromozomlar, 1 ve 0 değerlerini alabilen p uzunluğunda bir dizi ile temsil edilirler. Kromozom üzerindeki 1 değerleri, değerin bulunduğu indeksteki ekip eşlemesinin (kolon) çözüm kümesi içinde olduğunu, 0 ise olmadığını göstermektedir. Örnek bir gösterim Şekil 2.2'deki gibidir.

P_1	p_2	p_3	p_4	p_5	p_6	p_7	P_8
1	0	0	1	0	1	1	0

Şekil 2.2 : Kolon temelli gösterim.

Şekil 2.2'de sunulduğu gibi temsil edilen bir bireyin (kromozom) uygunluk (fitness) hesabı c_j ekip eşlemesi maliyeti x_j ise ikili karar değişkenini göstermek üzere Denklem 2.1'deki gibi hesaplanır:

$$\sum_{j=1}^{8} c_j x_j \tag{2.1}$$

Kolon temelli gösterimde, ilk populasyonun üretiminden sonraki algoritmanın daha ileri aşamalarında operasyonlar sonucunda ortaya çıkan yeni kromozomların çözüme uygun olması (feasibility) garanti edilemez. Yani algoritma çalışırken, kullanılan operatörler genler üzerinde değişiklik yaptığı sürece, tüm uçuşların (satırların) kapsanması kısıtını ihlal eden kromozomlar ortaya çıkabilmektedir. Bu sebeple, algoritmada uygun olmayan kromozomların uygun hale getirilebilmesi için bir düzeltme mekanizmasına ihtiyaç vardır. Bunu sağlayabilmek için kullanılabilecek en kolay yol, uygun olmayan tüm kromozomları çözüm kümesi dışına atmaktır ki bu şekilde hareket edilirse hiç uygun sonucun bulunamaması riski doğar. İkinci yol ise bu tip uygun olmayan kromozomların uygunluk fonksiyonunda yüksek ceza değerleri uygulamaktır ve Ozdemir ve Mohan (2001)'da bu yaklaşım kullanılmıştır. Son ve ideal yol ise uygun olmayan kromozomların belirli bir kurallar çerçecesinde bir onarım sürecinden geçirilerek uygun hale getirilmeleridir. Beasley ve Chu

(1996)'da, kolon temelli gösterimde kromozom onarma stratejisi uygulanarak, uygun olmayan kromozomları, uygun kromozomlar haline getirmek için bir kural tabanlı sezgisel operator (heuristic feasibility operator) uygulanmıştır.

2.2.1.2 Satır temelli gösterim (Row-based representation)

Satır temelli gösterimde, kromozomlar, uçuş (satır) sayısı kadar genden oluşmaktadır. Her genin indeksi bir uçuşu (satır), değeri ise o uçuşu kapsayan ekip eşlemesini (kolon) temsil etmektedir. Örnek bir gösterim Şekil 2.3'teki gibidir.

f_1	f_2	f_3	f_4	f_5	f_6
2	5	7	4	4	1

Şekil 2.3 : Satır temelli gösterim.

Şekil 2.3'te görüldüğü üzere, 1. uçuşu 2. ekip eşlemesi, 2. uçuşu 5. ekip eşlemesi kapsamaktadır. Satır temelli gösterimde, kolon temelli gösterimdeki uygun olmayan (infeasible) kromozomların oluşması problemi ortaya çıkmamaktadır. Fakat bu yöntemde, aynı kolon birden fazla gen tarafından temsil edilebileceğinden dolayı aynı çözümün farklı biçimlerde temsil edilebilmesi mümkün olabilmektedir. Bu da, uygunluk fonksiyonunun hesaplanmasını belirsizleştirmekte, farklı kromozomların aynı çözümü üretmesini ve farklı kromozomların aynı maliyet değerine sahip olmalarını mümkün kılmaktadır. Yöntemin diğer bir dezavantajı da, kolon tabanlı gösterimde iyi sonuç veren uniform çaprazlama operatörünün burada kullanılması durumunda fazla sayıda mükerrer satır (pas uçuş) temsili ortaya çıkarmasıdır. Bu durum, tek nokta çaprazlama (one point crossover) operatörü veya çok nokta çaprazlama (multi point crossover) operatörü ile aşılabilse de bu operatörler de hızlı yakınsama konusunda, kolon temelli tasarımda uygulanan yaklaşıma (füzyon çaprazlama) göre çok yavaş kalmaktadırlar. Aynı şekilde, kromozomda belirli genlerin değiştirilmesi ile uygulanan mutasyon operatörü de mükerrer satır (pas uçuş) temsilinin artmasına sebep olmaktadır.

2.2.2 Seçim (Selection)

GA'da seçim aşaması, çaprazlama işleminde kullanılacak ebeveynlerin seçiminin yapılması için kullanılan önemli bir aşamadır. Literatürde, ebeveynlerin uygunluk değerlerine göre seçildiği rulet (roulette wheel) seçimi, turnuva (tournament) seçimi, stokastik evrensel (stochastic universal) seçim gibi farklı yöntemler önerilmektedir.

Bu çalışmada özellikle performanslı ve hızlı bir şekilde hesaplamalarının yapılabilmesi ve diğer seçim yöntemleri ile karşılaştırıldığında benzer sonuçlar vermesi sebebi ile ikili turnuva (binary tournament) seçimi yöntemi kullanılmıştır. Bu yöntemde, populasyondan iki adet birey seçilir ve uygunluk değeri iyi olan birey birinci ebeveyn olmak üzere saklanır. Aynı işlem tekrarlanarak ikinci ebeveyn seçilir ve seçilen bu kromozomlar yeni kromozomlar üretmek üzere çaprazlama operatörüne sokulurlar.

2.2.3 Çaprazlama (Crossover) operatörü

Çaprazlama operatörü, genetik bilginin yeni nesillere aktarıldığı yani çocuk kromozomların üretildiği aşamadır. Seçim işleminden sonra seçilen yüksek uygunluk değerine sahip iki kromozom kullanılarak, çaprazlama operatörü vasıtasıyla duruma göre bir veya daha fazla yeni çocuk oluşturulabilmektedir. Beasley ve Chu (1996)'da kromozomların uygunluk (fitness) değerlerinin kullanıldığı füzyon (fusion) çaprazlama operatörü önerilmiştir. Çaprazlama operatörlerinden tek nokta, iki nokta ve uniform çaprazlama gibi literatürde önerilen diğer metodlar da denenmiş ve algoritmanın en iyi performansı, füzyon çaprazlama operatörü ile gösterdiği saptanmıştır. Çaprazlanacak olan kromozomlar $K_1[i]$ ve $K_2[i]$, uygunluk değerleri u_{K1} ve u_{K2}, populasyondaki en kötü uygunluk değeri u_{ek} ve çocuk kromozom $C[i]$ olmak üzere işlem Şekil 2.4'teki gibi gerçekleştirilmektedir.

1. $i = 1$
2. Eğer $K_1[i] == K_1[i]$ ise $C[i] = K_1[i] = K_2[i]$
3. Eğer $K_1[i] <> K_1[i]$ ise
    1. $o = (u_{K1} - u_{ek}) / (u_{K1} - u_{ek} + u_{K2} - u_{ek})$ olasılığı ile $C[i] = K_1[i]$
    2. 1 - o olasılığı ile $C[i] = K_2[i]$
4. Eğer $i < p$ ise $i = i + 1$ ve 2. adıma dön.

Şekil 2.4 : Füzyon çaprazlama operatörü.

2.2.4 Mutasyon (Mutation) operatörü

Mutasyon opertatörü, algoritmanın lokal maksimum ve minimumlara yakalanmaması için çaprazlama işlemi ile oluşturulan çocuk kromozomlarda belli bir oranda gerçekleştirilir ve sistem içinde, küçük çaplı rastlantısal bir arama/sıçrama mekanizması sağlar. Genelde kromozom üzerindeki belli sayıda genin, küçük bir olasılık ile değiştirilmesi şeklinde uygulanır. Bu çalışmada, kromozom üzerindeki her gen belirli bir mutasyon oranı (mutation rate) olasılığı ile değiştirilmektedir ve bu

oran Kornilakis ve Stamatopoulos (2002)'de önerildiği şekilde, populasyondaki en sağlıklı kromozomdaki 1 değerine sahip olan genlerin sayısının, kromozomun toplam uzunluğuna oranı ile hesaplanmıştır.

2.2.5 Uygunluk (Feasibility) operatörü

Daha önceki bölümlerde bahsedildiği üzere çaprazlama ve mutasyon operatörlerinin uygulanmasından sonra elde edilen yeni bir kromozomun uygun (feasible) olma garantisi yoktur. Dolayısıyla bu çalışmada Beasley ve Chu (1996)'da belirtildiği gibi, oluşabilecek bu tip kromozomların onarılması yoluna gidilmiştir. Buna ek olarak, algoritmayı daha efektif ve kararlı yapabilmek için bir lokal optimizasyon adımı uygulanmıştır.

Uygun olmayan bir kromozomu uygun hale getirmek için öncelikle, çözüm kümesi tarafından kapsanamamış uçuşları ve bu uçuşların kapsanabilmesi için çözüm kümesine ilave edilebilecek olan ekip eşlemelerinin belirlenmesi gerekmektedir. Eksik olan uçuşları kapsamak üzere hangi ekip eşlemesinin çözüm kümesine dahil edileceğine Denklem 2.2'deki oran ile karar verilmektedir.

$$\textit{Ekip eşlemesinin maliyeti / İçerdiği kapsanamamış uçuş sayısı} \qquad (2.2)$$

Kromozom uygun (feasible) hale getirildikten sonra aşırı kapsamaya sebep olan (redundant) ekip eşlemelerinin çözüm kümesi dışına çıkarıldığı bir lokal optimizasyon adımı uygulanmıştır. Bunun için çözüm kümesi dışına çıkarılsa dahi kromozomun uygunluğunun bozulmadığı ekip eşlemeleri tespit edilmektedir. Algoritma Şekil 2.5'teki gibi çalışmaktadır.

Algoritmada 1. ve 2. aşamalarda kapsanamayan uçuşlar tespit edilmekte, 3. aşamada kromozom uygun hale getirilmekte ve 4. aşamada ise aşırı kapsamaya sebep olan ekip eşlemeleri çözüm kümesinden çıkarılmaktadır.

F = Tüm uçuşların (satırların) kümesi

P = Tüm ekip eşlemelerinin (kolonların) kümesi

α_f = f uçuşunu kapsayan ekip eşlemekerinin kümesi, $f \in F$

β_p = p ekip eşlemesinin kapsadığı uçuşların kümesi, $p \in P$

S = Bir çözüm kümesini oluşturan ekip eşlemeleri kümesi, $S \subset P$

U = Kapsanmamış uçuşların kümesi, $U \subset F$

w_f = f uçuşunu kapsayan ekip eşlemelerinin sayısı, $f \in F$ in S

olmak üzere;

1. w_f için ilk değerleri $w_f = \left|S \cap \alpha_f\right|, \forall f \in F$ olarak ayarla.
2. U için ilk değerleri $U = \left\{f \mid w_f = 0, \forall f \in F\right\}$ olarak ayarla.
3. U kümesi içindeki her f için:
   a. $c_p \Big/ \left|U \cap \beta_p\right|$ oranını minimize eden ilk $p \in \alpha_f$ ekip eşlemesini bul.
   b. $S = S + p$; $w_f = w_f + 1$, $\forall f \in \beta_p$; $U = U - \beta_p$
   c. $w_f = w_f + 1$, $\forall f \in \beta_p$
4. S kümesi içindeki her ekip eşlemesi p için
   a. Eğer $w_f > 1$, $\forall f \in \beta_p$ ise $S = S - p$; $w_f = w_f - 1$, $\forall f \in \beta_p$

Şekil 2.5 : Uygunluk operatörü.

2.2.6 Kromozom iyileştirme (Improvement) operatörü

Bu çalışmada, diğerlerinden farklı olarak, kromozom uygunluk operatörünü takiben çalıştırılmak üzere sezgisel bir kromozom iyileştirme operatörü uygulanmıştır. Bu operatörde, öncelikle çözüm kümesi içindeki bir veya daha fazla ekip eşlemesi çözüm kümesinden çıkartılarak kromozom uygun olmayan (infeasible) hale getirilir ve ardından uygunluk operatörüne benzer şekilde, çıkartılan ekip eşlemesi yerine konabilecek maliyeti düşüren bir veya birden fazla alternatif ekip eşlemesi olup olmadığı aranır. Kromozom uygun olmayan hale getirildiği için bu işlem bir nevi bozuntu (perturbation) tekniği olarak düşünülebilir. Eğer arama başarılı ise alternatif olan ekip eşlemeleri çözüm kümesi içine alınır, arama başarısız olur ise kromozom ilk durumdaki haline getirilir.

Bu operatör, üretilmiş olan çocuk populasyonundaki rastgele seçilen bir kromozoma, uygunluk operatörünü takiben uygulanmaktadır. Ve yapılan testler sonucunda, bu operatörün uygulanması sayesinde çok daha az iterasyon ile çok daha

iyi noktalara ulaşıldığı ve aynı zamanda genetik algoritmanın rastlantısal doğasına rağmen her çalıştırmada istikrarlı (kararlı) sonuçlar elde edildiği görülmüştür.

2.2.7 Populasyon yenileme (Population replacement)

Genetik algoritmadaki her iterasyonun son adımı populasyon yenileme aşamasıdır. Bu aşamada genetik operatörler ile üretilen ve iyileştirilen yeni çocuk kromozomların yeni populasyona nasıl aktarılacağına karar verilmektedir. Bu çalışmada elitist yaklaşım uygulanmıştır ve aşağıdaki gibi çalışmaktadır.

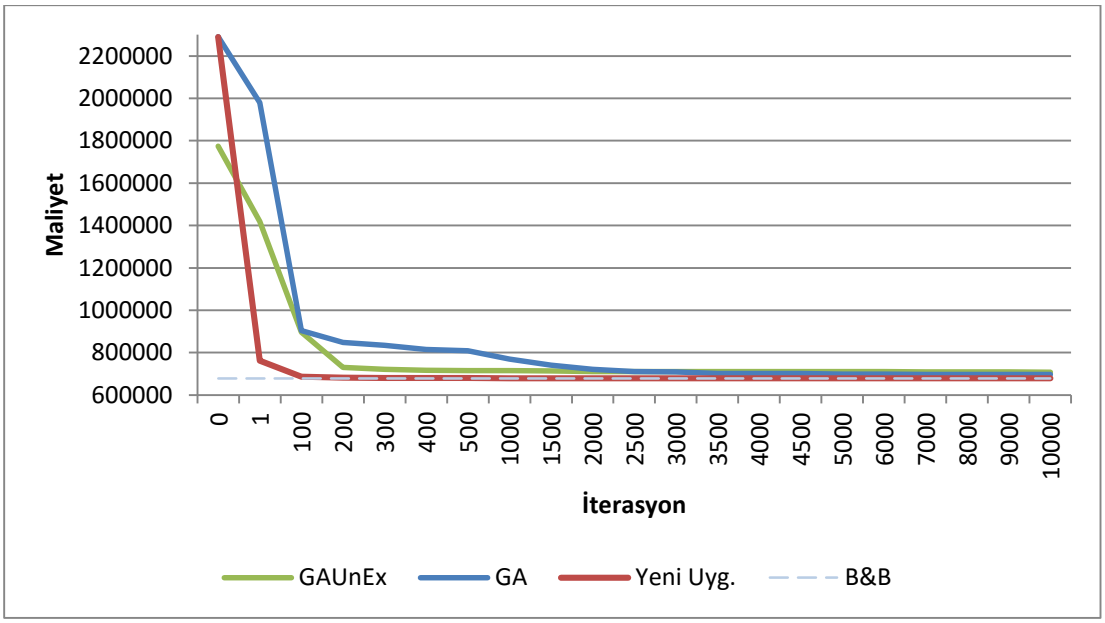
1. Çocuk populasyon ile ebeveyn populasyon bir araya toplanır.
2. Uygunluk fonksiyonlarına göre dizilir.
3. En iyi ilk 20 birey bir sonraki iterasyon için saklanır kalanlar ise atılır.

Elitist yaklaşım dışında, tüm çocukların populasyondaki tüm ebeveynler ile doğrudan yer değiştirildiği nesilsel yaklaşım (generational approach) ve her çocuk kromozomun uygunluk derecelerine göre rastgele seçilen bir ebeveyn ile yer değiştirildiği steady-state yaklaşım da test edilmiştir. Fakat bu çalışmada elitist yaklaşımın diğer yöntemlere göre daha iyi sonuç verdiği görülmüştür.

2.2.8 Deneysel sonuçlar ve değerlendirme

Bu çalışmada test amaçlı kullanılan uçuş verisi, 714 uçuşa sahip olan Türk Hava Yolları A310 filosunun tarife verisinden alınmıştır. Elde edilen sonuçlar ise Java platformunda geliştirilen bir kod ile üretilmiş ve daha detaylı bir şekilde Zeren ve Ozkol (2012) çalışmasında sunulmuştur.

Çözümde IST merkezli olmak üzere toplam 3646 adet ekip eşlemesi üretilmiştir. Sırasıyla; Park ve Ryu (2006) (GAUnEx), Beasley ve Chu (1996) (GA), yeni operatörler ile geliştirilmiş GA (Yeni Uyg.) ve Dal-sınır algoritması çözümlerinden elde edilen sonuçların karşılaştırılması Şekil 2.6’daki gibidir.



Şekil 2.6 : İterasyon ve maliyet değişimi karşılaştırması.

Şekil 2.6'da görüleceği üzere geliştirilmiş GA çok daha az iterasyon ile ve çok daha kısa zamanda optimum noktaya yakınsayarak diğer uygulamalara göre yaklaşık dört kat daha iyi performans sağlamaktadır. Ayrıca onbin iterasyon sonunda (GA) ile problemin global optimum değerini veren matematiksel dal-sınır (B&B) algoritması arasında % 2,6 lık bir fark elde edildiği halde bu çalışmada geliştirilen GA (Yeni uyg.) ile dal-sınır (B&B) algoritması arasında sadece % 0.036 lük bir fark bulunmuştur. Bu sonuçlar, ek operatörler ile geliştirilen GA (Yeni uyg.) yaklaşımının benzer büyüklükteki filolar için matematiksel metodları aratmayacak performansta ve kalitede çözüm üretebilme yeteneğine sahip olduğunu göstermektedir.

2.3 Paralel Genetik Algoritma Konfigürasyonu

Geliştirilen GA konfigurasyonu aylık 714 uçuşa sahip küçük bir filo için yüksek kaliteli çözümler üretse de çözülmek istenen problemin boyutu büyüdükçe problemin arama uzayı da üssel olarak büyümekte ve bu durum GA cephesinde çok daha büyük popülasyonlar kullanmayı gerektirmektedir. Bu bağlamda, optimum sonuca hızlı ulaşabilmek için, çok işlemcili makinalarda paralel algoritmalar kullanarak problemin çözümü, önemli bir araştırma alanı haline gelmektedir. Paralelleştirme çalışmasına yönelik birden fazla popülasyonun aynı anda koşması, popülasyonlar arası birey transferi gibi düşük seviye geliştirmeler yapılmıştır. Paralel genetik

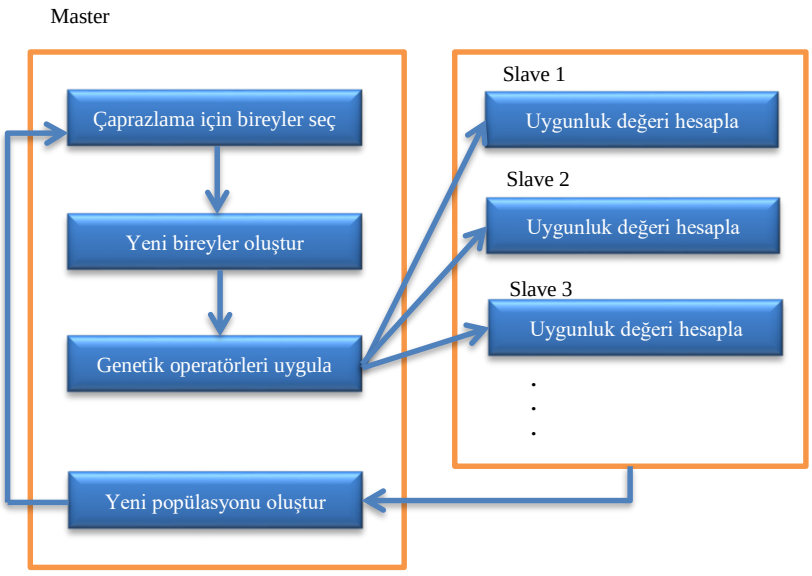
algoritmalar temelde üç farklı mimaride tasarlanırlar (Luque ve Alba, 2011; Cant´u-Paz, 2007).

- Global populasyon modeli (Master-Slave)
- Ada modeli (Island/Coarse-Grained)
- Hücresel (Cellular/Fine-Grained)

2.3.1 PGA modelleri

2.3.1.1 Global populasyon modeli (Master-Slave)

Bu model, büyük bir populasyonun yüksek işlem gücü isteyen uygunluk fonksiyonu hesaplaması gibi işlemlerinin birden fazla işlemciye (CPU) dağıtılması prensibine dayanır. Bu yapıda master bileşen, genetik algoritmanın koştuğu ana bileşendir. Burada populasyon üzerindeki fitness fonksiyon değerlendirme dışındaki crossover, mutasyon gibi tüm genetik algoritma işlemleri gerçekleştirilmektedir. Fitness fonksiyon hesaplama işlemleri ise farklı işlemcilerde koşan slave bileşenler kullanarak çözülmektedir.



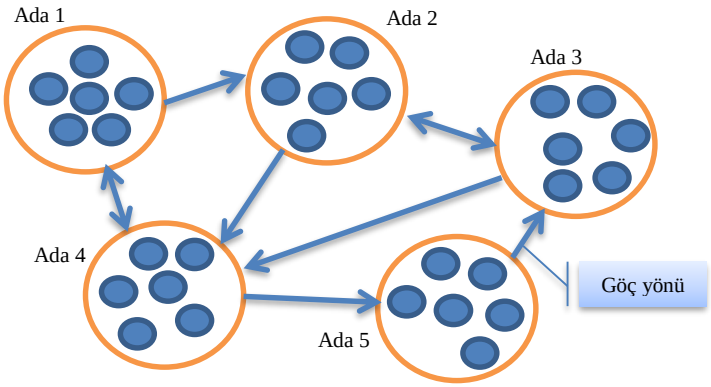
Şekil 2.7 : Global populasyon modeli (Master-Slave).

Master-slave model uygulaması en kolay paralel genetik algoritma modelidir. Fakat yapısı gereği algoritma koşarken çok yüksek bir frekansta slave düğümlere erişmesi gerekeceğinden ağ üzerinde çalışması istendiğinde oluşturacağı trafik ve işlem yükü büyük olmaktadır. Dolayısıyla ağ üzerindeki kullanımlarda efektif değildir. Sadece

karmaşık ve hesaplaması zor uygunluk fonksiyonlarının kullanıldığı problemlerde ve çok büyük ölçekte paralel (massively parallel) sistemlerde verimli birşekilde çalışabilir (Luque ve Alba, 2011; Cant´u-Paz, 2007).

2.3.1.2 Ada modeli (Island/Coarse-Grained)

Bu modelde, her populasyon bir ada olarak tanımlanmakta ve birden fazla populasyon (adalar) birbirleri ile kromozom alışverişi yaparak optimum çözüme yaklaşmaktadırlar. Bu model, en popüler ve üzerinde en çok çalışılan modeldir. Algoritmanın çalışmasını etkileyen iki temel parametre göç oranı (migration rate) ve göç frekansıdır (migration frequency). Bu parametreler için kullanılan değerler yakınsama hızını etkilediği gibi eğer algoritma ağ üzerinde çalışılıyor ise ağ trafiğini de etkilemektedirler. Özellikle çok sunuculu yapıyı daha esnek desteklediği için bu çalışmada bu model kullanılmıştır.



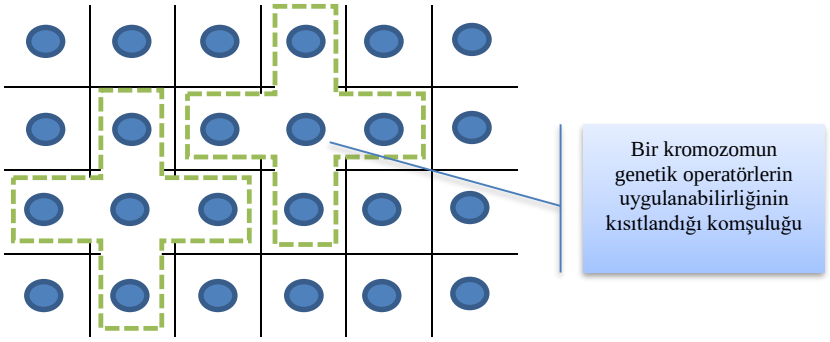
Şekil 2.8 : Ada modeli (Island/Coarse-Grained).

Bu modeldeki bir diğer önemli kavram da göç topolojisidir (migration topology). Bu kavram hangi populasyonların birbirlerine kromozom göç ettireceği ile ilgili bir çizgeyi (graph) ifade etmektedir. Tam bağlı çizge (full connected graph) en iyi performası vermektedir (Luque ve Alba, 2011; Cant´u-Paz, 2007).

2.3.1.3 Hücresel model (Cellular/Fine-Grained)

Bu modelde uzaysal olarak yapılandırılmış (spatially structured) tek bir populasyon bulunmaktadır. Büyük ölçekte paralel (massively parallel) sistemlerde çalışmak üzere tasarlanmış bir paralel işlem yapısıdır. Bu modelde populasyon içindeki kromozomlar iki boyutlu bir düzlem üzerindeymiş gibi konunlandırılırlar ve seleksiyon, crossover gibi işlemler iki boyutlu düzlemdeki yakın komuşuluklara sahip küçük gruplarla sınırlandırılmaktadır. Dolayısıyla optimizasyon işlemi

süresince gerçekleşen bir iyileşme öncelikle, iki boyutlu düzlem üzerindeki yakın komşular ile başlayıp tüm populasyona yayılmaktadır. Bu model üzerinde az çalışılan bir modeldir ve master-slave gibi ağ üzerinde uygulaması zordur (Luque ve Alba, 2011).



Şekil 2.9 : Hücresel model (Cellular/Fine-Grained).

2.3.2 Paralel genetik algoritma uygulaması

Önceki bölümlerde de bahsedilen nedenler ve popularitesinden dolayı bu çalışmada ada modeli kullanılmıştır. Geliştirilen mimari, birden fazla işlemciye sahip olan birden fazla sunucuda çalışabilir şekilde tasarlanmıştır.

Genetik algoritmalarda, kromozom tasarımından sonraki en önemli parametrelerden biri populasyon büyüklüğüdür. Genetik algoritmalardan istenilen performansın alınabilmesi için kullanılması gereken populasyon büyüklüğü parametresi ile ilgili çeşitli çalışmalar yapılmıştır. Harik ve diğ. (1999) çalışmasında, çözülmek istenen problemin karakteristik özelliklerine göre istenen performansın yakalanabilmesi için gereken populasyon büyüklüğünün hesaplandığı bir dizi formül sunulmuştur. Fakat havayolu ekip planlama probleminde, optimizasyon probleminin boyutları ve dolayısıyla kromozom boyutları çok büyük olduğundan bu tip fonksiyonlar ile elde edilen değerler pratikte uygulanması mümkün olmayan sayılara ulaşmaktadır. Dolayısıyla populasyon büyüklüğü ve sayısı donanımın imkan verdiği ölçüde büyük tutulmalıdır. Yine Luque ve Alba (2011) ve Cant´u-Paz (2007) referanslı çalışmalarda, paralel genetik algoritmalarda parametre tercihi ile ilgili detaylı çalışmalar sunmuştur. Bu çalışmalarda populasyon büyüklüğünün tespitine ek olarak kaç adet kromozomun populasyonlar arasında transfer edileceğini belirten göç oranı (migration rate), kaç iterasyonda bir göç işleminin tetikleneceğini belirten göç frekansı (migration frequency) ve populasyon bağlantı şekilleri (migration topology) konularında da çeşitli formulasyonlar sunulmuş ve test sonuçları paylaşılmıştır.

Tongchim ve Chongstitvatana (2000) ise ek olarak göç senkronizasyonu ile ilgili sonuçlar sunmakta ve asenktron göç uygulaması önermektedirler.

Yukarıdaki belirtilen kaynaklardan hareketle kendi çalışmamızda tam bağlı (fully-connected) bir populasyon kümesi kullanılmış ve bu popülasyonlar arasında asenkron göç (asynchronous migration) uygulanmıştır. Göç oranı (migration rate) değeri ise bazı testleri müteakip 4 olarak belirlenmiştir. Bu değerin arttırılması, aynı iterasyon sayısında daha başarılı sonuç verse de optimizasyon esnasında kullanılan ağ yükü ve göç işlemlerine harcanan efor toplam işlem süresini artmaktadır. Göç frekansı (migration rate) için de, yapılan testler sonucunda 20 civarında bir değerin kullanılmasının performans ve kalite açısından daha verimli olduğu gözlenmiştir.

2.3.3 PGA test sonuçları ve değerlendirme

Çalışmanın PGA kısmında Türk Hava Yolları'nın iki büyük filosu olan Airbus 320 ve Boeing 737 filolarının verileri kullanılmıştır. Türk Hava Yolları'nın bu filoları, hem uçak sayısı hem de uçuş sayısı açısından büyük bir gerçek dünya optimizasyon problemi modeli örneğidir.

Çizelge 2.1 : PGA verisetleri.

Veriseti	Uçuş sayısı	Ekip eşlemesi sayısı
Airbus 320	7251	341372
Boeing 737	13379	1102426

Airbus 320 filosunun kapsanan zaman periodu içinde 7251 adet uçuşu vardır. Ve bu uçuş listesi kullanılarak problemin çözümü için 341372 adet ekip eşlemesi üretilmiştir. Aynı şekilde Boeing 737 filosunun da 13379 uçuşu için 1102426 adet ekip eşlemesi üretilmiştir. Ekip eşlemelerinin üretimi konusunda Zeren ve Özkol (2012) çalışması incelenebilir.

Çizelge 2.1'deki iki veriseti karşılaştırıldığında görülmektedir ki, uçuş sayısı arttıkça üretilen ekip eşlemesi sayısı da artmakta ve milyonlu sütun sayılarına ulaşan çok daha büyük ve seyrek (sparse) bir optimizasyon problemi ortaya çıkmaktadır.

Detayları önceki bölümlerde belirtilmiş olan PGA çalışmasının sonuçları İstanbul Teknik Üniversitesi bünyesinde bulunan Ulusal Yüksek Başarımlı Hesaplama Merkezi'nin desteği ile üretilmiştir. Geliştirilen paralel genetik algoritma kodu her iki veriseti için de ayrı ayrı UYBHM sisteminde 20 işlemciye kadar 3 er kez 30

dakika boyunca çalıştırılmış ve elde edilen sonuçların ortalaması aşağıda sunulmuştur.

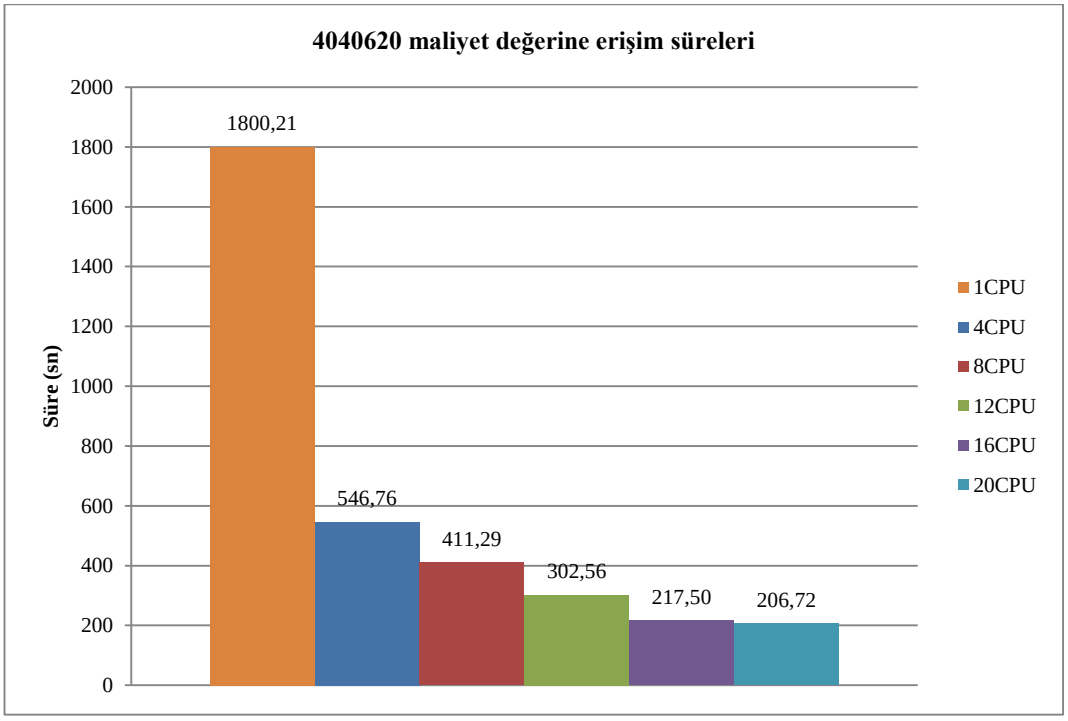
2.3.3.1 320 filosu PGA sonuçları

Çizelge 2.2'de görüleceği gibi geliştirilen PGA kodu 320 filosu için 20 CPU ile 30 dakikalık bir çalışma sonucunda %7,7 lik bir fark ile dal-sınır algoritması ile bulduğumuz optimum değere yaklaşmıştır.

Çizelge 2.2 : 320 filosu PGA sonuçları.

CPU	Itr	Ekip eşlemesi sayısı	Pas sayısı	Maliyet	Fark
1	4119	2337	358	4040620	0.196
4	3561	2327	274	3917287	0.16
8	3223	2262	220	3773538	0.117
12	3823	2225	232	3720356	0.101
16	3953	2242	230	3697594	0.095
20	4363	2169	207	3636743	0.077

Farklı sayıdaki CPU kullanımı ile ilgili değişim ise Şekil 2.10'daki gibidir. Şekil 2.10'daki grafik, 320 filosu verisetinin çözümünde 1 CPU ile 30 dakika içinde erişilen maliyet değerine 4, 8, 12, 16 ve 20 CPU kullanıldığında ne kadar sürede elde edildiğini göstermektedir. Aşağıdaki sonuçtan görüldüğü üzere Tongchim ve Chongstitvatana (2000) de yayınlandığı gibi lineer üstü bir hızlanma olmasa da 20 CPU kullanıldığında aynı değere 6 kat daha kısa sürede elde erişilebilmektedir.



Şekil 2.10 : Farklı CPU sayıları ile 320 çözümü performans karşılaştırması.

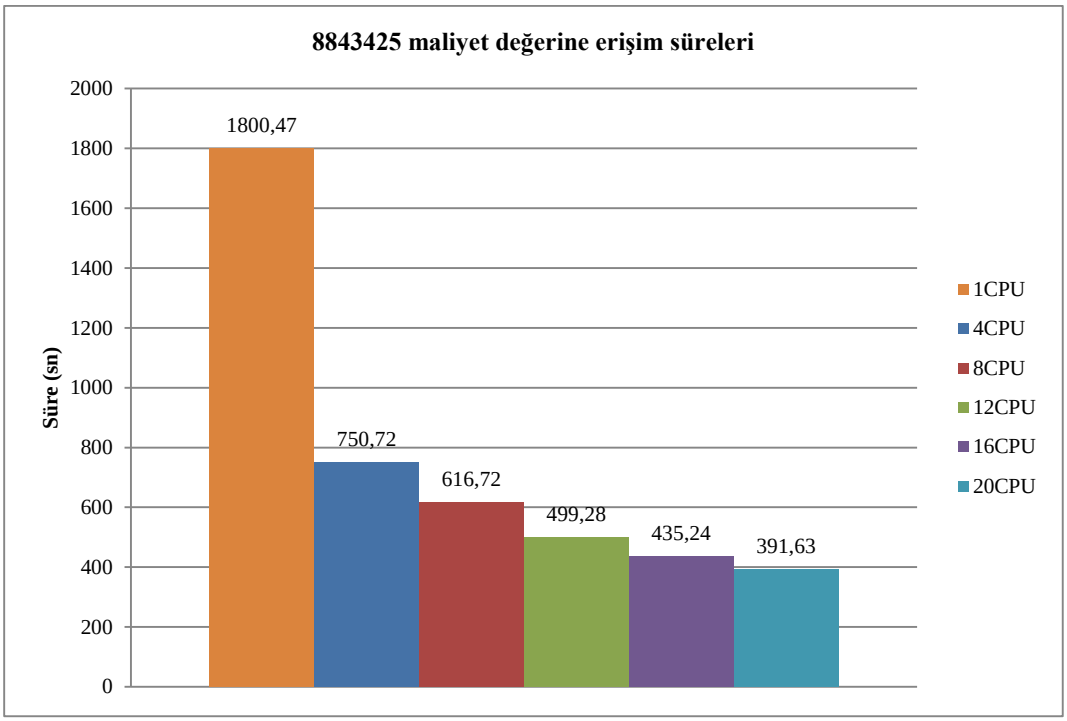
2.3.3.2 737 filosu PGA sonuçları

737 filosu için ise 20 CPU ile 30 dakikalık bir çalışma sonucunda 13% lük bir fark ile dal-sınır algoritması ile bulunan optimum değere yaklaşmaktadır.

Çizelge 2.3 : 737 filosu PGA sonuçları.

CPU	Itr	Ekip eşlemesi sayısı	Pas sayısı	Maliyet	Fark
1	1495	3338	1545	8843425	0.21
4	1099	3337	1434	8629867	0.18
8	1333	3351	1318	8498654	0.162
12	1118	3344	1317	8452680	0.156
16	1695	3367	1275	8386595	0.147
20	1754	3366	1226	8266532	0.131

Aynı şekilde 737 filosu için farklı sayıda CPU kullanımı ile ilgili değişim ise Şekil 2.11'deki gibidir.



Şekil 2.11 : Farklı CPU sayıları ile 737 çözümü performans karşılaştırması.

Sonuçlardan da görüldüğü üzere büyük filolar ile yapılan testlerde paralelleştirilmiş GA, tek kanallı uygulamaya göre gözle görülür bir iyileşme sağlasa da, matematiksel dal-sınır metodu ile elde edilen sonuçlara en fazla % 7.7 oranında yaklaşabilmiştir. Bu nedenle problemin çözümünde matematiksel kolon türetme temelli yöntemlerin halen en cazip ve kaliteli çözüm üretebilen teknikler olduğu görülmektedir. Büyük filolar için kesin (exact) çözüm üreten teknikler çok uzun sürelerde sonuç üretebiliyor olsa da bu çalışmada önerilen stratejide olduğu gibi kolon türetme

tekniğinin sezgisel yaklaşımlar ile harmanlanarak kullanımı ile elde edilen yaklaşıksal (approximal) metodlar global optimuma çok yakın yüksek kalitede sonuçları kabul edilebilir bir süre içinde üretebilmektedirler.

3. ARKAPLAN

3.1 Doğrusal Programlama (Linear Programming)

Doğrusal programlama (DP) bu çalışmada kullanılan ana yaklaşım olan kolon türetme tekniğinin dayandığı temel disiplinlerden birisidir. Doğrusal programlama, doğrusal denklemler ve eşitsizlikler kullanılarak ifade edilebilen optimizasyon problemleri için en uygun çözümleri bulmak amacıyla kullanılan matematiksel bir tekniktir. Eğer bir gerçek dünya problemi doğru bir şekilde doğrusal bir programın matematiksel denklemleri olacak şekilde ifade edilebilirse DP problem için en iyi çözümü bulur. Az sayıda karmaşık gerçek dünya problemi doğrusal bir fonksiyon grubu olarak doğru bir şekilde ifade edilebilmektedir. Yine de doğrusal programlar, matematiksel fonksiyonların yaratıcı bir üslup ile kullanılması sayesinde bir çok gerçek dünya problemi için oldukça makul, gerçekçi bir temsil imkanı sağlamaktadır. (Lasdon, 2002)

Bir doğrusal program; bir dizi değişkenden, her değişkenin istenilen çıktıdaki katkısını gösteren doğrusal bir amaç fonksiyonundan ve değişkenlerin sınırlarını belirleyen doğrusal kısıtlamalardan oluşur. Doğrusal bir programın sonucu, uygulanan kısıtlamaların çizdiği sınırlar içinde amaç fonksiyonundan en iyi çıktıyı üretebilecek değişken değerleri kümesidir. Bir problem doğrusal program olarak formüle edildikten sonra, bir bilgisayar programı vasıtası ile çözülebilir.

3.1.1 Doğrusal eşitlikler (Linear equations)

Bir doğrusal programdaki tüm eşitlikler ve eşitsizlikler tanımı gereği doğrusal olmalıdır. Örnek doğrusal bir fonksiyon Denklem 3.1'deki biçimdedir.

$$\sum_{i=1}^{n} a_i x_i \tag{3.1}$$

Genel olarak a'lar katsayılar veya parametreler olarak isimlendirilirler. Çözülen problemin doğasına uygun sabit değerler alırlar. Atama veya kapsama tipi

problemlerde sadece 0 veya 1 gibi sadece tamsayı değerler aldıkları gibi bir üretim optimizasyonu probleminde üretim için gerekli hammadde miktarları gibi büyük değerler de alabilirler. *x*'ler ise karar değişkenleri olarak adlandırılırlar ve kısıtlamaların belirlediği sınırlar içerisinde değerler alabilirler.

3.1.2 Karar değişkenleri (Decision variables)

Karar değişkenleri, bir doğrusal programın çözülmesi için tespit edilmesi gereken değerler kümesidir. Doğrusal problemler, bu değişkenlerin alabileceği en iyi değerlerin tespit edilmesi ile çözülür. Karar değişkenlerinin alabileceği değer aralıkları yine doğrusal kısıtlamalar ile belirlenir. Atama veya kapsama tipi problemlerde 0 ve 1 arasında bir değer alabildikleri gibi bir üretim optimizasyonu probleminde üretim için gerekli ürün miktarı gibi büyük gerçel değerleri de temsil edebilirler.

3.1.3 Amaç fonksiyonu (Objective function)

Doğrusal bir problemin nihai amacı sayısal bir değerin minimize veya maksimize edilmesidir. Bu değer minimize edilmesi gereken bir maliyet değeri olabileceği gibi maksimize edilmesi gereken bir karlılık değeri de olabilir. Amaç fonksiyonu her değişkenin optimize edilmesi istenen değeri nasıl etkilediğini gösteren doğrusal bir ifadedir.

$$\text{Max/Min} \qquad \sum_{i=1}^{n} c_i x_i \tag{3.2}$$

Denklem 3.2'deki genelleştirilmiş amaç fonksiyonu ifadesinde *x*'ler karar değişkenlerini, *c*'ler ise her karar değişkeninin bir biriminin amaç fonksiyonuna olan katkısını (gelir, maliyet vb.) gösterir.

3.1.4 Kısıtlamalar (Constraints)

Kısıtlamalar, bir doğrusal programlama problemindeki karar değişkenlerinin alabileceği değer aralıklarını tanımlarlar. Problemin tipine göre bu kullanılacak bir kaynağın (hammadde) üst miktarını veya üretilmesi gereken bir ürünün alt limitini ifade edebilirler. Kısıtlamalar Denklem 3.3'deki gibi bir genel formda gösterilirler.

$$\sum_{i=1}^{n} a_{j,i} x_i \leq b_j \qquad \text{j=1,2,...,m} \qquad (3.3)$$

Denklem 3.3'de *m* kısıt sayısını, *x*'ler karar değişkenlerini göstermektedir. *a*'lar ise her x_i değişkeninin ilgili kısıtlamaya olan etkisini göstermektedir. Örneğin bir üretim optimizasyonu probleminde *a*'lar her x_i ürününün üretilebilmesi için gerekli olan hammadde miktarını, eşitliğin sağ tarafındaki *b* değerleri ise kullanılabilecek toplam hammadde miktarı gibi bir değeri temsil edebilir.

3.1.5 Negatiflik önleyici kısıtlamalar (Non-negativity constraints)

Standart doğrusal programlama formülasyonunda, karar değişkenleri daima sadece sıfır ve üzeri (non-negative) değerler alabilir. Bu şartı sağlamak için problem tasarımında her değişkenin negatif değer almasını engellemek için Denklem 3.4'teki gibi negatiflik önleyici kısıtlamalar mutlaka kullanılmaktadır.

$$x_i \geq 0 \qquad \text{i=1,2,...,m} \qquad (3.4)$$

3.1.6 Genelleştirilmiş kanonik doğrusal programlama formu

Doğrusal programlama problemleri en genel formda cebirsel olarak Denklem 3.5 ve 3.6'daki gibi ifade edilir.

$$\text{Min } c^T x \qquad (3.5)$$

$$Ax \geq b, \qquad x \geq 0 \qquad (3.6)$$

3.1.7 Doğrusal programlama problemlerindeki temel özellikler

DP'nın altı temel varsayımı bulunmaktadır. Bunlar:

Belirlilik

Bir DP modelinde yer alan *A*, c^T ve *b* matrislerinin bilindiği ve değişmediği kabul edilir. Yani, birim başına gelir ya da maliyetlerin (c^T), her faaliyet için gerekli olan kaynak miktarlarının (*A*) ve mevcut kaynak miktarlarının (*b*) sabit olduğu varsayılır.

Doğrusallık

Bir DP modelinin amaç fonksiyonu ve kısıt denklemleri doğrusal olmalıdır. Bir başka deyişle *x*'ler birinci dereceden olmalıdır.

Negatif olmama

DP'deki tüm değişkenlerin negatif olmayan değerler alması gerekmektedir. Negatif üretimden söz edilemeyeceği için değişkenlerin pozitif ya da en azından sıfıra eşit olması gerekmektedir.

Toplanabilirlik

Amaç fonksiyonunun ve kısıt denklemlerinin değerlerine yapılan toplam katkı, her bir katkının ayrı ayrı toplanması ile elde edilir. Örneğin, bir ürün üç metre, diğer bir ürün iki metre hammadde ile üretiliyorsa iki ürünü birden üretmek beş metre hammadde gerektirir.

Orantısallık

Her bir karar değişkeninin amaç fonksiyonuna ve kısıt denklemlerinin sol tarafına yapacağı katkı karar değişkeninin değeri ile orantılıdır. Örnek olarak bir ürünün maliyeti 2 tl ise 3 ürünün maliyeti 6 tl olacaktır.

Bölünebilirlik

Belirlilik varsayımı ile karar değişkenlerinin gerçel (reel) değerler alabileceği kabul edilir. Dolayısıyla bir DP probleminin çözümünde 3.2 adet masa üretileceği gibi bir sonuç çıkabilir. Bu tip sonuçlar DP probleminin optimal çözümünün ardından tamsayı programlama tekniklerinin kullanılması ile tamsayılaştırılır.

3.2 Doğrusal Programlama Çıktıları

Doğrusal programlama problemlerinin çözümlerinde asıl hedef en iyi karar değişkeni kümesini bulmak olsa da her çözümde önemli ek bazı yorumlanmasında fayda olan değerler de elde edilmektedir.

3.2.1 İndirgenmiş maliyet (Reduced-cost)

DP çözümleri sonucunda her değişken için bir indirgenmiş maliyet değeri de elde edilir. İndirgenmiş maliyet değeri, bir karar değişkeninin pozitif değer alabilmesi için amaç fonksiyonu üzerindeki ilgili değişkenin katsayısındaki gerçekleşmesi gereken iyileşme miktarıdır. Bir diğer ifade ile temel olmayan (nonbasic) bir değişkenin

temel (basic) bir değişken olabilmesi için amaç fonksiyonundaki katsayısının değerinde gerçekleşmesi gereken iyileşme miktarıdır.

Örneğin, DP probleminin bir üretimde gelir maksimizasyonu problemi olması durumunda karar değişkenleri üretim miktarlarını, amaç fonksiyonu katsayıları ise satış fiyatlarını ifade edecektir. Bu durumda bir ürünün indirgenmiş maliyet değeri, o ürünün üretilebilmesinin mümkün olabilmesi için birim fiyatında gerçekleşmesi gereken iyileşmeyi (artış) ifade edecektir.

Eğer çözülmek istenen DP problemi bir küme-kapsama problemi ise karar değişkenleri bağlı oldukları kolonların (ekip eşlemesi) çözüm içinde olup olmadığını gösteren 0 ile 1 arasındaki gerçel sayıları, amaç fonksiyonu katsayıları ise ilgili kolonların maliyetini ifade edecektir. Bu durumda bir kolonun indirgenmiş maliyet değeri, o kolonun çözüm içinde bulunabilmesi için maliyetinde yapılması gereken iyileşmeyi (indirim) ifade edecektir. Diğer bir değişle ilgili kolonun çözüm içine dahil edilmesi durumunda amaç fonksiyonu değerinde meydana gelecek olan artışı gösterecektir.

3.2.2 Arttıran/eksilten yapay değişken (Slack/Surplus variable)

DP çözümleri sonucunda her kısıtlama için bir arttıran/eksilten yapay değişken değeri de elde edilir. Arttıran yapay değişkenler, küçük ve eşittir kısıtlamalarında ortaya çıkarlar ve kısıtlama ifadesinden elde edilen değerin kısıtlamanın sağ tarafındaki limit değerinden ne kadar küçük olduğunu gösterirler. Örneğin bir kaynağın kullanımı limitlenmek istendiğinde elde edilen optimum çözümdeki arttıran yapay değişkenler ilgili kaynağın daha ne kadar kullanılabileceğini yani atıl kapasiteyi gösterir. Eksilten yapay değişkenler ise büyüktür ve eşittir kısıtlamalarında ortaya çıkarlar. Örnek olarak bir üretim maliyeti optimizasyonunda üretilecek minimum ürün miktarı için bir değer girildiğinde ilgili eksilten yapay değişken minimum seviyenin ne kadar üzerinde üretim yapılacağını gösterecektir.

3.2.3 Gölge fiyat (Shadow-price/Dual variable)

Gölge fiyat değerleri, kolon türetme yaklaşımları için kritik olan en önemli DP çıktılarından birisidir. DP sonunda her kısıtlama için bir gölge fiyat (dual) değeri elde edilir. Gölge fiyat değeri bir kısıtlamanın sağ taraftaki limit değerinin bir birim arttırılması ile amaç fonksiyon değerinde meydana gelen değişimi ifade eder. Bir

maliyet minimizasyonu problemi için üretim kısıtlarından birini bir birim arttırdığımızda ilgili kısıtlamanın gölge fiyat değeri, amaç fonksiyonundaki toplam artışı (maliyet) verir. Bir gelir maksimizasyonu problemi için ise kullanılacak bir kaynağın miktarını sınırlayan bir kısıtlamanın sağ taraftaki limit değerinin bir birim artışı ile amaç fonksiyonunda ortaya çıkacak artışı gösterir. DP modelindeki her karar değişkeni için tanımlanan negatiflik önleyici (non-negativity) kısıtlamalara bağlı olan gölge fiyat değerleri de ilgili karar değişkeninin indirgenmiş maliyet değerini vermektedir.

3.3 Dualite (Duality)

Her DP probleminin kendisi ile sıkı matematiksel bağlantıları bulunan bir dual (ikil) versiyonu bulunmaktadır. Duali alınan DP problemi ise primal olarak isimlendirilir. Bir maksimizasyon primal probleminin duali minimizasyon problemi, bir minimizasyon primal probleminin duali de maksimizasyon problemidir. Standart formdaki bir maksimizasyon primal problemi ile bu problemin duali Denklem 3.7, 3.8, 3.9 ve 3.10'daki gibidir.

Primal: $\text{Max } c^T x$ (3.7)

$$Ax \leq b, \qquad x \geq 0 \tag{3.8}$$

Dual: $\text{Min } y^T b$ (3.9)

$$y^T A \geq c^T, \qquad y \geq 0 \tag{3.10}$$

Primal problemdeki kısıtlama limiti vektörü b dual problemde amaç fonksiyonu katsayıları ve primal problemdeki amaç fonksiyonu katsayıları vektörü c ise dual problemde kısıtlama limit değerleri olarak denklemlere yansır. Primal problemdeki kısıtlama katsayı matrisi A ise dual problemde transpozesi alınarak kullanılır. Bir DP probleminin iki kez duali alınırsa tekrar problemin kendisi elde edilir. Eğer primal problem sınırlı değil ise (unbounded) dual problem olurlu değildir (infeasible), dual sınırlı değil ise primal olurlu değildir. Standart formdaki bir maksimizasyon primal problemdeki küçük eşit kısıt denklemleri dual problemde yön değiştirerek büyük eşit

kısıtlara dönüşür. Minimizasyon primal problemindeki büyük eşit kısıtlar da dual problemde yön değiştirerek küçük eşit kısıtlara dönüşür.

Bir DP probleminin dualinin alınması problemin çözümünü kolaylaştırmasından ziyade, problemin ekonomik yapısı, ihtivası ve yorumlanması için gereken oldukça önemli bilgiler içermektedir. Bu bilgilerden ilki dual problemin çözümünün primal için primal problemin çözümünün de dual problem için bir limit değeri ihtiva etmesidir. Zayıf (weak) dualite teoremine göre bir primal probleminin herhangi bir uygun çözümünün *z* değeri en fazla dual problemin herhangi bir uygun çözümünün *w* değeri kadar iyi olabilir ($z \leq w$). Güçlü (strong) dualite teoremine göre ise uygun bir optimal çözümü olan primal ve dual problemlerin çözümlerinin değerleri birbirlerine eşittir. (Lasdon, 2002)

Dual problemin ikinci ve belki de en önemli çıktısı dual değişkenler y^T dir. Dual değişkenler primal probleminde bulunan her kısıt için elde edilen optimum gölge fiyat değerlerini verirler. Bu değerler özellikle primal problemdeki hangi kısıtın önem arzettiğini ve arttırılması veya azaltılması gerektiği yönünde önemli bilgiler içermektedir. Dual değişkenler (gölge fiyat) aynı zamanda bu çalışmada kullanılan kolon türetme tekniğinin hesaplama süreçlerinde de önemli rol oynamaktadırlar.

3.4 Doğrusal Programlama Çözüm Yöntemleri

DP çözümü için kullanılan en basit yöntem grafik yöntemdir. Grafik metodda DP kısıtlamaları bir eksen takımı üzerinde çizilerek olurlu (feasible) bölge işaretlenir. Ve olurlu bölge üzerindeki kısıtlamaların kesiştiği noktalar değerlendirilerek optimum nokta ve bu noktayı veren değişken değerleri belirlenir. Grafik yöntemin, DP probleminde ikiden fazla değişken olması durumunda kısıt grafiğinin çizimi ve değerlendirilmesi oldukça zor bir hal almaktadır. Bu kısıt grafik yöntemin gerçek dünya problemlerinde kullanılmasını zorlaştırmaktadır. Fakat grafik metod DP problemlerinin öğrenilmesi ve araştırılması konusunda oldukça önemli ve kullanışlı bir araçtır.

DP çözümü için kullanılan diğer bir yöntem tipi, kısıtlama politopunun sınır noktalarının üzerinde arama yaparak optimum noktayı bulmaya çalışan yöntemlerdir. Bu yöntemler bir temel çözüm noktasına ulaştıklarında bir sonraki arama yönünü belirlemek için pivot kuralları kullanırlar. Eğer bulunulan noktadaki temel çözümün

kalitesini daha iyiye götürecek yeni bir arama istikameti bulunamaz ise bulunmuş olan temel çözüm optimum çözüm olarak değerlendirilir. George Dantzig 1947 yılında bu stratejiyi kullanarak Simplex metodunu geliştirmiştir. Bu metod, grafik metoda göre çok daha üstün ve pratikte son derece iyi bir performans ile kullanılsa da polinominal zamanda çözülebilen herhangi bir pivot kuralının olup olmadığı kesin değildir. Greenberg (1997) gibi bazı çalışmalarda simplex yöntemi ile çözümünün üssel zaman gerektirdiği bazı örnekler verilmiştir. Simplex metodunun; standart, dual, revize edilmiş (revised) isimlerinde türevleri de bulunmaktadır.

Polinominal zamanlı ilk DP algoritması Shor, Nemirovsky ve Yudin'in elipsoid metodundan hareketle geliştirilmiştir. Bu algoritma bir DP problemi için uygun çözümlerin elipsler ile çevrelenerek kapsanmasına dayanmaktadır. Bir elipsin belirlenmesinin ardından merkez noktasının uygun bir çözüm olup olmadığına bakılır. Eğer uygun bir çözüm değil ise ayırma (seperation) işlemi ile ihlal edilen kısıtlama bulunur. Ardından içinde uygun çözümlerin bulunduğu elipsin yarısı kullanılarak yeni bir elips oluşturulur. Bu yeni elips bir önceki iterasyondaki elipsten en az belirlenen sabit bir değerin katı kadar daha küçük olacağı için iterasyonlar süresince bu adımlar tekrarlanarak çözüm uzayı üssel bir hız ile küçültülür. Khachiyan (1979), DP problemleri için polinominal zamanda uygun çözümler bulmak için bu tekniği adapte etmiştir. Geliştirilen bu yaklaşım teoride çok önemli bir adım olsa da pratikte simplex metodundan daha çok zaman almaktadır.

DP problemleri çözmek için kullanılan bir diğer algoritma sınıfı ise iç nokta (interior point) metodlarıdır. İsminden de anlaşılacağı üzere bu algoritmalar kısıt politopunda bir iç nokta bularak çözüme başlar ve optimum sonuca politopun içinde hareket ederek devam ederler. İlk iç nokta metodu Karmarkar (1984) tarafından ortaya atılmıştır. Karmarkar yaklaşımı elipsoid metodu gibi polinominal zamanlı olmasının yanında pratik uygulamalarda simplex metodundan daha hızlı çözüm üretebilme kabiliyetine sahiptir.

Piyasada bulunan Cplex gibi ticari optimizasyon kütüphanelerinde özellikle büyük boyutlu problemlerde iç nokta metodu simplex ve türevi metodlara göre daha hızlı çözüm üretebilmektedir. Özellikle kolon türetme uygulamalarında hesaplanmasına ihtiyaç duyulan dual değerlerin iç nokta metodları ile hesaplanmasının daha kaliteli kolonların üretilmesini sağladığı görülmüştür. İç nokta metodu iterasyonları sonucunda son elde edilen çözümün temel (basic) çözüm olabilmesi için son

aşamada temel çözüm elde edebilmek için kısa bir crossover (push phase) ve simplex çalışmasına ihtiyaç duyulmaktadır. Fakat dual değerlerin temel değişken olması zorunluluğunun olmaması sebebi ile iç nokta metodu ile daha hızlı hesaplanmaları mümkün olabilmektedir.

3.5 Kolon Türetme (Column Generation)

Kolon türetme, küme-kapsama ve küme-ayırma problemlerinin matematiksel yöntemler ile çözümünde olduğu gibi bu çalışmada önerilen yaklaşımın da dayandığı temel tekniklerden birisidir. Özellikle küme-kapsama veya küme-ayırma modeli kullanılarak çözülen büyük ölçekli DP problemlerinde bulunan satırları (kısıtlar veya uçuşlar) kapsayacak mümkün olan tüm kolonların (ekip eşlemeleri) üretilmesi hesaplamasal ve süresel açıdan mümkün olmamaktadır. Kolon türetme tekniği, çözüm uzayındaki tüm kolon kombinasyonlarını üretmek yerine dinamik bir şekilde algoritma süresince sadece çözümü daha ileri götürebilecek kolonların üretilmesine olanak sağlar. Bu sayede büyük problemlerde bile az sayıda kolon üretilerek çözüme ulaşmak mümkün olmaktadır.

Kolon türetme yaklaşımı iki aşamadan oluşur. Birinci aşama ana (master) problem, ikinci aşama ise ücretlendirme/alt (pricing/sub) problem olarak isimlendirilir. Ana problem aşamasında, problemde bulunan tüm satırlar (uçuşlar) ve üretilmiş olan tüm kolonlar (ekip eşlemesi) kullanılarak bir DP modeli oluşturulur ve bu modelin çözümü gerçekleştirilir. Çözüm için kullanılan bu DP modeline doğrusal gevşetme (linear relaxation/LP) çözümü veya problem çözüm uzayındaki tüm olası kolonlar üretilmeden çözüldüğü için kısıtlı ana problem (restricted master problem/RMP) olarak ta isimlendirilmektedir. Ücretlendirme aşamasında ise ana problemin çözümü ile elde edilen çıktılar (dual değişkenler) kullanılarak problemin amaç fonksiyonunu daha iyiye götürecek yeni kolonlar (ekip eşlemesi) üretilir. (Desrosiers ve Lübbecke, 2005)

$$Min\ z = \sum_{j=1}^{n} c_j x_j \tag{3.11}$$

$$\sum_{j=1}^{n} a_j x_j \geq b\,, \qquad x_j \geq 0, \quad j = 1, \ldots, n \tag{3.12}$$

Denklem 3.11 ve 3.12' ifade edilen sistem tipik bir minimizasyon ana problemi biçimindedir. Burada a_j ve b, m bileşenli vektörlerdir ($m<n$). Elde edilen ilk temel uygun (initial basic feasible) çözüm için hesaplanan x_b, ilişkili temel matrisi B ve maliyet matrisi c_b'nin biliniyor olduğu varsayılırsa problemin kısıtları ile ilişkili olan dual değişkenler (gölge fiyatlar) Denklem 3.13 vasıtası ile hesaplanabilir. (Lasdon, 2002)

$$y = c_b B^{-1} \tag{3.13}$$

Ulaşılan en son temel uygun (basic feasible) çözümü daha ileriye götürmek için temelde olmayan (nonbasic) tüm değişkenler gölge fiyatlar kullanılarak Denklem 3.14 vasıtası ile ücretlendirilir. Bu hesaplanan değere indirgenmiş maliyet (reduced cost). Dolayısıyla her simplex iterasyonunda elde olan temel değişken kümesi değerlerine göre yeni bir dual vektör hesaplanır.

$$\bar{c}_j = c_j - y a_j \tag{3.14}$$

$$\min_j \bar{c}_j = \bar{c}_s < 0 \tag{3.15}$$

Denklem 3.15'te ifade edildiği gibi temelde olmayan tüm değişkenler için Denklem 3.14'teki indirgenmiş maliyet değerini minimize eden x_s değişkeninin simplex pivot işlemi ile temele girmesiyle problem daha iyi bir amaç fonksiyonu değerine ulaşacaktır. (Lasdon, 2002)

Eğer DP modelindeki kolon sayısı binler, milyonlar mertebesinde ise kolonların büyük çoğunluğu temel dışı (nonbasic) olacak ve her kolon için indirgenmiş maliyet hesaplaması algoritma için meşakkatli bir hal alabilir. Bu nedenle özellkile büyük ölçekli problemler için mümkün olan tüm kolon kombinasyonlarının üretilerek DP modelinin çözülmesi mümkün olmamaktadır. Bu tip büyük ölçekli problemlerde tüm kolonların kümesi genellikle Dantzig-Wolfe ayrıştırmasına olanak saylayacak şekilde, birbirinden hemen hemen bağımsız ücretlendirme alt problemlerine (pricing

sub problem)bölünmeye uygun düzgün yapısal özelliklere sahiptir. Dantzig-Wolfe ayrıştırmasında DP modeli, hemen hemen bağımsız kısımlardan oluşan ücretlendirme alt problemleri ve bu alt problemleri bağlayan ana (main) probleme ayrıştırılır. En iyi sonucu elde edebilmek için ana ve alt problem ardışık olarak birçok kez çözülür. İlk olarak ana problem çözülür, bunun sonuçlarından yararlanılarak her alt problem için amaç fonksiyonu oluşturulur. Daha sonra bu alt problemler çözülür ve çözümlerden elde edilen yeni sütunlar temel probleme ilave edilir. Bu süreç, optimum koşullar sağlanıncaya kadar devam eder (Dantzig, 1963).

İlerideki bölümlerde de anlatılacağı üzere bu çalışmada önerilen yaklaşımda ana problem küme-kapsama problemi, Denklem 3.14 ve 3.15'te ifade edilen ücretlendirme alt problemi ise en kısa yol problemi olarak modellenmiştir.

3.6 Stabilizasyon

Standart kolon türetme uygulamalarında algoritma çalışması esnasında bazı zorluklar ortaya çıkmaktadır. Bunlardan bazıları; yavaş yakınsama (tailing-off effect), bir uç noktadan diğer uç noktaya sıçrayabilen dual değişkenlerin stabil olmaması durumu (bang-bang effect), özellikle ilk iterasyonlarda elde edilen dual bilginin zayıf olmasından dolayı ihtiyaç duyulmayan kolonların türetilmesi durumu (heading-in effect) gibi. Bu problemler ile başa çıkabilmek için stabilize kolon türetme yaklaşımlarının kullanılması gerekmektedir. Bu çalışmada, Merle ve diğ. (1999) çalışmasında önerilen yaklaşım takip edilmiştir. Sözkonusu yaklaşım, doğrusal programlama çerçevesinden çıkmadan, ana DP modeline ek terimler (yeni değişkenler ve kısıtlamalar) ekleyerek yüksek seviyeli osilasyonları penaltı uygulayarak azaltmaya dayanmaktadır. Stabilize kolon türetme yaklaşımı ile birlikte, sözkonusu zorlukların aşılmasının yanında algoritmanın ivmelenmesinde gözle görülür bir artış sağlandığı dolayısıyla da toplam iterasyon sayısında azalma olduğu görülmüştür. Stabilize kolon türetme yaklaşımları ile ilgili daha detaylı bilgi için Lübbecke (2011), Lübbecke ve Desrosiers (2005), Merle ve diğ. (1999), Amor ve diğ. (2009) çalışmaları incelenebilir.

3.7 Dal-sınır Algoritması (Branch&Bound Algorithm)

Kolon türetme yaklaşımı tamamen kesirli DP çözümlerine dayansa da Küme-kapsama, küme-ayırma problemleri gibi bir çok gerçek dünya problemi tamsayı (integer) çözüm gerektirmektedirler. Dolayısıyla her ne kadar kolon türetme ana probleminin çözümünde ve dual değişkenlerin hesaplanmasında DP çözümlerine ihtiyaç duyulsa da nihai tamsayı çözümün elde edilebilmesi için son kolon türetme iterasyonunun akabinde ana problemin tamsayı programlama modeli dal-sınır algoritması ile çözülür. Kolon türetme iterasyonları boyunca optimizasyon modeline eklenecek yeni kolonların üretilmesi için oluşturulan alt problem modelleri de dal-sınır algoritması ile çözülmektedir.

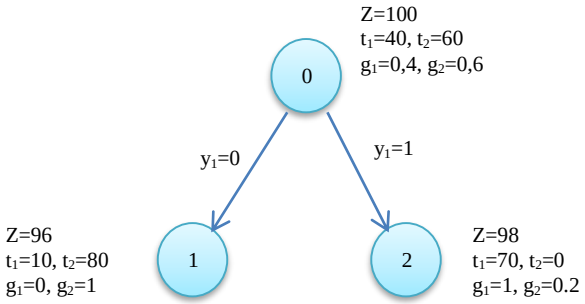
Dal-sınır algoritması tamsayı programlama problemlerinin çözümünde yaygın olarak kullanılır (Land ve Doig, 1960). Dal-sınır algoritmasında başlangıçta ana problemin DP gevşetmesi çözülerek ilk optimal çözüm bulunur. Eğer bulunan DPG çözümü tamsayı ise bu çözüm TP için de optimal bir çözümdür. Eğer bulunan DPG çözümü tamsayı değil ise ana problem yeni kısıtlar eklenerek bir ağaç yapısı üzerinde alt problemlere ayrıştırılır. Oluşturulan alt problemler için de yine yeni DPG problemleri oluşturulur, çözülür ve elde edilen sonuçlar değerlendirilir. Ağaç üzerindeki her düğümde probleme yeni bir kısıt eklenerek oluşturulan DPG problemleri tamsayı çözüm bulunana kadar her düğüm için tekrar tekrar çözülür. Dal-sınır algoritmasında her dal aşağıdaki durumlarda sonlandırılır:

- Tamsayı değer alması gereken değişkenlerin tümü tamamen veya kısmen gevşetilmiş problemde tamsayı değer alır ve bu çözüme tamsayı çözüm adı verilerek en iyi amaç fonsiyonu değeri gerekli olursa güncellenir,
- Bir düğümdeki amaç fonksiyonu değeri güncel olan en iyi amaç fonksiyonu değerinden daha kötüdür,
- Bir düğümdeki problemin çözümü kısıtlardan bazılarını sağlamamaktadır.

Bu kriterlere uyularak problem için dal-sınır ağacı oluşturulur ve tüm dallar sonlandırıldıktan sonra en iyi çözüme ulaşılır.

Örnek olarak Şekil 3.1'deki modelde, dal-sınır algoritmasının ilk problemi tamsayılı değişkenler olan g_1 ve g_2'nin gevşetilmesiyle oluşturulur. Bu durumda elde edilen DP modeli çözülür ve Şekil 3.1'de 0 numaralı düğümdeki çözüm elde edilir. Bu

düğümde elde edilen amaç fonsiyonu değeri 100'dür ve bu problemde en iyi çözüm için bir üst sınır verir. Bu çözümde tamsayılı değişkenlerin değerlerinin 0,4 ve 0,6 olduğunu görürüz. Bu düğümden iki yeni dal üzerinde yeni düğümler oluşturulmaktadır. Bu amaçla tamsayılı değişkenlerden birisini seçmemiz gerekir. Bu problemde rastgele bir seçimle g_1 değişkeni seçilmiş ve 1 ve 2 numaralı düğümler g_1'in 0'a ve 1'e eşitlenmesiyle elde edilmiştir. 1 düğümünde elde edilen sonuç tamsayı (integer) bir çözüm olduğu için problemin yeni alt sınırı 96 olarak belirlenir. 2 düğümünde ise daha iyi bir amaç fonksiyonu değeri g_2 değişkeninin gevşetilmesi ile elde edilmiştir. Bu noktadan sonra algoritma, 2 numaralı düğümden hareketle g_2 değişkeninin sabitleneceği yeni dallar (branch) oluşturulması ile benzer süreçleri nihai tamsayı sonucu bulunana kadar tekrarlar.



Şekil 3.1 : Örnek dal-sınır düğümü.

4. YENİ KOLON TÜRETME STRATEJİSİ

Bu başlık altında önerilen stratejinin içerdiği süreçler detaylandırılacaktır.

4.1 Sistemin Ayağa Kaldırılması

Algoritma süreçleri içerisinde kullanılacak veriyi ve tüm sistemi, ana kolon türetme döngüsü için hazır hale getiren rutinleri içeren aşama, sistemin ayağa kaldırılma aşaması olarak adlandırılmaktadır. Bu aşamanın ilk adımı, önerilen ücretlendirme ağının oluşturulabilmesi için kritik bir işlem olan görev üretme aşamasıdır.

4.1.1 Görev üretme

Bu çalışmada önerilen görev-uçuş yatı bağlantısı ağının oluşturulmasından önce verilen havayolu tarifesindeki uçuşlar kullanılarak gerçekleştirilmesi mümkün olan tüm görev alternatiflerinin üretilmesi gerekmektedir. Tüm görevlerin üretilmesi için, Şekil 3.1'deki örnek kodda gösterildiği gibi bir dept-first arama kodu geliştirilmiştir. Kod içerisindeki *Gorev_Ara* prosedürü, algoritmanın mümkün olan tüm uçuş bağlantılarını aramasını sağlayan ana tekrarlamalı (recursive) fonksiyondur. 2 ve 3 numaralı satırlarda, uygun uçuşlar aranır ve *AktifGörev*'e eklenir. 4 ve 5 numaralı satırlarda, *AktifGörev* denetlenir ve eğer legal ise *GörevListesi*'ne eklenir. 6 ve 7 numaralı satırlarda, eğer *AktifGörev* daha fazla uçuş eklenmeye müsait ise aramayı daha derinleştirmek için tekrarlamalı fonsiyon kendi içerisinden tekrar çağrılır.

Şekil 4.1'deki örnek kodda, *GörevListesi* tekrarlamalı fonksiyon tarafından üretilen tüm legal görevlerin listesidir. *AktifGörev* ise 4 numaralı satırda denetlenen ve legal bir uçuş görevini temsil eden uçuş dizisidir.

```
1 procedure Gorev_Ara(AktifGorev, GorevListesi)
2         for AktifGorev'in vardığı limandan, vardığı saatten sonra kalkan tüm uçuşlar için do
3                 Bulunan Uçuş'u AktifGorev'e ekle;
4                 if AktifGorev legal ise
5                         AktifGorev'i GorevListesi'ne ekle;
6                 if AktifGorev daha fazla uçuş içerebilir ise
7                         Gorev_Ara(AktifGorev, GorevListesi);
8                 Bulunan Uçuş'u AktifGorev'den çıkar;
```

Şekil 4.1 : Görev oluşturma örnek kodu.

GörevListesi'nin boyutu tamamen havayolu tarifesinin büyüklüğüne ve görevlerin denetlenmesi için kullanılan kural kümesine bağlıdır. Örneğin görev içerisindeki uçuşların aralarındaki minimum bağlantı süresi kısaltıldığında daha fazla görev, daha büyük bir ücretlendirme ağı ve dolayısıyla da daha zor bir problem ile karşılaşılacaktır. Bu çalışmada uygulanan kural kümesi Türk Hava Yolları, ekip planlama bölümünden elde edilmiştir. Sözkonusu kural kümesi ile aylık ortalama 17000 uçuşu bulunan bir filo için ortalama 420000 görev üretilmiştir.

4.1.2 Görev-uçuş yatı bağlantısı ağı (Duty-flight overnight connection network)

Tüm legal uçuş görevlerinin üretiminin ardından, ücretlendirme alt probleminin çözümünde kullanılacak olan ücretlendirme ağı oluşturulur. Ekip eşleme probleminin çözümü için kullanılan iki temel ağ modeli bulunmaktadır (Vance ve diğ, 1997). Bunlardan ilki uçuş ağıdır (flight network). Bu modelde bağlantı çizgileri (arc) uçuşları, düğümler (node) ise kalkış limanları, varış limanları, kaynak (source) ve sonlanma (sink) noktalarını temsil eder. Uçuş ağı modeli günlük ekip planlama problemlerinin çözümü için daha uygun bir modeldir. Aynı zamanda havayolu firması tarafından talep edilen karmaşık kural ihtiyaçlarının uçuş ağı üzerinde modellenmesi oldukça zordur.

İkinci model, görev ağı (duty network) modelidir. Bu modelde ise bağlantı çizgileri görevleri, düğümler (node) ise görev başlangıç limanları, görev bitiş limanları, kaynak (source) ve sonlanma (sink) noktalarını temsil eder. Bu model haftalık ve aylık ekip eşleme problemlerinin çözümü için daha uygun bir modeldir ve görev ağı üretilmeden önce tüm uçuş görevlerinin üretilmiş olmasına ihtiyaç duyar. Görev ağı modelinin kullanılması durumunda, tek bir görev periyodu için uygulanması istenen her türlü kural ve denetim; herhangi bir programlama dilinde geliştirilebilecek düz ve yalın bir kod ile görev üretimi aşamasında gerçekleştirilebilir. Fakat bu model, büyük boyutlu aylık ekip eşleme problemleri için görevler arasında çok büyük miktarda bağlantı üretmekte ve bu durum büyük hafıza (memory) ihtiyacı ve performans sorunları ortaya çıkarmaktadır. Bu sorunları çözmek için bir gevşetilmiş görev ağı (relaxed duty network) modeli önerilmiştir (Gustafsson, 1999). Gevşetilmiş görev ağı modelinde ağın boyutu başarılı bir şekilde düşürülmüş fakat görevler arası bağlantılar ile ilgili kurallar model üzerinde açık bir şekilde temsil edilmediği için illegal ekip eşlemesi üretebilmektedir. İllegal ekip eşlemelerinin üretilmesi

durumunda ise ağın hatalı ekip eşlemesini tekrar üretmeyecek şekilde modifiye edildiği bir strateji önerilmiştir.

Bu çalışma önerilen ücretlendirme ağı modeli, görev-uçuş yatı bağlantısı ağı olarak adlandırılmaktadır. Ağı oluşturmak için tüm görevden göreve bağlantıları ilgili kuralları kullanarak kontrol eden düz ve yalın bir kod geliştirilmiştir. Bu ağ modelinde iki tip bağlantı ve iki tip düğüm bulunmaktadır. Düğüm tiplerinden ilki, ağın başlangıç noktaları olan uçuş görevlerinden oluşan görev düğümleridir ve bu düğümler oluşturulacak olan ekip eşlemelerinin ilk başlangıç görevlerini temsil edecektir. Bağlantı tiplerinden ilki, aralarında yatı bağlantısı bulunan görev-uçuş çiftleri tarafından oluşmaktadır ve bu tip bağlantılar görev düğümlerini takip etmektedir. Düğüm tiplerinden ikincisi, görev-uçuş bağlantılarını izleyen, yatı periyodundan sonraki ilk görevin ilk uçuşunu temsil eden düğümlerdir. Bağlantı tiplerinden ikincisi ise uçuş düğümlerini takip eden ve uçuş düğümünü oluşturan uçuş ile başlayan görevleri temsil eden uçuş-görev çiftlerinden oluşmaktadır.

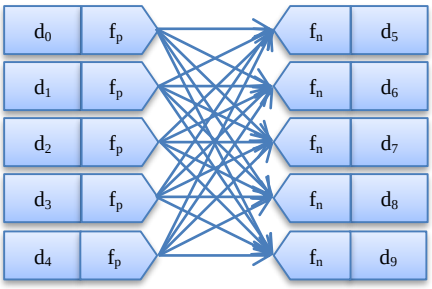


Şekil 4.2 : Görev-uçuş bağlantısı ağında bulunan düğüm ve bağlantılar.

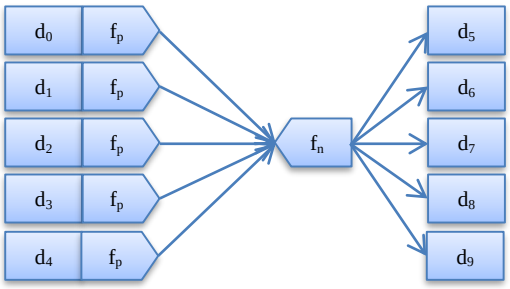
Şekil 4.2'de görüldüğü gibi, d_p bir görev düğümü ve ağın başlangıç noktasıdır. d_p'yi takip eden *görev-uçuş yatı bağlantısı*, f_n ile başlayacak olan bir sonraki göreve kadar sürecek olan legal bir yatı periyodunu temsil eder. f_n uçuş düğümü ise f_n ile başlayan tüm görevler için bir bağlantı noktasını temsil eder. Bu ağ modeli, özellikle yüksek miktarda yatı uçuş bağlantısı içeren problemlerde veri tekrarını azalttığı için tam-bağlı (fully-connected) bir görev ağına oranla daha sıkı (compact) bir temsil imkanı sunmaktadır. Dolayısıyla da ihtiyaç duyulan hafıza miktarını azaltmakta ve algoritmanın performansını yükseltmektedir.

Önerilen modelin, tam-bağlı görev ağı modeli ile daha iyi bir karşılaştırması için, aynı uçuş f_p ile biten 5 adet uçuş görevi ve bu 5 görevi legal bir yatı bağlantısı ile takip edebilecek olan bir uçuş f_n ile başlayan farklı 5 adet uçuş görevi daha düşünülsün. Sözkonusu senaryoyu temsil eden tam-bağlı bir görev ağı Şekil 4.3'te temsil edilmiştir. Şekil 4.3'te görüldüğü üzere ağ üzerinde toplam 25 adet görevden göreve bağlantı bulunmaktadır. Bu da problemin tam-bağlı görev ağı üzerinde çözülmek istenmesi durumunda inşaa edilecek olan en kısa yol tamsayı modelinde toplam 25 adet değişkene ihtiyaç duyulacağı anlamına gelmektedir.

Aynı problem görev-uçuş yatı bağlantısı ağı kullanılarak Şekil 4.4'teki gibi temsil edilebilir. Şekil 4.4'ten görüleceği üzere önerilen model ile sadece 10 adet bağlantı kullanılarak aynı problem temsil edilebilmektedir. Aynı zamanda Gustafson (1999) çalışmasının aksine, önerilen modelde görevler arası bağlantıları ilgilendiren kurallar (örn. min yatı süresi) ağ üzerinde açık bir şekilde temsil edildiği için algoritmanın görevler arası bağlantı kurallarını ihlal eden herhangi bir ekip eşlemesi üretme ihtimali ortadan kalkmaktadır.



Şekil 4.3 : Tam-bağlı görev ağı örneği.



Şekil 4.4 : Görev-uçuş yatı bağlantısı ağı örneği.

4.1.3 İlk ekip eşlemelerinin üretimi (Initial crew pairing generation)

Ana kolon türetme iterasyonlarına geçmeden önceki son aşama, ilk ekip eşlemelerinin üretilmesi aşamasıdır. Bu aşama ile birlikte sistemin ve veri yapılarının hazırlanma süreci tamamlanmış olur ve ardından algoritmanın ana döngüsü çalışmaya başlar. İlk ekip eşlemelerinin üretimi, görev üretimi aşamasında kullanılan mekanizmaya çok benzer bir dept-first arama kodu ile gerçekleştirilmektedir. Görev üretme aşamasından farklı olarak bu aşamada girdi olarak görev listesi kullanılır ve çıktı olarak ise legal ekip eşlemeleri üretilir. Görev üretme ile bu aşama arasındaki en önemli farklardan birisi de bu aşamada mümkün olan tüm ekip eşlemelerinin üretilmemesidir. Aksine algoritma sadece tarifedeki tüm uçuşları kapsamaya çalışmakta ve eğer bir uçuş zaten kapsanmış ise tekrar kapsamaya çalışmamaktadır. Algoritma, aylık ortalama 17000 uçuşu olan bir filo için ortalama 8200 ekip eşlemesi

üretmektedir. İlk ekip eşlemelerinin üretimi aşaması tamamlandıktan sonra, bir ana DP koşumu başlatılır ve tarifedeki tüm uçuşlar ve ekip eşlemeleri için ilk primal ve dual bilgi elde edilir. Böylece ana kolon türetme döngüsü için sistem ve veri hazırlanması aşaması tamamlanmış olur.

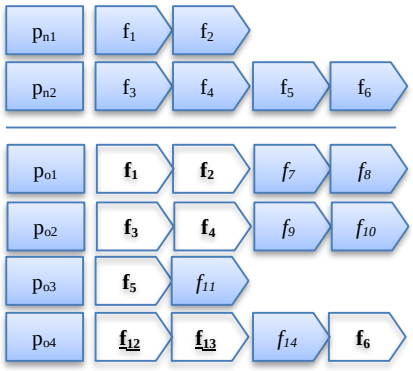
4.2 Kısmi Çözüm Üretme Yaklaşımı

Kolon türetme yaklaşımlarının uygulanması sırasında verilmesi gereken en önemli kararlardan birisi hangi uçuş ve/veya hangi gün için ekip eşlemesi üretileceğidir. Bu çalışmada önerilen kolon türetme stratejisi *kısmi çözüm üretme yaklaşımı* olarak adlandırılmıştır. Bu yaklaşım, tüm problemin bir kısmı için, anlamlı bir kısmi çözüm alternatifi oluşturacak olan yeni ekip eşlemelerini üretmek için bir önceki iterasyonda üretilen ekip eşlemelerinden elde edilen uçuş bağlantıları bilgisini kullanır. Üretilen kısmi çözümler, havayolu tarifesindeki tüm uçuşları kapsamak zorunda değildir. Dolayısıyla herhangi bir kolon türetme iterasyonunun sonunda elde edilen kısmi çözüm yalnız başına tüm çözüm alanını temsil edemez. Aksine, üretilen tüm kısmi çözümlerde bulunan tüm ekip eşlemeleri bir ana ekip eşlemesi havuzunda toplanır ve bu havuz ana DP problemin veri kaynağını oluşturur.

Her kolon türetme iterasyonunun sonunda bir kısmi çözüm üretilir ve bu çözüm içerisinde bulunan tüm ekip eşlemeleri ana ekip eşlemesi havuzuna koyulur. İşlem süresince, yeni ekip eşlemeleri üretilmesi gereken uçuşlar, anlamlı yönler (directions) bulabilmek için, yeni üretilen ekip eşlemeleri daha önceki iterasyonlarda üretilmiş ekip eşlemeleri ile karşılaştırılır. Bu karşılaştırma sürecini başlatabilmek için en az bir adet yeni bir ekip eşlemesine ihtiyaç duyulmaktadır. İterasyonu başlatması gereken bu ilk ekip eşlemelerini üretebilmek için, sezgisel bir şekilde bir veya daha fazla ekip eşlemesi üretebilen bir dept-first arama kodu geliştirilmiştir. Ardından, yeni ekip eşlemesi üretilmesi gereken yeni bir uçuş bulunamayana kadar, kesin (exact) ücretlendirme modeli tekrar tekrar çözülür.

İşlem süresince, kısmi çözüm tamamen üretilene yani aktif olan kolon türetme iterasyonu bitene kadar, kısmi çözüm tarafından kapsanmış olan uçuşları içeren tüm görevler, sonraki ücretlendirme koşumları için hesaplamalara (ağa) dahil edilmezler. Bu yaklaşım, işlem süresince ücretlendirme ağındaki gerçekleşecek küçülme sebebi ile algoritmanın performansını arttırmakta ve final çözümdeki pas (deadhead) zamanını gözle görülür oranda azaltmaktadır.

Yeni ekip eşlemelerinin üretilmesi için yeni uçuşların belirlenmesini sağlayan mekanizma Şekil 4.5'teki gibi gösterilmiştir.



Şekil 4.5 : İterasyonlar arası ekip eşleme karşılaştırması.

Şekil 4.5'te devam eden kolon türetme iterasyonu dahilinde üretilmiş olduğu varsayılan iki adet ekip eşlemesi (p_{n1}, p_{n2}) bulunmaktadır. Algoritma öncelikle bu yeni ekip eşlemelerinin içerdiği uçuşlardan herhangi birini kapsayan eski ekip eşlemelerini arar. Bulunan bu eski ekip eşlemelerinin aynı zamanda bir önceki DP çözümünde bulunması yani primal değerlerinin belli bir eşik seviyesinin üstünde olması gereklidir. Sözkonusu eski ekip eşlemelerinin (p_{o1}, p_{o2}, p_{o3}, p_{o4}) tespit edilmesinden sonra f_7, f_8, f_9, f_{10}, f_{11}, f_{14} uçuşları için ücretlendirme problemi çözülür ve bu uçuşları kapsayan yeni ekip eşlemeleri üretilir ve ardından tekrar yeni uçuşlar tespit edilerek döngüye devam edilir. f_{12} ve f_{13} uçuşları, aktif kolon türetme iterasyonunun daha önceki evrelerinde üretilen bir veya birden fazla ekip eşlemesi tarafından kapsandığı varsayıldığından işlemlere dahil edilmemişlerdir. Kısmi çözüm üretme işlemi yeni bir ekip eşlemesi üretilmesi gereken bir uçuş bulunamayana kadar devam eder ve işlem süresince bir çok ekip eşlemesi üretilir. Her kolon türetme iterasyonunda birden fazla ekip eşlemesi üretilmesinin, elde edilen dual değerlerin stabilizasyonu konusunda da olumlu bir etkisi bulunmaktadır.

Şekil 4.6'da kısmi çözüm üretme yaklaşımını betimleyen bir örnek kod bulunmaktadır. 1 numaralı satırdan anlaşılacağı üzere, algoritma input olarak son DP koşumundan elde edilen dual değişken bilgisi ve o ana kadar üretilmiş tüm ekip eşlemelerini içeren global eşleme listesini girdi olarak almaktadır. 2 numaralı satırda sezgisel ekip eşleme üretimi kodu çalıştırılır ve iterasyonun ilk ekip eşlemeleri üretilir. 3 numaralı satırda sezgisel metod ile üretilen ekip eşlemeleri global ekip eşleme listesine eklenir ve 4 numaralı satırda yeni üretilen ekip eşlemeleri ile eski

iterasyonlardan gelenler karşılaştırılarak ekip eşlemesi üretilecek yeni uçuşlar tespit edilir. 6 numaralı satırdan 8 numaralı satıra kadar benzer prosedür kesin (exact) ekip eşleme üretme metodu ile yeni uçuşlar tespit edildiği sürece devam ettirilir.

Bu yaklaşım, bu çalışmada sunulan kolon türetme stratejisinin önerdiği kolon türetme iterasyonlarının ana parçasını oluşturmaktadır. DP koşumları ve kesirli değerlerin elimine edilmesini yolu ile tamsayı çözüm üretilmesini sağlayan bağlantı sabitleme sezgiseli bu aşamayı takiben devreye girmektedirler.

```
1 procedure Kısmi_Çözüm_Üret(DualBilgi, EşlemeListesi)
2         Eşlemeler = Sezgisel_Metod_İle_Eşleme_Üret(DualBilgi);
3         Eşlemeler'i EşlemeListesi'ne ekle;
4         Uçuşlar = Eşleme_Üretimi_İçin_Yeni_Uçuş_Tespit_Et(EşlemeListesi, Eşlemeler);
5         while Uçuşlar boş değil ise do
6                 Eşlemeler = Kesin_Metod_İle_Eşleme_Üret(DualBilgi, Uçuşlar);
7                 Eşlemeler'i EşlemeListesi'ne ekle;
8                 Uçuşlar = Eşleme_Üretimi_İçin_Yeni_Uçuş_Tespit_Et(EşlemeListesi, Eşlemeler);
```

Şekil 4.6 : Kısmi çözüm üretimi yaklaşımı örnek kodu.

Benzer büyüklükte ekip eşleme problemlerini yine benzer şekilde bir ekip eşleme havuzu kullanarak çözen farklı bir tekniğin sunulduğu Erdoğan ve diğ. (2015) çalışması da başvurulabilecek önemli bir kaynak olma özelliği taşımaktadır. Sözkonusu çalışma küme-ayırma modeli ile matematiksel ve metasezgisellerin birlikte kullanıldığı bir optimizasyon güdümlü sezgisel (optimization driven heuristics) yaklaşımı kullanmaktadır.

4.2.1 Ücretlendirme probleminin sezgisel çözümü

Her kolon türetme iterasyonu, o iterasyonun ilk ekip eşlemelerinin elde edildiği bir sezgisel arama kodu ile başlamaktadır. Sözkonusu sezgisel arama, negatif indirgenmiş maliyetli ekip eşlemeleri arayan bir dept-first arama algoritması ile gerçekleştirilmektedir. Kolon türetmede indirgenmiş maliyetin hasaplanması ile ilgili daha detaylı teorik arka plan için Desrosiers ve Lübbecke (2005), Lübbecke (2011), Lübbecke ve Desrosiers (2005), Vanderbeck (2005) çalışmaları incelenebilir. Bu sezgisel arama global minimumu bulmayı garanti etmemektedir. Aksine, bu sezgiselin görevi, mümkün olduğu kadar hızlı bir şekilde kaliteli ekip eşlemeleri bularak kısmi çözüm üretme işlemini başlatmasıdır. Bu ilk aşamada, ekip eşlemesi üretilecek uçuş bilgisinin eksikliği sebebi ile kesin bir ücretlendirme modelinin kullanımı zordur. Bu bilgi olmadan tüm ücretlendirme ağının oluşturulması ve çözülmesi gerekmektedir ve bu da büyük ölçekli veri sebebi ile pratikte gerçekleştirilmesi zor bir işlemdir.

```
1  procedure Sezgisel_Metod_İle_Eşleme_Üret(UçuşListesi, EşlemeListesi)
2          for UçuşListesi'ndeki her Uçuş için do
3                  for ikamet merkezinden başlayan ve Uçuş'u içeren her Görev için do
4                          Eşleme = Yeni bir ekip eşlemesi oluştur;
5                          Görev'i Eşleme'ye ekle;
6                          Eşleme_Ara(Eşleme, EşlemeListesi);
7
8  subprocedure Eşleme_Ara(Eşleme, EşlemeListesi)
9          if Eşleme legal ise
10                 if Eşleme tamamlanmış ise
11                         if Eşleme kalitesi yeterli ise
12                                 Eşleme'yi EşlemeListesi'ne ekle;
13                 else
14                         if Eşleme'ye daha fazla görev eklenebilir ise
15                                 for Eşleme'nin bittiği limandan başlayan her Görev için do
16                                         Görev'i Eşleme'ye ekle;
17                                         Eşleme_Ara(Eşleme, EşlemeListesi);
18                                         Görev'i Eşleme'den çıkar;
```

Şekil 4.7 : Ücretlendirme probleminin sezgisel çözümü örnek kodu.

Şekil 4.7'de ücretlendirme probleminin sezgisel çözümü örnek kod ile betimlenmiştir. 1 ve 6 numaralı satırlar arasında, her uçuş ve görev için dept-first arama mekanizması tetiklenmektedir. 2 numaralı satırda, ekip eşlemesi üretilmesi gereken uçuşlar bulunmaktadır. 3 numaralı satırda, seçilen uçuşu içeren görevler aranmakta ve filtrelenmektedir. 6 numaralı satırda, ise seçilen görev ile seçilen uçuş için ekip eşleme araması tetiklenmektedir.

8 ve 18 numaralı satırlarda, tekrarlamalı dept-first arama mekanizması betimlenmiştir. 9 numaralı satırda, üretilen *Eşleme*'nin legal olup olmadığı kontrol edilmektedir. 10 numaralı satırda, üretilen *Eşleme*'nin ikamet merkezine dönmüş olup olmadığı kontrol edilmektedir. 11 numaralı satırda, üretilen *Eşleme*'nin indirgenmiş maliyet değeri bir eşik değeri ile kontrol edilir yani *Eşleme*'nin kalitesi kontrol edilir. Bu eşik değeri sezgisel aramanın başında 0 "sıfır" dır ve arama ilerledikçe üretilen ekip eşlemelerinin indirgenmiş maliyet değerlerine göre dinamik olarak güncellenir. Eğer üretilen *Eşleme*'nin kalitesi bu eşik değerinin üstünde ise ise 12 numaralı satırda global ekip eşleme listesine eklenir. 14 ve 18 numaralı satırlar arasında, *Eşleme*'ye eklenebilecek görev adayları bulunur ve tekrarlamalı fonksiyon tekrar çağrılır.

Dept-first arama işleminin sonunda, üretilen yeni ekip eşlemeleri son bir sezgisel operasyona tabi tutulurlar. Bu son operasyon, üretilen yeni ekip eşlemelerinin içinden birden fazla kez kapsanan uçuş olmasına müsaade etmeyecek şekilde eliminasyon yapar. Bu eliminasyon için de ekip eşlemelerinin indirgenmiş maliyet değerleri ölçü olarak kullanılır. Bir uçuş için indirgenmiş maliyet değeri en düşük olan ekip eşlemeleri listede kalırken diğerleri elimine edilir. Bu sayede tarife boyunca daha iyi bir şekilde (homojen) dağılmış ekip eşleme kümesi elde edilir.

Aynı zamanda, bu aşamada üretilen ekip eşlemeleri tarafından kapsanmış uçuşları içeren görevler, daha ileriki kesin ücretlendirme çözümlerine dahil edilmemektedir. Bu sayede, ileriki aşamada görev-uçuş yatı bağlantı ağı kullanılarak oluşturulacak kesin ücretlendirme en kısa yol modellerinin boyutunun küçülmesi ve gözle görülür bir performans artışı sağlanmaktadır.

4.2.2 Ücretlendirme probleminin kesin çözümü

Kısmi çözüm üretme aşamasının sezgisel ekip eşleme üretimi yaklaşımı ile başlatılmasının ardından yeni bir ekip eşlemesi üretecek uçuş bulunamayana kadar kesin ücretlendirme modeli tekrar tekrar oluşturulup çözülerek yeni eşlemeler üretilir. Önceki bölümde anlatıldığı gibi sözkonusu uçuşların tespiti için yeni üretilen ekip eşlemeleri ile önceki iterasyonlarda üretilmiş olan ekip eşlemelerini karşılaştırılması yolu ile çalışan bir mekanizma kullanılmaktadır.

Kesin ücretlendirme alt problemi, bir en kısa yol tamsayı problemi olarak modellenmiş ve her koşumda birden fazla ekip eşlemesi üretebilecek şekilde tasarlanmıştır. Her koşum öncesinde, yeni uçuşlar daha önce anlatıldığı gibi belirlenir ve bu çalışmada önerilen görev-uçuş yatı bağlantısı ağı üzerinde yalın bir kod ile görev-uçuş bağlantıları aranır. Bu aramada temelde, ekip eşlemesi üretilmesi gereken sözkonusu uçuşları içeren görevler bulunur ve bulunan bu görevlerle yatı bağlantısı olan, birlikte kullanılmaları durumunda ekip eşlemesi üretilebilecek diğer bağlantılı görevler aranır. Bu arama; taramanın başlatılacağı kaynak görevlerin ikamet merkezinde başlayıp bitme durumuna göre ileri ve/veya geri doğrultuda, ana sistem kurallarından birisi olarak belirlenmiş olan maksimum ekip eşlemesi uzunluğu parametresine uygun olarak gerçekleştirilir. Aynı zamanda bu arama süresince, kısmi çözüm tarafından daha önce kapsanmış uçuşları içeren görevler gözardı edilir.

Her koşum sonrasında, yeni üretilen ekip eşlemeleri kural denetiminden geçirilir. Görevler arasındaki bağlantılar ile ilgili kurallar, önerilen görev-uçuş yatı bağlantısı ağında tam olarak ifade edilebildiği için illegal ekip eşlemesi sadece ve sadece maksimum eşleme uzunluğunu geçmek, birden fazla erken görev bulundurmak durumları gibi ekip eşlemesi seviyesindeki kuralların ihlali sebebi ile oluşabilmektedir. Herhangi bir illegal ekip eşlemesi oluşması durumunda, ücretlendirme ağı üzerinde tekrar aynı ekip eşlemesinin oluşturulmasını engelleyecek

bazı bağlantıların ağdan çıkarılması gibi modifikasyonlar yapılarak model tekrar çalıştırılır.

```
1  procedure Yeni_Eşleme_Oluştur_Ve_Denetle(GörevUçuşBağlantısıListesi, İhlalMevcut, EşlemeListesi)
2          İhlalMevcut = false; EşlemeListesi = {};
3          while true do
4                  [Görev, Uçuş] = GörevUçuşBağlantısıListesi'nden ikamet merkezinden başlayan ilk görevi
                                   ve bağlantılı uçuşunu bul.
5                          if ikamet merkez başlangıçlı bir görev bulunamaz ise
                                   Prosedürden çık;
6                          else
7                                  Eşleme = Yeni bir ekip eşlemesi oluştur;
8                          while true do
9                                  Görev'i GörevUçuşBağlantısıListesi'nden çıkar;
10                                 Görev'i Eşleme'ye ekle;
11                                 if Eşleme legal ise
12                                         if Eşleme ikamet merkezinde bitiyor ise
13                                                 Eşleme'yi EşlemeListesi'ne ekle;
14                                                 4. satıra dön;
15                                         else
16                                                 [Görev, Uçuş] = GörevUçuşBağlantısıListesi'nden Uçuş
                                                                   ile başlayan görev ve bağlantılı uçuşunu bul;
17                                 else
18                                         İlgili görev-uçuş yatı bağlantısını ücretlendirme ağından çıkar;
19                                         İhlalMevcut = true;
20
```

Şekil 4.8 : Kesin ücretlendirme modelinin çıktısı ile ekip eşleme üretimi.

Bu sezgisel kod Şekil 4.8'de görüldüğü üzere, kesin ücretlendirme koşumundan elde edilen görev-uçuş bağlantıları listesini girdi olarak almakta ve daha sonra kullanmak üzere iki adet parametreye kod içinde değer atamaktadır. Bu parametrelerden ilki herhangi bir kural ihlali olup olmadığını gösteren *İhlalMevcut* parametresi, diğeri ise üretilen yeni ekip eşlemelerinin tutulacağı bir liste olan *YeniEşlemeListesi*'dir. 3 ve 4 numaralı satırlarda, ikamet merkezinden başlayan görevler aranır ve bu ana döngü ikamet merkez kalkışlı herhangi bir görev kalmayana kadar tekrar eder. Eğer ikamet merkezinden başlayan herhangi bir görev bulunamaması, aramanın tamamlandığı ve daha sonraki süreçler gerçekleştirilmek üzere prosedürden çıkılması gerektiği anlamına gelmektedir. Eğer ikamet merkezinden başlayan bir görev bulunursa 8 numaralı satırda yeni ve boş bir ekip eşlemesi üretilir, 10 ve 11 numaralı satırlarda sözkonusu uçuş görevi girdi listesinden çıkarılıp üretilen ekip eşlemesi *Eşleme*'ye eklenir. 12 numaralı satırda eklenen her uçuş görevi sonrası *Eşleme* kural denetiminden geçirilir ve herhangi bir kural ihlali mevcutsa veya maksimum eşleme uzunluğu geçiliyor ise 19 ve 20 numaralı satırlarda en son eklenen görev-uçuş bağlantısı ücretlendirme ağından çıkarılır ve *İhlalMevcut* parametresi işaretlenir. Bu sayede üst seviyedeki kod illegallik durumuna göre ücretlendirme problemini tekrar çalıştırma kararı verebilir. Eğer herhangi bir kural ihlali mevcut değil ise *Eşleme*'nin tam olup olmama durumuna göre, algoritma *Eşleme*'yi *EşlemeListesi*'ne ekler ve

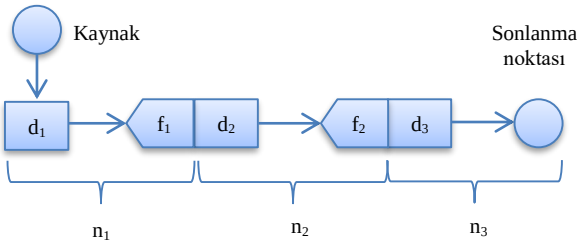
yeni ekip eşlemelerinin olup olmadığını kontrol etmek üzere 4 numaralı satıra döner veya *Eşleme* tamamlanmamış ise 17 numaralı satırdan devam ederek *Eşleme*'ye eklenecek yeni uçuş görevleri arar.

Fakat yine de bu sezgisel kod, ağ güncellemesi ardından illegal ekip eşlemesi üretiminin engelleneceğini tamamen garanti etmemektedir. Bu nedenle sözkonusu ağ güncellemesi tekrarlarını limitlemek için ek bir parametre kullanılmış ve 3 değeri belirlenmiştir. Eğer üç denemenin ardından da illegal ekip eşlemesi üretilmiş ise illegal olan ekip eşlemeleri gözardı edilir.

Ekip eşlemesi seviyesindeki kurallar, tüm kural uygulamasının çok küçük bir kısmını oluşturmaktadır ve bu nedenle de elde edilen tecrübeler üç denemenin karşılaşılan illegalliklerin büyük çoğunluğunu çözmek için yeterli olduğu göstermiştir. Aynı zamanda kısmi çözüm üretme yaklaşımı bazı uçuşların açıkta kalmasına olanak sağlamaktadır.

Kesin (Exact) ekip eşleme üretme modeli

Önerilen modelde her değişken bir görev-uçuş yatı bağlantısını temsi etmektedir. Bu modele göre üç adet göreve sahip bir ekip eşlemesinin temsili Şekil 4.9'daki gibidir.



Şekil 4.9 : Örnek ekip eşlemesi.

Şekil 4.9'da görüleceği üzere, örnek ekip eşlemesi üç adet görev-uçuş bağlantısı içermektedir (n_1, n_2, n_3). İlk iki görev f_1 ve f_2 uçuşlarına yatı bağlantısı ile bağlıdırlar. En son görev d_3 ikamet merkezinde sonlanmakta ve dolayısıyla da bir uçuş bağlantısı yerine sonlanma noktasına uzanan bir bağlantıya sahiptir.

N'nin ücretlendirme alt problemindeki değişkenleri (görev-uçuş bağlantıları) temsil eden bir indis kümesi ve x_n'nin ikili karar değişkenleri olduğu varsayılırsa, ücretlendirme alt problemi modeli aşağıdaki gibi ifade edilir.

$$x_n = \begin{cases} 1, & eğer\ görev - uçuş\ bağlantısı\ n\ seçili\ ise, \\ 0, & aksi\ takdirde, \end{cases}$$

$$a(d_n, f) = \begin{cases} 1, & eğer\ görev\ d_n, f\ uçuşunu\ \ içeriyor\ ise \\ 0, & aksi\ takdirde, \end{cases}$$

$$b(n, f) = \begin{cases} -1, & eğer\ uçuş\ f, görev - uçuş\ bağlantısı\ n\ deki \\ & \quad yatı\ bağlantısı\ uçuşu\ ise, \\ 0, & aksi\ takdirde, \end{cases}$$

$$c(d_n, f) = \begin{cases} 1, & eğer\ uçuş\ f, görev\ d_n\ deki\ ilk\ uçuş\ ise, \\ 0, & aksi\ takdirde, \end{cases}$$

$$src(d_n) = \begin{cases} -1, & eğer\ görev\ d_n\ ikamet\ merkezinden\ başlıyor\ ise, \\ 0, & aksi\ takdirde, \end{cases}$$

$$sink(d_n) = \begin{cases} 1, & eğer\ görev\ d_n\ ikamet\ merkezinde\ bitiyor\ ise, \\ 0, & aksi\ takdirde, \end{cases}$$

$$Min \sum_{n \in N} r_n \ x_n \tag{4.1}$$

$$\sum_{n \in N_f} a(d_n, f)\ x_n \leq 1, \qquad \forall f \in F \tag{4.2}$$

$$\sum_{n \in N_f} \big(b(n, f) + c(d_n, f)\big)\ x_n = 0, \qquad \forall f \in F \tag{4.3}$$

$$\sum_{n \in N} \big(src(d_n) + sink(d_n)\big)\ x_n = 0 \tag{4.4}$$

$$x_n \in \{0, 1\}, \qquad \forall n \in N \tag{4.5}$$

Denklem 4.1'de r_n görev-uçuş bağlantısı n'nin indirgenmiş maliyet değerini temsil etmektedir. r_n, bitiminde yatı bağlantısı bulunan görevler için ek olarak yatı maliyetlerini de barındırmaktadır. Denklem 4.2'de d_n, görev-uçuş bağlantısı n'ye ait olan görevi, N_f ise f uçuşunu kapsayan görev-uçuş bağlantıları kümesini temsil etmektedir. Bu ifade her uçuşun en fazla bir kez kapsanacağını garanti eder. Böylece ücretlendirme modelinde pas (deadhead) uçuşlara izin verilmez. Denklem 4.3, ekip eşlemelerinin yatı bağlantıları boyunca sürekliliğini garanti eden akış kontrolü (flow conservation) kısıtlamasıdır. Denklem 4.4, her ekip eşlemesinin doğru bir şekilde sonlanmasını ve her koşumda birden fazla ekip eşlemesi üretilmesini sağlamaktadır. Bu sayede algoritma indirgenmiş maliyet değeri minimum olan sadece bir adet ekip eşlemesi bulmak yerine, tüm probleme kısmi bir çözüm oluşturabilecek birden fazla ekip eşlemesinin anlamlı kombinasyonlarını üretmektedir.

Daha önce de belirtildiği üzere kesin ücretlendirme alt problemi, ekip eşlemesi üretilmesi gereken yeni bir uçuş bulunamayana kadar tekrarlı bir şekilde çözülür.

4.2.3 Bağlantı sabitleme (Connection fixing) sezgiseli

Her kolon türetme iterasyonunun son aşamasında, uçuş görevi ve ekip eşlemesi eliminasyonu yapan ve bu sayede de ana DP, ücretlendirme alt problemi ve nihai ana TP çözümlerinin performanslarını arttıran bir bağlantı sabitleme sezgiseli çalıştırılır. Her ne kadar nihai çözümün tamsayı olması istense de standart kolon türetme uygulamaları DP çözümlerine dayanmaktadır. Eğer problem simetrik bir yapıya sahip ise en son aşamada tamsayı çözüm elde etmek için çalıştırılan dal-sınır algoritması kötü bir performans sergilemektedir. Bunun nedeni, dal-sınır ağacındaki dallanmanın ardından problemin çok değişmesidir ve bu simetrinin giderilmesi için bir formül üretilmesi zorunludur (Barnhart ve diğ, 1998).

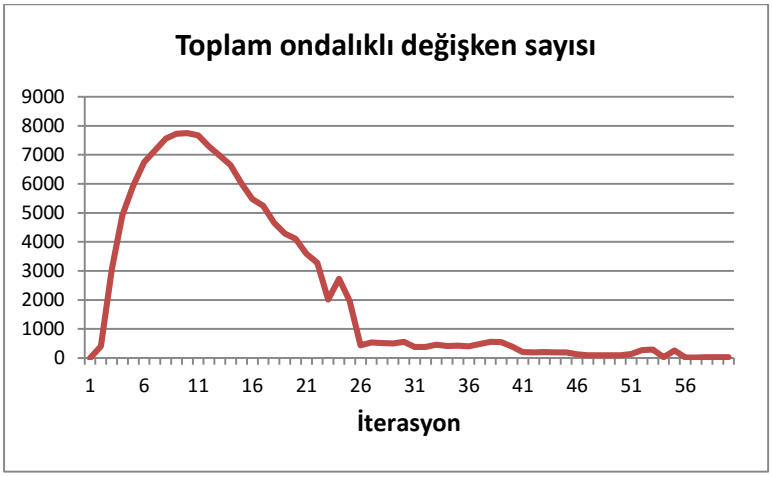
Özellikle büyük ölçekli ekip eşleme problemlerinde, nihai dal-sınır algoritması koşumu sırasında çok ciddi performans problemleri yaşanmaktadır. Bu problemin çözümü için bağlantı sabitleme sezgiseli olarak adlandırılan özel bir kod geliştirilmiştir. Bu sezgisel, her kolon türetme aşamasının sonunda DP çözümünün akabinde çalışır ve öncelikle DP çözümünde ondalıklı değer alan ekip eşlemelerinde bulunan uçuşları bulur. Ardından bulunan uçuşların kaç farklı ekip eşlemesinde bulunduğu ve kaç farklı uçuş ile bağlantı yaptığı listelenir. Bulunan bu uçuş bağlantılarına, DP koşumundan elde edilen primal değerler kullanılarak bir önem ölçüsü değeri atanır. Her iterasyonda, bulunan bu bağlantılar önem derecesine göre dizilir ve yüksek primal değerlere sahip olanlar sabitlenir. Sabitlenme işlemi ile birlikte, sözkonusu bağlantıya sahip olmayan ekip eşlemeleri ve görevler sistemden çıkarılır. Uygulanan metod Ryan ve Foster (1981) çalışmasında önerilen tekniğe yakın bir yaklaşımdır. Ryan ve Foster (1981) çalışmasında sunulan fikir Vance ve diğ. (1997) çalışmasında önerilen dal-ücret algoritmasında dallanma mekanizması olarak kullanılmıştır. Bu çalışmada ise dallanma yerine, bağlantıları sabitleme ve bu sabitlenen bağlantıları ihlal eden görev ve ekip eşlemelerinin sistemden çıkarılması amacı ile kullanılmıştır.

Her iterasyonda kaç adet görev elimine edileceğine dair Denklem 4.6'daki gibi bir formül kullanılmıştır.

$$\sqrt{\frac{Ondalıklılar}{Eşlemeler}}\ \frac{\left(\frac{Uçuşlar}{Günler}\right)^2\ \ln(Itr)^2\ \ Görevler}{Uçuşlar}\ EtkiKatsayısı \tag{4.6}$$

Denklem 4.6'da *Ondalıklılar*, ondalıklı primal değere sahip olan toplam değişken (ekip eşlemesi) sayısını temsil etmektedir. *Eşlemeler* toplam aktif (elimine edilmemiş) eşleme sayısını, *Uçuşlar* toplam uçuş sayısı, *Günler* ise plandaki toplam gün sayısını ifade eder. *Itr* aktif iterasyon numarasını ve *Görevler* ise sistemdeki aktif (elimine edilmemiş) görevlerin sayısını temsil eder. *EtkiKatsayısı*, sözkonusu sezgiselin etki gücünü ve dolayısıyla da toplam çözüm süresini ayarlamak için kullanılan ve keyfi seçilen bir katsayıdır. Etki katsayısı için bu çalışmada 0.00035 değeri kullanılmıştır. Daha yüksek değerler optimizasyonun daha yüksek maliyet değerleri ile daha erken sonlanmasına sebep olacaktır. Daha düşük değerler ise daha uzun çalışma süreleri ve daha iyi amaç fonksiyonu değerleri ile sonlanacaktır. Bu nedenle makul bir süre içinde tatmin edici bir çözüm kalitesine ulaşmak için bu değerin optimum olarak berlirlenmesi önem arzetmektedir.

Birçok farklı havayolu firması farklı büyüklüklere sahip olduklarından dolayı, bu formül farklı filo tipleri için herhangi bir iç değişiklik gerektirmeyecek şekilde tasarlanmıştır. Daha büyük filolar için *Uçuşlar* ve *Görevler* daha büyük değerlere sahip olacak ve dolayısıyla da algoritma daha fazla görev eleyecektir. Denklem 4.6 aynı zamanda optimizasyon koşumu esnasındaki sayısal değişimlere duyarlıdır. Yine daha büyük *Ondalıklılar/Eşlemeler* oranı, *Itr* numarası optimizasyon esnasında daha fazla görev eliminasyonunu sağlamaktadır.



Şekil 4.10 : Optimizasyon süresince ondalıklı değişken sayısının değişimi.

Şekil 4.10'da görüldüğü üzere, 11 numaralı iterasyondan sonra ondalıklı değişkenlerin sayısı önemli ölçüde düşmekte ve zamanla denklem daha az görev elimine edilmesine izin vermektedir. Özellikle 26 numaralı iterasyondan sonra

ondalıklı değişkenlerin sayısı neredeyse sıfır "0" olmakta ve takip eden iterasyonlarda çok daha az görev elimine edilmektedir. Farklı filo büyüklükleri ve farklı yoğunluklara sahip tarifeler için probleme daha uygun alternatif denklemler tasarlanıp kullanılabilir. Denklem 4.6 aracılığı ile belirli miktarda görev elimine edildikten hemen sonra ilgili görevleri içeren ve sabitlenen bağlantılar ihlal eden ekip eşlemeleri de sistemden silinir.

Bu sezgiselin uygulanması ile birlikte, zamanla, ana DP'nin ürettiği ondalıklı değişkenlerin sayısı git gide azalmakta ve DP çözümü TP çözümüne yaklaşmaktadır. Yapılan testlerde, bazı ana DP'lerin en son aşamada hiç ondalıklı değişken değeri üretmediği ve TP çözümünün gerekmediği durular ile karşılaşılmıştır. Algoritmanın bu özelliği, nihai dal-sınır algoritmasının hızlı bir şekilde TP çözümünü üretmesini sağlamaktadır.

Şekil 4.10'da aylık ortalama 17000 uçuşu olan bir filonun tarifesi kullanılarak çalıştırılan bir optimizasyon koşumundaki ondalıklı primal değişkenlerin sayısının koşum süresince değişimi görülmektedir. Grafikten açıkça görüleceği üzere ilk iterasyonlarda ondalıklı değişkenlerin sayısı hızlı bir şekilde artmaktadır. Belli bir eşik değerinden sonra hızlı bir şekilde bu değer azalmakta ve dal-sınır algoritmasının kolay bir şekilde çözüm üretebileceği seviyelere inmektedir.

Bağlantı sabitleme sezgiseli ile ilgili önemli bir konu da, bağlantı sabitleme kararı verilirken, sadece tek bir DP çözümünden elde edilen değerleri kullanılması yerine iki veya daha fazla iterasyonun DP değerlerinin gözönüne alınması, uçuş bağlantılarının süreklilik arzettiğini ve sabitlemenin daha güvenli olacağını gösterdiğinden daha sağlıklı sonuçlar üretmektedir.

4.3 Çözüm Stratejisine Genel Bakış

Bu çalışmada önerilen ekip eşleme problemi çözümü yaklaşımı Şekil 4.11'deki gibi bir örnek kodda betimlenmiştir. Şekil 4.11'den görüleceği üzere algoritma tarife verisi olan *Uçuşlar* girdi olarak almaktadır. 2 numaralı satırda tarife verisindeki uçuşlar kullanılarak mümkün olan tüm uçuş görevleri üretilmektedir, 3 numaralı satırda üretilen uçuş görevleri kullanılarak görev-uçuş yatı bağlantısı ağı oluşturulmaktadır ve 4 numaralı satırda ilk ekip eşleme üretimi aşaması gerçekleştirilmektedir. 5 numaralı satırda çözüm yaklaşımının ana kolon türetme

döngüsü başlatılır. 6 numaralı satırda ana DP çözülerek son çözümün primal ve dual bilgisi saklanır. 7 numaralı satırda son üretilen çözümün kalitesi kontrol edilir ve eğer tatmin edici bir seviyeye ulaşılmış ise algoritma dal-sınır algoritması kullanarak TP çözümünü çalıştırır ve elde edilen tamsayı çözüm kaydedilerek optimizasyon süreci sonlandırılır. Eğer çözüm kalitesi yeterli seviyede değil ise 12 numaralı satırda bağlantı sabitleme sezgiseli çalıştırılır ve 13 numaralı satırda kısmi çözüm üretme yaklaşımı yeni ekip eşlemeleri üretmek için başlatılır.

```
1  procedure Ekip_Eşleme_Problemini_Çöz(Uçuşlar)
2    GörevListesi = Tüm_Görevleri_Üret(Uçuşlar);
3    ÜcretlendirmeAğı = Ücretlendirme_Ağını_Oluştur(GörevListesi);
4    EşlemeListesi = İlk_Ekip_Eşlemelerini_Oluştur(Uçuşlar, ÜcretlendirmeAğı);
5    while true do
6        [PrimalBilgi, DualBilgi] = Ana_DP_yi_Çöz(EşlemeListesi, Uçuşlar);
7        if çözüm kalitesi yeterli ise
8            TpÇözümü = Ana_TP_yi_Çöz(EşlemeListesi, Uçuşlar);
9            Çözümü_Kaydet(TpÇözümü);
10           Çık;
11       else
12           Bağlantı_Sabitleme_Sezgiselini_Uygula(EşlemeListesi, GörevListesi, PrimalBilgi);
13           Kısmi_Çözüm_Üretme_Yaklaşımını_Uygula(EşlemeListesi, Uçuşlar, ÜcretlendirmeAğı, DualBilgi);
```

Şekil 4.11 : Önerilen çözüm stratejinin genel şeması.

5. DENEYSEL SONUÇLAR VE TAVSİYELER

Bu çalışmada sununlan test sonuçları, Java 7 platformunda geliştirilen bir kod ile üretilmiştir. DP ve TP çözücüsü olarak ise ILOG'un Cplex kütüphanesi kullanılmıştır. Donanım olarak ise 1.9 ghz hızında bir Intel Core i5 işlemcisine sahip tek işlemcili bir dizüstü bilgisayar kullanılmıştır.

Kullanılan test verisi olarak ise Türk Hava Yolları'nın 2014 yılında uyguladığı ilk altı aylık tarife verisi kullanılmıştır. Türk Hava Yolları, dört adet kokpit ve üç adet kabin filosuna sahiptir. Bu çalışmada karmaşık kural gereksinimleri ve en büyük kokpit filosu olması sebebi ile kokpit 320 filosu kullanılmıştır. Bu filo aynı zamanda kabin dar gövde filosuna çok yakın özellikler ihtiva eden tipik bir filodur.

Çizelge 5.1 : Test verisi.

Month	Uçuşlar	Görevler	İlk ekip eşlemeleri
Ocak	17268	416153	8190
Şubat	15738	393285	7458
Mart	17318	390973	8200
Nisan	16833	439661	7961
Mayıs	16127	406233	7733
Haziran	15656	401527	7437

Çizelge 5.1'de kullanılan tüm veri kümelerinin detayları görülmektedir. Çizelge 5.1'de bulunan her satır planlama ayının tüm uçuşlarını ve bir sonraki aydan da 10 günlük bir tarife verisini içermektedir. Özellikle planlama yapılan ayın son günlerindeki uçuşları kapsayan ekip eşlemelerinin bir sonraki ayda sonlanması gerekebileceğinden, bir sonraki ayın tarife verisinin kullanılması gerekmektedir. Bu nedenle üretilen aylık ekip eşleme çözümleri girdi kümesindeki uçuş listesinden daha az uçuş kapsamaktadır.

Çizelge 5.1'deki *Görevler* sütunu, görev üretme aşamasında her ay için toplam kaç adet uçuş görevi üretildiğini göstermektedir. Bu üretilen görevler aynı zamanda çalışmada kullanılan görev-uçuş yatı bağlantısı ağının oluşturulmasında da kullanılmaktadır. *İlk ekip eşlemeleri* sütunu ise ana kolon türetme döngüsüne girilmeden önce sistemin hazırlandığı aşamalardan olan ilk ekip eşlemeleri üretme aşamasında üretilen toplan ekip eşlemesi sayısını göstermektedir.

Türk Hava Yolları'nda hali hazırda kullanılan sistemlerde, optimizasyon amaç fonksiyonu olarak karmaşık bir yapı kullanılmaktadır. Bu çalışmada da, farklı önem derecelerine sahip birçok terim içeren aynı amaç fonksiyonu uygulanmıştır. Terimlerin birçoğunun etkisinin küçük seviyelerde olması ve özellikle bu çalışmanın kapsamına daha rahat odaklanabilmek için, test sonuçlarının sunulduğu karşılaştırma tablolarında nu maliyet katsayılarından en büyük etkiye ve öneme sahip olan üç tanesi gösterilmiştir. Bu maliyet katsayıları Türk Hava Yolları'nda aktif olarak kullanılmakta olan üç farklı parametre kümesinden elde edilmiştir.

Çizelge 5.2 : Paremetre setleri.

Parametre kümesi	Adam gün	Pas (deadhead) saati	Yatı
PK 1	60000	500000	10000
PK 2	200000	100000	10000
PK 3	60000	100000	100000

Çizelge 5.2'de görüleceği üzere bu maliyet katsayıları her görev günü, her pas saati ve her otel yatısı gecesi için uygulanmaktadır. Üretilen nihai ekip eşleme çözümlerinden bir toplam maliyet değeri bulunuyor olsa da, Türk Hava Yolları'nda hali hazırdaki uygulanan pratiklerde, üretilen ekip eşleme çözümlerinin değerlendirilmesi aşamasında çıktı raporlarındaki bu kalite kriterleri kullanılır. Operasyonel zorunluluklar sebebiyle maalesef çözümler arasında gerçek bir finansal maliyet karşılaştırması yapılmamaktadır. Ekip planlama bölümü genellikle yüksek yoğunluklu dönemlerde toplam pas (deadhead) süresinin, ekip kullanımının maksimum olması gerektiği, müsait ekip bulmanın güç olduğu dönemlerde ise toplam adam gün sayısının minimize edilmesine odaklanır.

Önerilen çözüm yaklaşımı ile Türk Hava Yolları'nda kullanılan yaklaşımın karşılaştırıldığı detaylı tablolar Çizelge 5.3, Çizelge 5.4 ve Çizelge 5.5'teki gibidir.

Çizelge 5.3 : Parametre kümesi 1 için kalite kriterleri ve maliyet raporu.

	Ocak		Şubat		Mart		Nisan		Mayıs		Haziran	
KRİTER	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**
Eşleme sayısı	5053	5052	4468	4483	4978	4975	4755	4743	4545	4563	4520	4481
Görev sayısı	5886	5884	5199	5203	5862	5860	5702	5704	5636	5627	5474	5449
Kapsanan uçuş sayısı	13811	13799	12307	12307	13783	13775	13346	13351	12916	12915	12449	12432
Pas (deadhead) sayısı	66	64	56	54	84	80	56	54	76	72	72	74
Adam gün	14824	14826	13102	13112	14726	14730	14406	14412	14358	14358	14028	13954
Yurtiçi yatı sayısı	786	808	692	680	718	720	580	628	722	754	668	706
Yurtdışı yatı sayısı	1338	1336	1170	1170	1518	1524	1792	1790	1936	1892	1670	1660
Uçuş süresi (sa)	68518.40	68487.40	60598.40	60599.30	67966.20	67938.40	65634.10	65655.00	64493.00	64483.30	62745.10	62668.00
Görev süresi (sa)	107528.10	107268.30	95520.30	95227.20	107066.50	106803.20	103385.40	103436.10	101678.20	101368.30	98086.00	97662.20
Yatı süresi (sa)	30531.30	30761.10	27030.50	26761.20	33080.40	33509.20	35401.50	36199.30	41844.00	41149.50	38104.00	38444.30
Pas süresi (sa)	162.00	157.20	142.50	137.00	216.50	212.40	157.50	153.40	203.40	190.50	195.10	194.20
Eşleme süresi (sa)	138059.40	138029.40	122551.20	121988.40	140147.30	140312.40	138787.30	139635.40	143522.20	142518.20	136190.00	136106.50
Gün başına uçuş saati	4.37	4.37	4.37	4.37	4.36	4.36	4.33	4.33	4.29	4.29	4.28	4.29
Koşum süresi (dk)	23.59	25.18	33.35	29.31	29.52	26.23	53.37	30.02	57.12	31.30	35.59	33.13
MALİYET												
Adam gün maliyeti	44844000	44850000	39642000	39672000	44514000	44526000	43374000	43392000	43074000	43074000	42108000	41886000
Otel maliyeti	2094000	2114000	1840000	1829000	2206000	2216000	2352000	2398000	2637000	2626000	2310000	2337000
Otel transfer maliyeti	7904	7928	6936	6856	8368	8376	8920	8376	9848	9624	8544	8560
Pas maliyeti	8099996	7866663	7141663	6849997	10841663	10633329	7891665	7683332	10183331	9541665	9758331	9716664
Sanal maliyetler	54124801	54202312	47920076	47952197	53528965	53667513	51059411	51249799	51329634	51411712	50304289	50092492
Toplam maliyet	109170701	109040903	96550675	96310050	111098996	111051218	104685996	104731507	107233813	106663001	104489164	104040716

Çizelge 5.4 : Parametre kümesi 2 için kalite kriterleri ve maliyet raporu.

	Ocak		Şubat		Mart		Nisan		Mayıs		Haziran	
KRİTER	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**
Eşleme sayısı	4456	4392	3932	3909	4500	4437	4337	4352	4312	4305	4199	4195
Görev sayısı	5697	5678	5044	5029	5710	5689	5548	5549	5529	5508	5335	5326
Kapsanan uçuş sayısı	13874	13871	12393	12377	13803	13812	13404	13401	12955	12945	12463	12458
Pas (deadhead) sayısı	170	186	154	144	170	220	158	152	236	202	222	220
Adam gün	14396	14362	12730	12718	14350	14312	14024	14038	14072	14054	13680	13690
Yurtiçi yatı sayısı	1310	1454	1176	1284	1232	1372	1030	998	896	978	984	1064
Yurtdışı yatı sayısı	1762	1782	1574	1560	1798	1802	1916	1918	1998	1932	1760	1720
Uçuş süresi (sa)	68838.00	68792.40	60996.10	60906.30	68061.20	68086.30	65901.40	65900.30	64669.10	64620.20	62820.10	62805.20
Görev süresi (sa)	108438.40	108136.20	96660.20	96155.10	107682.00	107614.50	103495.40	103356.10	101858.10	101633.10	98019.50	98004.40
Yatı süresi (sa)	48369.30	50667.10	43107.50	44191.50	46719.40	48656.40	46500.10	46266.50	46402.40	46331.30	45397.10	45714.20
Pas süresi (sa)	310.20	326.50	299.00	276.50	346.30	405.00	316.10	283.50	444.50	382.50	398.30	401.50
Eşleme süresi (sa)	156808.10	158803.30	139768.10	140347.00	154401.40	156271.30	149995.50	149623.00	148260.50	147964.40	143417.00	143719.00
Gün başına uçuş saati	4.46	4.47	4.47	4.47	4.44	4.45	4.41	4.41	4.35	4.35	4.35	4.35
Koşum süresi (dk)	24.54	34.13	35.45	37.27	20.55	29.52	27.13	33.05	31.25	33.48	23.48	35.55
MALİYET												
Adam gün maliyeti	145200000	144860000	128420000	128300000	144620000	144240000	140760000	140900000	140720000	140540000	136880000	136980000
Otel maliyeti	3019000	3184000	2696000	2795000	2973000	3117000	2923000	2894000	2872000	2889000	2717000	2757000
Otel transfer maliyeti	10872	11256	9792	9864	10720	11024	10376	10240	10520	10432	9664	9648
Pas maliyeti	3103284	3268280	2989956	2768294	3464950	4049934	3161636	2838302	4448274	3828284	3984948	4018284
Sanal maliyetler	160950778	160948695	142381286	142514994	159525007	159583148	153974277	154172687	153511336	153481931	149856798	150153961
Toplam maliyet	312283934	312272231	276497034	276388152	310593677	311001106	300829289	300815229	301562130	300749647	293448410	293918893

Çizelge 5.5 : Parametre kümesi 3 için kalite kriterleri ve maliyet raporu.

	Ocak		Şubat		Mart		Nisan		Mayıs		Haziran	
KRİTER	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**	**TK**	**Yeni strateji**
Eşleme sayısı	5276	5294	4672	4683	5296	5304	4840	4851	4765	4747	4661	4656
Görev sayısı	5926	5932	5240	5248	5942	5939	5707	5724	5657	5633	5485	5463
Kapsanan uçuş sayısı	13786	13779	12289	12291	13766	13758	13332	13342	12905	12898	12426	12422
Pas (deadhead) sayısı	112	104	88	86	136	136	102	90	152	162	148	148
Adam gün	15038	15062	13294	13310	15012	15022	14428	14472	14448	14420	14108	14098
Yurtiçi yatı sayısı	496	494	416	418	432	444	438	460	388	458	446	488
Yurtdışı yatı sayısı	1290	1280	1138	1138	1350	1334	1752	1760	1844	1794	1608	1574
Uçuş süresi (sa)	68426.40	68406.20	60509.00	60509.30	67888.20	67851.30	65576.50	65631.20	64441.40	64408.40	62654.20	62641.40
Görev süresi (sa)	107239.50	107144.00	95197.20	95001.50	106758.00	106711.10	103060.10	103168.20	101140.00	101188.50	97630.00	97587.40
Yatı süresi (sa)	24674.00	24615.00	21735.40	21813.10	24553.10	24440.30	32032.20	32574.10	33769.10	33711.10	33124.10	33101.40
Pas süresi (sa)	218.10	204.40	180.50	174.50	283.50	283.10	243.30	217.50	331.40	340.10	283.10	287.10
Eşleme süresi (sa)	131913.50	131759.00	116933.00	116815.00	131311.10	131151.40	135092.30	135742.30	134909.10	134900.00	130754.10	130689.20
Gün başına uçuş saati	4.33	4.32	4.33	4.32	4.31	4.31	4.32	4.32	4.27	4.27	4.26	4.26
Koşum süresi (dk)	17.52	34.01	19.28	22.34	29.26	17.32	46.20	25.25	31.59	26.54	34.50	20.20
MALİYET												
Adam gün maliyeti	45486000	45558000	40218000	40266000	45372000	45402000	43440000	43572000	43344000	43260000	42348000	42318000
Otel maliyeti	17810000	17740000	15500000	15530000	17770000	17770000	21720000	22040000	22170000	22390000	20380000	20490000
Otel transfer maliyeti	6800	6704	5888	5872	6792	6712	8280	8328	8336	8288	7520	7384
Pas maliyeti	2181628	2046632	1808304	1748304	2838292	2831630	2434978	2178312	3316626	3401630	2831638	2871638
Sanal maliyetler	54598642	54779993	48335351	48437415	53847830	53955006	51019136	51318641	50975177	50967367	50298788	50405386
Toplam maliyet	120083070	120131329	105867543	105987591	119834914	119965348	118622394	119117281	119814139	120027285	115865946	116092408

Karşılaştırma tablolarından da açıkça görüleceği üzere bu çalışma çerçevesinde geliştirilen strateji, Türk Hava Yolları'nda halihazırda kullanılan en ileri teknolojilerin, çözücülerin ve tamsayı algoritmalarının kullanıldığı sistemler (TK) ile karşılaştırıldığında, son derece rekabetçi sonuçlar üretebilme kabiliyetine sahiptir.

Çizelge 5.6 : Kalite kriterleri yüzdelik değişimleri özet karşılaştırma raporu.

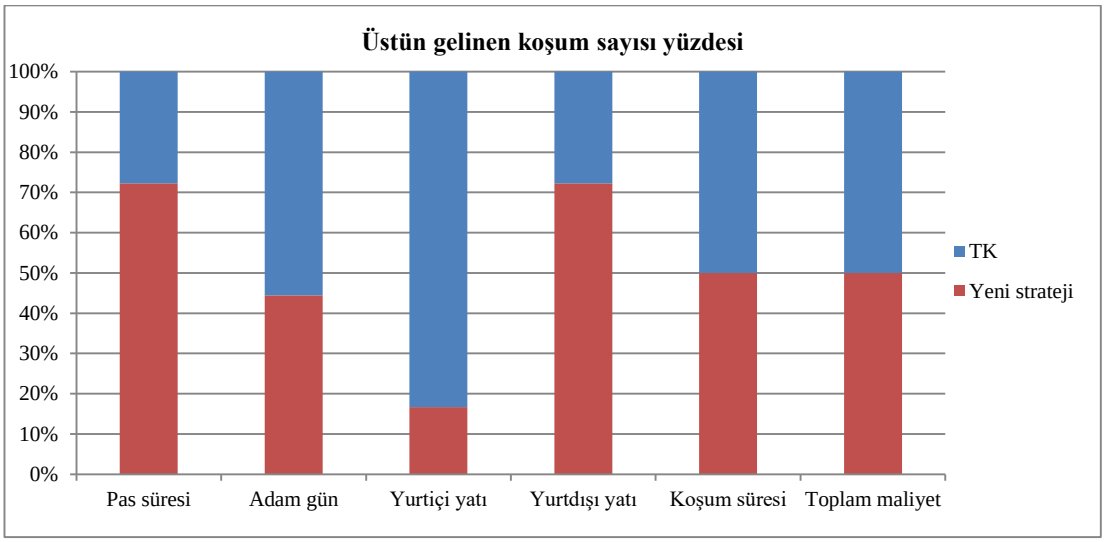
		Ocak	Şubat	Mart	Nisan	Mayıs	Haziran
PK 1	Pas süresi (sa)	2.96%	3.86%	1.89%	2.60%	6.34%	0.46%
	Adam gün	-0.01%	-0.08%	-0.03%	-0.04%	0.00%	0.53%
	Yurtiçi yatı sayısı	-2.80%	1.73%	-0.28%	-8.28%	-4.43%	-5.69%
	Yurtdışı yatı sayısı	0.15%	0.00%	-0.40%	0.11%	2.27%	0.60%
	Toplam maliyet	0.12%	0.25%	0.04%	-0.04%	0.53%	0.43%
PK 2	Pas süresi (sa)	-5.25%	7.53%	-16.95%	10.31%	13.95%	-0.80%
	Adam gün	0.24%	0.09%	0.26%	-0.10%	0.13%	-0.07%
	Yurtiçi yatı sayısı	-10.99%	-9.18%	-11.36%	3.11%	-9.15%	-8.13%
	Yurtdışı yatı sayısı	-1.14%	0.89%	-0.22%	-0.10%	3.30%	2.27%
	Toplam maliyet	0.00%	0.04%	-0.13%	0.00%	0.27%	-0.16%
PK 3	Pas süresi (sa)	6.28%	3.32%	0.14%	10.60%	-2.63%	-1.41%
	Adam gün	-0.16%	-0.12%	-0.07%	-0.30%	0.19%	0.07%
	Yurtiçi yatı sayısı	0.40%	-0.48%	-2.78%	-5.02%	-18.04%	-9.42%
	Yurtdışı yatı sayısı	0.78%	0.00%	1.19%	-0.46%	2.71%	2.11%
	Toplam maliyet	-0.04%	-0.11%	-0.11%	-0.42%	-0.18%	-0.20%

Çizelge 5.6'da tüm ekip eşleme çözümleri arasındaki kritik plan değerlendirme kriterlerinin değişimleri görülmektedir. Yeşil arkaplan rengi ile gösterilen pozitif değerler bu çalışmada önerilen stratejinin kullanılması ile elde edilen iyileşme miktarlarının yüzdelik ifadeleri göstermektedirler. Özet tablodan da görüleceği üzere önerilen yaklaşımın özellikle yoğun sezonlarda ekip elemanlarının satılabilecek yolcu koltuklarını işgal etmeleri ile havayolu şirketinin gelir kaybetmesine sebep olan *pas (deadhead) süresi* kriteri için daha iyi değerler üretebilme konusunda daha başarılı olduğu gözükmektedir. Önerilen strateji, özellikle parametre kümesi 1'de (*PK 1*) pas süresi kriterinde tüm aylarda daha iyi sonuçlar üretmiştir. *Adam gün* değerleri de özellikle havayolu şirketinin müsait pilot bulmak konusunda zorlandığı dönemler için oldukça önemli bir diğer kriterdir. Bu durum ile özellikle her yılın Aralık aylarında pilotlardan bir kısmının yıllık uçuş limitlerini doldurmaları sebebiyle karşılaşılır. *Adam gün* kriteri gözönüne alındığında da önerilen strateji ve TK sisteminin birbirine çok yakın, rekabetçi değerler ürettiği görülmektedir. *Yatı sayısı* kriteri incelendiğinde ise önerilen stratejinin, yurtiçi yatılar haricinde yine TK sistemine göre daha iyi değerler üretebildiği görülmektedir. Yurtiçi yatılar ise yurtdışı yatılara kıyasla operasyonel olarak daha kolay ve maliyeti düşük olduğu için yüksekliği tercih edilebilir bir kriterdir.

Toplam maliyet kriteri sistemlerin genel optimizasyon performansını göstermektedir. Sonuçlardan görüleceği üzere önerilen strateji, daha yüksek yatı cezası katsayısı uygulanan parametre kümesi 3 (*PK 3*) haricindeki diğer tüm parametre kümelerinde

daha iyi değerler üretebilmektedir. Bunun temel sebebi ise önerilen strateji tarafından üretilen çözümlerin daha yüksek yurtiçi yatı sayılarına sahip olmasıdır ve bu durum önerilen stratejinin tek dezavantajı olarak gözükmektedir. Küçük farklılıklar olsa da tüm *toplam maliyet* değerleri, aralarındaki farkın % 0.5 i geçmediği, birbirine çok yakın değerler olduğu görülmektedir. Dolayısıyla üretilen ekip eşleme çözümlerinin değerlendirilmesinde, içinde bulunulan şartlar altında operasyonel olarak önem arzeden kalite kriterlerinin kullanılması daha makul gözükmektedir.

Tüm toplam koşum süreleri makul değerler olarak gözükmektedir. Ekip planlama bölümü, farklı senaryoların sonuçlarını görmek ve değerlendirmek için bir iş günü içerisinde bir çok çözüm üretebilmektedirler. Optimizasyon koşumlarının yarısında, önerilen strateji daha iyi *toplam koşum süresi* performansı göstermiştir. Toplam koşum süreleri karşılaştırılırken, önerilen stratejinin tek kanallı (single threaded) bir kod ile bir dizüstü bilgisayar üzerinde sonuç ürettiği halde TK sisteminin dört paralel kanallı koşumlar ile güçlü parallelik özellikleri olan bir donanım üzerinde sonuç ürettiği gözönüne alınmalıdır.



Şekil 5.1 : Toplam üstün gelinen koşum sayısı yüzdesi.

Şekil 5.1'deki üstün gelinen koşumların sayılarının toplam koşum sayısına oranının yüzde cinsinden gösterildiği grafikte görüleceği üzere, önerilen strateji koşumları %70 inden fazlasında daha iyi *pas süresi* ve *yurtdışı yatı* değerleri üretmiştir. *Adam gün*, *koşum süresi* ve *toplam maliyet* kriterleri gözönüne alındığında ise her iki sistem de ortalama olarak tüm koşumların yarısında daha iyi değerler üretmişlerdir. *Yurtiçi*

yatı kriteri için ise TK sistemi koşumların ortalama %80 inde daha iyi değerler üretmiştir.

6. SONUÇ VE VİZYON

Sunulan bu çalışmada, literatürde yoğun bir şekilde çalışılmış olan havayolu ekip eşleme problemi yeniden ele alınmış ve problemin çözümü için bir kolon türetme stratejisi önerilmiştir. Ekip eşleme, ekip planlama sürecinin en önemli maliyet belirleyici aşamasıdır. Bu sebeple ekip eşlemelerinin optimize bir şekilde üretilmesi, ekip maliyetlerinin düşürülmesi ve uçucu ekiplerin zamanlarının verimli kullanımının arttırılması için oldukça önemlidir. Bu çalışmada sunulan problemlerde olduğu gibi gerçek dünya problemlerinde, gerçek finansal maliyetlerin yanında, ekip eşleme optimizasyonunun adam gün, pas süresi, yatı sayısı gibi ölçütler üzerinden yürütülmesi ve değerlendirilmesi özellikle havayolu şirketlerinin karşılaştığı operasyonel zorlukları aşabilmesi için oldukça önemlidir.

Bu çalışmada sunulan çözüm yaklaşımının geliştirilmesi sürecinde; var olan yaklaşımlar değerlendirilmiş, bir kısmı adapte edilmiş, yeni modeller ve stratejiler geliştirilmiştir. İki sistem arasında sağlıklı ve objektif test sonuçları ve karşılaştırmalar elde edilebilmesi için Türk Hava Yolları'nda aktif olarak kullanılan sistemin sahip olduğu amaç fonksiyonu modeli ve kural kümesi, önerilen sistem üzerinde aynen kodlanmıştır. Test sonuçları bölümünde de görüleceği üzere önerilen strateji, Türk Hava Yolları'nda halihazırda kullanılan en ileri teknolojilerin, çözücülerin ve tamsayı algoritmalarının kullanıldığı sistemler (TK) ile karşılaştırıldığında, son derece rekabetçi ve kaliteli çözümler üretebilme kabiliyetine sahiptir. Önerilen strateji özellikle pas süresi ve yurtdışı yatı kriterlerinin düşürülmesinde daha başarılıdır. Toplam optimizasyon koşum süresi açısından ise birbirine çok yakın değerler elde edilmiştir. Burada Türk Hava Yolları'nda kullanılan sistemlerden elde edilen test sonuçlarının, gelişmiş sunucular üzerinde dört kanallı optimizasyon koşumları ile elde edilmiş olması gözönüne alınmalıdır.

Başarılı sonuç raporlarının yanında, bu çalışmanın literatüre üç önemli katkısı bulunmaktadır:

- Görev-uçuş yatı bağlantısı ağı (Duty-flight overnight connection network)

- Kısmi çözüm üretme yaklaşımı (Partial solution generation approach)
- Bağlantı sabitleme sezgiseli (Connection fixing heuristics)

Bunlardan ilki yeni bir ağ tasarımı olan görev-uçuş yatı bağlantısı ağı olarak adlandırılan yapıdır. 4.1.2 ve 4.2.2' de anlatıldığı üzere önerilen ağ tasarımı, sunulan problemlerin çözümünde ihtiyaç duyulan ücretlendirme ağını daha küçük bilgisayar hafızasına ihtiyaç duyan, sıkı (compact) bir temsil imkanı sağlamaktadır. Önerilen ağ modeli aynı zamanda görevler arası bağlantılar ile ilgili kuralların model üzerinde tam olarak temsil edilebilmesini sağlamakta ve bu da, daha az sayıda illegal ekip eşlemesi üretmesini sağlamaktadır. Dolayısıyla diğer yaklaşımlara göre daha az model modifikasyonu gerektirmekte ve bu durum önerilen modelin daha verimli çalışmasını sağlamaktadır.

İkinci önemli katkı ise önerilen stratejinin yüksek kalitede çözüm üretebilmesinde ana rolü oynayan kısmi çözüm üretme yaklaşımıdır. Bu teknik; ücretlendirme ağının oluşturulması, ekip eşlemelerinin üretilmesi konularında açık bir strateji sağlar. 4.2'de bahsedildiği üzere bu yaklaşım, önceki iterasyonlarda üretilen ekip eşlemeleri ile ilgili bilgiyi kullanmakta ve tarife boyunca daha iyi bir şekilde (homojen) dağılmış ekip eşlemeleri üretilmesini sağlamaktadır. Bu muntazam dağılım, aynı zamanda dual değer stabilizasyonu mekanizmasının daha verimli çalışmasını sağlamaktadır.

Bağlantı sabitleme sezgiseli ise üçüncü önemli katkıdır. Bu sezgisel 4.2.3'te açıklandığı üzere uçuş bağlantılarını önem derecelerine uygun bir şekilde sabitleyerek uçuş görevlerinin ve ekip eşlemelerinin bir kısmını koşum süresince elimine eder ve bu sayede bir ağaç yapısı üzerinde optimizasyon koşumunun yönünü tayin eder. Özellikle nihai TP çözümü için kullanılan dal-sınır algoritmasının çalışma performansını önemli ölçüde arttırır. Bazı testlerde, son iterasyonlardaki DP koşumlarında tamsayı çözüm elde edilebildiği ve nihai dal-sınır algoritmasının çalıştırılması ihtiyacını ortadan kaldırdığı görülmüştür. Aynı zamanda, ücretlendirme ağının boyutunun optimizasyon süresince azalttığı için, DP ve ücretlendirme alt problemlerinde de performans artışı sağlamaktadır.

Elde edilen karşılaştırma sonuçları, havayolu firmalarının operasyonlarına ve ihtiyaçlarına daha uyumlu, gürbüz (robust) ekip eşlemesi çözümleri üretilebilmesi için matematiksel ve/veya sezgisel teknikler üzerinde daha ileri iyileştirmeler,

geliştirmeler yapılması konusunda önemli bir kapı açmaktadır. Geliştirilen strateji ve elde edilen bu ilk sonuçlar, bizleri havayolu işletmeciliği alanı içerisindeki diğer optimizasyon problemleri konusunda da çalışmaya teşvik eden başarılı ve önemli bir adımdır olma niteliğindedir.

Aynı zamanda stratejik ekip planlama yazılımlarının dünya çapında sadece bir iki yabancı firma tarafından geliştiriliyor olması ve Türk Hava Yolları gibi büyük filolara sahip firmaların ihtiyaçlarını karşılayan yerli bir optimizasyon çözümünün olmaması bu çalışmanın yapılmasındaki temel motivasyon unsurlarındandır. Elde edilen başarılı sonuçlar da benzeri ve hatta daha iyi bir algoritmanın/sistemin yerli insan kaynağı ve bilgi birikimi ile rahatlıkla üretilebileceğini göstermektedir.

Önümüzdeki dönemde, bu çalışmada sunulan stratejinin paralelleştirilmesi konusunda yapılacak çalışmalar, daha iyi koşum süresi değerleri elde edilebilecek önemli bir araştırma ve geliştirme konusu adayıdır. Algoritmanın performansını iyileştirmek, özellikle de daha az yurtiçi yatı sayısı veren çözümler üretilebilmek üzere geliştirmeler yapılması da önemli bir ilerleme noktasıdır. Bu çalışma aynı zamanda ekip eşleme aşamasının ardından yürütülen ekip atama aşaması için de bir ön çalışma niteliğindedir. Ekip atama aşamasının başarısı büyük oranda üretilen ekip eşleme çözümlerinin kalitesine bağlıdır.

KAYNAKLAR

Aickelin, U. (2002). An indirect genetic algorithm for set covering problems. *Journal of the Operational Research Society*, *53*(10), 1118–1126.

Amor, B. H., Frangioni, A. & Desrosiers, J. (2009). On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, *157*(6), 1167–1184.

Andersson, E., Housos, E., Kohl, N. & Wedelin, D. (1997). Crew Pairing Optimization. In G. Yu (ed.), *Operations Research in the Airline Industry* (pp. 228–258). Norwell, MA: Kluwer Academic.

Baker, E., Bodin, L., Finnegan, W. & Ponder, R. (1979). Efficient heuristic solutions to an airline crew scheduling problem. *AIIE Transactions*, *11*(2), 79–85.

Barnhart, C., Cohn, A. M., Johnson, E. L., Klabjan, D., Nemhauser, G. L. & Vance, P. H. (2003). Airline Crew Scheduling, *Handbook of Transportation Science* (2nd ed., pp. 517-560). Boston, MA: Springer US.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P. & Vance, P. H. (1998). Branch-and-price: column generation for solving huge integer programs. *Operations Research*, *46*, 316-329.

Beasley, J. E., & Chu, P. C. (1996). A genetic algorithm for the set covering problem. *European Journal of Operational Research*, *94*(2), 392–404.

Cant´u-Paz, E. (2007). Parameter Setting in Parallel Genetic Algorithms. *Studies in Computational Intelligence 54*, 259–276.

Dantzig, G. B. (1949). Programming in a linear structure. *Econometrica*, *17*, 73–74.

Dantzig, G. B. (1963). *Linear Programming and Extensions*. Princeton, NJ: Princeton University Press.

Desaulniers, G. & Soumis, F. (2010). Airline Crew Scheduling by Column Generation. *CIRRELT Spring School on Combinatorial Optimization in Logistics*. Montréal, Canada.

Desrosiers, J. & Lübbecke, M. E. (2005). A primer in column generation. *Column Generation* (pp. 1-32), Boston, MA: Springer US.

Erdoğan, G., Haouari, M., Örmeci, M. & Özener, O. Ö. (2015). Solving a large-scale crew pairing problem. *Journal of the Operational Research Society*, *66*(10), 1742-1754.

Garey, M. R. & Johnson, D. S. (1979). *Computers and Intractibility; A Guide to the Theory of NP-Completeness*. New York, NY: H. Freeman & Company.

Greenberg, H. J. (1997). *Klee-Minty Polytope Shows Exponential Time Complexity of the Simplex Method*, University of Colorado, Denver, US.

Gustafsson, T. (1999). *A heuristic approach to column generation for airline crew scheduling, (Lisansüstü tezi)*. Department of Mathematics, Chalmers University of Technology, Gothenburg, Sweden.

Gonzales Hernandez, L. F. (1996). Evolutionary Divide and Conquer for the Set-Covering Problem. *Lecture Notes In Computer Science, 1143*(198-208).

Harik, G., Cantu´-Paz, E., Goldberg, D. E. & Miller, B. L. (1999). The Gambler's Ruin Problem. Genetic Algorithms, and the Sizing of Populations, *Evolutionary Computation, 7*(3), 231-253.

Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. MIT Press.

ILOG Cplex Optimization Studio. http://www-01.ibm.com/software/info/ilog/

İTÜ. Ulusal Yüksek Başarımlı Hesaplama Merkezi. http://uybhm.itu.edu.tr

Java. http://www.oracle.com/technetwork/java/index.html

Karmarkar, N. (1984). A New Polynomial Time Algorithm for Linear Programming. *Combinatorica, 4*(4), 373–395.

Khachiyan, L. G. (1979). A Polynomial Algorithm in Linear Programming. *Doklady Akademiia Nauk SSSR, 244*,1093–1096. (English translation: *Soviet Mathematics Doklady, 20*(1), 191–194).

Klabjan, D. , Johnson, E. L. , Nemhauser, G. L. , Gelman, E. , & Ramaswamy, S. (2001). Solving Large Airline Crew Scheduling Problems: Random Pairing Generation and Strong Branching. *Computational Optimization and Applications, 20*(1), 73–91.

Kornilakis, H. & Stamatopoulos, P. (2002). Crew Pairing Optimization with Genetic Algorithms. *Lecture Notes In Computer Science, 2308*, 109–120.

Land, A. H. & Doig, A. G. (1960). An automatic method for solving discrete programming problems. *Econometrica, 28*, 497-520.

Lasdon, L. S. (2002). Optimization Theory for Large Systems. Mineola, NY: *Dover*.

Luque, G. & Alba, E. (2011). Parallel Models for Genetic Algorithms. *Parallel Genetic Algorithms* (SCI 367, pp. 15–30), Boston, MA: Springer US.

Luque, G. & Alba, E. (2011). Theoretical Models of Selection Pressure for Distributed Gas. *Parallel Genetic Algorithms* (SCI 367, pp. 15–30). Boston, MA: Springer US.

Lübbecke, M.E. (2011). Column generation. *Wiley Encyclopedia of Operations Research and Management Science* (pp. 1-19). New York, NY: John Wiley & Sons Inc.

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

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

Merle, O. D., Villeneuve, D., Desrosiers, J. & Hansen, P. (1999). Stabilized column generation. *Discrete Math*, *194*, 229–237.

Nemhauser, G.L. & Wolsey, L. A. (1988). Integer and Combinatorial optimization. New York, NY: John Wiley & Sons Inc.

Ozdemir, H. T. & Mohan, C. K. (2001). Flight graph based genetic algorithm for crew scheduling in airlines. *Information Sciences*, *133*(3-4), 165–173.

Park, T. & Ryu, K. R. (2006). Crew pairing optimization by a genetic algorithm with unexpressed genes. *Journal of Intelligent Manufacturing*, *17*(4), 375-383.

Ryan, D. M. & Foster, B. A. (1981). An integer programming approach to scheduling. *Computer Scheduling of Public Transport: Urban Passenger Vehicle and Crew Scheduling*, (pp. 269–280). Amsterdam: Elsevier Science Publishers BV.

SHGM Sivil havacılık Talimatı. (2014). Uçucu Ekip Uçuş Görev ve Dinlenme Süreleri ile Uygulama Esasları Talimatı. *Türkiye Cumhuriyeti Sivil Havacılık Genel Müdürlüğü*,

http://web.shgm.gov.tr/documents/sivilhavacilik/files/mevzuat/sektorel/talimatlar/sht_6a_50_rev06.pdf.

Tongchim, S. & Chongstitvatana, P. (2000, January 26-28). Parallel genetic programming: synchronous and asynchronous migration. *5th International Symposium on Artificial Life and Robotics*. Oita, Japan.

Vance, P., Atamtürk, A., Barnhart, C., Gelman, E., Johnson, E., Krishna, A., Mahidhara, D., Nemhauser, G. & Rebello, R. (1997). *A heuristic branch-and-price approach for the airline crew pairing problem* (Technical Report). Georgia Institute of Technology.

Vanderbeck, F. (2005). Implementing mixed integer column generation. *Column Generation* (pp. 331-358). Boston, MA: Springer US.

Wedelin, D. (1995). An algorithm for large scale 0-1 integer programming with application to airline crew scheduling. *Annals of Operations Research*, *57*, 283-301.

Zeren, B., & Özkol, I. (2012). An improved genetic algorithm for crew pairing optimization. *Journal of Intelligent Learning Systems and Applications*, *4*(1), 70–80.

Zeren, B., & Özkol, I. (2016). A novel column generation strategy for large scale airline crew pairing problems. *Expert Systems with Applications*, *55*, 133-144.

ÖZGEÇMİŞ



Ad-Soyad **:**Bahadır ZEREN

Doğum Tarihi ve Yeri **:**1978, İstanbul

E-posta **:** bzeren@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** **:** 2001, İstanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri Fakültesi, Uçak Mühendisliği Bölümü
- **Yükseklisans** **:** 2007, İstanbul Teknik Üniversitesi, Uçak ve Uzay Bilimleri Fakültesi, Uçak Mühendisliği Bölümü

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Zeren, B.**, Özkol, İ., (2016). A Novel Column Generation Strategy for Large Scale Airline Crew Pairing Problems, *Expert Systems with Applications*, *55*, 133-144.
- **Zeren, B.**, Özkol, İ., (2012). An Improved Genetic Algorithm for Crew Pairing Optimization, *Journal of Intelligent Learning Systems and Applications*, *4*(1), 70-80.