

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMİNDE
TEKRARSIZ KROMOZOM DESTEKLİ
PARALEL GENETİK ALGORİTMA UYGULAMASI**

DOKTORA TEZİ

Şafak EBESEK

Mimarlık Anabilim Dalı

Yapı Bilimleri Programı

MAYIS 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMİNDE
TEKRARSIZ KROMOZOM DESTEKLİ
PARALEL GENETİK ALGORİTMA UYGULAMASI**

DOKTORA TEZİ

**Şafak EBESEK
(502922205)**

Mimarlık Anabilim Dalı

Yapı Bilimleri Programı

Tez Danışmanı: Prof. Dr. Hakan YAMAN

MAYIS 2019

İTÜ, Fen Bilimleri Enstitüsü'nün 502922205 numaralı Doktora Öğrencisi Şafak EBESEK, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMİNDE TEKRARSIZ KROMOZOM DESTEKLİ PARALEL GENETİK ALGORİTMA UYGULAMASI” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

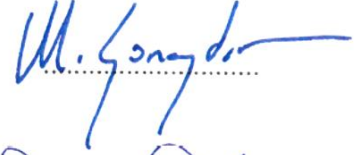
Tez Danışmanı :

Prof. Dr. Hakan YAMAN
İstanbul Teknik Üniversitesi



Jüri Üyeleri :

Prof. Dr. Murat GÜNAYDIN
İstanbul Teknik Üniversitesi



Doç. Dr. Emrah ACAR
İstanbul Teknik Üniversitesi



Prof. Dr. Sema ERGÖNÜL
Mimar Sinan Güzel Sanatlar Üniversitesi



Doç. Dr. Candan ÇINAR ÇITAK
Yıldız Teknik Üniversitesi



Teslim Tarihi : 26 Mart 2019
Savunma Tarihi : 24 Mayıs 2019





Ailem'e



ÖNSÖZ

Değerli katkıları için Danışmanım Sayın Prof. Dr. Hakan Yaman’a, görüş ve değerlendirmeleri ile destek olan Jüri üyeleri Sayın Prof. Dr. Murat GÜNAYDIN’a ve Sayın Doç. Dr. Emrah ACAR’a teşekkür ederim.

Bu çalışma İstanbul Teknik Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimince Desteklenmiştir. [Proje Numarası: 36237]

Katkıda bulunan herkese teşekkürlerimle...

Mart 2019

Şafak EBESEK
(Araştırmacı)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER.....	ix
KISALTMALAR.....	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
ÖZET.....	xvii
SUMMARY.....	xix
1. GİRİŞ.....	1
1.1 Arka Plan	1
1.2 Motivasyon	5
1.3 Amaç.....	5
1.4 Bilime Katkı.....	6
1.5 Yöntem ve Araçlar	6
1.6 Tezin Organizasyonu	7
1.7 Bölüm Özeti	8
2. LİTERATÜRDE KAYNAK KISITLI PROJE ÇİZELGELEME	11
2.1 Sınıflandırma.....	12
2.2 Çözüm Yöntemleri	16
2.2.1 Kesin yöntemler	18
2.2.1.1 CPM ve PERT	18
2.2.1.2 Lineer ve tamsayılı programlama	20
2.2.1.3 Dal ve sınır algoritması	20
2.2.2 Sezgisel algoritmalar	21
2.2.2.1 Çizelge oluşturma algoritmaları	21
2.2.2.2 Öncelik kuralları	23
2.2.2.3 Yerel arama algoritmaları	24
2.2.3 Meta-sezgisel algoritmalar	26
2.2.3.1 Genetik algoritma	28
2.2.3.2 Tavlama benzetimi algoritması	29
2.2.3.3 Tabu arama	31
2.3 Çizelge Gösterim Biçimleri	32
2.4 KKPÇP Çözümüne Yönelik Çalışmalar	35
2.5 Bölüm Özeti	42
3. GENETİK ALGORİTMAYA BAKIŞ	45
3.1 Meta-sezgisel Optimizasyon	45
3.2 No Free Lunch Teorisi.....	48
3.3 Genetik Algoritma	50
3.3.1 Genetik algoritmaya giriş.....	50
3.3.2 Genetik algoritmaya ilişkin temel kavramlar	53
3.3.3 Genetik algoritma operatörleri	54

3.3.4 Genetik algoritmanın işleyişi.....	56
3.4 Paralel Genetik Algoritma.....	56
3.5 Bölüm Özeti.....	59
4. HİBRİT PARALEL GENETİK ALGORİTMA UYGULAMASI.....	61
4.1 Hibrit Paralel Genetik Algoritma Uygulamasına İlişkin Kavramlar	62
4.2 Uygulamada Yararlanılan Güçlendirme ve Destekler	63
4.2.1 Tekrarsız kromozom güçlendirmesi.....	63
4.2.1.1 GA'ya özel tekrarsız kromozom güçlendirme	64
4.2.1.2 Probleme özel tekrarsız kromozom güçlendirme.....	66
4.2.2 Paralel genetik algoritma desteği	67
4.2.3 Hibrit algoritma desteği.....	67
4.3 Uygulamada Kullanılan Algoritma	68
4.3.1 Uygulama ortamı	69
4.3.2 Uygulama algoritmasının ana hatları	72
4.3.3 Uygulama algoritmasının akış şeması üzerinden açıklanması	74
4.3.4 Uygulama algoritmasının sınanması	85
4.4 Uygulama Sonuçları ve Değerlendirilmesi	86
5. SONUÇ VE ÖNERİLER.....	93
5.1 Tezin Bütünlüğü.....	94
5.2 Değerlendirme	95
5.3 Öneriler.....	95
KAYNAKLAR.....	97
EKLER.....	109
ÖZGEÇMİŞ.....	113

KISALTMALAR

ACO	: Karınca Kolonisi Algoritması (<i>Ant-Colony Optimization</i>)
AON	: D�ğ�mde Yer Alan Aktivite (<i>Activity On Node</i>)
CPM	: Kritik Y�r�nge Metodu (<i>The Critical Path Method</i>)
EF	: Aktivite En Erken Bitiř Zamanı (<i>Early Finish</i>)
ES	: Aktivite En Erken Bařlama Zamanı (<i>Early Start</i>)
GA	: Genetik Algoritma
GERT	: Grafik Deęerlendirme ve G�zden Geirme Teknięi (<i>Graphical Evaluation and Review Technique</i>)
IP	: Tamsayılı Programlama (<i>Integer Programming</i>)
KKPP	: Kaynak Kısıtlı Proje izelgeleme Problemi
LP	: Lineer Programlama
MILP	: Karma Tamsayılı Programlama (<i>Mixed Integer Linear Programming</i>)
NFL	: Bedava Yemek Yok Teoremi (<i>No-Free-Lunch Theorem</i>)
NP-Hard:	: Polinomsal zamanda bir �z�m� olduęu ispatlanamayan optimizasyon problemleri (<i>Nondeterministic Polynomial Time</i>)
NRCPGA	: Tekrarsız Kromozom Destekli Paralel Genetik Algoritma (<i>Nonrepetitive Chromosome Improvement for Parallel Genetic Algorithm</i>)
PERT	: Program Deęerlendirme ve G�zden Geirme Teknięi (<i>Program Evaluation and Review Technique</i>)
PGA	: Paralel Genetik Algoritma
PSO	: Paracık S�r� Optimizasyonu (<i>Particle Swarm Optimization</i>)
PSPLIB	: izelgeleme Problemleri Kitaplıęı (<i>Project Scheduling Problem Library</i>)
SA	: Tavlama Benzetimi Algoritması (<i>Simulated Annealing</i>)
TS	: Tabu Arama Algoritması (<i>Tabu Search Algorithm</i>)



ÇİZELGE LİSTESİ

Sayfa

Çizelge 1.1 : KKPÇ’de yaygın kullanılan meta–sezgisel algoritmalar.	5
Çizelge 2.1 : KKPÇP için öncelik kuralları.	24
Çizelge 2.2 : Sık rastlanan meta–sezgisel algoritmaların tarihsel gelişimi.	27
Çizelge 4.1 : HP Z420 iş istasyonu konfigürasyonu	69
Çizelge 4.2 : Çözüm sonucunda elde edilen ortalama sapma değerleri.	87
Çizelge 4.3 : PSPLIB J30 Ortalama sapma değerleri.	89
Çizelge 4.4 : PSPLIB J60 Ortalama sapma değerleri.	90
Çizelge 4.5 : PSPLIB J120 Ortalama sapma değerleri.	91
Çizelge A.1 : KKPÇP için PSPLIB çözümü bildiren önemli çalışmalar.....	111
Çizelge A.2 : KKPÇP için Ulusal Tez Merkezi’nden erişilen yeni çalışmalar.	112



ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Sorgu:RCPSP or PSPLIB– Yıllara göre yayın sayısı	3
Şekil 1.2 : Sorgu:RCPSP or PSPLIB– Yıllara göre alıntı sayısı.....	3
Şekil 1.3 : Sorgu:Metaheuristic– Yıllara göre yayın sayısı	4
Şekil 1.4 : Sorgu:Metaheuristic– Yıllara göre alıntı sayısı.....	4
Şekil 2.1 : KKPCP Geniş Sınıflandırma.	15
Şekil 2.2 : Yerel arama	25
Şekil 2.3 : GA kavramsal akış şeması	28
Şekil 2.4 : SA akış şeması.....	31
Şekil 2.5 : TS kavramsal akış şeması.....	32
Şekil 2.6 : AON çizge	33
Şekil 3.1 : Optimizasyon metotları	46
Şekil 3.2 : GA Akış Şeması	57
Şekil 3.3 : PGA akış şeması.	58
Şekil 4.1 : GA’ya özel tekrarsız kromozom güçlendirme	65
Şekil 4.2 : Çeşitlendirmede hibrit SA desteği.	68
Şekil 4.3 : Uygulama akış şeması.....	76
Şekil 4.4 : PGA– GA bellek pozisyonları.....	77
Şekil 4.5 : Çaprazlama.	81
Şekil 4.6 : Yerel arama: Başka pozisyona yerleştirme	82
Şekil 4.7 : Yerel arama: Bir başka genle karşılıklı olarak yer değiştirme	82
Şekil 4.8 : Yerel arama: Rastgele bir değer atama	82
Şekil 4.9 : Yerel arama: Aşırı değer atama.	83
Şekil 4.10 : AON çizgenin kaynak kısıtları altında çözümü.....	84



KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMİNDE TEKRARSIZ KROMOZOM DESTEKLİ PARALEL GENETİK ALGORİTMA UYGULAMASI

ÖZET

Kaynak kısıtlı proje çizelgeleme bir kombinatorial optimizasyon problemidir ve kaynak optimizasyonuna yönelik olarak Proje ve Yapım Yönetimi alanında kullanılan önemli bir araçtır. Projeyi oluşturan aktivitelerin süreleri, kaynak ihtiyaçları, öncelik–sonralık ilişkileri ve kullanılacak kaynakların sınırları belirlenmiştir. Proje tamamlanma süresinin minimize edilmesi hedeflenir. Tanımı basitçe yapılsa da, kaynak kısıtlı proje çizelgeleme problemi, NP–Hard optimizasyon problemlerinin sınıfına aittir ve aslında pratikte en zorlu klasik problemlerden biridir. Kaynak kısıtlı proje çizelgeleme problemlerinin çözümü için geliştirilen kesin yöntemler, özellikle büyük problemlerin çözümünde –pratikte kullanılamayacak kadar çok– zaman alıcıdır. Meta–sezgisel algoritmalar ise büyük boyutlara ve çok karmaşık kısıtlamalara sahip problemlere makul hesaplama zamanlarında yeteri kadar iyi çözümler üretebilirler.

Kaynak kısıtlı proje çizelgeleme problemlerinin meta–sezgisel algoritmalarla çözümü yoğun çalışılan bir alandır. Literatür taraması ve boşluk analizi sırasında, Genetik Algoritma’ya yönelik güçlendirme ve desteklerin etkili olabileceği görülmüştür. Doktora tezinde uygulanan güçlendirme ve desteklerin ardındaki ana fikir: tekrarsızlığın sağlanması ve çeşitlendirilmenin arttırılmasıdır. Genetik Algoritma’ya yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi: Tekrarlanan kromozomların uzaklaştırılması ve yerlerine –çeşitliliği sağlamak üzere– yenilerinin konması esasına dayanmaktadır. Kaynak kısıtlı proje çizelgeleme problemine yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi ise farklı görünen ama aynı kaynak profilini üreten kromozomların elenmesi ve yerlerine yenilerinin konulmasını sağlamaktadır. Hibrit Simulated Annealing desteği: yerel iyiden kaçmak ve daha iyi çözümler içerebilecek patikalara sıçramak için kullanılmaktadır. Paralel Genetik Algoritma desteği, Genetik Algoritma’ya göre daha hızlı sonuca ulaşmak, daha küçük popülasyonla çalışmak, daha yüksek çeşitlilik sağlamak ve daha yüksek çözüm kalitesi elde etmek için kullanılmaktadır.

Doktora tezinde geliştirilen uygulama MATLAB ortamında hazırlanmıştır ve kaynak kısıtlı proje çizelgeleme problemlerinin başarımlarının ölçülmesinde standart olarak kullanılan PSPLIB problemleri üzerinde sınanmıştır. Problemlerinin çözümü sonucunda elde edilen ortalama sapma değerleri: J30 ve J60 problemlerinde aynı çözüm değerlerini veren diğer çalışmalarla birlikte en üst bandta, J120 problemlerinde altıncı sırada yer almaktadır. Çözümler sırasında J120_35_5 numaralı problemde, –daha önceki çalışmalarda elde edilmemiş– en iyi değere erişilmiştir. Problemin çözümü onaylanmış ve PSPLIB problemlerini barındıran Münih Teknik Üniversitesi resmi web sitesinde yer almıştır. Doktora tezinde uygulanan

güçlendirmeler ve destekler, kaynak kısıtlı proje çizelgeleme problemlerine yönelik olarak yapılacak yeni çalışmalarda yol gösterici olarak kullanılabilir.



A PARALLEL GENETIC ALGORITHM APPLICATION WITH NONREPETITIVE CHROMOSOME IMPROVEMENT FOR RESOURCE CONSTRAINED PROJECT SCHEDULING PROBLEM

SUMMARY

Resource constrained project scheduling is a combinatorial optimization problem and is an important tool used in Project and Construction Management for resource optimization. The duration of the activities, resource requests, precedence relations and the limits of the resources to be used were strictly determined. The goal is minimizing the makespan of project. Despite the simplicity of its definition, the resource constrained project scheduling problems belongs to the class of NP–Hard optimization problems and is actually one of the most intractable classical problems in practice. The exact methods developed to solve resource constrained project scheduling problems are time consuming and useless, especially when solving large problems. Metaheuristics are mostly useful to reach good quality solutions in reasonable computational times and are suitable for practical problems, which often have large dimensions and very complex constraints.

The solution of resource constrained project scheduling problems with Meta–heuristics is one of the intensive working areas. It was determined during the literature review and gap analysis what improvements and supports for Genetic Algorithm could be effective. The main idea behind the improvement and support applied in this PhD Thesis is to achieve nonrepetition and increase diversification. Nonrepetitive chromosome improvement for the Genetic Algorithm: It is based on the removal of repetitive chromosomes and the replacement of new ones in order to provide diversity. The nonrepetitive chromosome improvement for the resource constrained project-scheduling problem allows the elimination of chromosomes producing the same source profile and replacing them with new ones. Hybrid Simulated Annealing support: used to escape from the local optimum and to leap forward into pathways that may include better solutions. Parallel Genetic Algorithm support is used to achieve faster results than Genetic Algorithm, to work with smaller populations, to provide higher diversity and to achieve higher solution quality.

The application in this PhD Thesis has been developed on MATLAB environment and has been tested on PSPLIB problems, which are used as standard for measuring the performance values of resource constrained project scheduling problems. The average deviation values obtained as a result of the solutions of the problems: J30 and J60 problems are at the top band with other top studies having equal results, J120 problems are ranked as sixth and have minor gaps comparing with other top studies. The best new solution has been reached on problem J120_35_5. The solution has been approved and appended on the official website of the Technical University of Munich, which hosted the PSPLIB problems. Improvements and supports applied in this PhD Thesis can be used as a guide for new studies on resource constrained project-scheduling problems.

The aim of the doctoral thesis is to develop a genetic algorithm for the solution of the resource constrained project scheduling problems, which shows high performance value compared to the current studies examined in the literature.

This thesis consists of the following five chapters. In the first chapter, background, motivation, purpose, contribution to science, methods-tools and organization of thesis are explained.

In the second chapter, a recent literature survey on resource constrained project scheduling problems is presented. The literature research is presented under the headings of classification, solution methods, schedule presentation schemes and work towards solution.

In the third chapter, GA was examined in detail. This chapter is presented under the headings of metaheuristic optimization, No Free Lunch Theory, GA and PGA.

In the fourth chapter, hybrid parallel GA application which was prepared for resource constrained project scheduling problems was introduced. The conceptual principles behind the application, the algorithms used and the results obtained are discussed. Two improvement approaches, one for GA and the other for resource constrained project scheduling problems were given, followed by PGA and hybrid SA supports. At the end of the fourth chapter, the results obtained from the application are compared with the studies in the literature.

Finally, Chapter five concludes the research of this thesis by summarizing its significant technical contributions in the domain of resource constrained project scheduling problems research produced during this study and the major conclusions that can be drawn from the experiments conducted. Also, some conceivable directions for further research are also suggested.

Metaheuristic algorithms are techniques that independent of the problem structure and aim to obtain good quality solutions at reasonable computational times. Metaheuristic algorithms are generally well suited for practical problems with large dimensions and very complex constraints. The use of metaheuristic algorithms is not efficient when solution already known problems. If the search space is too big, the complexity of the problem is high, the mathematical modeling of the problem and the solution is very difficult, then metaheuristic algorithms can be used. Pure metaheuristic algorithms can produce good results or be inadequate for solving hard combinatorial optimization problems. In order to achieve better results, the problem specific neighborhood structures, search space constraints and guiding rules should be studied carefully and any improvements that are accessible should be used. In particular, the solution of linear equations, constraints or simply solvable nonlinear equations should be done within the model, and solutions of such problems should not be left to the metaheuristic algorithms. If it is possible, the solution design should be based on searching in feasible areas. Although it is possible to use penalty functions for the unacceptable areas, it should be considered as the last option. In this case, the value of the penalty function should be designed in such a way that it can point towards the feasible region. Whenever possible, the power of exact solution algorithms must be used, and metaheuristic algorithms should be used to efficiently navigate in search space and produce new solutions.

GA works to improve the solution quality of population, as opposed to the local search strategies that work to improve the single solution. After creating an initial population, the operators carry out successive generation and evaluate the results.

Once the new generation has been produced, the most appropriate solutions are kept to create the next generation, while the bad solutions are eliminated. The fitness value is used to measure the quality of the solution and is usually calculated from the objective function of the optimization problem to be solved.

The PGA is used to shorten the long-lasting solution time in GA's wide search spaces and provides gains in terms of both performance and solution quality. Parallel running hill climbing algorithms cannot share information among themselves, whereas the lower GAs belonging to the PGA can share solutions depending on the chosen migration mechanism. Thus, both the permeability and diversity processes are enriched in a balanced manner.

Instead of over-optimization, simple but effective, guiding rules should be sought. In general, no superiority of any metaheuristic algorithm can be determined. Some algorithms seem to be appropriate for some specific problem structures, but there is no significant difference in long-term averages that will show dominance. Recent studies on the No Free Lunch theorem suggest that problem-specific metaheuristic algorithms can be more successful than general metaheuristic algorithms. The hybrid algorithm approach that has emerged in recent years can benefit in achieving better results. If possible, different algorithms should be combined. Hyper metaheuristics can be an important area of study for selecting algorithms to be used in problem solving. Algorithms that are not ambitious about achieving the best results but approaching reasonable results quickly should not be ignored. In many cases, the cost of a unit recovery may be too extreme to be tolerated in practice.

Resource constrained project scheduling problems with repeated activities or repeated subprojects may be an interesting area of study. New presentation and calculation methods can be developed for such projects. Although the number of activities increases a lot, activity blocks remain the same. Algorithms to be developed over the same sequence of activity blocks can significantly reduce the search space.

It is important to pay attention to hold on the a false assumptions made, when working on the resource constrained project scheduling problems. Activity times are predetermined and are based on significant assumptions. Resource constraints may change throughout the project. A small improvement in resource constraints can significantly reduce project completion time. Similarly, the divisibility of activities should be examined. By dividing the activity into several activities connected in series, it can provide better planning and modeling facilities. If the activity does not have to continue uninterruptedly or there is no cost for intermittent execution, resource balancing process options will be increased and a shorter total project completion time can be obtained. The resource requirement can be concentrated before activity or at the beginning of the activity and may appear as a single load at any time. These features should be examined well and should not be considered as a uniformly distributed load with an extremely rough approach. Sensitivity analysis for the source and time relationship may be guiding.

There should be a reasonable point between extreme detailism and extreme roughness in the implementation of the RCPSP, and should concentrate on places that have a leverage effect on the behavior of the system. If possible, the resource limit should be increased and the risks on the expected resource supply profile should be reduced by using time and resource buffers. Simulation approaches, which are

entered into the system according to the probability distributions of both activity periods and resources, can help with more accurate estimations of possible reality.

It may be useful to use the improvements and supports applied in the RCPSP for the following purposes: To prevent the presence of chromosomes in the population that appear different but represent the same solution, to provide greater diversity in the population, to reach a solution with a smaller population, to reduce the time to reach a solution and improving the quality of the solution.

The performance values of the application developed in thesis are evaluated through the black-box approach. The contribution of each of the improvements and supports on the solution performance can be transformed into the white-box approach and explained together with the performance values to be obtained through different test problems.



1. GİRİŞ

1.1 Arka Plan

Kaynak kısıtlı proje çizelgeleme, pratikte birçok uygulamayı modellemek için kullanılabilen genel bir çizelgeleme problemidir. Amaç, kaynak sınırları olan bir projenin aktivitelerini, seçilen amaç doğrultusunda optimize etmektir. Sınırlı kaynaklara örnek olarak: makinalar, insanlar, malzemeler, aletler, mekânlar, para ve enerji gösterilebilir. Proje tamamlanma süresini, teslim tarihlerinden sapmayı, kaynak maliyetlerini veya kaynak dalgalanmalarını en aza indirmek, örnek olarak verilebilecek optimizasyon hedefleridir. Pratikte karşılaşılan KKPÇP (Kaynak Kısıtlı Proje Çizelgeleme Problemi) uygulama alanlarından bazıları aşağıda verilmektedir (Brucker & Knust, 2012).

- Yapım yönetiminde çizelgeleme (*Scheduling in construction management*)
- Proses tipi (sürekli) üretimin söz konusu olduğu endüstrilerde toplu çizelgeleme (*Batch scheduling in process industries*),
- Mikroişlemci yükü çizelgelenmesi (*microprocessor scheduling*),
- Eğitim sektöründe ders programı hazırlama,
- Levha veya rulo haldeki standart boyutlu malzemelerin küçük parçalara ayrılması (*A cutting-stock problem*),
- Uçak bakımı,
- Havaalanı yer hizmetlerinin çizelgelenmesi,
- Uçuş kapısı çizelgelenmesi,
- Spor liglerinin fikstürlerinin düzenlenmesi.

Uygulama kaynaklı bir bakış açısıyla, proje yönetiminin kritik bir parçası olarak çizelgelemenin iyileştirilmesi, projenin başarıyla tamamlanmasını ve maliyetlerin önemli ölçüde azalmasını sağlayabilir. Teorik perspektiften, proje planlaması,

yöneylem alanında birçok araştırmacının dikkatini çeken ilginç optimizasyon konularından biri olarak kabul edilmektedir (Habibi ve diğ., 2018).

KKPÇP birçok araştırmacı tarafından genel bir çerçeve içerisinde tanımlanmış, çözüm modelleri ve algoritmalar detaylandırılarak incelenmiştir (Brucker & Knust, 2012; Kolisch & Hartmann, 2006; Habibi ve diğ., 2018; Abdolshah, 2014)

KKPÇP, NP–Hard (Polinomsal zamanda bir çözümü olduğu ispatlanamayan optimizasyon problemleri, *Nondeterministic Polynomial Time*) sınıfında, en zorlu problemlerden biridir (Blazewicz ve diğ., 1983). Bu alanda yapılan yeni çalışmalarla, referans problemlerinin çözümünde, gittikçe artan daha iyi başarımlar elde edilmektedir.

Çizelgeleme problemlerinin çözümü için birçok çalışma yapılmaktadır. Bu çalışmalarda: Kesin, sezgisel (heuristik) ve meta–sezgisel/sezgi(sel) ötesi (metaheuristik) çözüm yolları önerilmiştir. Kesin yöntemlere dayanan ve en iyi çözümleri bulabilen çeşitli modeller geliştirilmiştir. Kesin yöntemler, özellikle büyük problemlerin çözümünde zaman alıcıdır (Demeulemeester & Herroelen, 1992; Jalilvand ve diğ., 2005). Uzmanların bilgi ve deneyimlerine dayanan ve probleme özgü olan sezgisel yaklaşımlar ise, arama uzayını daraltarak, kısa zamanda olurlu çözümler bulunmasına yardımcı olmaktadır (Gavrilas, 2010). Öte yandan, meta–sezgisel çözüm yolları, problemin herhangi bir özelliğinden yararlanmayan, problem bağımsız tekniklerdir (Beheshti ve Shamsuddin, 2013). Hem sezgisel hem de meta–sezgisel yaklaşımlar, karmaşık sorunları başarıyla çözen güçlü ve esnek arama mekanizmalarıdır.

Bu algoritmalar, makul hesaplama zamanlarında iyi kalitede çözümler elde etmeyi amaçlar ve genellikle büyük boyutlara ve çok karmaşık kısıtlamalara sahip pratik problemler için oldukça uygundur (Das ve Acharyya, 2011; Kolisch ve Hartmann, 1999).

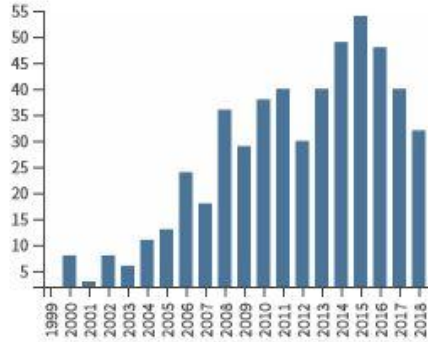
KKPÇP için geliştirilen algoritmalarının çözüm performanslarının karşılaştırılmasında çok yaygın olarak kullanılan sınama problemleri, 2040 problem içeren tek modlu PSPLIB (Çizelgeleme Problemleri Kitaplığı, *Project Scheduling Problem Library*) setinden seçilmektedir. (Kolisch ve Sprecher, 1996). Sınama işlemi sırasında, her problem için elde edilen tamamlanma süresi, kaynak kısıtlaması olmadan elde edilen en kısa tamamlanma süresine bölünmekte ve problemin

yüzdesel sapma miktarı bulunmaktadır. Sonraki adımda tüm problemlerin sapma miktarlarının ortalaması alınmaktadır. PSPLIB problemlerinin çözümü sonucunda elde edilen ortalama sapma değerleri, geliştirilen algoritmanın başarımlarını değerlendiren bir ölçüt olarak anılmaktadır.

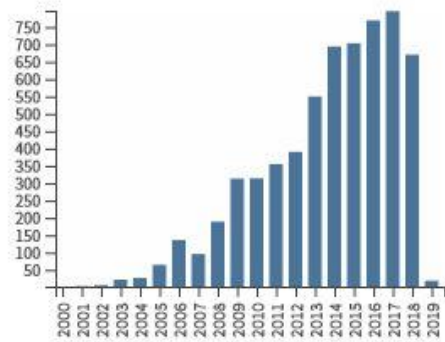
KKPÇP veya PSPLIB anahtar kelimeleri kullanılarak Web of Science üzerinde yapılan bibliyometrik analiz sorgu cümlesi ve sorgu sonucu Şekil 1.1’de ve Şekil 1.2’de görülmektedir:

Sorgu: ALL FIELDS: (rcpsp) OR ALL FIELDS: (psplib)

Timespan=2000–2018. Indexes=SCI–EXPANDED, SSCI, A&HCI, CPCI–S, CPCI–SSH, BKCI–S, BKCI–SSH, ESCI.



Şekil 1.1 : Sorgu:RCPSP or PSPLIB–
Yıllara göre yayın sayısı.



Şekil 1.2 : Sorgu:RCPSP or PSPLIB–
Yıllara göre alıntı sayısı.

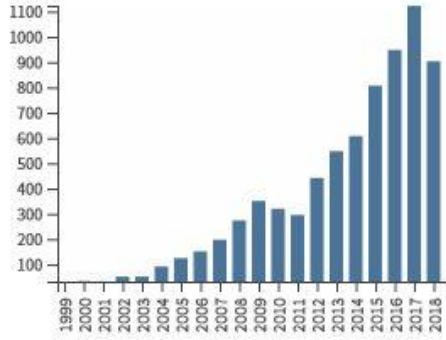
Şekil 1.1 ve Şekil 1.2’de görselleştirilen sorgulama sonuçları değerlendirildiğinde:

2000’li yılların başında başlayıp, 2010’lu yıllarda yoğunlaşan çalışmalar, 2016 yılına kadar süren belirgin bir tırmanış göstermektedir. 2010–2018 Aralığında gözlenen eğilim ortalaması, yüksek plato yapısındadır. Sorgulamada yer alan çalışmalardan yapılan alıntılar ise, 2000’li yıllardan başlayarak devam eden kararlı bir tırmanış görünümündedir.

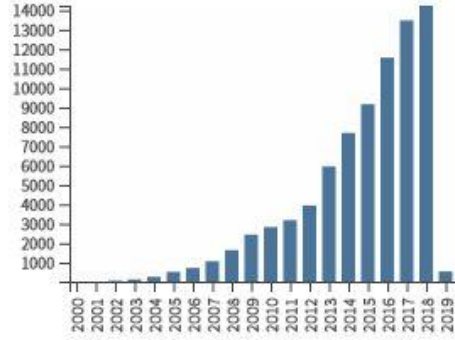
“Metaheuristic” anahtar kelimesi kullanılarak Web of Science üzerinde yapılan bibliyometrik analiz sorgu cümlesi ve sorgu sonucu Şekil 1.3’te ve Şekil 1.4’de görülmektedir:

Sorgu: ALL FIELDS: (metaheuristic)

Timespan=2000–2018. Indexes=SCI–EXPANDED, SSCI, A&HCI, CPCI–S, CPCI–SSH, BKCI–S, BKCI–SSH, ESCI.



Şekil 1.3 : Sorgu:Metaheuristic– Yıllara göre yayın sayısı.



Şekil 1.4 : Sorgu:Metaheuristic– Yıllara göre alıntı sayısı.

Şekil 1.3 ve Şekil 1.4 de görselleştirilen sorgulama sonuçları değerlendirildiğinde:

Hem sorgulamada yer alan çalışmaların hem de bu çalışmalardan yapılan alıntılarının miktarı, 2000’li yıllardan başlayarak devam eden kararlı bir üstel tırmanış görünümündedir.

Çizelge 1.1’de görülmekte olan GA (Genetik algoritma), SA (Tavlama Benzetimi Algoritması, *Simulated Annealing*), TS (Tabu Arama Algoritması, *Tabu Search Algorithm*), ACO (Karınca Kolonisi Algoritması, *Ant-Colony Optimization*), PSO (Parçacık Sürü Optimizasyonu, *Particle Swarm Optimization*), literatürde en çok karşılaşılan meta–sezgisel yöntemlerdir ve çizelgeleme probleminin çözümünde yaygın olarak kullanılmaktadır. Doğal ve biyolojik modellerden esinlenen meta–sezgisel algoritmalar arasında yer alan GA, ACO, PSO, çoklu başlangıç çözümü geliştirerek ilerlemektedir. Mantıksal modellerden esinlenen meta–sezgisel algoritmalar arasında yer alan TS, önceden denenmiş çözümlere geri dönüşleri engelleyen yasaklama listeleri kullanmakta ve tek başlangıç çözümü geliştirerek ilerlemektedir. Fiziksel modellerden esinlenen meta–sezgisel algoritmalar arasında yer alan SA, başlangıç çözümü geliştirerek ilerlemektedir. Sağladıkları esneklik ve performans üstünlükleri sayesinde, farklı esinlerle geliştirilen algoritmaların bir arada tasarlandığı hibrit yaklaşımlar, son yıllarda gittikçe artan kullanım alanı bulmaktadır.

Çizelge 1.1 : KKPÇ’de yaygın kullanılan meta–sezgisel algoritmalar.

Algoritma	Geliştiren
Genetic Algorithm (GA)	Goldberg, 1989
Simulated Annealing (SA)	Kirkpatrick ve diğ., 1983
Tabu Search (TS)	Glover, 1990
Ant Colony Optimization (ACO)	Dorigo ve Blum, 2005
Particle Swarm Optimization (PSO)	Eberhart ve diğ., 2001

1.2 Motivasyon

Literatür araştırması ve çözüm algoritmalarının anlamlandırılması sırasında elde edilen bulgular ve değerlendirmeler sonucunda, aşağıda değinilen alanlara yönelik iyileştirme olanaklarının mümkün olabileceği belirlenmiştir.

- KKPÇ alanında: tekrarsız kromozom gösterimi ile arama uzayının verimli kullanılması,
- GA alanında: Erken yakınsama ve baskılama sonucu oluşan çeşitlilik kaybının engellenmesi,
- GA alanında: PGA (Paralel Genetik Algoritma) desteği ile çeşitlik artışı ve arama uzayının verimli kullanılması,
- GA alanında: Hibrit SA desteği ile yerel iyiden kaçış.

KKPÇP’ne yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi ana fikri Debels ve diğ. (2006) tarafından ortaya atılmıştır. Ancak bu fikrin geliştirilmesine ve uygulanmasına yönelik bir başka bir çalışmaya rastlanmamıştır.

Tez çalışmasının şekillendirilmesinde şimdiye kadar bir arada kullanılmamış olan iyileştirme olanaklarının yansıtıldığı bir algoritmanın PSPLIB test problemleri üzerinde uygulanması sonucu elde edilebilecek başarımların, literatürde yer alan çalışmalar arasında üst sıralara yer alabilmesi olanağı, yüksek bir motivasyon unsuru olarak değerlendirilmiştir

1.3 Amaç

Doktora tezinin amacı: KKPÇP’nin çözümüne yönelik olarak güçlendirilmiş bir genetik algoritma ile literatürde incelemiş olan mevcut çalışmalarla karşılaştırılabilen yüksek başarımların elde edilmesidir.

1.4 Bilime Katkı

Literatür taraması sonucunda, hem geliştirilecek güçlendirici algoritmalara, hem de bunların PGA içinde kullanılarak KKPÇP sına ma problemlerine uygulanmasına yönelik hiçbir çalışmaya rastlanmamıştır.

Güçlendirme ve desteklerin ardındaki ana fikir, tekrarsızlığın sağlanması ve çeşitlendirilmenin artırılmasıdır.

Geliştirilen algoritma sonucunda elde edilen değerlerin olumlu çıkması durumunda, konu alanında şimdiye kadar yapılmış diğer çalışmaları zenginleştirmeye katkıda bulunabileceği değerlendirilmiştir. Sınama sonuçlarının yeteri kadar iyi olmaması durumunda ise, gelecekte bu konu alanında yapılacak çalışmalarda verimsiz yolların farkına varılması bakımından araştırmacılara yol göstermek ve düzeltici geliştirmeler için zemin hazırlamak üzere bu çalışmadan yararlanılabileceği öngörülmüştür.

1.5 Yöntem ve Araçlar

Doktora tezi aşağıdaki yöntem izlenerek gerçekleştirilmiştir:

- Probleme ve ilişkili görülen yardımcı disiplinlere yönelik literatür araştırması yapılması,
- Problemin çözümü için kullanılan yaklaşımların, algoritma anlamlandırma ve yazılım geliştirme bakışıyla incelenmesi,
- Ön yazılım geliştirilmesi ve örnek problemlerin çözülmesi,
- Boşluk analizi ve güçlendirme çalışmaları yapılması,
- Güçlendirilmiş algoritmayı yansıtan yazılımın hazırlanması,
- PSPLIB problem setindeki sınama problemlerinin çözülmesi ve sonuçların değerlendirilmesi.

Referans yönetimi yazılımı olarak “Mendeley” yazılımı kullanılmıştır (Mendeley, 2019). Mendeley, araştırma belgelerini paylaşmak, yönetmek, keşfetmek ve çevrim içi iş birliği için oluşturulmuş, İnternet üzerinde çalışan bir masaüstü programdır.

Kaynak tarama ve bibliometrik analiz için “Publish or Perish” yazılımı kullanılmıştır. Publish or Perish akademik alıntıları İnternet üzerinden araştıran, alan ve analiz eden bir yazılımdır (Publish or Perish, 2019).

Doktora tezinde hem araç hem de amaç olarak kullanılan sınama problemleri, KKPÇP için geliştirilen algoritmalarının çözüm performanslarının karşılaştırılmasında çok yaygın olarak kullanılan, tek modlu PSPLIB setinden (J30, J60, J90, J120) seçilmiştir (Kolischve Sprecher, 1996).

Sınama için kullanılan yazılım “MATLAB” uygulama ortamında geliştirilmiştir. MATLAB, araştırma ve mühendislik alanlarında karşılaşılan problemlere: pratik, tutarlı, güvenilir, kolay kontrol edilebilir, hızlı sonuçlar sunmakta ve tüm dünyada endüstri, devlet kurumları ve akademik kurumlarda yaygın olarak kullanılmaktadır (Mathworks, 2019).

Yazılım HP Z420 iş istasyonu üzerinde çalıştırılmıştır. İş istasyonları, diğer masaüstü bilgisayarlara oranla yüksek yükler altında uzun süreler, hatasız ve kesintisiz çalışabilmektedir.

Sınama için kullanılan yazılım ve donanım bilgileri, Bölüm 4.3.1’de ayrıntılı olarak görülmektedir.

1.6 Tezin Organizasyonu

Doktora tezi aşağıda verilen beş bölümden oluşmaktadır:

Bölüm 1: Giriş

Bölüm 2: Literatürde Kaynak Kısıtlı Proje Çizelgeleme

Bölüm 3: Genetik Algoritmaya Bakış

Bölüm 4: Hibrit Paralel Genetik Algoritma Uygulaması

Bölüm 5: Sonuç

Birinci bölümde doktora tezinin arka planı, motivasyon kaynakları, amacı, bilime katkısı, tezde kullanılmış olan yöntem ve araçlar ve doktora tezi organizasyonu ele alınmaktadır.

İkinci bölümde, kaynak kısıtlı proje çizelgeleme problemi hakkında güncel literatür araştırması sunulmaktadır. Literatür araştırması sınıflandırma, çözüm yöntemleri,

çizelge gösterim biçimleri ve KKPÇP çözümüne yönelik çalışmalar başlıkları altında ele alınmaktadır. Bu bölümde, KKPÇP ayrıntılı olarak incelenmekte ve dördüncü bölümde yer alan, hibrit paralel GA uygulaması konusuna hazırlık yapılmaktadır.

Üçüncü bölümde genetik algoritma konusu incelenmektedir. GA incelemesi: meta-sezgisel optimizasyon, No Free Lunch teorisi, GA ve PGA başlıkları altında ele alınmaktadır. Bu bölümde, GA ayrıntılı olarak incelenmekte, ve dördüncü bölümde yer alan, hibrit paralel GA uygulaması konusuna hazırlık yapılmaktadır.

Dördüncü bölümde KKPÇP'ne yönelik olarak hazırlanan hibrit paralel GA uygulaması tanıtılmaktadır. Uygulamanın ardındaki kavramsal esaslar, yararlanılan güçlendirme ve destekler, kullanılan algoritmalar, elde edilen sonuçlar ve bu sonuçların değerlendirilmesi ele alınmaktadır. Biri genel olarak GA'ya, diğeri özel olarak KKPÇP'ne yönelik iki güçlendirme yaklaşımı verilmekte, ardından PGA ve hibrit SA destekleri açıklanmaktadır. Önceki bölümlerde yapılan hazırlıkla ulaşılan ve uygulamanın ele alındığı dördüncü bölüm doktora tezinin esasını oluşturmaktadır. Bölüm sonunda uygulamadan elde edilen sonuçlar, literatürde yer alan çalışmalarla karşılaştırılarak verilmektedir.

Beşinci bölümde, Sonuç başlığı altında, doktora tezi değerlendirilmekte, görüş ve öneriler verilmektedir.

1.7 Bölüm Özeti

Kaynak kısıtlı proje çizelgeleme, pratikte birçok uygulamayı modellemek için kullanılabilen genel bir çizelgeleme problemidir. Kaynak kısıtlı proje çizelgeleme bir kombinatorial optimizasyon problemidir ve kaynak optimizasyonu için Proje ve Yapım Yönetimi alanında kullanılan önemli bir araçtır. Projeyi oluşturan aktivitelerin süreleri, kaynak ihtiyaçları, öncelik–sonralık ilişkileri ve kullanılacak kaynakların sınırları belirlenmektedir. Proje tamamlanma süresinin minimize edilmesi ulaşılmak istenen hedeftir. Tanımı basitçe yapılsa da, kaynak kısıtlı proje çizelgeleme problemi, NP–Hard optimizasyon problemleri sınıfına girer ve aslında pratikte en zorlu klasik problemlerden biridir. KKPÇP'nin çözümü için geliştirilen kesin yöntemler, özellikle büyük problemlerin çözümünde –pratikte kullanılamayacak kadar çok– zaman alıcıdır. Meta-sezgisel çözüm yolları ise, büyük boyutlara ve çok karmaşık

kısıtlamalara sahip problemlere makul hesaplama zamanlarında yeteri kadar iyi çözümler üretebilmektedir.

KKPÇP'nin meta-sezgisel çözüm yolları yardımıyla çözüm bulunması, içinde bulunduğumuz dönemde üzerinde olarak yoğun çalışılan bir alandır. Literatür taraması ve boşluk analizi sırasında, Genetik Algoritma'ya yönelik güçlendirme ve desteklerin etkili olabileceği görülmüştür. Doktora tezinde uygulanan güçlendirme ve desteklerin ardındaki ana fikir, tekrarsızlığın sağlanması ve çeşitlendirilmenin artırılmasıdır.

Doktora tezinin amacı: KKPÇP'nin çözümüne yönelik olarak güçlendirilmiş bir genetik algoritma ile literatürde incelemiş olan mevcut çalışmalarla karşılaştırılabilen yüksek başarımlı değerlerine ulaşmaktır. Geliştirilen algoritmanın referans problemler üzerinde sınanmasına yönelik olarak bir yazılım hazırlanacak ve bu yazılım, PSPLIB problemlerinin çözülmesinde kullanılacaktır. Sınama sonuçların iyi olması durumunun, sonraki çalışmaları zenginleştirmeye katkıda bulunabileceği değerlendirilmiştir.

2. LİTERATÜRDE KAYNAK KISITLI PROJE ÇİZELGELEME

Günümüzde proje kaynakları yönetimi giderek daha önemli bir hal almaktadır. Bir aktivitenin çıktılarının elde edilebilmesi için kullanılan girdiler kaynaklar olarak adlandırılırlar. Kaynaklar çoğunlukla: işgücü, araçlar, gereçler, dışarıdan alınan hizmetler ve finansal giderler olarak gruplanırlar. Öncelik–sonralık şartlarını sağlayan ve başlayabilir duruma gelen aktivitelerin girdilerini oluşturan kaynakların temin edilememiş olası durumu kaynak sınırlılığını gösterir. Bu durumda paralel yürütülebilen aktivitelerden en az biri kaynaklar yeterli oluncaya kadar ertelenmelidir. Kaynak sınırları dikkate alınmadan yapılan planlama ve programlama faaliyetleri, yeteri kadar gerçekçi olmama riski taşımaktadır. Sistem davranışı, özellikle çoklu görevlerin paralel gerçekleştirildiği otomasyon ve robotlaşma sürecinde, basit kestirimlerle belirlenemeyecek kadar karışık olabilir. Kaynak girişlerinin dalgalanan bir yapı göstermesi durumunda, dinamik süre belirleme gereksinimi kaçınılmaz hale gelir. Eğer aktivite süreleri statik olarak verilmişse, tasarlanan sistem davranışıyla gerçekleşen sistem davranışı arasındaki boşluk giderek artar. Kaynak sınırlamalarının göz önüne alınması durumunda zor çözümlü bir optimizasyon problemi ortaya çıkar (Garey ve Johnson, 1979).

Standart KKPÇP, bir kombinatorial optimizasyon problemidir ve karar değişkenleri, kısıtlamalar ve amaç fonksiyonlar ile ifade edilir (Artigues ve diğ., 2010). Projeyi oluşturan aktivitelerin süreleri, kaynak ihtiyaçları, öncelik–sonralık ilişkileri ve kaynakların sınırları belirlenmiştir. Proje tamamlanma süresinin minimize edilmesi hedeflenir. Kaynak profili dalgalanmasının minimize edilmesi gibi ikincil hedefler de belirlenebilir.

Tanımı basitçe yapılsa da, KKPÇP, NP–Hard optimizasyon problemleri sınıfına girer ve aslında pratikte en zorlu klasik problemlerden biridir (Blazewicz ve diğ., 1983).

Genel olarak KKPÇP konusuna ilişkin varsayımlar aşağıda verilmektedir (Krichen ve Chaouachi, 2014) :

1. Projeyi oluşturan aktivitelerin süreleri bilinmektedir.

2. Aktiviteler arasındaki öncelik–sonralık ilişkileri bilinmektedir.
3. Bir aktivite öncülleri bittikten sonra başlayabilir.
4. Başlayan aktivite kesintisiz olarak tamamlanır.
5. Tüm aktiviteler tamamlanır.
6. Aktivitelerin kaynak kullanım değerleri bilinmektedir.
7. Kaynak sınırları bilinmektedir.
8. Bir kaynak, aktivite boyunca atanmış olarak kalır.
9. Kaynaklar yenilenebilir özelliktedir.
10. Bir aktivitenin birim zamanda kullandığı kaynak miktarı sabittir.
11. Amaç fonksiyon, proje tamamlanma süresini en aza indirmektir.

İlk yedi varsayım genel KKPCP’ni, son dört maddenin eklenmesiyle elde edilen varsayımlar listesi, –literatürde sık rastlanan– tek modlu KKPCP’ni tarif eder. Öncelik–sonralık ve kaynak sınırı kısıtlamalarını sağlayan aday çözümler olurlu olarak adlandırılırlar.

2.1 Sınıflandırma

KKPCP konusunda sıklıkla kullanılan sınıflamalar aşağıda verilmektedir. Dar sınıflandırma olarak adlandırılan ilk sınıflandırma Yang ve diğ. (2001) tarafından verilmektedir ve beş başlıktan oluşmaktadır. Geniş sınıflandırma olarak adlandırılan ikinci sınıflandırma Habibi ve diğ. (2018) tarafından verilmektedir ve ağaç yapısı biçiminde hazırlanmıştır.

Dar Sınıflandırma:

- Tek Modlu,
- Çok Modlu,
- Düzenli olmayan amaç fonksiyonlara sahip olan,
- Stokastik ve
- Çoklu kaynak kısıtlı olarak bilinmektedir.

Eğer bir aktivite için farklı süre ve kaynak kullanım seçenekleri varsa, bu aktivite çok modlu olarak adlandırılır. Her mod bir seçenek olmaktadır. Çizelgeleme sırasında bu seçeneklerden sadece biri kullanılabilir. Tek modlu aktiviteler, verilen süre ve kaynak kullanım değeri ile yürütülürler.

Bölüm 1.5’de belirtildiği gibi, doktora tezi sınama problemleri, KKPÇP için geliştirilen algoritmalarının çözüm performanslarının karşılaştırılmasında çok yaygın olarak kullanılan, tek modlu PSPLIB setinden seçilmiştir.

Aktivite sürelerindeki azalmanın, toplam proje tamamlanma süresini artırması beklenmez. Finansal optimizasyon istenen KKPÇP’nde, aktivite süresinin azalması, toplam proje süresinin artmasına sebep olabilir. Aktivite süresinin azalması, finansal taleplerin –kaynak sınırların aşacak şekilde– yoğunlaşmasına yol açabilir. Karşıt bakışla, aktivite sürelerinin artırılması, kaynak sınırlarının altında kalınmasına, diğer aktivitelerin paralel olarak çalışabilmesine ve böylece toplam proje tamamlanma süresinin kısılmasına yol açabilir. Açıklanan davranışı ortaya çıkartan amaç fonksiyonlar “*düzenli olmayan amaç fonksiyonlar*” olarak adlandırılır.

Aktivite süreleri olasılık dağılımlarından gelen problemler, “*Stokastik KKPÇP*” olarak adlandırılır. Bu durumda, proje süresi yerine, beklenen proje süresi minimize edilmektedir. Aktivite sürelerinin varsayımlara göre belirlenmesi, peş peşe gelen aktivitelerde giderek biriken bir varsayım taşmasına yol açabilir. Aktivite sürelerinin olasılık dağılımlarından gelmesi, daha gerçekçi proje tamamlanma sürelerini olanaklı kılarsa da, çok daha ağır bir hesaplama yükü maliyeti oluşmasına sebep olur.

Bir aktivitenin kullandığı kaynaklar, kendi aralarında öncelik–sonralık ilişkisine sahip olabilirler. Bu tip aktivitelerden oluşan problemler “*çoklu KKPÇP*” olarak adlandırılır.

Çoğunlukla, başlatılan aktivitenin kesintisiz olarak (*nonpreemptive*) tamamlanması öngörülür. Bazı durumlarda başlatılan aktivite bir veya daha çok kez durdurulabilir. Eğer aktivitenin durma–başlama zamanları önceden biliniyorsa, bu aktiviteyi seri olarak bağlı birkaç aktiviteye bölerek planlamak, hesap ve modelleme kolaylıkları sağlayabilir. Eğer aktivitenin kesintisiz olarak devam etme zorunluluğu yoksa veya kesintili yürütmenin bir maliyeti bulunmuyorsa, kaynak dengeleme işlemi seçenekleri artar, daha kısa toplam proje tamamlanma süresi elde edilebilir.

KKPÇP’nde kullanılan kaynaklar: yenilenebilir, yenilenemez ve çift yönden kısıtlı olarak üç başlık altında ele alınabilir. Yenilenebilir kaynaklar tükenmezler. Atanmış olduğu aktivite tamamlandığında serbest kalırlar. Örnek olarak işgücü gösterilebilir. Biyolojik ve mekanik işgücü varlıklar, aktivitenin tamamlanmasında sonra, yeni bir aktiviteye atanmak üzere hazır durumdadırlar. Bu tanımda, kullanım sırasında ortaya çıkabilecek küçük yıpranmalar göz ardı edilmektedir. Yenilenebilir kaynakların birim zaman içerisinde kullanım miktarları sınırlı olabilir. Yenilenemez kaynakların aktivite sonundaki miktarı, aktivite başındaki miktarından küçüktür. Tüketilen kaynağın stok seviyesi düşer. Kaynağın kümülatif kullanım miktarı tüm aktivitelerdeki toplam talebi karşılamalıdır. Yenilenemez kaynak miktarı, atandığı aktivitenin talebinden küçükse, aktivite başlatılamaz. Bir kaynak, hem tüm projede kullanım açısından, hem de aktivitelerdeki kullanımı açısından sınırlıysa çift yönden kısıtlı kaynak olarak adlandırılır.

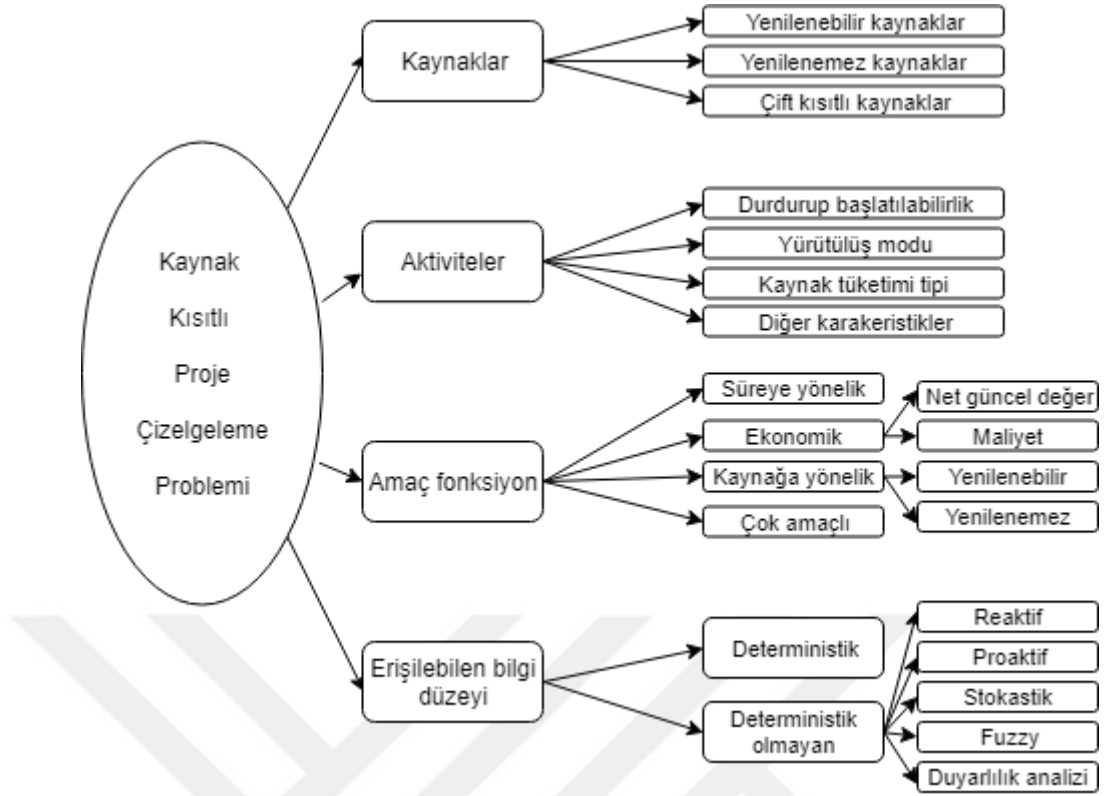
Geniş Sınıflandırma:

Geniş sınıflandırma, dar sınıflamada verilen başlıkları, genişletilmiş bir ağaç yapısı şeklinde ele almaktadır. Geniş sınıflandırma Şekil 2.1’de görülmektedir. Geniş sınıflandırmanın, dar sınıflandırmadan farklılaşan başlıkları aşağıda açıklanmaktadır.

Reaktif çizelgeleme: Bu yaklaşımda, belirsizlikler çizelgeleme sırasında ele alınmamakta, beklenmedik bir olay meydana geldiğinde yeniden çizelgeleme yapılmaktadır. Aşağıda verilen iki sorunun cevapları, reaktif çizelgelemede yol göstericidir:

- Yeni çizelgeleme işlemi ne zaman ve nasıl tetiklenir?
- Yeni çizelgelemenin çıktısı nasıl olmalı?

Proactive–Robust çizelgeleme: Reaktif çizelgelemenin aksine, projenin tamamlanma süresini geciktirebilecek beklenmedik olaylar, çizelgeleme sırasında ele alınır. Temel amaç, proje çıktıları üzerindeki dalgalanmanın en (küçüklenmesi) aza indirilmesidir. Proje giriş parametrelerinin özel bir haline bağlı olarak elde edilebilen en kısa proje tamamlanma süresi yerine, proje giriş parametrelerinde ve projenin içinde bulunduğu ortam parametrelerinde ortaya çıkabilecek olası dalgalanmalara rağmen, en az dalgalanan proje tamamlanma süresi aranır. Proje çıktısı çoğunlukla en kısa tamamlanma süresi olsa da, problemin yapısına bağlı olarak, kaynak dalgalanmasının en küçüklenmesi gibi çıktılar, birincil veya çoklu olarak en küçüklenebilir.



Şekil 2.1 : KKPÇP Geniş Sınıflandırma (Habibi ve diğ. 2018).

Duyarlılık analizi: Proactive–Robust çizelgelemeye ek olarak, proje giriş parametrelerinin ve projenin içinde bulunduğu ortam parametrelerinin, proje çıktılarındaki dalgalanması üzerindeki etkileri araştırılır. Temel amaç, proje çıktıları üzerindeki dalgalanmanın sönümlendirilmesidir. Çıktılarda büyük dalgalanma yaratabilecek girdilerin sönümlendirilmesine öncelik verilir. Bunun için senaryo analizleri veya birçok senaryonun sonucunu bir olasılık dağılımı olarak ifade eden simülasyon teknikleri kullanılabilir.

Bulanık (*Fuzzy*) çizelgeleme: Bazı durumlarda, tarihsel veriler mevcut değildir ve aktivite süresi için olasılık dağılımı elde edilemez. Ayrıca, aktivitelerin süreleri –genellikle projenin benzersiz koşullar altında gerçekleştirilmesi ve tekrar edilemezliği nedeniyle– uzmanlar tarafından tahmin edilir. Aktivite sürelerinin birden çok seçenek içerisinden doğru olarak seçilmesinin oluşturduğu kaygı, farklı seçeneklerin peş peşe dizilmesi sonrasında ortaya çıkabilecek olan proje tamamlanma süresi belirsizliğinden daha yoğundur. Rastgele sayılarla yapılan simülasyon yerine, üyelik fonksiyonlarını kullanan bulanık çizelgeleme, düşük sapma hatası içeren çizelgeler üretme konusunda yol gösterici olabilir.

2.2 Çözüm Yöntemleri

Bir problem senaryosunun ya da çevrenin (genellikle matematiksel olarak) ifade edilmesi olarak tanımlanan karar modellemesi, belirleyici (*Deterministik*) modelleme ve olasılığa dayalı (*Stokastik*) modelleme olarak iki modelleme türü içermektedir. Belirleyici sistemin gelecekteki durumlarının gelişmesinde rastgelelik bulunmaz ve belirli bir başlangıç koşulundan veya başlangıç durumundan yola çıkarak her zaman aynı çıktıyı üretir. Olasılığa dayalı modeller ise sistemin herhangi bir yerinde rastgelelik barındırırlar ve aynı şartlar altında tekrarlanan deneylerde farklı sonuçlar üretirler.

KKPÇP'ne yönelik çözüm yöntemleri: Kesin, sezgisel ve meta-sezgisel yöntemler olarak üç genel başlık altında incelenmektedir. Belirleyici modeller başlığı altında yer alan kesin yöntemler, arama uzayının dar olduğu küçük problemlerde en iyi sonuçlara (*global optimum*) ulaşabilirler. Eğer arama uzayı genişlerse, çözüm süresi –pratikte kabul edilemeyecek kadar çok– uzar. Belirleyici modeller altında yer alan sezgisel yöntemler, basit yol gösterici kurallarla, çabuk fakat en iyi çözümden uzak çözümlerler üretirler. Olasılığa dayalı modeller altında yer alan meta-sezgisel yöntemler ise bünyelerinde rastlantısallık barındırırlar, arama uzayının geniş olduğu problemlere uygulanabilirler ve makul bir süre içerisinde yeteri kadar iyi bir sonuca (*local optimum*) ulaşabilirler.

KKPÇP'nin karmaşıklık (*complexity*) düzeyi, yüksek zorluk anlamına gelen NP-Hard olarak ispatlanmıştır (Blazewicz ve diğ., 1983). Aktivite sayısı az ise kesin çözüm yöntemleri uygulanabilir. Aktivite sayısı arttıkça, sezgisel ve meta-sezgisel algoritmaların kullanımı kaçınılmaz hale gelir. Doktora tezinde kullanılan PSPLIB problemleri j30, j60, j90, j120 problem setlerinden oluşmaktadır. Bu setler sırasıyla 30, 60, 90 ve 120 aktivite içermektedirler. Her problem setinde çözümü kolay ve zor problemler bulunsa da, genel olarak j30 seti, kesin yöntemler ile çözülebilir olma sınırında kalmaktadır.

KKPÇP çözümünde sık rastlanan yöntem ve algoritmalar, ana başlıkları ile aşağıda verilmektedir (Brucker ve Knust, 2012):

- En kısa yol algoritması,
- Lineer programlama,

- Network akış algoritması,
- Dal sınır algoritması,
- Kısıt programlama,
- Dinamik programlama,
- Sezgisel algoritmalar,
- Meta-sezgisel algoritmalar,
- Diğer yöntemler.

Optimizasyon işleminin çıktısı çoğunlukla projenin en erken tamamlanma süresidir. Bazı projelerde en erken tamamlanma süresi yanında veya birincil amaç olarak kaynak profilinde düşük oynaklık elde edilmesi amaçlanabilir. Elde edilen çözüm, öncelik sonralık kurallarını ve kaynak kısıtlarını tam olarak sağlamış olmalıdır. Sağlama için çözüm sırasında oluşan kaynak profili ve aktivitelerin başlama sıralaması incelenebilir. Bu inceleme, doğrudan sayısal olarak veya hazırlanacak grafikler aracılığı ile görsel olarak yapılabilir.

Optimizasyon işleminin girdisi olarak kullanılabilecek veri tabloları aşağıda verilmektedir:

- Öncelik–sonralık tablosu,
- Aktivite süreleri tablosu,
- Aktivite kaynak kullanımı tablosu,
- Kaynak sınırlaması tablosu.

Tablolar ilişkisel veri tabanı içerisinde oluşturulabilirler. Ancak kullanımda çoklukla bir kez okuma işlemi için erişilip belleğe aktarılırlar. Özel bir ilişkilendirme gerektirmeyen küçük dosyalar halinde de tutulabilirler. Tablolardan birkaçı bir araya getirilerek, birleştirilmiş bir tablo da hazırlanabilir. Bu durumda, okuma ve yazma işlemleri –bir araya gelme işlemini göz önüne alacak şekilde– özelleşmektedir.

Girdilerin hazırlanmasında, *yasaklı birliktelik tablosu*, ek bir tablo olarak kullanılabilir. Yasaklı birliktelik tablosunun tüm yasaklamaları içermesi durumunda aşırı büyümesi söz konusu olabilir. Bu durumda bir araya gelemeyecek aktivitelerin bir kısmı seçilebilir. Bu tablodaki bilgilerin yol göstericiliğinde, bir araya

gelemeyecek olan aktiviteleri içeren aday çözümler üretilmezler. Öncü algoritma tarafından belirlenen yasaklı birliktelikler setinin varlığı, arama uzayını sınırlandırarak, hızlı ve kaliteli bir çözüm desteği sağlayabilir.

2.2.1 Kesin yöntemler

Bir projenin öncelik–sonralık kurallarını sağlayan olurlu çözüm, basit algoritmalarla kolayca bulunabilir. Probleme kaynak sınırlaması eklenirse karmaşıklık artar ve NP–Hard seviyesi ortaya çıkar. KKPÇP’nin matematik modeli çeşitli araştırmacılar tarafından ayrıntılı olarak verilmiştir (Christofides ve diğ., 1987; Kolisch ve Sprecher, 1996; Kolisch ve Hartmann, 1999). Kesin yöntemler, düşük karmaşıklık düzeyinde ve az aktiviteli projelerde makul sürelerde çözüm üretebilirler.

Kesin yöntemler genel olarak üç başlık altında incelenebilirler:

- Kaynak kısıtlarının bulunmadığı projelerde kullanılan CPM (Kritik Yörünge Metodu, *The Critical Path Method*) ve PERT (Program Değerlendirme ve Gözden Geçirme Tekniği, *Program Evaluation and Review Technique*),
- Küçük projelerde kullanılabilen ve saf matematiksel optimizasyon tekniği olan Lineer ve tamsayılı programlama,
- Büyük projelerde kısmen kullanılabilen ve bir arama ağacı yardımıyla, matematiksel programlama tekniklerini yinelemeli olarak değerlendiren dal ve sınır algoritması.

2.2.1.1 CPM ve PERT

CPM ve PERT, proje zamanlama problemini çözmek için etkili ve yaygın olarak kullanılan tekniklerdir (Kelley Jr ve Walker, 1959; Malcolm ve diğ., 1959). Eğer aktivite süresi –kesin olduğu varsayılan– tek değerle ifade edilebiliyorsa CPM, bir olasılık dağılımından geliyorsa PERT yöntemi kullanılır. Her iki teknik de kaynak kısıtlarını girdi olarak kullanmaz. Karmaşıklıkları çok düşüktür ve basitçe hesaplanabilirler. Hesaplama sonunda aşağıdaki çıktılar elde edilir:

- Projenin en erken tamamlanma süresi,
- Her aktivitenin en erken/geç başlama/bitiş zamanları,
- Her aktivitenin gecikebilme esnekliğini gösteren bolluk değerleri,

- Gecikemez, kritik aktiviteleri içeren kritik yörünge veya yörüngeler.

Toplam bolluğu sıfır olan aktivitelerden oluşan kritik yörünge üzerinde bir birim gecikme, projenin en erken tamamlanma süresini en az bir birim arttırır. Ancak, kritik yörünge üzerinde bulunan aktivitelerden birinin bir birim kısaltılması, proje süresini bir birim kısaltmayabilir. Birden çok kritik yolun varlığı dikkatlice incelenmeli, kısaltma işlemi her kritik yörünge için ayrı ayrı yapılmalıdır. Proje süre başarımı için kritik yörünge üzerindeki aktiviteler dikkatle izlenmelidir. Aktiviteler toplam bolluk değerine göre küçükten büyüğe doğru artan şekilde sıralanırlarsa, elde edilen liste kritik olmaya karşı duyarlı aktiviteleri gösterir. Toplam bolluğu küçük olan aktiviteler de kritik aktiviteler gibi dikkatlice izlenmelidir. İki veya daha çok aktivitenin mantıksal olarak bitişini toplayan düğümlerin hemen önüne, zaman tamponu olarak kullanılmak üzere yeni bir aktivite yerleştirilmesi, risk yönetimi açısından kolayca kullanılabilecek bir yaklaşımdır. PERT tekniği uygulandığında elde edilen aktivite varyans değerleri, aktivite tamamlanma riskini ölçmek için kullanılabilir. Varyansı büyük olan aktiviteler dikkatlice izlenmelidir. Uygulama öncesi varyansı büyük aktiviteler için hazırlanan olasılık dağılımının yeniden gözden geçirilmesi önerilir. Faaliyet sürelerinin varsayıma dayandığı akılda tutulmalı, bu varsayımların sonucu olarak ortaya çıkan program, yol gösterici bir esneklikle değerlendirilmelidir.

Bilgisayarlı hesaplamanın gelişmesiyle birlikte, daha çok hesap yükü gerektiren ve daha berrak bir gözlem olanağı sunan simülasyon teknikleri kullanıma girmiştir (Rezaie ve diğ., 2007). Monte Carlo simülasyon tekniği, olasılık dağılımdan gelen aktivite sürelerini girdi olarak kullanır ve çıktı olarak proje tamamlanma süresini olasılık dağılımı olarak ifade eder.

Esnek bir genelleştirilmiş ağ aracı olan GERT (Grafik Değerlendirme ve Gözden Geçirme Tekniği, *Graphical Evaluation and Review Technique*) dikkat çekici özellikler taşımaktadır (Taylor, 1978). GERT olasılıklı (stokastik) dallanma, ağ döngüleri, çoklu hedef düğümleri ve PERT/CPM'de bulunmayan tekrarlanan düğüm gerçekleştirme gibi özellikleri içerir. GERT özellikleri, kullanıcıya projeleri ve sistemleri çok genel bir formda modelleme ve analiz etme olanağı sağlar (Moore ve Clayton, 1976).

2.2.1.2 Lineer ve tamsayılı programlama

Lineer programlama ve tamsayılı programlama, ek yaklaşımlar gerektirmeyen saf matematik yöntemlerdir (Garfinkel ve Nemhauser, 1972). Geleneksel MILP (Karma Tamsayılı Programlama – *Mixed Integer Linear Programming*) yapısını kullanarak gerçekleştirilen erken araştırmalar, çoğunlukla planlama problemlerinin 0–1 (*binary*) olarak formüle edilmesine ve matematiksel optimizasyon problemi olarak çözülmesine odaklanmıştır. (Pritsker ve diğ., 1969) tarafından önerilen IP/LP (Tamsayılı Programlama – *Integer Programming*) (*Linear Programming*) bazlı orijinal yöntemler, (Oguz ve Bala, 1994) tarafından da kullanılmıştır. Patterson (1984) proje çizelgeleme için en iyi çözüm (*global optimum*) yöntemlerine genel bir bakış hazırlamıştır. Berthold ve diğ. (2010), Koné ve diğ. (2013), Damay ve diğ. (2007) klasik KKPÇP için var olan MILP modelinin genelleştirilmesini ve olay (*event*) kavramına dayanan yeni modelleri sunmuşlardır.

Modern matematiksel programlama araçları IP/LP problemlerini tek çatı altında MILP olarak değerlendirmektedir. Bu programlar, tamsayı değişkenler kümesinin boş verilmesi durumunda LP olarak çalışırlar. KKPÇP için geliştirilen MILP modelleri kavramsal olarak benzerlik gösterirler. Bu modeller çok sayıda 0–1 değişkenle kurulurlar. Problem boyutu büyüdükçe, çok daha hızlı büyüyen değişken sayıları, gerçek uygulama problemlerinde kullanılamayacak kadar uzun –çoğunlukla olanaksız– çözüm süresi gerektirir. MILP modelleri, çizelgeleme probleminin doğal kısıtları ve boyut artışını başarılı bir şekilde ölçeklendiremez. Problem boyutundaki birim artış, MILP modelinde büyük bir artışa sebep olur. MILP modelleri şimdiki olanaklarla küçük KKPÇP’leri için kullanılabilir.

2.2.1.3 Dal ve sınır algoritması

Dal ve sınır algoritmaları, bir arama ağacı oluşturarak arama uzayını keşfetmek için kullanılır. İlk örnekler Demeulemeester ve Herroelen (1992) tarafından sunulmuştur. Aktivite sayısının artışı karmaşıklığı önemli ölçüde artırmaktadır. Diğer yanda, tüm kısıtları karşılamadan, makul bir sürede iyi çözümler üreten algoritmalar önerilmiştir. (Li ve diğ., 1997; Lupetti ve Zagorodnov, 2006). Cheng ve Wu (2006) zaman kısıtlaması içeren bir hibrit dal sınır algoritması geliştirmişlerdir. Simülasyon sonuçları, bu algoritmanın optimizasyon başarımının önceki algoritmalarından daha iyi olduğunu göstermiştir.

Held ve Karp (1962) tarafından açıklanan dinamik çizelgeleme yaklaşımında, iki aktivite ile başlayıp ardından tüm aktiviteler tamamlanana kadar, kalan aktiviteleri ekleyerek optimal bir çözüm geliştirilmektedir.

2.2.2 Sezgisel algoritmalar

Sezgisel algoritmalar, tüm arama uzayı içerisinde iyi bir çözüm (*local optimum*) bulabilirler. Ancak en iyi çözüme ulaşma garantisi yoktur (Gigerenzer ve diğ., 2011). Sezgisel algoritmalar, yaklaşık sonuç veren algoritmalar olarak düşünülebilir. Kesin yöntemlere göre hızlı çalışırlar, işlem yükleri düşüktür ve daha az bellek kullanırlar. Sezgisel ve meta-sezgisel terimleri, birbirlerinin yerine kullanılabilir. İlk çalışmalarda sadece sezgisel terimi kullanılırken, ilerleyen çalışmalarda meta-sezgisel terimi ağırlık kazanmıştır. Genel olarak sezgisel algoritmalar, probleme özel öncelik kuralları ya da yol göstericiler olarak tanımlanmaktadır. Meta-sezgisel algoritmalar ise problemten bağımsız –probleme göre özelleştirme olanağı da bulunan– ileri düzey akıllı görünümlü algoritmalar olarak tanımlanabilirler. Meta-sezgisel algoritmalar biyolojik, fiziksel, sosyal, vs. esinli veya mantıksal–matematiksel yol göstericiler olabilirler.

Sezgisel algoritmalar, uygulanma biçimine göre tek geçişli ve çok geçişli olarak iki başlık altında incelenmektedirler. Yol gösterici kural ya da kural setinin (sezgisel), verilen sıraya göre, projeyi oluşturan tüm aktivitelere bir kez uygulanması, tek geçiş tekniği olarak adlandırılır. Çok geçişli tekniklerde ise, yol gösterici kural ya da kural seti, daha fazla iyileştirme sağlanana kadar tekrar tekrar uygulanır. İyileştirme geçişleri genel olarak olurlu bir çözümle başlar ve iyileşme sürdükçe devam eder (Agarwal ve diğ., 2006). Sık kullanılan çok geçişli teknikler: çoklu öncelik kuralları, ileri–geri çizelgeleme ve örnekleme yaklaşımı olarak sıralanabilir.

2.2.2.1 Çizelge oluşturma algoritmaları

Eğer kaynak sınırlaması yoksa projenin öncelik–sonralık kurallarını sağlayan olurlu tamamlanma süresi, basit algoritmalarla kolayca bulunabilir. Kaynak sınırlaması probleme eklenirse, hem öncelik–sonralık kurallarını hem de kaynak sınırlaması koşullarını sağlayan çizelge oluşturma algoritmaları kullanılmalıdır. Çizelge oluşturma algoritmaları, kurucu algoritmalar olarak da adlandırılmakta ve makul bir sürede olurlu çizelgeler üretebilmektedir. Üretilen çizelgeler çoğunlukla diğer

algoritmalar tarafından başlangıç çözümü olarak kullanılmaktadır. Çizelge oluşturma algoritmaları Artigues ve diğ. (2010) tarafından dört başlıkta incelenmektedir:

- Standart liste çizelge kurucuları,
- Genel ekleme tabanlı liste çizelge kurucusu,
- Set tipi çizelge kurucuları,
- İleri-geri, çift yaslama kurucular.

Standart liste çizelge kurucu algoritmalar ise iki başlık altında öne çıkmaktadır (Brucker ve Knust, 2012):

- Aktivite odaklı seri çizelgeleme,
- Zaman odaklı paralel çizelgeleme.

Liste çizelge kurucu algoritma her adımda, başlama koşullarını sağlayan aday aktiviteleri belirler, aday aktivitelerden birini uygun bir öncelik yöntemiyle seçer ve başlatılmış aktiviteleri gösteren listeye ekler. Öncelik yöntemi, probleme özel olabilir veya rastgele sayılardan oluşabilir.

Seri Çizelgeleme:

Her adımda bir aktivite başlatılabilir. Seri çizelgeleme algoritmasında çalışma boyunca güncellenen üç küme kullanılmaktadır:

- Karar kümesi,
- Tamamlanmış faaliyetler kümesi,
- Kalan aktiviteler kümesi.

Karar kümesinde yer alan aktivitelerin öncülleri bitmiştir ve çizelgeleme işlemi için kullanılabilirler.

Tamamlanmış faaliyetler kümesinde yer alan aktiviteler karar kümesinden seçilmiş ve çizelgelenmişlerdir.

Kalan aktiviteler kümesi, karar kümesi ve tamamlanmış faaliyetler kümesi dışında kalan aktivitelerden oluşmaktadır. Bu aktiviteler henüz karar kümesine girmemişlerdir.

Algoritma, her adımda tamamlanmış aktiviteler kümesine ve kalan aktiviteler kümesine bakar ve karar kümesini oluşturur. Karar kümesi içindeki aday aktivitelerden biri uygun bir öncelik yöntemiyle seçilir. Seçilen aktivite çizelgelenir ve tamamlanmış aktiviteler kümesine aktarılır. Öncelik değerine göre başlama hakkı kazanan aktivite, eğer kaynak yeterliyse öncüllerinin bitişinden hemen sonra başlayabilir, Kaynak yetersiz ise başlamak için kaynağın uygun olduğu en yakın yere kadar ötelenir. Bu işlem tüm aktiviteler çizelgeleninceye kadar sürer.

Paralel Çizelgeleme:

Her adımda birden çok aktivite başlatılabilir. Paralel çizelgeleme algoritmasında çalışma boyunca güncellenen dört küme kullanmaktadır:

- Karar kümesi,
- Tamamlanmış faaliyetler kümesi,
- Başlatılmış ancak henüz tamamlanmamış faaliyetler kümesi,
- Kalan faaliyetler kümesi.

Başlatılmış ancak henüz tamamlanmamış faaliyetler kümesi, paralel çizelgeleme sırasında gerçekleştirilen çizelgeleme adımının birinde başlatılmış, ancak henüz tamamlanmamış faaliyetleri içerir.

Paralel çizelgeleme, zaman çizgisi üzerinde ilerleyerek çalışır. Başlama hakkı kazanan aktiviteler önceliklerine göre sıralanırlar. Eğer kaynak yeterli ise, öncül aktivitelerinin bitişinden hemen sonra başlarlar. Kaynak yetersiz ise, başlamak için kaynağın uygun olduğu en yakın yere kadar ötelenirler. Başlatılmış aktivitelerden en erken tamamlanan bulunur. Zaman çizgisi işaretçisi bu değere ötelenir. İşlem tüm aktiviteler çizelgeleninceye kadar sürer.

2.2.2.2 Öncelik kuralları

Öncelik kuralları, kolay uygulanma ve hızlı sonuç üretme avantajları nedeniyle, kaynak kısıtlı çizelgeleme probleminde yaygın olarak kullanılmaktadır. İlk örnekleri Giffler ve Thompson (1960) tarafından verilmiş, Kolisch (1996) tarafından ayrıntılı olarak incelenmiştir.

Öncelik kurallarıyla çalışan bir algoritma oluşturmak için, öncelik kuralları ve çizelge oluşturma algoritmaları bir araya getirilir. Ortaya çıkan sezgisel yöntem, tek

veya çok geçişli olabilir. KKPÇP'ne özel öncelik kurallarından örnekler Çizelge 2.1'de verilmiştir (Brucker ve Knust, 2012).

Öncelik kurallarına dayalı yaklaşımlar ve çözümle için ayrıntılı bilgi Abdolshah (2014) tarafından verilmektedir. Problemin özelliklerinden faydalanılarak geliştirilen öncelik kuralları çizelge oluşturma algoritmalarında yol gösterici olarak kullanılmaktadır.

Rastgele sayılar, probleme özel olarak geliştirilen öncelik kuralları yerine, probleminden bağımsız öncelik belirleyiciler olarak kullanılabilir. Bu durumda rastgele sayılar, probleme özel öncelik kuralları yerine geçmektedir.

Çizelge 2.1 : KKPÇP için öncelik kuralları.

Aktivite Çıkışı	
SPT	En kısa işlem süresi
LPT	En uzun işlem süresi
Network çıkışı	
MIS	En çok tetiklenmiş ardıl sayısı
LIS	En az tetiklenmiş ardıl sayısı
MTS	En çok toplam ardıl sayısı
LTS	En az toplam ardıl sayısı
GRPW	En büyük konumsal ağırlık
Kritik Yörünge Çıkışı	
EST	En erken başlama zamanı
EFT	En erken tamamlanma zamanı
LST	En geç başlama zamanı
LFT	En geç tamamlanma zamanı
MSLK	En küçük bolluk süresi
Kaynak çıkışı	
GRR	En yüksek kaynak talebi

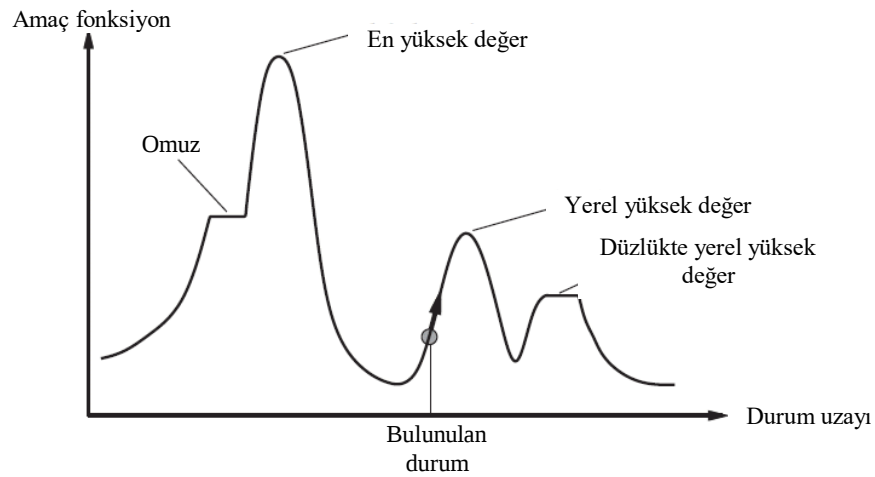
2.2.2.3 Yerel arama algoritmaları

Yerel arama, ilerleyen bir optimizasyon sürecinde elde edilen son olurlu iyi çözümün yakın çevresinin, daha iyi bir çözüm için araştırılmasıdır. Optimizasyon algoritması arama uzayında başka bir bölgeye sıçramadan önce yapılan yerel arama işlemi, çok

yakında bulunabilecek daha iyi bir çözümün yakalanmasını sağlayabilir. Arama, genellikle mevcut komşuluk yapısı üzerinde küçük hareketlerle gerçekleştirilir.

Yerel arama algoritmaları Russell ve Norvig (2009) tarafından detaylı şekilde incelenmiştir. Yerel arama uzayı ve karşılaşılan durumlar Şekil 2.2’de gösterilmektedir. Sık rastlanan yerel arama algoritmaları, Tepe Tırmanma (*Hill Climbing*), Tavlama Benzetimi (*Simulated Annealing*), Tabu Araştırma (*Tabu Search*) olarak sıralanabilir.

Tepe tırmanma algoritması, dögüsel halde daha iyi bir noktaya tırmanmaya çalışır ve daha iyi bir yer bulamadığında durur. Başlangıç çözümünde tüm komşuları kontrol edip en iyi seçeneğe yönelebilir veya ilk bulduğu yeni iyiden devam edebilir. Son derece hızlı ilerler ve bellekte tek çözüm tutar. Arama uzayında yer alan olası diğer çözümlere ulaşmak için iki mekanizma kullanılabilir. Birincisi, yerel iyiye ulaştıktan sonra rastgele bir yerden yeniden başlamaktır. Sonsuz döngüyü engellemek için bir durdurma kriteri eklenmelidir. Durdurma kriteri, toplam deneme sayısı veya amaç fonksiyon için yeterli görülen bir hedef değer olabilir. İkincisi ise, bir noktadan başlamak yerine çok noktadan başlamak ve paralel ilerlemek olabilir. Algoritma, tüm paralel parçalar kendi yerel iyilerine ulaştınca durur.



Şekil 2.2 : Yerel arama. (Russell ve Norvig, 2009).

Tepe tırmanma algoritmasına stokastik bir ekleme yapılabilir. Bir tepeye ulaşıp takılma gerçekleştiğinde, küçük bir olasılıkla kötü yönde harekete izin verilir. Bu geçiş, yeni bir patika oluşturabilir ve başka bir tırmanış başlar. Tepe tırmanma algoritması, bu düzenlemeyle SA haline dönüşür. Geri düşüş sırasında eskiden ziyaret edilmiş yerlerin yeniden ziyaret edilmesi gereksizdir. Yakın geçmişte ziyaret

edilmiş yerlerin bellekte tutulması, aynı yerlere dönüşü kısıtlar. Algoritmanın son durumu Tabu Destekli SA algoritmasıdır.

Meta-sezgisel algoritmalar hızla yakınsayacak şekilde yapılandırılarak, bir yerel arama algoritması haline dönüştürülebilirler. Bunun için arama uzayında yeni yerlere geçişin ve çeşitliliğin sınırlandırılması yeterlidir.

2.2.3 Meta-sezgisel algoritmalar

Büyük KKPÇP'i, kesin yöntemlerle, makul bir hesaplama süresi içinde çözülememektedir. Büyük problemlerin, makul sürelerde, yeteri kadar iyi çözümlerine ulaşabilmek için sezgisel ve meta-sezgisel tekniklere başvurulmaktadır. Sezgisel ve meta-sezgisel teknikler, makul bir süre içinde çözümler üretseler de, bunların etkililiğini ve verimliliğini arttırmak için daha fazla araştırma yapılması gerekmektedir (Michalewicz, 1996; Widmer ve diğ., 2008).

Meta-sezgisel algoritmalar, problemten bağımsız –probleme göre özelleştirme olanağı da bulunan– ileri düzey akıllı görünümlü algoritmalar olarak tanımlanabilirler. Meta-sezgisel algoritmalar biyolojik, fiziksel, sosyal, doğa vs. esinli veya mantıksal–matematiksel yol göstericiler olabilirler.

GA, SA, TS, ACO, PSO, iyi bilinen meta-sezgisel yöntemlerdir ve çizelgeleme probleminin çözümünde yaygın olarak kullanılmaktadırlar. Doğal ve biyolojik modellerden esinlenen meta-sezgisel algoritmalar arasında yer alan GA, ACO, PSO, çoklu başlangıç çözümü geliştirerek ilerlerler. Mantıksal modellerden esinlenen meta-sezgisel algoritmalar arasında yer alan TS, geri dönüşleri engelleyen yasaklama listeleri kullanır ve tek başlangıç çözümü geliştirerek ilerler. Fiziksel modellerden esinlenen meta-sezgisel algoritmalar arasında yer alan SA, tek başlangıç çözümünü geliştirerek ilerler. Farklı esinlerle geliştirilen algoritmaların bir arada tasarlandığı hibrit modeller, sağladıkları esneklik ve performans üstünlükleri sayesinde artmakta olan kullanım alanı bulmaktadırlar.

Sık rastlanan Meta-sezgisel algoritmaların tarihsel gelişimi Çizelge 2.2'de görülmektedir (Madni ve diğ., Latiff, 2016).

KKPÇP’nde meta–sezgisel algoritmalarla yapılan çalışmalar detaylı olarak Brucker ve Knust (2012), Kolisch ve Hartmann (2006), Habibi ve diğ. (2018) ve Abdolshah (2014) tarafından verilmiştir.

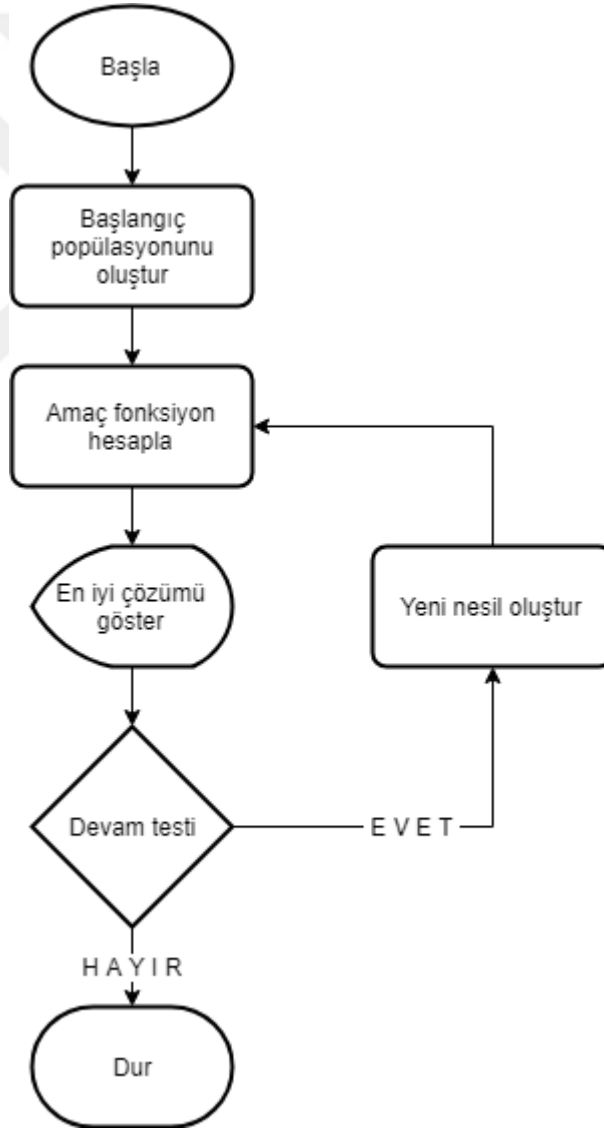
Çizelge 2.2 : Sık rastlanan meta–sezgisel algoritmaların tarihsel gelişimi.

Algoritma	Geliştiren
Evolution Process (EP)	Barricelli, 1954
Random Search (RS)	Karnopp, 1963
Random Optimization (RO)	Matyas, 1965
Evolutionary Programming (EP)	Fogel, 1993
Graph Partitioning Method (GPM)	Kernighan ve Lin, 1970
Genetic Algorithm (GA)	Goldberg, 1989
Scatter Search (SS)	Glover, 1977
Metaplan	Mercer & Sampson, 1978
Simulated Annealing (SA)	Kirkpatrick ve diğ., 1983
Tabu Search (TS)	Glover, 1990
Memetic Algorithm (MA)	Moscato ve diğ., 1989
Ant Colony Optimization (ACO)	Dorigo ve Blum, 2005
Immune algorithm (IA)	Mori ve diğ., 1993
Multi–Objective GA (MOGA)	Fonseca ve Fleming, 1993
Reactive Search Optimization (RSO)	Battiti ve Brunato, 2010
Particle Swarm Optimization (PSO)	Eberhart ve diğ., 2001
Differential Evolution (DE)	Storn ve Price, 1997
Harmony Search (HS)	Geem ve diğ., 2001
Artificial Bee Colony Algorithm (ABC)	Karaboga ve Basturk, 2007
Glowworm Swarm Optimization (GSO)	Krishnanand ve Ghose, 2009
Honey bee Mating Optimization (HbMO)	Haddad ve diğ., 2006
Imperialist Competitive Algorithm (ICA)	Atashpaz ve Lucas, 2007
Intelligent Water Drops (IWD)	Hosseini, 2007
Firefly Algorithm (FA)	Yang, 2009
Cuckoo Search (CS)	Yang ve Deb, 2009
Gravitational Search Algorithm (GSA)	Rashedi, 2009
League Championship Algorithm (LCA)”	Kashan, 2009
Bat Algorithm (BA)	Yang, 2010
Galaxy based Search Algorithm (GbSA)	Hosseini, 2011
Spiral Optimization (SO)	Tamura ve Yasuda, 2011
Teaching Learning Based Optimization (TLBO)	Rao ve diğ., 2011
Krill Herd (KH)	Gandomi, 2012
Coral Reefs Optimization (CRO)	Sanz ve Del, 2013
Swallow Swarm Optimization (SSO)	Neshat, 2013
Interior Search Algorithm (ISA)	Gandomi, 2014
Gradient Evolution Algorithm (GEA)	Kuo ve Zulvia, 2015
Lion Optimization Algorithm (LOA)	Yazdani ve Jolai, 2016
Optics Inspired Optimization (OIO)	Kashan, 2015
Water Wave Optimization (WWO)	Zheng, 2015

2.2.3.1 Genetik algoritma

Biyolojik evrim sürecinden esinlenen Genetik Algoritmalar (GA), Holland (1975) tarafından tanıtılmış ve Goldberg (1989) tarafından detaylandırılmıştır. GA, yerel arama stratejilerinin aksine, sadece bir çözüm yerine, çözüm topluluğunu dikkate alır. Bir başlangıç popülasyonu oluşturduktan sonra, çaprazlama ve mutasyon mekanizmalarını kullanarak yeni çözümler üretilir.

Yeni çözümler üretildikten sonra, en uygun çözümler gelecek nesli oluştururken, diğer uçta kalan kötü çözümler elenirler. Uygunluk değeri, çözümün kalitesini ölçmek için kullanılır ve genellikle çözülecek optimizasyon probleminin amaç fonksiyonu üzerinden hesaplanır. GA kavramsal akış şeması Şekil 2.3'te verilmiştir.



Şekil 2.3 : GA kavramsal akış şeması.

GA kombinatorial optimizasyon problemleri için uygun görülmektedir (Hartmann, 1999). GA, Bölüm 3.3’de ayrıntılı olarak incelenmektedir.

2.2.3.2 Tavlama benzetimi algoritması

Tavlama Benzetimi Algoritması (Simulated Annealing) (SA), erimiş bir katı maddenin –düşük enerjili bir duruma soğutulduğu– fiziksel tavlama işleminden esinlenir. Fiziksel algoritmalar, fiziksel bir süreçten esinlenen algoritmalar ve genellikle meta–sezgisel ve hesaplamalı zekâ alanlarına aittirler. Bu algoritmalar, doğa esinli olarak adlandırılırsalar da, GA, ACO, PSO gibi biyolojik esinli algoritmalarla farklı bir sınıftadırlar. Metalürji, müzik ve su hareketleri gibi karmaşık dinamik sistemler, fiziksel algoritmalarla esin kaynağı olabilirler. SA, yerel arama ve küresel arama tekniklerini bir arada bulunduran stokastik optimizasyon algoritmasıdır.

SA, Kirkpatrick ve diğ. (1983) tarafından sunulmuştur. (Granville ve diğ., 1994) SA için global optimuma yakınsama kanıtı sağlamıştır. “Simulated Annealing: Theory and Applications” (Laarhoven ve Aarts, 1987) ve “Simulated Annealing: Parallelization Techniques” (Azencott, 1992) önemli başvuru kaynakları arasında gösterilir (Brownlee, 2011).

SA algoritmasında, başlangıç çözümüne yerel arama uygulanarak bir komşu çözüm üretilir. Yeni çözüm mevcut olandan daha iyi olursa kabul edilir ve arama bu yeni çözümden başlar. Aksi takdirde, kötü olan yeni çözüm, uzaklaşmanın büyüklüğüne ve sıcaklık olarak adlandırılan bir parametreye bağlı olarak hesaplanan küçük bir olasılıkla kabul edilir. Daha kötü çözümlerin kabul edilmesi, yerel iyiden kaçmak ve en iyiye ulaşmak amacını taşır. Algoritma ilerledikçe, daha kötü komşu çözümleri kabul etme olasılığını azaltmak için sıcaklık azaltılır. . SA, herhangi bir bozulmayı reddeden, daha iyi bir komşu çözümü –düşük bir olasılıkla kötü komşu çözümü– kabul eden, basit açgözlü (*Simple Greedy*) algoritmanın uzantısı olarak görülebilir (Hartmann, 1999). SA algoritması, taksonometri, esinlenme, benzetim ve strateji alt başlıklarında incelenecektir.

Taksonomi:

SA, stokastik süreç izleyen küresel bir meta-sezgisel optimizasyon algoritmasıdır ve Monte Carlo algoritmasından izler taşır. Paralel SA, Hızlı SA ve Adaptif SA gibi genişletilmiş algoritmalar için bir altlık oluşturmaktadır.

Esinlenme:

SA, metalürjide kullanılan tavlama işleminden esinlenilerek geliştirilmiştir. Bu işlemde bir malzeme ısıtılır ve malzemenin kusurlarını azaltmak için kontrollü koşullar altında yavaşça soğutulur. Tavlama işlemi malzemenin dayanıklılığını iyileştirir.

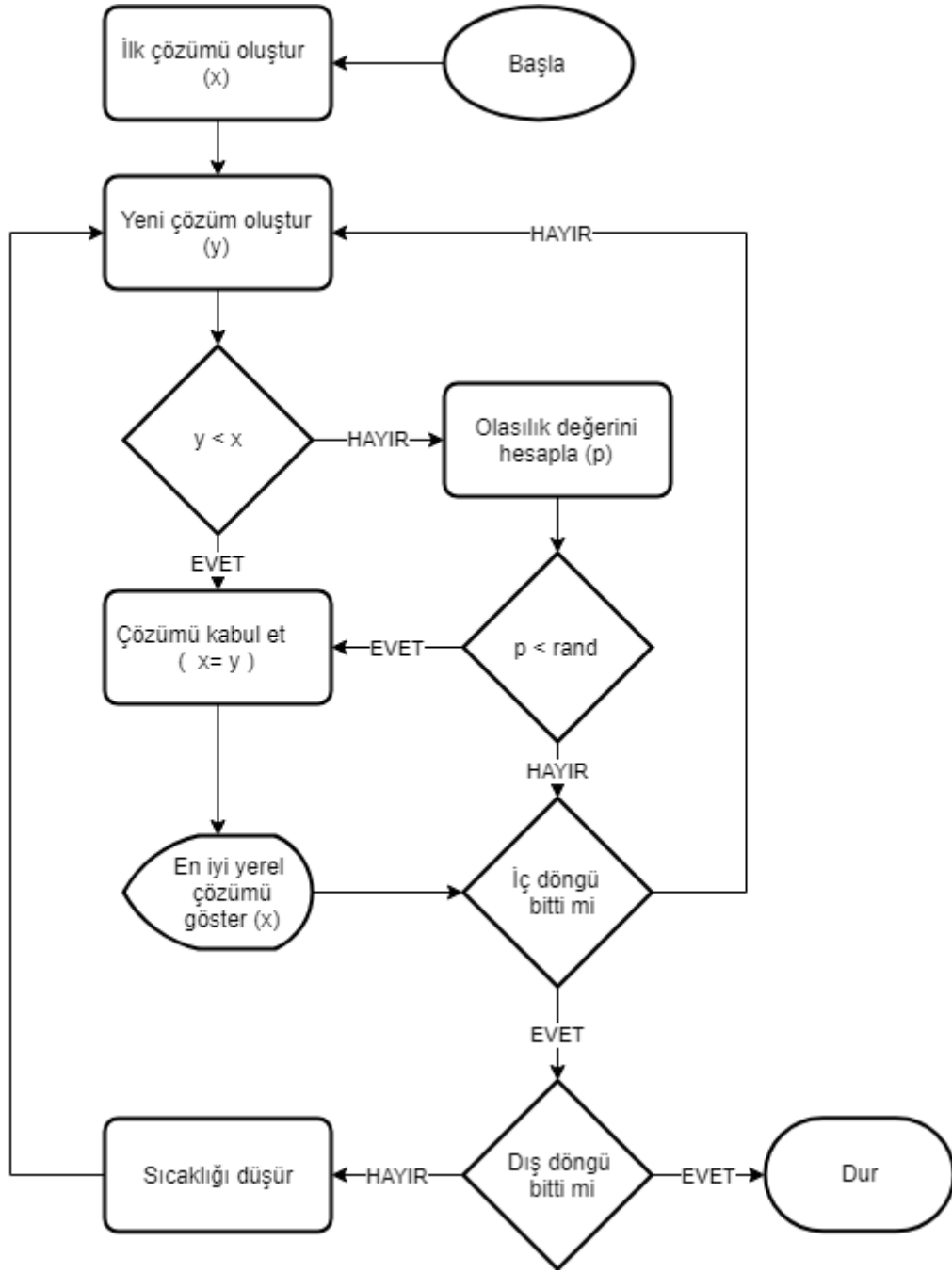
Benzetim

SA algoritması, yerel arama sınırlarını belirlemek için tavlama sırasında uygulanan soğutma işlemini taklit eder. Sürecin başında ısı yüksektir ve yerel arama için kullanılan komşuluk ilişkisi sınırları geniş tutulur. İşlem ilerlerken ısı yavaşça düşürülür ve yerel arama sınırları daralır. İşlem teorik olarak en iyi çözümle (*Global Optimum*) veya ona yakın bir yerel iyi (*Local Optimum*) ile sonuçlanır.

Strateji:

Algoritmanın hedefi, arama işleminin minimum maliyetle yapılmasını sağlamaktır. Yeni elde edilen çözümün kabul olasılığı, termodinamik sistemlerden ödünç alınan bir fonksiyon tarafından değerlendirilir. Süreç ilerledikçe olasılık düşer.

SA hem kombinatorial hem de sürekli fonksiyonların optimizasyonunda kullanılabilir. SA'nın yeteri kadar uzun bir soğutma işlemi sonunda en iyi çözüme ulaşacağı ispatlanmıştır. Diğer yandan bu süre, tüm arama alanının dolaşılması için harcanacak süreden uzun olabilir. Performansı arttırmada: soğutma işlemi parametrelerinin ayarlanması, komşuluk yapısı düzenlemeleri ve probleme özel sezgisel algoritmalar yardımcı olabilir. Yakınsamanın yavaşlaması durumunda, elde edilen en iyi sonucu kullanarak soğutmanın yeniden başlatılması –yerel arama sınırlarını genişleterek– daha iyi sonuçlara ulaşmayı sağlayabilir. SA akış şeması Şekil 2.4'de verilmiştir.

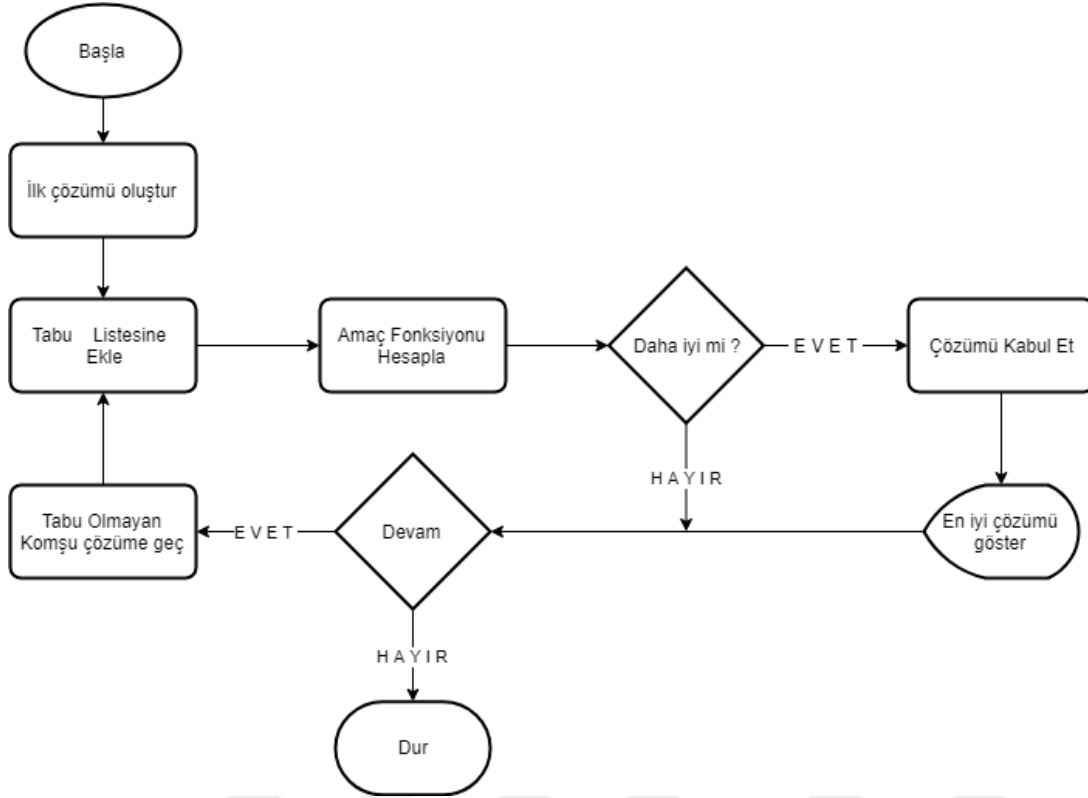


Şekil 2.4 : SA akış şeması.

2.2.3.3 Tabu arama

Tabu Arama Algoritması (Tabu Search Algorithm) (TS) Glover (1990) tarafından geliştirilmiştir. Tüm komşu çözümleri değerlendirir ve en iyisini seçer. Bu yaklaşım tekrarlanarak (*iteratively*) uygulandığında TS döngüye girme olasılığı taşır. Çözüm olarak, tabu listesi kavramı oluşturulmuştur. Genellikle, tabu listesi, yakın zamanda ziyaret edilen çözümleri kaydetmek ve yeniden aynı çözüme götürecek olan

komşuluk hareketlerini yasaklamak için kullanılır (Hartmann, 1999). TS kavramsal akış şeması Şekil 2.5'te verilmiştir.



Şekil 2.5 : TS kavramsal akış şeması.

2.3 Çizelge Gösterim Biçimleri

Çizelge gösterim biçimleri: Çizelge oluşturma algoritma girdisi olarak program üretmek ve optimizasyon algoritması girdisi olarak yeni çizelge üretmek gibi iki amaçla kullanılabilirler.

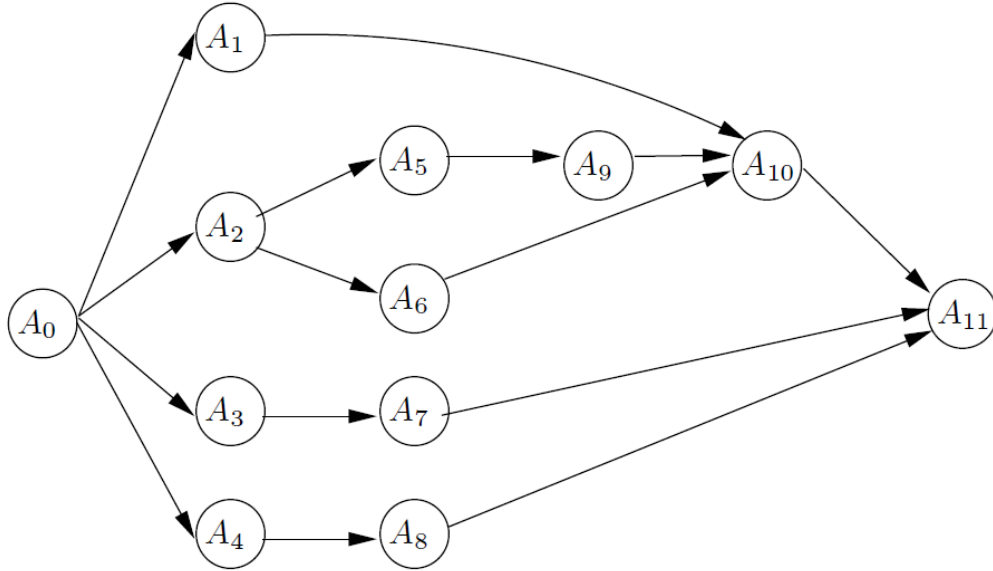
Çizelge kurucu algoritma, verilen listeyi girdi olarak kullanarak olurlu bir çizelge/çözüm oluşturur. Çözüm, aktivitelerin başlama/bitiş zamanlarını veya sadece proje tamamlanma zamanını gösterir. Bazı durumlarda verilen liste üzerinde düzeltici işlem gerçekleştirir. Gerek program gerekse düzeltilmiş liste, çözümü temsil etmektedir.

Optimizasyon algoritması verilen listeyi girdi olarak kullanır ve yeni bir liste oluşturur. Yeni liste kurucu algoritma tarafından işlenir ve çizelge/çözüm oluşturulur. Yeni liste ve çözüm, optimizasyon algoritması tarafından işleme alınır.

Optimizasyon sürecinde kaynak kısıtlı bir projenin olası aday çözümlerini temsil etmek için kullanılan liste yapılı çizelge gösterim biçimleri aşağıda verilmektedir (Kolisch ve Hartmann, 1999):

- Aktivite listesi gösterimi,
- Rastgele anahtar gösterimi,
- Öncelik kuralları gösterimi,
- Kaydırma vektörü gösterimi,
- Diğer dizilimler.

Liste yapılarının örneklenmesinde kullanılan AON (Düğümde Yer Alan Aktivite, *Activity On Node*) çizge Şekil 2.6'da verilmiştir.



Şekil 2.6 : AON çizge (Artigues ve diğ. 2010).

Aktivite Listesi Gösterimi:

Aktivite listesi gösterimi, kolay kurulum, hızlı çözüm ve kolay değiştirilme özellikleri nedeniyle birçok araştırmada yaygın olarak kullanılmaktadır (Moumene ve Ferland, 2008).

Aktiviteler arası öncelik–sonralık koşullarını sağlayan ve aktivite numaralarıyla gösterilen kombinatorial bir dizilim, aktivite listesi olarak kullanılabilir. Bu liste aktivitelerin tetiklenme sırasını gösterir. Çizelge oluşturma algoritması listedeki

aktiviteleri sırayla tetikler. Eğer kaynak yeterli değil ise, kaynağın yeterli olduğu yere kadar ilerlenir, pozisyon belirlendikten sonra aktivite başlatılır ve kaynak kullanımı güncellenir. Liste tamamlandığında tüm aktivitelerin başlama/bitiş zamanları bilinmektedir. En büyük tamamlanma zamanı, projenin tamamlanma zamanı olarak kullanılır.

Olurlu bir çözümü gösteren aktivite listesi üzerinde optimizasyon algoritması tarafından uygulanacak değişiklik işlemi sırasında, öncelik sonralık kurallarının korunmasına dikkat edilmelidir.

Şekil 2.6'da verilen çizgeden oluşturulan ve öncelik–sonralık kısıtlarını sağlayan olurlu aktivite listesi gösterim örnekleri aşağıda verilmiştir.

$$L=[0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11]$$

$$L=[0 \ 4 \ 3 \ 2 \ 1 \ 8 \ 7 \ 6 \ 5 \ 9 \ 10 \ 11]$$

$$L=[0 \ 2 \ 5 \ 9 \ 1 \ 6 \ 10 \ 4 \ 8 \ 3 \ 7 \ 11]$$

Olurlu liste oluşturmak için 0 numaralı aktiviteden başlayarak 11 numaralı aktiviteye ilerlenebilir veya ters yönde 11 numaralı aktiviteden başlanarak 0 numaralı aktiviteye ulaşılabilir. Sondan çıkışlı aktivite listesi örneği aşağıda verilmiştir.

$$L=[11 \ 10 \ 1 \ 9 \ 5 \ 6 \ 8 \ 7 \ 3 \ 4 \ 2 \ 0]$$

Rastgele Anahtar Gösterimi:

Rasgele anahtar (ya da öncelik değeri) gösterimi ilk olarak Bean (1994) tarafından önerilmiştir. Kolisch ve Hartmann (1999) rastgele anahtar gösteriminin öncelik kurallarının farklı bir ifadesi olduğunu belirtmiştir. Rastgele anahtar gösterimleri, birçok araştırmada kullanılmıştır (Mendes ve diğ., 2009; Abdolshah, 2014).

Rastgele anahtar gösteriminde öncelik–sonralık koşullarını sağlama zorunluluğu yoktur. Öncelik–sonralık kısıtları, çizelge üretme algoritması tarafından kendiliğinden sağlanır. Başlama hakkı elde eden aktiviteler içinden en küçük (veya en büyük) anahtar değerine sahip olan çizelgelenir. Rastgele anahtar listesi, aktivitelerin –diğerleri arasındaki– önceliğini gösterir ve paralel yürütülebilen aktiviteler arasından seçim yapılabilmesini sağlar.

Oluru bir çözümü gösteren rastgele anahtar listesi üzerinde, optimizasyon algoritması tarafından uygulanacak değişiklik işlemi sırasında, öncelik sonralık kurallarının

korunması gerekmez. Liste her zaman olurludur. Bu özellik hesap yükünü azaltır ve uygulama kolaylığı sağlar.

Anahtar değeri genellikle sıfırdan büyük ve birden küçük olarak seçilir. Şekil 2.6'da görülen çizgiden oluşturulan rastgele anahtar listesi gösterimi aşağıda verilmiştir.

$$L=[0,00 \ 0,42 \ 0,56 \ 0,12 \ 0,88 \ 0,34 \ 0,62 \ 0,04 \ 0,28 \ 0,48 \ 1,00]$$

Popülasyon tabanlı bir optimizasyon algoritmasında, birbirinden farklı görünen fakat aynı öncelik dizilimini içeren listeler üretilmiş olabilir. Bu konu boşluk analizi sırasında gözlemlenmiş olup, önerilen güçlendirme yaklaşımı Bölüm 4.2.1.2'de ele alınmaktadır.

Öncelik Kuralları Gösterimi:

Öncelik kuralları, kolay uygulanma ve hızlı sonuç üretme avantajları nedeniyle, kaynak kısıtlı çizelgeleme probleminde yaygın olarak kullanılmaktadır. İlk örnekleri Giffler ve Thompson (1960) tarafından verilmiş, Kolisch (1996) tarafından ayrıntılı olarak incelenmiştir. KKPÇP'nde kullanılan öncelik kurallarından bazıları Bölüm 2.2.2.2'de verilmiştir.

Öncelik kuralları listesinde, probleme özel öncelik kurallarını temsil eden tanıtıcı değerler yer alır. Öncelik kuralları listesi gösterim örneği aşağıda verilmiştir.

$$L=[LFT \ EST \ EFT \ LST \ SPT \ LPT \ MIS \ LIS \ LFT \ EFT \ LFT]$$

Kaydırma Vektörü Gösterimi:

Kaydırma vektörü gösterimi, pozitif tam sayılardan oluşan bir listedir. Aktivitelerin başlangıç zamanı listede verilen değer kadar ötelenir (Sampson ve Weiss, 1993). Liste doğrudan işletilirse kaynak kısıtları göz ardı edilmiş olur. Kaynak kısıtlarını dikkate alan olurlu çözüm için düzeltici işlem yapılabilir. Listede sırası gelen aktivite başlama hakkı kazanır ve en az verilen değer kadar ötelenir. Eğer kaynak yeterli değil ise, kaynağın yeterli olduğu yere kadar ilerlenir, pozisyon belirlendikten sonra aktivite başlatılır ve kaynak kullanımı güncellenir. KKPÇP'nde sık kullanılmayan kaydırma vektörü gösterimi, probleme özel olarak daha verimli hale getirilebilir.

2.4 KKPÇP Çözümüne Yönelik Çalışmalar

KKPÇP birçok araştırmacı tarafından genel bir çerçeve içerisinde tanımlanmış, çözüm modelleri ve algoritmalar detaylandırılarak incelenmiştir.

Kolisch ve Hartmann (2006), '*Experimental investigation of heuristics for resource-constrained project scheduling*' başlıklı makalelerinde, 2000 yılında yaptıkları çalışmayı yenilemişler, sezgisel algoritmaları sınıflandırmışlar, KKÇP çözümünde kullanılan algoritmalar alanında ortaya çıkan gelişmeleri aktarmışlardır.

Brucker ve Knust (2012), '*Complex Scheduling*' başlıklı çalışmalarında, çizelgeleme modellerini, çizelgeleme algoritmalarını ve KKÇP'ni ayrıntılı olarak incelemişlerdir.

Abdolshah (2014), '*A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions*' başlıklı makalesine, saygın dergilerde yer alan 200 makaleyi incelemiş, KKÇP'ni sınıflandırmış ve KKÇP çözümünde kullanılan algoritmalar alanında ortaya çıkan gelişmeleri aktarmıştır.

Habibi ve diğ. (2018), '*Resource-constrained project scheduling problem: review of past and recent developments*' başlıklı makalelerinde, 1980–2017 yılları arasında yayınlanmış 216 çalışmayı değerlendirmişler, KKÇP'ni sınıflandırmışlar ve KKÇP çözümünde kullanılan algoritmalar alanında ortaya çıkan gelişmeleri aktarmışlardır.

KKÇP için PSPLIB çözüm performansı bildirilen ve karşılaştırma tablolarında kullanılan önemli çalışmalar Çizelge A.1'de, YÖK Ulusal Tez Merkezi'nden erişilen çalışmalar Çizelge A.2'de görülmektedir.

KKÇP çözümüne yönelik çalışmalar hakkında yapılan literatür araştırmasının sonuçları aşağıdaki sıralamaya göre ele alınmaktadır:

- Paralel Genetik Algoritmalar,
- Tavlama Benzetimi Algoritmaları,
- Hibrit yaklaşım,
- PSPLIB performans tablolarında yer alan Genetik Algoritma çalışmaları.

Mutalik ve diğ. (1992), '*Solving combinatorial optimization problems using parallel simulated annealing and parallel genetic algorithms*' başlıklı çalışmalarında, PGA, Paralel SA ve GA'yı iki kombinatorial optimizasyon problemine uygulamışlar ve aldıkları sonuçları kıyaslamışlardır. Her iki problemde de Paralel SA son sırada yer almıştır. Bir problemde PGA, diğer problemde GA ilk sırayı almıştır. Yazarlar, Paralel SA'nın gelişim halinde bulunduğunu ve üzerinde çalışmalar ilerledikçe daha

iyi sonuçlar alınabileceğini belirtmişlerdir. İşlem süresi açısından paralel algoritmalar, GA'ya kıyasla beş kat daha hızlı sonuç üretmişlerdir.

Mühlenbein (1992), '*Parallel Genetic Algorithms in Combinatorial Optimization*' başlıklı çalışmada, PGA'yı iki optimizasyon problemine uygulamış, alınan sonuçları tepe tırmanma algoritmaları ile karşılaştırmıştır. PGA diğer algoritmalarından çok daha iyi sonuçlar üretmiştir.

Kohlmorgen ve diğ. (1999), '*Experiences with fine-grained parallel genetic algorithms*' başlıklı çalışmalarında, PGA ve GA'yı bir çok kombinatorial optimizasyon problemine uygulamışlar ve aldıkları sonuçları karşılaştırmışlardır. PGA hem çözüm kalitesi hem de sonuç üretme hızı açısından GA'ya göre çok daha iyi sonuçlar üretmiştir. Yazarlar, PGA'nın çeşitliliğin korunması açısından da başarılı olduğunu belirtmişlerdir.

Helzer (2001), '*An empirical study of parallel genetic algorithms for the traveling salesman and job-shop scheduling problems*' başlıklı çalışmada, kombinatorial optimizasyon problemlerinde PGA ve GA'yı karşılaştırmıştır. Tüm problemlerde PGA, GA'ya göre daha yüksek çözüm kalitesi daha kısa sürede çözüm başarısı sağlamıştır. PGA, GA'nın 100 birim zamanda ürettiği sonuçlardan daha yüksek kaliteli sonuçlara, 15 birim zamanın altında ulaşmıştır.

Kandil ve El-Rayes (2006), '*Parallel genetic algorithms for optimizing resource utilization in large-scale construction projects*' başlıklı çalışmalarında: 180, 360 ve 720 aktivite içere 3 projeyi, PGA parametrelerin kombinasyonundan oluşan 183 farklı durumda denemişler ve PGA'ların yüksek çözüm kalitesi, kısa sürede çözüm başarısı ve kolay uygulanabilirlik açısından üstün özellikler gösterdiklerini belirtmişlerdir. Çalışmaların içeriğinde, başlıkta yer alan '*in large-scale construction projects*' ifadesine yönelik açıklayıcı veya destekleyici bir açıklamaya rastlanmamıştır.

Bouleimen ve Lecocq (2003), '*A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version*' başlıklı çalışmalarında, SA'ı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi iki GA sonucun ardından, üçüncü sırada yer almıştır. Yazarlar, elde edilen sonuçların SA parametre ayarlarına duyarlı olduğuna dikkat çekmişlerdir.

Lim ve diğ. (2013), '*New meta-heuristics for the resource-constrained project scheduling problem*' başlıklı çalışmalarına, SA benzeri bir soğuma sürecini, GA içerisinde hibrit olarak kullanmışlar ve PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar sıralamasında, üçüncü sırada yer almıştır İşlemci süresinin sınırsız bırakılması durumunda, J60 ve J120 problemlerinde en iyi sonuca ulaşılmıştır.

Wang ve Zheng (2001), '*An effective hybrid optimization strategy for job-shop scheduling problems*' başlıklı çalışmalarında, SA'yı, GA içerisinde hibrit olarak kullanarak kombinatorial optimizasyon problemlerine uygulamışlar, alınan sonuçları GA, SA ve önceki çalışmalarda kullanılan sonuçlar ile karşılaştırmıştır. GA içerisinde hibrit olarak kullanılan SA, GA'dan, SA'den ve önceki çalışmalardan daha iyi sonuçlar üretmiştir. Yazarlar, GA ve SA'in erken yakınsayarak durduğunu, geliştirilen hibrit algoritmanın çalışmaya devam ederek daha iyi çözüm sonuçları ürettiğini belirtmişlerdir.

Bettemir ve Sonmez (2014), '*Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling*' başlıklı çalışmalarında, GA içerisinde hibrit SA kullanarak geliştirdikleri algoritmayı test problemlerine uygulamışlar, aynı problemlerin çözümünde Primavera ve MS Project programlarının verdiği sonuçlarla karşılaştırmışlardır. Yazarlar, hibrit algoritmanın, GA'dan daha iyi sonuç ürettiğini belirtmişlerdir. Hibrit algoritma tüm test problemlerinde Primavera ve MS Project'den çok daha iyi sonuçlar üretmiştir. Yazarlar, işlem süresinin kısaltılması için paralelleştirmenin kullanılmasının faydalı olabileceği yönünde görüş bildirmişlerdir.

Kochetov ve Stolyar (2003), '*Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem*' başlıklı çalışmalarında, GA, '*path relinking*' ve TS kullanarak geliştirdikleri algoritmayı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen sonuçlar sıralamasında birinci sırada yer almıştır. Yazarlar, iyi sonuçların iyi sonuçların yanında, kötü sonuçların kötü sonuçların yanında kümелendiğini gözlemlemişlerdir.

Alcaraz v diğ. (2004), '*Improving the performance of genetic algorithms for the RCPS problem*' başlıklı çalışmalarında, bir önceki çalışmalarda elde ettikleri

değerleri iyileştirmişlerdir. Alcaraz ve Maroto (2001), '*A Robust Genetic Algorithm for Resource Allocation in Project Scheduling*' başlıklı çalışmalarında, GA kullanarak geliştirdikleri algoritmayı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen sonuçlar sıralamasında birinci sırada yer almıştır. Çalışmada ileri geri yönde ve geri çözümler hesaplanmış ve her iki yönde yapılan çözümün oluşturabileceği avantajlardan faydalanılmıştır.

Alcaraz ve Maroto (2001), '*A Robust Genetic Algorithm for Resource Allocation in Project Scheduling*' çalışmalarında, GA kullanarak geliştirdikleri algoritmayı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen sonuçlar sıralamasında birinci sırada yer almıştır. Çalışmada ileri geri yönde ve geri çözümler hesaplanmış ve her iki yönde yapılan çözümün oluşturabileceği avantajlardan faydalanılmıştır.

Valls ve diğ. (2008), '*A hybrid genetic algorithm for the resource-constrained project scheduling problem*' başlıklı çalışmalarında Hybrid GA kullanarak geliştirdikleri algoritmayı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar sıralamasında, j30 problemlerinde ikinci, J60 ve J120 problemlerinde birinci sırada yer almıştır. Yazarlar kaliteli çözümlerin iyi yanlarını birleştiren 'peak crossover operator'ünü geliştirmişler ve ileri geri hesaplama yaparak daha iyi çözümler elde etmişlerdir. Yazarlar daha önceki '*Justification and RCPSP: A technique that pays*' başlıklı çalışmalarında elde ettikleri değerleri iyileştirmişlerdir (Valls ve diğ., 2005).

Cervantes ve diğ. (2008), '*A dynamic population steady-state genetic algorithm for the resource-constrained project scheduling problem*' başlıklı çalışmalarında geliştirdikleri GA'yı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar sıralamasında, j30 problemlerinde ikinci, J60 ve J120 problemlerinde birinci sırada yer almıştır. Yazarlar değişken uzunlukta popülasyon kullanmışlar, daha iyi çözümlerin baskın hale gelmesini sağlayan baskılama mekanizmaları geliştirmişler ve ileri geri hesaplama yaparak daha iyi çözümler elde etmişlerdir.

Mendes ve diğ. (2009), '*A random key based genetic algorithm for the resource constrained project scheduling problem*' başlıklı çalışmalarında geliştirdikleri GA'yı

PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar sıralamasında, j30 problemlerinde ikinci, J60 problemlerinde birinci ve J120 problemlerinde üçüncü sırada yer almıştır. Yazarlar, aktivitelerin öncelik değerlerinin tutulduğu listenin kuyruğuna, aktivitelerin gecikme zamanının tutulduğu ikinci bir listeyi eklemişler ve diğer çalışmalarda kullanılan kromozom uzunluğunun iki katı uzunlukta kromozom kullanmışlardır.

Wang ve diğ. (2010), '*Efficient genetic algorithm for resource-constrained project scheduling problem*' başlıklı çalışmalarında geliştirdikleri GA'yı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar sıralamasında, j30 problemlerinde ikinci, J60 problemlerinde birinci ve J120 problemlerinde üçüncü sırada yer almıştır. Yazarlar rastgele anahtar gösterimini kullanmışlar ve ileri geri hesaplama yaparak daha iyi çözümler elde etmişlerdir. Ayrıca paralel ve seri çizelgeleme tekniklerini birlikte kullanmışlar, aktivite önceliklerini gösteren kromozom listesinin sonuna ekledikleri bir gen ile çizelgeleme tekniğini belirtmişler, aynı anda başlayan aktiviteler için eşit öncelik değeri düzenlemesi yapmışlardır.

Gonçalves ve diğ. (2011), '*A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem*' başlıklı çalışmalarında geliştirdikleri GA'yı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar arasında yer almaktadır. Yazarlar rastgele anahtar gösterimini kullanmışlar ve ileri geri hesaplama yaparak daha iyi çözümler elde etmişlerdir.

Proon ve Jin (2011), '*A genetic algorithm with neighborhood search for the resource-constrained project scheduling problem*' başlıklı çalışmalarında geliştirdikleri GA'yı PSPLIB problemlerine uygulamışlardır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar sıralamasında, j30 problemlerinde yirmidördüncü, J60 problemlerinde birinci ve J120 problemlerinde birinci sırada yer almıştır. Yazarlar yüksek kaliteli yerel arama işlemleri yapan, komşuluk yapası tarama operatörü aracılığı ile daha iyi çözümler elde etmişlerdir.

Zamani (2013), '*A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem*' başlıklı çalışmasında geliştirdiği

GA'yı PSPLIB problemlerine uygulamıştır. Çalışmada elde edilen sonuçlar, daha önce başka çalışmalarda bildirilen en iyi sonuçlar sıralamasında, j30 problemlerinde altıncı, J60 problemlerinde onuncu ve J120 problemlerinde onbirinci sırada yer almıştır. Yazar, ileri geri hesaplamanın daha iyi çözümler elde etmeye katkıda bulunduğunu belirtmiştir.

KKPÇP çözümüne yönelik çalışmalar rekabetçi bir ortamda yoğun bir şekilde sürmektedir. Yazarlar, farklı teknikleri ve yeni düşünüş biçimlerini bir araya getirerek, test problemleri üzerinde sınımışlar ve aldıkları sonuçları karşılaştırmalı olarak bildirmişlerdir. Yapılan çalışmaların incelenmesinde elde edilen gözlemler aşağıda verilmektedir:

- Hiç bir meta-sezgisel algoritmanın iyi sonuçları tek başına garanti etmediği,
- Hibrit algoritmaların kullanımında artış olduğu,
- Komşuluk yapısı ve kaliteli yeni çözüm üretme alanlarına yoğunlaşıldığı,
- Paralel algoritmalar konusunda –güçlü iyileştirici kanıtlar olmasına rağmen– üzerinde çok az çalışıldığı sonucuna varılmıştır.

Literatür araştırması ve çözüm algoritmalarının incelenmesi sonucunda elde edilen bulgulara göre, aşağıda değinilen alanlara yönelik iyileştirme olanaklarının mümkün olduğu değerlendirilmiştir:

- KKPÇ alanında: tekrarsız kromozom gösterimi ile arama uzayının verimli kullanılması,
- GA alanında: Erken yakınsama ve baskılama sonucu oluşan çeşitlilik kaybının engellenmesi,
- GA alanında: PGA desteği ile çeşitlik artışı ve arama uzayının verimli kullanılması,
- GA alanında: Hibrit SA desteği ile yerel iyiden kaçış.

Şimdiye kadar bir arada kullanılmamış iyileştirme olanaklarının yansıtıldığı bir genetik algoritmanın, test problemleri üzerinde uygulanması sonucu elde edilebilecek başarımların değerlerinin, literatürde yer alan çalışmalar arasında üst sıralara yer alabilmesi olanağına yönelik olarak aşağıda verilen alt amaçlar üzerinde çalışılacaktır.

- Motivasyon başlığı altında ele alınan güçlendirme ve destek algoritmalarının geliştirilmesi,
- Geliştirilen algoritmanın, PSPLIB problemleri üzerinde sınanması.

Literatür taramasında, hem geliştirilecek güçlendirici algoritmalara hem de bunların PGA içinde kullanılarak KKPÇP sınamaya problemlerine uygulanmasına yönelik hiçbir çalışmaya rastlanamamıştır. Sınama sonuçlarının yeteri kadar iyi olmaması durumunda ise, gelecekte bu konu alanında yapılacak çalışmalarda verimsiz yolların farkına varılması bakımından araştırmacılara yol göstermek ve düzeltici geliştirmeler için zemin hazırlamak üzere bu çalışmadan yararlanılabileceği öngörülmüştür.

2.5 Bölüm Özeti

İkinci bölümde, KKPÇP konu alanında yapılmış olan literatür araştırmasının bulgularına yer verilmiştir. Literatür araştırması, sınıflandırma, çözüm yöntemleri, çizelge gösterim biçimleri ve KKPÇP çözümüne yönelik çalışmalar başlıkları altında ele alınmıştır. Bu bölümde, KKPÇP konusu ayrıntılı olarak incelenmiş ve dördüncü bölümde yer alan, hibrit paralel GA uygulaması konusuna hazırlık yapılmıştır.

KKPÇP, bir kombinatorial optimizasyon problemidir ve karar değişkenleri, kısıtlamalar ve amaç fonksiyonlar ile ifade edilir. Projeyi oluşturan aktivitelerin süreleri, kaynak ihtiyaçları, öncelik–sonralık ilişkileri ve kaynakların sınırları belirlenmiştir. Proje tamamlanma süresinin minimize edilmesi hedeflenir. KKPÇP, NP–Hard optimizasyon problemleri sınıfına girer ve aslında pratikte en zorlu klasik problemlerden biridir.

Çözüm yöntemleri, kesin yöntemler ve stokastik yöntemler olarak iki başlık altında incelenmektedir. Kesin yöntemler, arama uzayının dar olduğu küçük problemlerde en iyi sonuçlara ulaşabilirler. Arama uzayı genişlerse, çözüm süresi –pratikte kabul edilemeyecek kadar çok– uzar. Stokastik yöntemler bünyelerinde rastlantısallık barındırırlar ve arama uzayının geniş olduğu problemlere uygulanabilirler. Makul bir süre içerisinde yeteri kadar iyi bir sonuca ulaşabilirler.

KKPÇP birçok araştırmacı tarafından genel bir çerçeve içerisinde tanımlanmış, çözüm modelleri ve algoritmalar detaylandırılarak incelenmiştir. Bu bölümde: KKPÇP’ne yönelik çalışmalar incelenmiş, literatür araştırmasından edinilen

gözlemler ve bu gözlemleri dayanan algoritmik iyileştirme olanakları ele alınmıştır. Söz konusu algoritmik iyileştirmelerin taşıyıcısı olarak kullanılacak olan Genetik algoritmaların incelenmesi üçüncü bölümde, geliştirilen algoritma, uygulanması ve elde edilen sonuçlar dördüncü bölümde ele alınmaktadır.



3. GENETİK ALGORİTMAYA BAKIŞ

Bu bölümde genetik algoritma konusu, bir üst kavram olan meta-sezgisel optimizasyon konusundan başlanarak incelenmektedir. Genetik algoritma, meta-sezgisel optimizasyon, No Free Lunch teorisi, GA ve PGA başlıkları altında ayrıntılı olarak ele alınmakta ve dördüncü bölümde yer alan, hibrit paralel GA uygulaması konusuna hazırlık yapılmaktadır.

Pratikte, NP-Hard yapısı nedeniyle saf matematiksel programlama teknikleri ile çözülemeyen KKPÇP, makul bir sürede, kesin olmayan fakat yeteri kadar iyi sonuçlar verebilen meta-sezgisel algoritmalar üzerinden çözülebilmektedir.

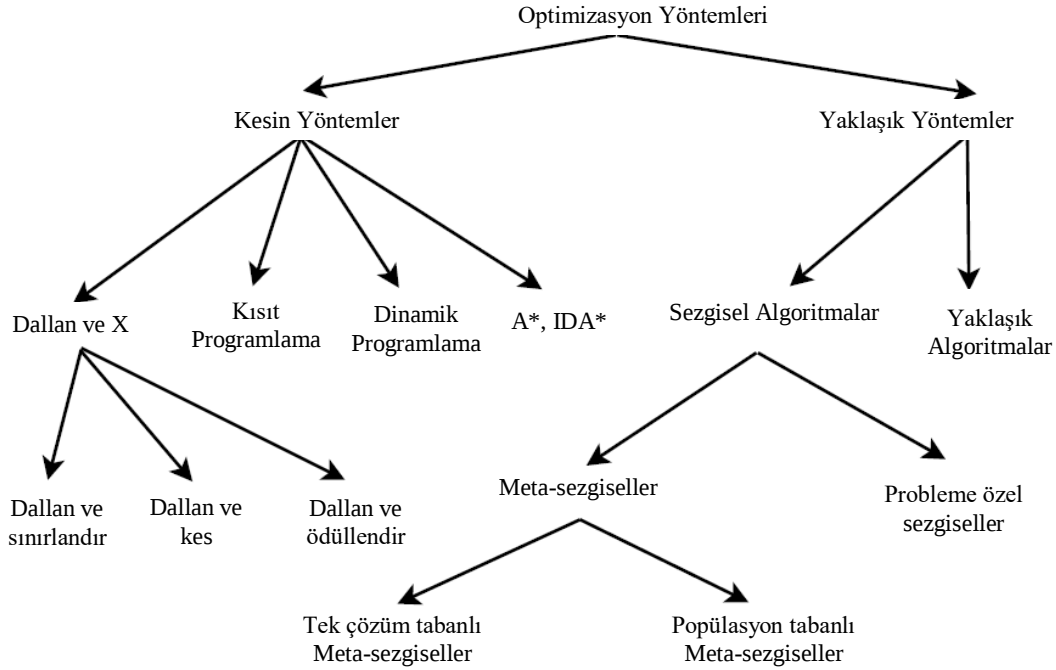
3.1 Meta-sezgisel Optimizasyon

Algoritma, kabaca belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yol olarak tanımlanabilir. Matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir.

Sezgisel sözcüğü kökeni eski Yunanca bir kelime olan “*heuriskein*”dir ve problemleri çözmek için yeni yollar bulmak anlamında kullanılmaktadır. Archimedes tarafından (*heurika*), “*buldum*” anlamında kullanılmıştır. Meta öneki ise Yunanca’da, “*üst düzey metodoloji*” anlamına gelir. Meta-sezgisel algoritmalar, optimizasyon problemlerin çözümünde sezgisel algoritmalarla yol gösterici üst düzey genel arama stratejileri olarak tanımlanabilir. Meta-sezgisel algoritmaların optimizasyon evrenindeki yeri, Talbi (2009) tarafından yapılan sınıflandırmayla Şekil 3.1’de verilmiştir.

Meta-sezgisel algoritmalar, problemin herhangi bir özelliğinden yararlanmayan, problem bağımsız tekniklerdir ve makul hesaplama zamanlarında iyi kalitede çözümler elde etmeyi amaçlamaktadırlar (Beheshti ve Shamsuddin, 2013). Meta-sezgisel algoritmalar, genellikle büyük boyutlara ve çok karmaşık

kısıtlamalara sahip pratik problemler için oldukça uygundur (Das ve Acharyya, 2011; Kolisch ve Hartmann, 1999).



Şekil 3.1 : Optimizasyon metotları (Talbi, 2009).

Çözüm yolları bilinen problemlerde meta-sezgisel algoritmaların kullanımı verimli değildir. Eğer arama uzayı çok büyük, problemin karmaşıklığı yüksek, problemin matematiksel olarak modellenmesi ve çözümü çok zor ise meta-sezgisel algoritmalar kullanılabilir. Saf meta-sezgisel algoritmalar probleminden bağımsız olarak kullanıldıklarında oldukça iyi sonuçlar üretebilir veya yetersiz kalabilirler. Probleme özel komşuluk yapıları, arama uzayı kısıtlamaları ve yol gösterici kurallar üzerinde özenle çalışılmalı ve erişilebilen her türlü iyileştirmeden faydalanılmalıdır. Özellikle lineer eşitliklerin, kısıtlamaların veya basitçe çözülebilen non-lineer denklemlerin çözümü model içerisinde yapılmalı, bu tip problemlerin çözümleri meta-sezgisel algoritmalarla bırakılmamalıdır. Çözüm tasarımı mümkün ise, olurlu alanlarda arama yapmak üzerine kurulmalıdır. Olursuz alanlar için ceza fonksiyonları kullanmak mümkün olsa da, son seçenek olarak değerlendirilmelidir. Bu durumda ceza fonksiyonunun değeri, olurlu bölgeye doğru gidişi işaret edebilir şekilde tasarlanmalıdır. Mümkün olan her durumda kesin çözüm algoritmalarının gücü kullanılmalı, meta-sezgisel algoritmalar arama uzayında verimli bir şekilde dolaşmak ve yeni çözümler üretmek için kullanılmalıdır.

Meta-sezgisel algoritmalar karşıt kavramlar üzerinden de değerlendirilebilirler (Talbi, 2009):

- Doğa esinli ve doğa esinli olmayan,
- Bellek kullanan ve bellek kullanmayan,
- Deterministik ve stokastik,
- Popülasyon tabanlı çözüm arayan ve tek çözüm arayan,
- Yinelemeli arama ve açgözlü arama.

Doğa esinli ve doğa esinli olmayan:

Meta-sezgisel optimizasyon algoritmaların birçoğu doğal süreçlerden esinlenmektedir. Evrimsel algoritmalar ve yapay bağışıklık sistemleri biyolojiden, ACO ve PSO sosyal sürü zekâsından ve SA fiziksel süreçlerden esinlenerek geliştirilmiştir.

Bellek kullanan ve bellek kullanmayan: Bazı meta-sezgisel algoritmalar belleksizdir. Önceki adımların bilgisi, mevcut tek çözümün içindedir. SA belleksiz meta-sezgisel algoritmalarla örnek olarak gösterilebilir. Tek çözüm üzerinde çalışılır ve sürekli daha iyi aranır. Bazı meta-sezgisel algoritmalar, arama sırasında elde edilen bilgileri, ilerleyen adımlarda kullanmak üzere bellekte saklamaktadırlar. TS algoritması önceden ziyaret edilen yerlere yeniden gidişi yasaklayan listeyi bellekte tutar ve yeni hareketlerde bu listeyi engelleyici olarak kullanır.

Deterministik ve stokastik: Deterministik algoritmalar, her çalıştırmada, aynı başlangıç koşullarında, aynı yolları izleyerek, aynı sonucu üretirler. TS ve tepe tırmanma algoritması deterministik algoritmalarla örnek gösterilebilir. Meta-sezgisel algoritmalar rastgelelik barındırır ve her seferinde farklı bir patikadan ilerleyerek değişik sonuçlara ulaşabilirler. SA ve GA stokastik algoritmalarla örnek gösterilebilir.

Popülasyon tabanlı çözüm arayan ve tek çözüm arayan: GA ve PSO gibi popülasyon tabanlı algoritmalar bütün popülasyonu geliştirerek ilerlerler. Tepe tırmanma ve SA gibi tek çözüm tabanlı algoritmalar ise tek çözüm üzerinde çalışırlar ve bu çözümü geliştirirler. Popülasyon tabanlı algoritmalar arama uzayında çeşitlendirme ve yayılma yaklaşımı ile araştırma yaparken, tek çözüm arayan algoritmalar yoğunlaşma ve yakın komşuluk taraması yaklaşımını kullanırlar. Bu iki

tip algoritmanın bir arada kullanılması daha iyi çözümler üretilmesine yol açabilir. Böylece yayılma, çeşitlendirme ve yoğunlaşma işlemleri daha etkili kullanılabilir.

Yinelemeli arama ve açgözlü arama: Yinelemeli algoritmalarda her yinelemede elde edilen çıkış bilgileri, bir sonraki yinelemenin giriş bilgilerini etkiler. Her yinelemede çözüm iyileşebilir veya daha iyi bir çözüme götürebilecek patikalara geçilebilir. Bazı durumlarda platolarda veya bir yerel iyide takılma olabilir. Algoritma buradan kurtulmak için uygun olan mekanizmaları kullanır ve yineleme yeni bölgeden devam eder. Açgözlü algoritmalar, çözüm uzayını sıralar ve bu sıraya göre çözümü araştır. Sıralama, geriye dönmeden tüm arama uzayını taramayı garantiler. Daraltılmış açgözlü algoritmalar ise bir başlangıç noktasından hareketle –o durumda– en iyi görünen hareketi gerçekleştirir. Her durumda görünen en iyiyi seçmek, problemin en iyi çözümünden çok farklı bir yerde bitebilir. Tarama ufkunu genişletmek veya görünen en iyiden vazgeçebilmek, daha iyi çözümlere ulaşmak için farklı patikalar seçilmesine yardımcı olabilir.

3.2 No Free Lunch Teorisi

‘No Free Lunch’, on dokuzuncu yüzyılda salonların sunduğu “*ücretsiz öğle yemeği*” servisine uygulanan sözün gerçek örneklerinden evrimleşmiştir. On dokuzuncu ve yirminci yüzyılın başlarında, Amerika Birleşik Devletleri’ndeki birçok salon, en az bir içki satın alan müşterilere, öğlen yemeği büfesi seçimi sunmuştur. “*Ücretsiz öğle yemeği*”, salon sahipleri tarafından, müşteri sayısını, satın alınan içki miktarını ve toplam satışı artırmak için bir çekici öğe olarak kullanılmıştır. NFL, “*The Moon Is a Harsh Mistress*” adlı romanda yer almaktadır (Heinlein, 1966). “*There’s No Such Thing as a Free Lunch*”, Milton Friedman tarafından kendi yazdığı bir kitabın adı olarak kullanılmıştır (Friedman, 1975). NFL adlandırmasının tarihsel gelişimi ve kullanımı Quoteinvestigator.com (2018) tarafından verilmektedir.

Sistemler, içsel çalışması hakkında gözlemci tarafından sahip olunabilen bilgi açısından üç başlık altında incelenmektedirler:

- Kara kutu (Black Box) sistemler,
- Gri kutu (Grey Box) sistemler,
- Beyaz kutu (White Box) sistemler.

Kara kutu sistemlerin içsel çalışması hakkında hiçbir bilgi yoktur ve sistem sadece girdileri ve çıktıları açısından görülebilir durumdadır. Beyaz kutu sistemlerin ise içsel çalışmaları tam olarak bilinmektedir. Kara kutu sistemlerle beyaz kutu sistemler arasında farklı tonlarda yer alabilen gri sistemlerin içsel çalışmaları hakkında kısmi bilgi mevcuttur (Kuday, 2014).

Çözüm tekniği hakkında hiçbir bilgi olmayan veya çok az bilgi olan optimizasyon problemlerinin çözümünde kullanılan algoritmalar, kara kutu optimizasyon algoritmaları olarak adlandırılmaktadır. Girdi değerlerine karşılık çıktı değerlerini üreten amaç fonksiyon, optimizasyon probleminin kara kutusudur. Çözüm tekniği hakkında tam bilgi olan optimizasyon problemlerinin çözümünde kullanılan algoritmalar ise beyaz kutu optimizasyon algoritmaları olarak adlandırılmaktadır. Eğer bir kara kutu optimizasyon algoritmasında, problem alanının bazı parçaları beyaz kutu optimizasyon algoritmaları tarafından çözülebiliyorsa, bu algoritma gri optimizasyon algoritması olarak adlandırılmaktadır. Çoğunlukla meta-sezgisel optimizasyon algoritmalarının amaç fonksiyonu bir kara kutudur (Brownlee, 2011).

Wolpert ve Macready (1997) tarafından ele alınan '*No Free Lunch*', Teorisi (NFL) (Bedava Yemek Yok Teoremi, *No-Free-Lunch Theorem*), tüm kara kutu (*black box*) optimizasyon algoritmalarının, bir amaç fonksiyonun uç noktalarını araştırmak için katlandığı maliyetin, bütün olası problemler üzerinde ortalama alındığında aynı olduğunu öne sürmektedir (Brownlee, 2011). NFL teorisi, özellikle meta-sezgisel ve hesaplamalı zekâ algoritmalarının değerlendirilmesi ve karşılaştırılması ile ilgili olarak çok fazla karamsarlığa ve yanlış anlamalara neden olmuştur. NFL teorisinin anlamı, genel amaçlı siyah kutu optimizasyon algoritmalarından en iyisini bulmak için yapılan araştırmanın, teorik olarak –böyle bir sonucun mümkün olmaması yüzünden–, anlamsız olmasıdır. NFL teorisi zaman içinde, arama stratejilerini öğrenen ve ayarlayan algoritmaların yanı sıra, stokastik ve deterministik optimizasyon algoritmalarına da uygulanmıştır. Kullanılan performans ölçüsünden ve seçilen temsilden bağımsızdır. Wolpert ve Macready'nin orijinal makalesi, algoritmaların temsil veya iletişim üstünlüğü gibi genellemelerin çokça yapıldığı bir zamanda üretilmiştir. Teorinin pratik etkisi, uygulamacıların arama ve optimizasyon algoritmaları için –diğerlerine göre– yüksek performans iddialarını sınırlamaya yönlendirmektir. Wolpert ve Macready, pratik problem sınıflarına yönelik uygun algoritmaların geliştirilmesine yönelik çabanın yoğunlaştırılmasını desteklemektedir

(Brownlee, 2011). Tüm problemler için üstün performans gösteren evrensel bir arama algoritması olmadığından, geliştirilen meta-sezgisel optimizasyon algoritmaları seçilen problem sınıflarına yönelik olmalıdır (Igel, 2014).

NFL teoremi, hiçbir optimizasyon algoritmasının, bütün problemlere uygulandığında, diğer algoritmalarından daha iyi performans gösteremeyeceğini belirtmektedir. Diğer yandan, yeni yapılan çalışmalarda, Free Lunch'un var olduğu ve optimizasyon algoritmalarının, problem sınıflarına göre eşleştirildiğinde diğerlerinden daha iyi performans gösterebileceği bildirilmiştir (Joyce & Herrmann, 2018).

3.3 Genetik Algoritma

3.3.1 Genetik algoritmaya giriş

Evrimsel algoritmalar, adaptasyonun özelliklerini –deneme ve yanılma yoluyla yararlı çeşitlilikleri çoğaltarak bir araya getiren– yinelemeli bir süreçte ortaya koyarlar. Aday çözümler, –belirli bir amaç fonksiyon tarafından tanımlanan bir ortamda hayatta kalmak için uyum sağlamaya çalışan– sanal bir popülasyonun üyelerini temsil eder. Evrimsel süreç, genetik çaprazlama ve mutasyon gibi evrim mekanizmalarını kullanarak, aday çözümlerinin –amaç başarımına bağlı olarak biçimlenen– popülasyonuna uyumunu geliştirir. Evrim teorisi hakkında yüksek kaliteli birçok kaynak bulunabilir. Birinci sırada Darwin'in (1859) orijinal kaynağı ele alınabilir. Huxley'in kitabı (1942) Darwin'in doğal seleksiyonu ve Mendel'in genetik mekanizmalarını birleştiren evrimsel biyolojinin modern sentezi olarak görülebilir. Futuyma ve Kirkpatrick'in (2017) “*Evolution*” adlı çalışmaları evrim ile ilgili iyi bir ders kitabı olarak değerlendirilmektedir. Evrim üzerinde gen merkezli bir bakış açısı sunan “*The Selfish Gene*” (Dawkins, 1976) ve sürecin algoritmik özelliklerini ele alan “*Darwin's Dangerous Idea*” (Dennett, 1995) gibi çalışmalar bu alandaki popüler bilim kitapları arasında yer almaktadır.

Goldberg'in (1989) klasik çalışması GA için değerli bir kaynaktır. Holland'ın çalışması (1975), GA haline gelen uyarlamalı sistemler hakkında bilgi edinmek için kullanılabilir. Koza ve diğ., (2005) tarafından yapılan çalışma Genetik Programlama alanında önemli bir kaynaktır. Evrimsel hesaplama stratejiler ve algoritmalar hakkında Schwefel (1981) ve De Jong'dan (2006) faydalanılabilir (Brownlee, 2011).

GA, taksonomi, esinlenme, benzetim ve strateji alt başlıklarında incelenecektir.

Taksonomi: GA –uyum sağlama yaklaşımı üzerine kurulu– bir Evrimsel Algoritma ve Global Optimizasyon tekniğidir. GA, Evrimsel Hesaplama’nın altında, Genetik Programlama, Evrim Stratejileri ve Evrimsel Programlama gibi diğer Evrimsel Algoritmalar ile birlikte yer alır.

Esinlenme: GA, aşağıda verilen kavramlardan esinlenerek geliştirilmiştir:

- Popülasyon kaynaklı : evrim, kalıtım, gen sıklıkları
- Birey Kaynaklı : kromozomlar, genler
- Mekanizma kaynaklı : çaprazlama, mutasyon

Evrimsel biyolojinin bu yapay sentezi, popülasyon açısından Darwin’in, birey açısından Mendel’in görüşlerini yansıtmaktadır.

Benzetim: Toplumu oluşturan bireyler, kendi genetik varlıklarını, içinde bulundukları ortama uyum değerleri ile orantılı olarak yayarlar. Sonraki nesil, iki bireyin genetik varlıklarının, –mutasyon olarak adlandırılan küçük kopyalama hatalarını da içerecek şekilde– birbirine bağlanmasıyla ortaya çıkar.

Strateji: GA’nın amacı, aday çözümlerinin, bir amaç fonksiyon tarafından ölçülen kazanımlarını maksimize etmektir. Genetik algoritma stratejisi, aday çözümlerin oluşturduğu topluluk üzerinde çaprazlama ve mutasyon olarak adlandırılan genetik mekanizmaları sürdürmektir. Bir aday üzerinde ortaya çıkan değişiklik, bu adayın bir sonraki halinin çözüme yapacağı katkıyı yanlı olarak yönlendirir.

Çoğunlukla bir optimizasyon tekniği olarak kullanılan GA, aynı zamanda genel amaçlı bir adaptif bir strateji olarak kullanılabilir (Brownlee, 2011). GA’nın belli başlı özellikleri aşağıda verilmektedir:

- GA, parametrelerin kendisiyle değil, doğrudan parametre kodlarıyla uğraşır.
- GA tek noktadan değil, birçok noktadan arama yapar.
- GA sadece amaç fonksiyonundan elde edilen bilgiyi kullanır.
- GA’nın uygulamasında kullanılan operatörler rastlantısal yöntemlere dayanır.

GA'nın Avantajları:

GA birçok çözüm adayının oluşturulması ve geliştirilmesi yoluyla çalışır. Böylece çözüm uzayının farklı bölgeleri eşanlı olarak taranır. Arama işlemi, çözüm adaylarından biri yerel iyiye takılsa bile, diğer çözüm adayları üzerinden devam eder. Farklı çözüm adayların üreme işleminden, daha iyi sonuçlara ulaşabilecek yeni bir çözüm adayı oluşabilir.

GA problemden bağımsızdır. Bir çok optimizasyon problemine kolayca uygulanabilir. Hibrit kullanıma olanak tanır ve paralel çalışabilir. GA operatörleri, probleme göre değiştirilebilir ve geliştirilebilirler. GA'nın çözüm ilerlemesi kolayca izlenebilir.

GA amaç fonksiyon değerine göre çözüm üretir, türev ve benzeri yol göstericilere ihtiyaç duymaz. Mevcut kısıtlamalar, doğrudan amaç fonksiyona yansıtılabilir. Olurlu olmayan çözüm bölgelerinden kaçış, amaç fonksiyona eklenen ceza mekanizmasıyla düzenlenebilir.

Her biri çözüm adayı olan kromozomlar ikili, tamsayı, ayrık, reel ve sembolik olarak kodlanabilirler. Kodlama, amaç fonksiyon tarafından çözülür ve uygunluk değeri olarak hesaplanır. Böylece probleme özel birçok gösterim biçimi hazırlanabilir. Yeni ve yaratıcı çözüm yollarının GA'ya yansıtılması mümkündür.

GA çok amaç fonksiyonlu optimizasyon problemlerine başarıyla uygulanabilir. Her amaç fonksiyon ayrıca hesaplanabilir ve uygun bir değerlendirme mekanizmasıyla izlenebilir. Problemin alt problemlere indirgenebilmesi durumunda, her alt problem ayrı ayrı ve eşanlı çözülebilir.

GA'nın çok noktalı başlangıç ve ilerleme mekanizması sayesinde, çözüm sırasında oluşabilecek gürültünün sönmümlendirilmesi mümkün olabilir. GA amaç fonksiyon gürültüsüne ve başlangıç çözümlerindeki yetersizliklere tolerans gösterebilir.

GA çok değişkenli, çok kısıtlı ve çok amaç fonksiyonlu optimizasyon problemlerine kolaylıkla uygulanabilir.

GA'nın Dezavantajları:

GA en iyi çözüme ulaşmayı garanti edemez. Elde edilen sonuç, yerel iyilerden biri olabilir. GA, en iyi değer, arama uzayının ipucu vermeyen bir bölgesinde saklı

kaldığı ve problem sınırlarının çok sıkışık olduğu durumlarda, herhangi bir sonuca ulaşamayabilir.

GA rastlantısallık içerir ve bu yüzden her çalıştırmada farklı patikalardan ilerleyerek farklı sonuçlara ulaşabilir. Bir çalıştırmada yerel iyilerden birine takılabilir, başka bir çalıştırmada en iyi değeri bulabilir. İlerleme sırasında elde edilen aday çözümlerden birinin hemen yanı başında olan en iyi çözümü görmezden gelebilir ve bir daha aynı arama alanına dönmeyebilir.

GA, popülasyon büyüklüğü, çaprazlama oranı ve mutasyon oranı parametrelerinin uygunsuz seçimi, operatörlerin verimsiz düzenlenmesi veya problemin yetersiz tasarımı gibi nedenler yüzünden verimsiz çalışabilir.

GA, problemin kabul edilebilir sürede çözüm veren yollarının veya problem alt parçalarının karşılıklı ilişkilerinin göz ardı edildiği durumlarda, verimsiz çalışıyor olarak gözlemlenebilir.

3.3.2 Genetik aloritmaya ilişkin temel kavramlar

GA'ya ilişkin genel kavramlar şunlardır:

Kromozom: GA içerisinde çözüm adayını temsil eden bir diziliş olarak kullanılır. Her kromozom, bir aday çözümün kodlanmış halidir. Kromozom, aday çözüm hazırlama işleminin tasarımına göre, olurlu veya olursuz olarak elde edilebilir.

Gen: Kromozomu oluşturan bilgi blokları olarak tanımlanabilir. Bir kromozom, temsil şemasına göre bir araya gelmiş genlerden oluşur.

Popülasyon: Kromozomlar topluluğu olarak tarif edilebilir. Klasik GA operatörlerinin kullanıldığı optimizasyon problemlerinde, popülasyondaki kromozom sayısı en az iki olabilir. Popülasyon büyüdükçe: çözüm süresi, çeşitlilik ve en iyi çözüme yaklaşma ihtimali artar, popülasyon küçüldükçe çözüm süresi azalır, çeşitlilik azalır ve yerel iyiye takılma ihtimali artar.

Jenerasyon: GA'nın her yeni nesil üretme döngüsü yeni bir nesli temsil eder.

Amaç fonksiyon: Kromozomda bulunan aday çözümün hesaplanan sonucu olarak tarif edilebilir. Optimizasyon sürecinde amaç fonksiyonu en iyileyen kromozom veya kromozomlar aranır.

Uygunluk fonksiyonu: Amaç fonksiyonun ölçeklendirilmiş hali olarak tanımlanabilir.

Elitist seçim: Popülasyonda yer alan en iyi çözümlerin, yeni nesil üretme işlemi sırasında kaybolmaması ve üreme işlemi sonrasında elde edilen yeni popülasyonda yer alması amacıyla yapılan işlemler olarak tarif edilebilir. Elitist seçim, elde edilen en iyi çözümün kaybedilmeden saklanması veya çözümün ilerleyişinin eldeki en iyi çözümler yönünde baskılanması amaçları için kullanılabilir.

Çaprazlama: İki çözüm adayı kromozomdan yeni bir kromozom oluşturulması işlemidir. Matematiksel olarak ikiden daha çok kromozom kullanılarak yeni bir kromozom üretilir.

Mutasyon: Kromozom üzerinde rastgele oluşan bir değişim olarak tarif edilebilir. Yerel iyiye takılmayı veya popülasyonun aynılaşmasını önleyici bir etki yapması beklenir.

Yerleştirme: Üreme sonrasında elde edilen kromozomların popülasyona aktarılması ve mevcut kromozomların bir kısmının popülasyondan uzaklaştırılması işlemi olarak tarif edilebilir.

3.3.3 Genetik algoritma operatörleri

GA operatörleri, popülasyonun üreme yoluyla yenilenmesi işlemi gerçekleştirirler. GA yeni bir nesile ulaşmak için: seçim, çaprazlama ve mutasyon operatörlerini kullanır.

Seçim operatörü: Çiftleşecek kromozomların belirlenmesi işlemi yerine getirir. Seçim operatörü, aşağıdaki sorulara yol gösterici olarak tasarlanır ve kullanılır:

- Popülasyonun ne kadarı çiftleşebilir?
- Hangi kromozomlar çiftleşebilir?
- Hangi kromozomlar, ne kadar ağırlıkla baskın eş olabilir?
- Hangi kromozomlar değişmeden korunarak yeni nesile aktarılabilir?

Sık kullanılan seçim operatörleri: Roulette Wheel Selection, Random Selection, Rank Selection, Tournament Selection, Boltzmann Selection, Stochastic Universal Sampling olarak verilebilir (Sivanandam ve Deepa, 2008). Elitist seçim işlemi,

değişmeden korunarak yeni nesile aktarılabilecek kromozomların belirlenmesinde kullanılabilir.

Problemin çözümüne yönelik yeni veya karma operatörler geliştirilebilir.

Çaprazlama operatörü: Seçilen kromozomlardan yeni bir kromozom elde edilmesi işleminde kullanılır. Çaprazlama operatörü, aşağıdaki soruya yol gösterici olarak tasarlanır ve kullanılır:

Yeni kromozomda yer alacak olan değerler, diğer kromozomlara bakarak nasıl üretilcek?

Yeni kromozomun üretilmesi, her gen üzerinde veya peş peşe gelen genlerden oluşan gen blokları üzerinde yapılan bir işlem olabilir. Kaynak genlerin pozisyonları bir kural setine göre veya rastgele seçilebilir. Yeni gen, kaynak genlerden birisinin kopyası veya kaynak genlerin işleme tabi tutulması sonucunda elde edilen bir değer olabilir. Ayrıca yeni kromozomun olurlu olması için işlemler uygulanabilir. Çaprazlama operatörü, aşağıdaki kriterlere farklı ağırlıklar verilerek birçok şekilde geliştirilebilir:

- Kaynak genlerin pozisyonlarını belirlenmesi,
- Yeni genin içeriğinin ve pozisyonunun belirlenmesi,
- Olurluluğun sağlanması,
- Çaprazlama işleminin yoğunluğu.

Sık kullanılan çaprazlama operatörleri: Single Point Crossover, Two Point Crossover, Multi-Point Crossover, Uniform Crossover, Three Parent Crossover, Shuffle Crossover, Precedence Preservative Crossover, Ordered Crossover, Partially Matched Crossover olarak verilebilir (Sivanandam ve Deepa, 2008).

Mutasyon operatörü: Değişikliğe uğrayacak kromozomların ve genlerin belirlenmesi ve değiştirilmesi işlemini yerine getirir. Mutasyon operatörü, aşağıdaki sorulara yol gösterici olarak tasarlanır ve kullanılır:

- Hangi koşullar oluşunca mutasyon yapılacak?
- Hangi kromozomlar mutasyona uğrayacak?
- Seçilen kromozom üzerinde hangi genler mutasyona uğrayacak?

- Mutasyon işlemi nasıl yapılacaktır?

Sık kullanılan mutasyon operatörleri: Flipping, Interchanging, Reversing, Random Replacement olarak verilebilir (Sivanandam ve Deepa, 2008). Ayrıca yeni kromozomun olurlu olması için işlemler uygulanabilir. Mutasyon operatörü, çeşitliliği artıracak ve yerel iyiden kaçışı sağlayacak mekanizmalarla birlikte değerlendirilerek birçok şekilde geliştirilebilir. GA operatörleri bağımsız veya iç içe geçmiş şekilde geliştirilebilirler.

3.3.4 Genetik algoritmanın işleyişi

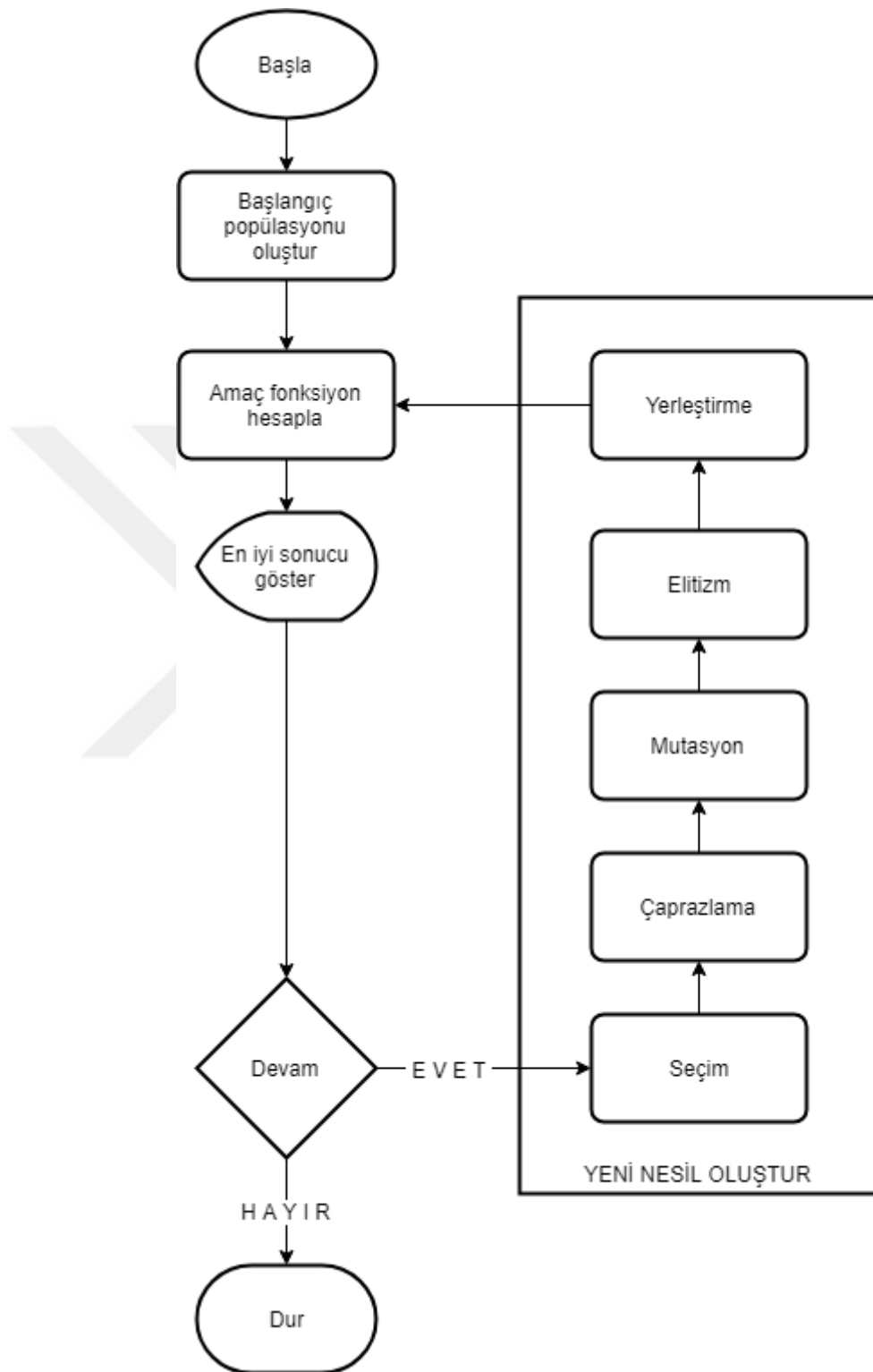
GA tek çözümün iyileştirilmesine yönelik çalışan yerel arama stratejilerinden farklı olarak, çözüm topluluğunun iyileştirilmesine yönelik olarak çalışır. Bir başlangıç popülasyonu oluşturduktan sonra, yeni nesil üretmeye ve elde edilen sonuçları değerlendirmeye yönelik operatörleri peş peşe yürütür. Yeni nesil üretildikten sonra, en uygun çözümler bir sonraki nesli oluşturmak için tutulurken, diğer uçta kalan kötü çözümler elenirler. Uygunluk değeri, çözümün kalitesini ölçmek için kullanılır ve genellikle çözülecek optimizasyon probleminin amaç fonksiyonu üzerinden hesaplanır. GA akış şeması Şekil 3.2’de görülmektedir.

3.4 Paralel Genetik Algoritma

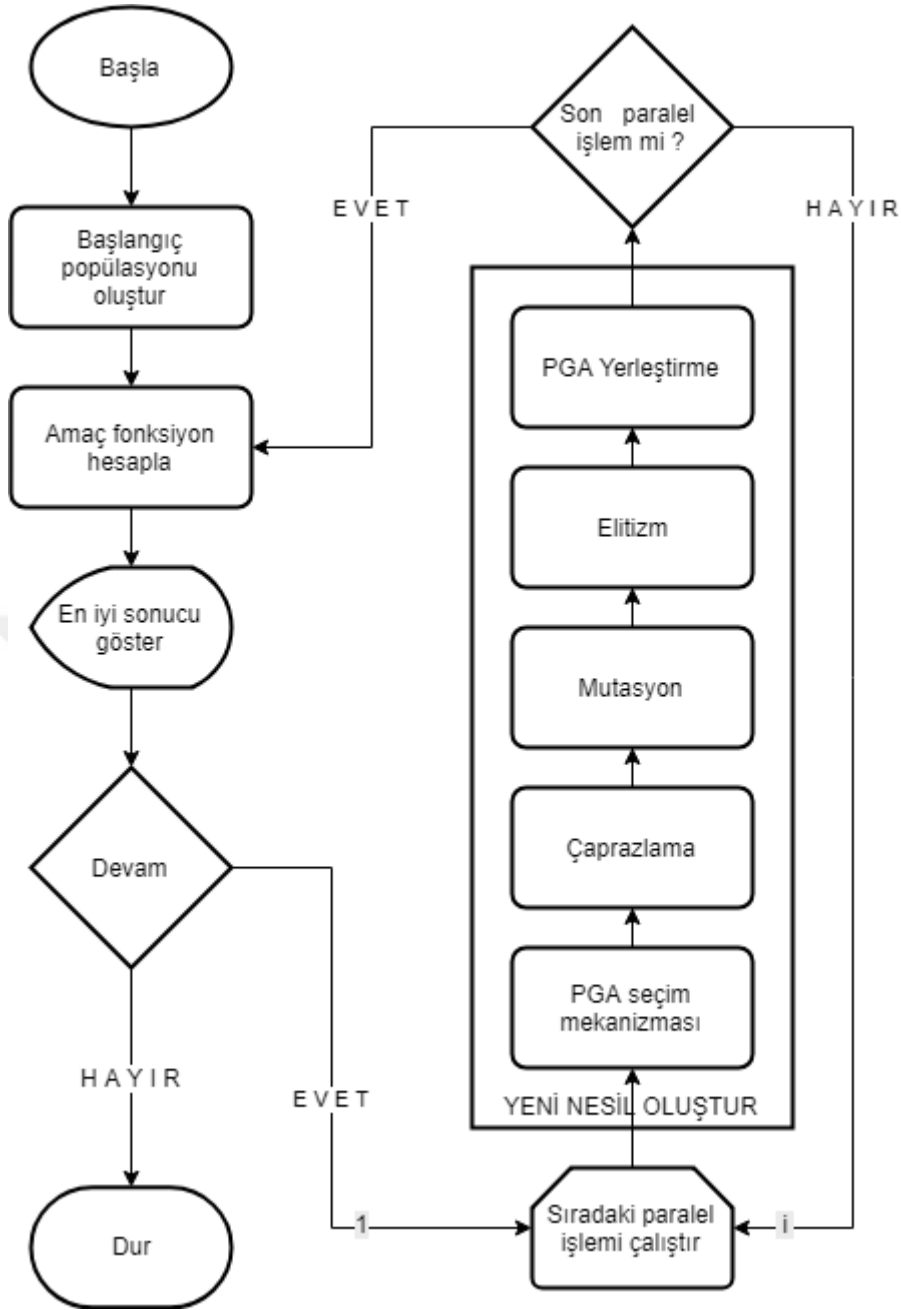
PGA, GA’nın geniş arama uzaylarında uzun süren çözüm süresini kısaltmak için kullanılır ve hem performans hem de çözüm kalitesi açısından kazanımlar sağlamaktadır (Sivanandam ve Deepa, 2008). PGA’nın avantajları aşağıda verilmektedir (Sumathi ve diğ., 2008):

- Kolay kodlanabilme,
- Kolay paralelleşme,
- Problemden bağımsızlık,
- Alternatif çözümler üretebilme,
- GA’lardan daha yüksek etkinlik,
- Diğer arama algoritmaları ile kolay işbirliği,
- Uzayda birden fazla noktadan paralel arama,

- Tek ortamda ya da çoklu ortamda çalışabilme.



Şekil 3.2 : GA Akış Şeması.



Şekil 3.3 : PGA akış şeması.

GA ile çözülen bir problem, aynı bellek büyüklüğünü koruyacak şekilde parçalara bölünebilir ve bu bellek bölgelerinin her biri üzerinde GA'lar çalıştırılabilir. Başlangıç popülasyonu seksen kromozom olan GA, parçalara ayrılarak dört kromozomlu yirmi GA haline dönüştürülebilir. GA'ların aynı anda çalıştırılması durumunda ortaya çıkan yapı PGA olarak adlandırılır. Bu durumda her bir GA hızlıca kendi yerel iyi değerine yakınsar. Bu yaklaşım, paralel çalışan tepe tırmanma algoritmalarının davranışına benzetilebilir. Paralel çalışan tepe tırmanma algoritmaları, kendi aralarında bilgi paylaşımında bulunamazken, PGA'ya ait alt

GA'lar seçilen geçişkenlik mekanizmasına bağlı olarak çözüm paylaşımında bulunabilirler. Böylece hem çeşitlendirme hem de yoğunlaşma süreçleri dengeli bir biçimde zenginleşir. PGA'lar, değinilen mekanizmaların katkısı ile GA'lara göre daha hızlı ve kaliteli çözümler üretebilirler. PGA'lar bir bütün halinde yayılma ve çeşitliliği korurken, alt GA'lar tekil olarak da yoğunlaşmacı bir yol izlerler. PGA içsel hibrit olarak adlandırılabilir. PGA akış şeması Şekil 3.3'te verilmiştir.

3.5 Bölüm Özeti

Üçüncü bölümde GA incelenmektedir. GA, meta-sezgisel optimizasyon, No Free Lunch teorisi, GA ve PGA başlıkları altında ele alınmıştır ve dördüncü bölümde yer alan, hibrit paralel GA uygulaması konusuna hazırlık yapılmıştır.

Algoritma, kabaca belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yol olarak tanımlanabilir. Matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir.

Meta-sezgisel algoritmalar, optimizasyon problemlerin çözümünde sezgisel algoritmalarla yol gösterici üst düzey genel arama stratejileri olarak tanımlanabilir. Meta-sezgisel algoritmalar, problemin herhangi bir özelliğinden yararlanmayan, problem bağımsız tekniklerdir ve makul hesaplama zamanlarında iyi kalitede çözümler elde etmeyi amaçlar. Çözüm yolları bilinen problemlerde meta-sezgisel algoritmaların kullanımı verimli değildir. Eğer, arama uzayı çok büyük, problemin karmaşıklığı yüksek, problemin matematiksel olarak modellenmesi ve çözümü çok zor ise, meta-sezgisel algoritmalar kullanılabilir. Mümkünse tasarım, olurlu alanlarda arama yapmak üzerine kurulmalıdır. Mümkün olan her durumda kesin çözüm algoritmalarının gücü kullanılmalı, meta-sezgisel algoritmalar arama uzayında verimli bir şekilde dolaşmak ve yeni çözümler üretmek için kullanılmalıdır.

NFL teoremi, hiçbir optimizasyon algoritmasının, bütün problemlere uygulandığında, diğer algoritmalarından daha iyi performans gösteremeyeceğini belirtmektedir. Diğer yandan, yeni yapılan çalışmalarda, Free Lunch'un var olduğu ve optimizasyon algoritmalarının, problem sınıflarına göre eşleştirildiğinde diğerlerinden daha iyi performans gösterebileceği bildirilmiştir.

GA, tek çözümün iyileştirilmesine yönelik çalışan yerel arama stratejilerinden farklı olarak, çözüm topluluğunun iyileştirilmesine yönelik olarak çalışır. Bir başlangıç popülasyonu oluşturduktan sonra, yeni nesil üretmeye ve elde edilen sonuçları değerlendirmeye yönelik operatörleri peş peşe yürütür. Yeni nesil üretildikten sonra, en uygun çözümler bir sonraki nesli oluşturmak için tutulurken, diğer uçta kalan kötü çözümler elenirler. Uygunluk değeri, çözümün kalitesini ölçmek için kullanılır ve genellikle çözülecek optimizasyon probleminin amaç fonksiyonu üzerinden hesaplanır.

PGA, GA'nın geniş arama uzaylarında uzun süren çözüm süresini kısaltmak için kullanılır ve hem performans hem de çözüm kalitesi açısından kazanımlar sağlamaktadır. Paralel çalışan tepe tırmanma algoritmaları, kendi aralarında bilgi paylaşımında bulunamazken, PGA'ya ait alt GA'lar seçilen geçişgenlik mekanizmasına bağlı olarak çözüm paylaşımında bulunabilirler. Böylece hem çeşitlendirme hem de yoğunlaşma süreçleri dengeli bir biçimde zenginleşir.

İkinci bölümde, KKPCP'ne yönelik çalışmalar incelenmiş, literatür araştırmasından edinilen gözlemler ve bu gözlemleri dayanan algoritmik iyileştirme olanakları ele alınmıştır. Bu bölümde, ikinci bölümde belirlenen algoritmik iyileştirmelerin taşıyıcısı olarak kullanılacak olan GA optimizasyon tekniği üzerinde durulmuş ve dördüncü bölümde ele alınan hibrit paralel genetik algoritma uygulamasına hazırlık yapılmıştır. Geliştirilen algoritma, uygulanması ve elde edilen sonuçlar dördüncü bölümde ele alınmaktadır.

4. HİBRİT PARALEL GENETİK ALGORİTMA UYGULAMASI

Kaynak kısıtlı proje çizelgeleme, pratikte birçok uygulamayı modellemek için kullanılabilen genel bir çizelgeleme problemidir. Amaç, kaynak sınırları olan bir projenin aktivitelerini, seçilen amaç doğrultusunda optimize etmektir.

Literatür araştırması ve çözüm algoritmalarının incelenmesi sırasında, aşağıda değinilen alanlara yönelik iyileştirme olanakları belirmiştir.

- KKPÇ: tekrarsız kromozom gösterimi ile arama uzayının verimli kullanılması,
- GA: Erken yakınsama ve baskılama sonucu oluşan çeşitlilik kaybının engellenmesi,
- GA: PGA desteği ile çeşitlik artışı ve arama uzayının verimli kullanılması,
- GA: Hibrit SA desteği ile yerel iyiden kaçış.

Gözlemlenen iyileştirme olanaklarından yola çıkarak, aşağıdaki algoritmalar yardımıyla söz konusu güçlendirme ve desteklerle daha iyi çözüm değerleri üretebileceği düşünülmüştür:

- GA'ya yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi,
- KKPÇP'ne yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi,
- Hibrit SA desteği,
- PGA desteği.

Doktora tezinde uygulanan güçlendirme ve desteklerin ardındaki ana fikir, tekrarsızlığın sağlanması ve çeşitlendirilmenin artırılmasıdır.

GA'ya yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi, düşen çeşitlilik ve hızlı yakınsama problemlerine karşı çözüm olarak kullanılabilecek olan bir mekanizmadır. Tekrarlanan kromozomların uzaklaştırılması ve yerlerine –çeşitliliği sağlamak üzere– yenilerinin konması esasına dayanmaktadır.

KKPÇP'ne yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi ise, farklı görünen ama aynı kaynak profilini üreten kromozomların elenmesi ve yerlerine yenilerinin konulmasını sağlamaktadır.

Hibrit SA desteği, yerel iyiden kaçmak ve daha iyi çözümler içerebilecek patikalara sıçramak için kullanılmaktadır.

PGA desteği, GA'ya göre daha hızlı sonuca ulaşmak, daha küçük popülasyonla çalışmak, daha yüksek çeşitlilik sağlamak ve daha yüksek çözüm kalitesi elde etmek amaçlarına ulaşmak için kullanılmaktadır.

Bu bölümde, KKPCP problemine uygulanmak üzere güçlendirilmiş bir GA yaklaşımı ele alınmaktadır. Güçlendirme ihtiyacının arkasındaki temel düşünce biçimleri ve –güçlendirmeye yönelik olarak geliştirilen– çözümler açıklanmış, ardından uygulama algoritması ayrıntılı akış şeması üzerinden tanıtılmıştır.

4.1 Hibrit Paralel Genetik Algoritma Uygulamasına İlişkin Kavramlar

Klasik GA bir çok optimizasyon probleminin çözümünde başarıyla uygulanabilir. Eğer probleme özgün bir düzenleme göz ardı edilirse, elde edilen sonuçlar yeteri kadar iyi olmaktan uzak kalabilir. Bilinen basit ve etkili kurallar, probleme özel gösterim biçimleri, komşuluk yapıları, arama uzayı daraltma araçları, dikkatle incelenmeli ve iyileştirme çalışması yapılmalıdır. GA yerel arama konusunda çok geç davranmaktadır. Bu yüzden yerel aramaların GA'ya bırakılması çözümü olumsuz yönde etkilemektedir. Çözümde yol gösterici olabilecek araçlar GA'ya eklenmelidir. Etkili bir yerel arama aracı geliştirilmesi, yakınsama hızını artırmakta ve daha kaliteli çözümlere kısa sürede ulaşılmasına yardımcı olmaktadır. Yerel arama basitçe komşuluk yapılarının taranması şeklinde olabilir veya başka bir meta-sezgisel algoritma hibrit olarak eklenebilir.

GA'nın ilerleyen aşamalarında, bazı kromozomlar baskın hale gelmekte, popülasyon –mutasyon mekanizmasına rağmen– erken yakınsama göstererek çeşitliliğini kaybetmektedir. GA yeniden başlatılmakta ve çözümün yeni bir patikada gerçekleşmesi umulmaktadır. Tekrarlanan denemelerde çeşitlilik kayboluyorsa popülasyonu artırmak bir çözüm olabilmektedir. Bu durumda işlem hızı yavaşlayabilmektedir.

GA'lar paralel olarak çalıştırılabilirler. Böylece hem farklı patikalarda çözüm arama çabası ilerlemekte, hem de –bir haberleşme mekanizması kurularak– iyi çözüm bölgelerine yönelik bilgiler paylaşılabilmektedir. Popülasyon sayısının çok düşük tutulduğu Micro–GA'lar, paralel yapı desteği ile dikkat çekici bir performans artışı gösterebilmektedirler.

GA parametrelerinin probleme yönelik olarak aşırı optimize edilmesi (overfitting) konusunda dikkatli davranılmadır. Parametre optimizasyonu, devingen olmayan bir veri setine yönelik öğrenme işleminde etkili olabilir. Eğer veri devingense veya farklılaşıyorsa, her problem için yeniden parametre optimizasyonu yapmak verimsiz bir çaba haline dönüşebilir. Basit ama etkili araçlar geliştirilmesine zaman ayrılmalıdır. Parametre optimizasyonu, küçük değişikliklerde kırılabilirlik yaratmayacak ve geniş aralıklarda geçerli olabilecek şekilde yapılmalı, optimizasyon algoritması tarafından henüz görülmemiş veri setlerinde sınanmalıdır.

4.2 Uygulamada Yararlanılan Güçlendirme ve Destekler

4.2.1 Tekrarsız kromozom güçlendirmesi

GA'nın çalışma süresi ilerledikçe, başarılı kromozomlar elde edilir ve en başarısız kromozomlar elenir. İlerleyen aşamalarda tüm kromozomlar, en başarılı kromozomlara benzemeye başlarlar. GA, benzerliği azaltmak ve arama uzayında farklı yerlere sıçramak için mutasyon mekanizmasını kullanır. Ancak problemin dar bir alan sıkışması durumunda uygulanan mutasyon işlemi yeni bir olurlu çözüm üretmez ve etkisizleşir. Tekrarlanan kromozomların engellenmesi veya yerlerine yenilerinin konulması, çeşitliliğin artmasına ve GA operatörlerinin yeni arama bölgelerine geçişine olanak sağlar. Ayrıca tekrarlanan veya çok benzer durumda olan kromozomlar, kazanç sağlamayan ve birbirinin aynı olan başarımlarını üreten amaç fonksiyonunun hesaplanması için fazladan işlemci yükü oluştururlar.

Popülasyonunda tekrar edilen kromozomların varlığı GA'nın çalışmasını sınırlandırır. Çeşitlilik düşer, dar bir arama uzayının sınırları dışına çıkılamaz ve yerel optimumdan kurtulmak olanaksız hale gelebilir.

Tekrarlanan kromozomlar üç biçimde oluşabilirler:

- Başlangıç popülasyonunun özensizliği,

- Baskın kromozomların oluşturduğu erken yakınsama,
- Farklı görünen kromozomların, probleme özel olarak aynı çözümü göstermesi.

İlk iki madde GA'ya özgüdür. Üçüncü madde ise probleme özgü olarak ortaya çıkar. Aşağıda, ilk adımda genel olarak GA'ya, ikinci adımda probleme özel olan güçlendirme konuları ele alınmaktadır.

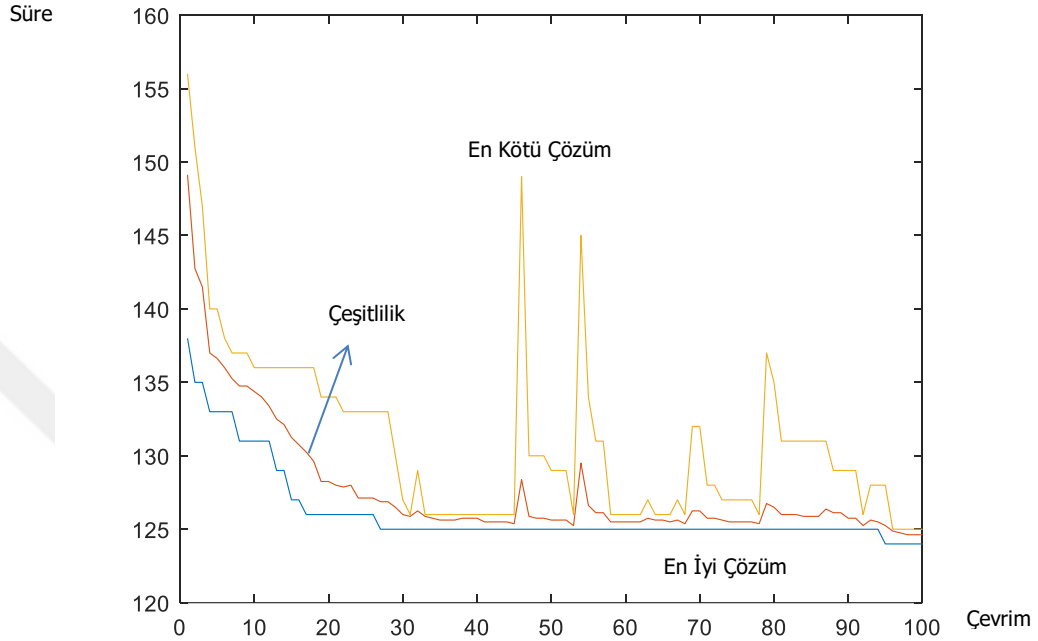
4.2.1.1 GA'ya özel tekrarsız kromozom güçlendirme

GA'ya özel güçlendirme için önerilen mekanizma, popülasyonun tekillik açısından kontrol edilmesi, varsa tekrar eden kromozomların uzaklaştırması ve yerlerine yenilerinin konulması biçimindedir. Tekrar eden kromozomlar kolayca bulunabilirler. Yerine koyma için aşağıdaki seçenekler kullanılabilir:

- Başlangıç popülasyonu hazırlama mekanizması,
- Rastgele olurlu bir çözüm,
- Aşırı mutasyona uğratma,
- Yeni bir patikaya girecek kadar geri çekilme,
- Paralel çalışan diğer işlemlerden, arama uzayının uzak bölgeleri için örnek alma,
- Diğer çeşitlendirme çalışmaları.

Aşırı mutasyona uğratma: Çok fazla gen üzerinde rastgele değişiklik yapma, rastgele seçilen iki genin yerini ardışık olarak değiştirme, rasgele seçilen genlere aşırı yüksek veya düşük değerler atama veya benzer yollarla yapılabilir. Popülasyonun tekillik açısından kontrol edilmesi işlemi, çeşitliliğin belli bir eşiğin altına inmesiyle veya rastgele olarak tetiklenebilir. Çeşitlilik, popülasyonun amaç fonksiyon sonuçlarının ortalamasının, mevcut en iyi çözüme yaklaşmasını izleyerek ölçülebilir. GA bir yerel çözüme takılabilir. Bu durumda çeşitliliğin ölçümü de takılıp kalır ve beklenen eşiğin altına inemez. Çeşitliliğin rastgele tetiklenmesi bu takılmayı kırmak için kullanılabilir. Rastgele tetikleme, çözümlerinin yeteri kadar olgunlaşmasına izin verecek kadar geç, uzun takılmaları engelleyecek kadar sık yapılmalıdır. Tekrarlı kromozomlar az sayıda ortaya çıksa bile, yapılan çeşitlendirme girişimi özel bir yük yaratmaz. Sadece karşılaştırmanın düşük maliyeti ortaya çıkar. Bu yüzden rastgele

çeşitlendirme tetikleme sıklığı için hassas bir ayar gerekmez. Çoğu kez önemli olan, hassas bir ayar yerine çözüm getiren mekanizmanın etkili varlığıdır. PSPLIB problemleri çözümü sırasında elde edilen GA'ya özel tekrarsız kromozom güçlendirme davranışı Şekil 4.1 'de görülmektedir.



Şekil 4.1 : GA'ya özel tekrarsız kromozom güçlendirme.

Şekilde yukarıdan aşağıya doğru sırayla:

- Popülasyonda yer alan en kötü çözüm,
- Çeşitliliği ölçen ortalama,
- Optimizasyon sırasında oluşan yakınsama (elde edilen en iyi çözüm)

değerleri görülmektedir. Yatay eksen optimizasyondaki ilerlemeyi gösteren çevrim sayısını, düşey eksen ise elde edilen çözüm değerlerini göstermektedir. Ortadaki eğrinin (çeşitlilik ölçüsü) alttaki eğriye yaklaşması çeşitliliğin azaldığını göstermektedir. Çeşitlilik kaybolunca GA'ya özel tekrarsız kromozom güçlendirme işlemi devreye girmektedir. İşlem sonunda çeşitlendirme yeniden artmakta ve optimizasyon çeşitlendirilmiş popülasyonla devam etmektedir.

4.2.1.2 Probleme özel tekrarsız kromozom güçlendirme

KKPÇP'ne yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi ana fikri Debels ve diğ. (2006) tarafından ortaya atılmıştır. Ancak bu fikrin geliştirilmesine ve uygulanmasına yönelik bir başka bir çalışmaya rastlanmamıştır.

Özellikle kombinatorial problemlerin kromozom gösterimlerinde, farklı görünen kromozomların aynı çözümü işaret etmesi mümkündür. KKPÇP probleminde, farklı görünen kromozomlar tamamen aynı kaynak profiline sahip olabilirler. Aslında çözüm uzayında aynı yer, tekrarlı olarak işleme alınmaktadır. Bu durumda çeşitlendirme düşer, işlem yükü artar. Şekil 2.6'da verilen çizgeden oluşturulan rastgele anahtar listesi gösterimi aşağıda yer almaktadır.

$$L=[0,00 \ 0,42 \ 0,56 \ 0,12 \ 0,88 \ 0,34 \ 0,62 \ 0,04 \ 0,28 \ 0,48 \ 1,00]$$

Listenin tüm elemanlarını sabit bir sayıya bölerek elde edilen yeni liste, ilk listeye aynı olan öncelik düzenini üretir.

$$L=[0,00 \ 0,21 \ 0,28 \ 0,06 \ 0,44 \ 0,17 \ 0,31 \ 0,02 \ 0,14 \ 0,24 \ 0,50]$$

Listedeki sayılar üzerinde oluşacak küçük değişiklikler yeni bir listeyi tarif etmeyebilir.

$$L=[0,000 \ 0,211 \ 0,284 \ 0,063 \ 0,449 \ 0,174 \ 0,313 \ 0,026 \ 0,146 \ 0,242 \ 0,501]$$

Çözüm olarak aşağıdaki adımlar önerilebilir:

- Kromozomda yer alan bilgilere göre çizelgeleme yapılır.
- Çözüm ES'a (Aktivite En Erken Başlama Zamanı, *Early Start*) göre sıralanır. Eğer iki aktivite aynı ES değerine sahip ise, aktivite numarası ikinci sıralama anahtarı olarak kullanılır.
- Bir önceki adımda elde edilen aktivite listesi, birden başlayan ve toplam aktivite sayısında biten tekrarsız tamsayılarla öncelik anahtarı gösterimine dönüştürülür.
- Çizelgelemeye esas kromozom silinir ve yerine yeni sıralama yerleştirilir.

L listesine yukarıdaki çözüm uygulandıktan sonra elde edilen düzeltilmiş liste aşağıda verilmektedir.

$$L=[1 \ 6 \ 8 \ 3 \ 10 \ 5 \ 9 \ 2 \ 4 \ 7 \ 11]$$

Düzeltilmiş listede üçüncü aktivitenin 8, dördüncü aktivitenin 3 öncelik değerine sahip olduğu görülebilir.

Popülasyon tabanlı bir optimizasyon algoritmasında, birbirinden farklı görünen fakat aynı öncelik dizilimini işaret eden listeler (kromozomlar) üremiş olabilir. Probleme özel güçlendirme işlemi, aynı sonucu üreten kromozomların farklı görünmesini engeller ve aynı görünmesini sağlar. Yerel arama algoritmalarının etkinliğinde komşuluk yapısı önemli bir rol oynar. Probleme özel güçlendirme işlemi, farklı seçeneklermiş gibi görünen komşuluk dizilişlerinin tekillenmesini sağlayarak, daha verimli bir arama uzayı hazırlar. Gösterilen düzeltmenin yapılmasından sonra, eş öncelik dizilimini gösteren listeler belirginleşir ve GA için önerilen güçlendirme işlemi, tekrar eden kromozomları uzaklaştırıp çeşitlendirmeyi tazeler.

4.2.2 Paralel genetik algoritma desteği

GA rastgelelik barındıran yapısı nedeniyle her çalıştırmada farklı patikalarda ilerleyebilir. Bir çalıştırmada çözümden çok uzaktayken takılıp kalabilir, başka bir çalıştırmada hızlı bir şekilde en iyi değere yaklaşabilir. Uzun süren ve yeni çözümler üretmeyen bir patikanın sonuna doğru benzerlik artar ve çeşitlilik düşer. Ayrıca GA yerel arama için yavaş davranır. PGA kullanımı bu tip sorunlara bir çözüm olabilir. Paralel olarak çalışan GA'lar, birbirlerinden farklı patikada ilerleyebilirler. PGA'dan bazıları takılma gösterirken, diğerleri daha iyi çözümler üretebilirler. Özellikle küçük popülasyona sahip alt GA'lar, bir yerel arama aracı olarak çalışıp kendi yolları üzerindeki yerel iyiye hızlıca ulaşabilirler. İyi çözümlere ulaşan alt GA'lar diğer GA'larla haberleşerek iyi çözümler içeren kromozomlarla eşleşme olanağı sağlayabilirler. GA'lardan biri erken yakınsama göstererek çeşitliliğini kaybederken, diğer GA'lardan gelen kromozomlarla çeşitlenip yeni çözümlere aday olabilir. Alt GA'daki içsel farklılık, geçişkenlik mekanizmalarıyla, diğer GA'lara aktarılır. Tekrarsız kromozom güçlendirmesi ve PGA desteği, çeşitliliğin korunmasına ve yakınsamanın hızlanmasına olanak sağlayabilir.

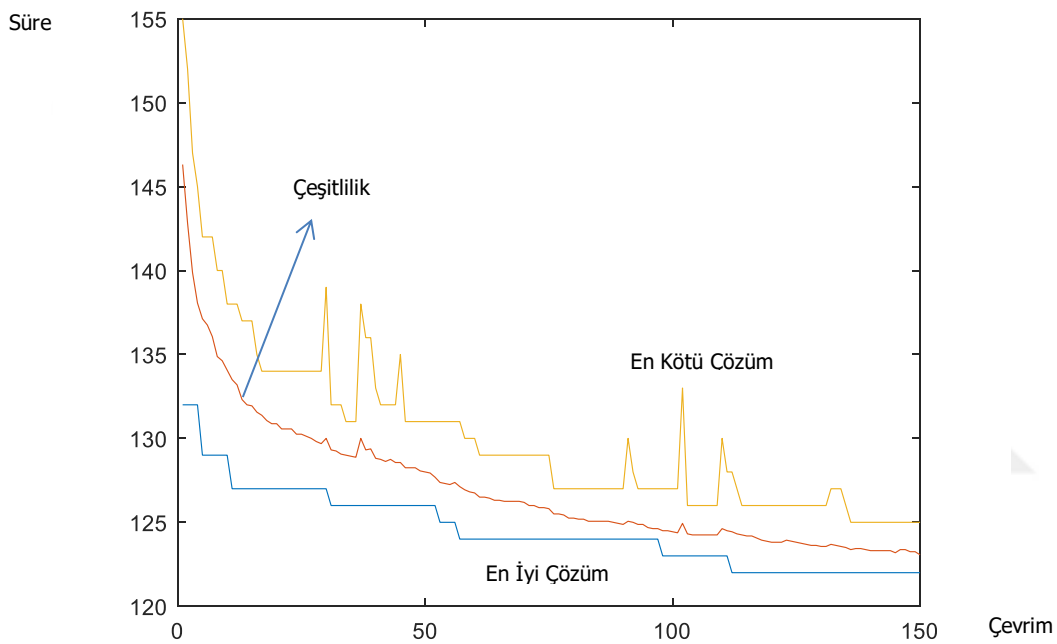
4.2.3 Hibrit algoritma desteği

Yerel optimumdan kaçış ve çeşitlendirme desteği için SA'dan ödünç alınan “*küçük bir olasılıkla daha kötüyü kabul etme*” yaklaşımı kullanılmıştır. Çaprazlama adımında elde edilen yeni kromozom, kendisini üreten kromozomlardan kötü ise, küçük bir olasılıkla popülasyona katılma şansı bulabilir. PSPLIB problemleri

çözümü sırasında elde edilen Hibrit SA desteği davranışı Şekil 4.2’de görülmektedir. Şekilde yukarıdan aşağıya doğru sırayla:

- Popülasyonda yer alan en kötü çözüm,
- Çeşitliliği ölçen ortalama,
- Optimizasyon sırasında oluşan yakınsama (elde edilen en iyi çözüm)

değerleri görülmektedir. Yatay eksen optimizasyondaki ilerlemeyi gösteren çevrim sayısını, dikey eksen ise elde edilen çözüm değerlerini göstermektedir.



Şekil 4.2 : Çeşitlendirmede hibrit SA desteği.

Ortadaki eğrinin (çeşitlilik ölçüsü) alttaki eğriye yaklaşması çeşitliliğin azaldığını göstermektedir. Kötü çözümlerin kabul edilmesi olasılığı gerçekleştiğinde hibrit SA desteği devreye girmektedir. Optimizasyon işlemi, yerel iyiden geri çekilme işlemi sonrasında –daha iyi bir çözüme ulaşacak patikaya girme beklentisiyle– devam etmektedir.

4.3 Uygulamada Kullanılan Algoritma

Uygulamada kullanılan algoritmaya ilişkin olarak sırasıyla:

- Uygulama ortamı ortaya konmakta,

- Uygulama algoritmasının ana hatlarına bakış atılmakta,
- Uygulama algoritmasının akış şeması açıklanmaktadır.
- Uygulama algoritmasının sınanması son adım olarak ele alınmaktadır.

4.3.1 Uygulama ortamı

Uygulama HP Z420 iş istasyonu üzerinde hazırlanmış ve çalıştırılmıştır. İş istasyonları diğer masaüstü bilgisayarlara oranla yüksek işlem yükleri altında uzun süreler hatasız ve kesintisiz çalışabilmektedir. İş istasyonu konfigürasyonu Çizelge 4.1’de görülmektedir.

Çizelge 4.1 : HP Z420 iş istasyonu konfigürasyonu

Model	HP Z420 İş İstasyonu
PS / Çipset	600W/ Intel C602, Sıvı Soğutma
İşlemci	Intel Xeon E5–1650 (6 Core) (3.20GHz, 12MB, 1600MHz)
RAM	32GB (4x8GB) DDR3 ECC 1600MHz
Ekran Kartı	nVidia Quadro K600 (2GB, Profesyonel 3D)
Sistem Diski	256GB SATA 3 SSD
Depolama Diski	2x2TB (7.200rpm) SATA 3 HDD
Optik Sürücü	16x DVD+/-RW
İşletim Sistemi	Windows 7 Pro 64bit
Monitör	HP 24" ZR2440w

Yazılım MATLAB uygulama ortamında geliştirilmiştir. MATLAB’ın özellikleri, diğer uygulama araçlarından farkları, avantajları ve dezavantajları aşağıda açıklanmaktadır (Mathworks, 2019).

MATLAB, araştırma ve mühendislik alanlarında karşılaşılan problemlere pratik, tutarlı, güvenilir, kolay kontrol edilebilir ve hızlı sonuçlar sunmakta ve tüm dünyada endüstri, devlet ve akademik kurumlarda yaygın olarak kullanılmaktadır.

MATLAB'ın kullanım alanları şunlardır:

- Matematik ve hesaplama işlemleri,
- Algoritma geliştirme,
- Modelleme, simülasyon ve prototip üretme,
- Veri analizi ve görsel sunumu,
- Bilimsel ve mühendislik grafikleri,
- Uygulama geliştirme.

MATLAB hesaplama işleri için çok popüler bir üst seviye dildir. Diğer popüler dillere göre kullanımı çok daha kolaydır. MATLAB, algoritmaların ve uygulamaların hızlı bir şekilde geliştirilmesini ve analizini sağlayan üst düzey bir programlama dili ve geliştirme araçları seti sunmaktadır. MATLAB dili, bilimsel ve mühendislik problemlerini çözmek için gerekli olan vektör ve matris işlemlerini içermektedir. Söz konusu işlemler, hem yazılım geliştirmeyi hem de yürütmeyi kolaylaştırmaktadır. MATLAB dili ile algoritmaları geleneksel dillere göre daha hızlı programlamak ve geliştirmek mümkündür. Çünkü değişkenleri bildirmek, veri türlerini belirlemek ve bellek atama gibi düşük seviyeli görevler otomatik olarak yapılabilmektedir. Birçok durumda, MATLAB 'for' döngülerine olan ihtiyacı ortadan kaldırmaktadır. Sonuç olarak, bir MATLAB kodu satırı, genellikle birkaç C veya C++ kodunun satırının yerini almaktadır. MATLAB kütüphanesinde gelen hazır fonksiyonlar, tek satırlık parametreler çağırısı ile sayfalarca C, C++ koduna karşılık gelen işlemi güvenilir ve hızlı bir şekilde yerine getirebilmektedir. Aynı zamanda, MATLAB, aritmetik operatörler, kontrol akışı, veri yapıları, veri tipleri, nesne yönelimli programlama ve hata ayıklama gibi geleneksel programlama dillerinin tüm özelliklerini sunmaktadır. MATLAB, komutları veya komut gruplarını, derleme olmadan tek tek çalıştırılmasına ve en uygun çözümü elde etmek için yürütmenin tekrarlanmasına olanak tanımaktadır. MATLAB, karmaşık vektör ve matris hesaplamalarını hızlıca gerçekleştirmek için, işlemciye özel optimize edilmiş kütüphanelerini kullanmaktadır. MATLAB, algoritmaları etkili bir şekilde geliştirmeye yardımcı olan

araçlar içermektedir. MATLAB fonksiyonlarını kullanarak grafiksel kullanıcı arayüzleri oluşturulabilmektedir. MATLAB platformu veri dosyalarına, diğer uygulamalara, veritabanlarına ve harici cihazlara verimli erişim sağlayabilmektedir. Microsoft Excel, ASCII metin dosyaları, ikili dosyalar, görüntü, ses ve video dosyaları, HDF, HDF5 dosyaları gibi çeşitli formatlarda depolanan verileri okuyabilmektedir. Düşük seviye giriş çıkış işlevleri yapılabilir. ODBC/JDBC veritabanlarına erişim sağlanabilmektedir. MATLAB, GIF, JPEG, BMP, EPS, TIFF, PNG, HDF, AVI ve PCX gibi bilinen tüm veri dosyası ve grafik formatlarıyla uyumludur. Sonuç olarak, MATLAB grafik dosyalarını Microsoft Word ve Microsoft PowerPoint veya masaüstü yayıncılık yazılımı gibi diğer uygulamalara aktarmak mümkündür. Dışa aktarmadan önce, yayın özelliklerine uymak için gereken tüm tasarım ayrıntılarını, yazı tiplerini, çizgi kalınlığını, stil şablonları oluşturmak mümkündür. MATLAB, yaygın olarak kullanılan işlemlerin çoğunu destekleyen matematiksel, istatistiksel ve mühendislik araçları içermektedir. Ek olarak, MATLAB, çalışmaları belgelemek ve paylaşmak için bir dizi işlev içermektedir. MATLAB kodu diğer diller ve uygulamalar ile entegre edilebilmektedir. Geliştirilen algoritmalar ve MATLAB uygulamaları, özerk programlar veya yazılım modülleri olarak dağıtılabilir. MATLAB, uygulamaların C, C++, FORTRAN, COM nesneleri ve Java ile entegre edilmesini sağlayan fonksiyonlar sunmaktadır. DLL'ler, Java sınıfları ve ActiveX denetimleri çağırılabilir. Ayrıca MATLAB kodları C, C++ veya FORTRAN kodlarından çağırılabilir. MATLAB'da oluşturulan kodlar diğer MATLAB kullanıcılarına veya MATLAB derleyicisini kullanarak oluşturulan bağımsız uygulamalar olarak MATLAB sahibi olmayan kullanıcılara dağıtılabilir. Son kullanıcılara yönelik olarak diğer dillerle geliştirecek programlar, MATLAB kullanarak hazırlanan prototipleri izleyerek geliştirilebilmekte ve sıranabilmektedir. Ek olarak, COM veya Microsoft Excel'den çağırılacak yazılım modülleri hazırlanabilir. MATLAB, veri yapıları, fonksiyonlar, akış kontrolleri, giriş çıkışlarının yönetimi ve nesne yönelimli programlama olanakları ile üst düzey bir programlama dili olarak kullanılabilir (López, 2014).

MATLAB, algoritmaları kodlamak, geliştirmek, sınamak ve raporlamak için hızlı üretim yapmaya olanak sağlayan bir ortam sunmaktadır. Yorumlayıcı olarak çalışması sayesinde, her adım hızlıca anlaşılabilir ve izlenebilir. Sahip

olduğu geniş fonksiyon kütüphanesi, çok sık kullanılan kodların yeniden yazılması ve test edilmesi yükünü ortadan kaldırmakta ve doğrudan algoritmaya yönelik çalışmanın yapılmasını sağlamaktadır. Ayrıca vektörel kullanım olanağı kodlamayı kısaltmaktadır. MATLAB çok iyi düzeyde dokümanite edilmiştir.

İnsan sağlığına yönelik yaşam kritik uygulamaların, çok kullanıcıli uygulamaların ve yüksek hız gerektiren uygulamaların MATLAB’la geliştirilmesi uygun değildir. Ancak bu uygulamaların algoritmaları ve prototipleri MATLAB’la hızlıca geliştirilebilmektedir.

Diğer sayısal hesaplama dilleri seçenekleri, Python, R, Julia olarak sıralanabilir. Python ve R dilleri de MATLAB gibi yorumlayıcı olarak çalışmaktadırlar. Julia yeni ve gelişmekte olan bir dildir. Henüz dokümantasyonu ve alt kitaplıkları tam olarak gelişmemiştir. Ancak modern ve derlenebilen bir dil özelliği taşımaktadır, kolay kodlanabilir ve son derece hızlı çalışmaktadır. R’ın bazı alt kitaplıkları iyi bir dokümantasyona sahip değildir. R, bazı test problemlerinde diğer dillere göre daha yavaş olarak görülmektedir. Python güçlü, kararlı ve iyi dokümente edilmiş bir dildir ve R gibi yaygın olarak kullanılmaktadır. Python, R, Julia ücretsiz olarak kullanılabilir. MATLAB lisans ücretleri kullanılacak kitaplıklara bağlı olarak değişiklik göstermektedir (Danielsson ve Fan, 2018).

MATLAB, Python ve R yaygın olarak kullanılmaktadır. Julia gittikçe artan bir kullanıcı sayısına sahiptir. Hesaplamalı dillerin seçiminde, geçmiş kullanım alışkanlıkları, kurum kültürü, lisans ücreti, algoritma geliştirilen alandaki alt kitaplıkların özellikleri ve dokümantasyon düzeyi kriterleri karar vermeye yardımcı olabilmektedir.

4.3.2 Uygulama algoritmasının ana hatları

GA hakkındaki tanımlamalar Bölüm 3.3’te ayrıntılı olarak incelenmiş bulunmaktadır. Uygulama algoritması aşağıda verilen başlıklar altında açıklanmaktadır:

- Başlangıç popülasyonu,
- Amaç fonksiyon,
- Üreme,
- Mutasyon,

- Yerleştirme,
- Durdurma.

Başlangıç Popülasyonu:

Tekrarsız kromozom güçlendirmesi ve PGA desteği, küçük popülasyonlarda bile hızlı bir yakınsama sağlamaktadır. Popülasyon küçükse, yerel aramaya daha çok zaman ayrılabilir. Başlangıç olarak az kromozomlu PGA çalıştırılmış, en iyi sonuçlara yeterli yakınsama sağlanamıyorsa kromozom sayısı ve PGA sayısı artırılmıştır. Başlangıç popülasyonu –tekrarsızlık kontrolü yapılarak– rastgele sayılarla hazırlanmıştır.

Amaç Fonksiyon:

Problemin temsilde seri çizelgeleme ve rastgele anahtar gösterimi kullanılmıştır. Kromozomda verilen öncelik sırasına göre başlatılan aktivitelerin, kaynak kısıtları altında çizelgelenmesi sonucu elde edilen en erken proje tamamlanma zamanı, o kromozom için amaç fonksiyon değeri olarak anılır.

Üreme:

GA havuzundaki her kromozom sırayla taranır. Sırası gelen kromozom için, öncelikle kendi popülasyon havuzundan veya, –düşük bir olasılıkla– paralel çalışan diğer GA kromozom havuzlarından bir eş kromozom seçilir. Çaprazlama olasılığı üstünde kalan genler birinci kromozomdan, altında kalan genler ikinci kromozomdan alınarak ve yeni bir kromozom oluşturulur. Üremede elde edilen yeni kromozoma yerel arama işlemi uygulanır.

Mutasyon:

Klasik GA’da kullanılan basit mutasyon mekanizması devre dışı bırakılmış, yerine tekrarsız kromozom güçlendirmesi konmuştur. Güçlendirme, çeşitlenmenin düşmesi veya düşük bir olasılığın ortaya çıkması durumunda tetiklenir. Hybrid SA katkısı mutasyona destek vermektedir.

Yerleştirme:

Yerel arama sonucu elde edilen kromozomun hesaplanan amaç fonksiyon değeri, birinci veya ikinci üretici kromozomdan iyi ise, en kötü değere sahip üretici kromozom uzaklaştırılır ve yerine yeni üretilen kromozom yerleştirilir. Yeni

kromozom her iki kromozomdan da kötü değere sahipse, küçük bir olasılıkla kötü değere sahip üretici kromozomun yerine yerleştirilir.

Durdurma:

Verilen çevrim sayısına veya istenen amaç fonksiyon değerine ulaşıncaya durma gerçekleşir.

4.3.3 Uygulama algoritmasının akış şeması üzerinden açıklanması

GA ve PGA temel kavramları Bölüm 3.3’de ve Bölüm 3.4’de ele alınmıştır. KKPCP’de tekrarsız kromozom destekli paralel genetik algoritma akış şeması Şekil 4.3’te verilmiştir. Akış şemasında üç ayrı bölüm görülmektedir. Solda yer alan ve gri renkle vurgulanmış olan bölüm ana blok, diğer bölüm ise PGA bloğu olarak adlandırılmaktadır. PGA bloğu kendi içerisinde paralel olarak çalışacak GA bloklarını barındırır.

Akış şemasına yönelik olarak verilen giriş bilgileri, aşağıda verilen kırılım yapısında incelenmektedir.

1. Ana blok

1.1. Başlama

1.2. Başlangıç popülasyonunun oluşturulması

1.3. Ana döngü

1.3.1. PGA bloğu → 2

1.4. Bitiş

Ana blok, başlangıç popülasyonunun oluşturulması ile başlar, Yinelemeli işlemleri yürüten döngü ile devam eder ve döngü sonunda durur. Her döngüde PGA bloğu çağırılır.

2. PGA bloğu

2.1. PGA yöneticisi

2.1.1. GA bloğu → 3

2.2. En iyi sonucun gösterimi

2.3. GA’ya özel tekrarsız kromozom güçlendirme

PGA bloğu üç alt işlemi çağırır. Birinci adımda PGA yöneticisi çalıştırılır. PGA yöneticisi paralel çalışma kuyruğunda bekleyen GA'ları sırayla çalıştırır. İkinci adımda en iyi sonuç gösterilir ve üçüncü adımda tekrarsız kromozom güçlendirme işlemi gerçekleştirilir.

3. GA Bloğu

3.1. Kromozom seçimi

3.2. Çaprazlama

3.3. Yerel arama

3.4. Probleme özel tekrarsız kromozom güçlendirme

3.5. Hibrit SA destekli kromozom kabulü

3.6. Yerleştirme

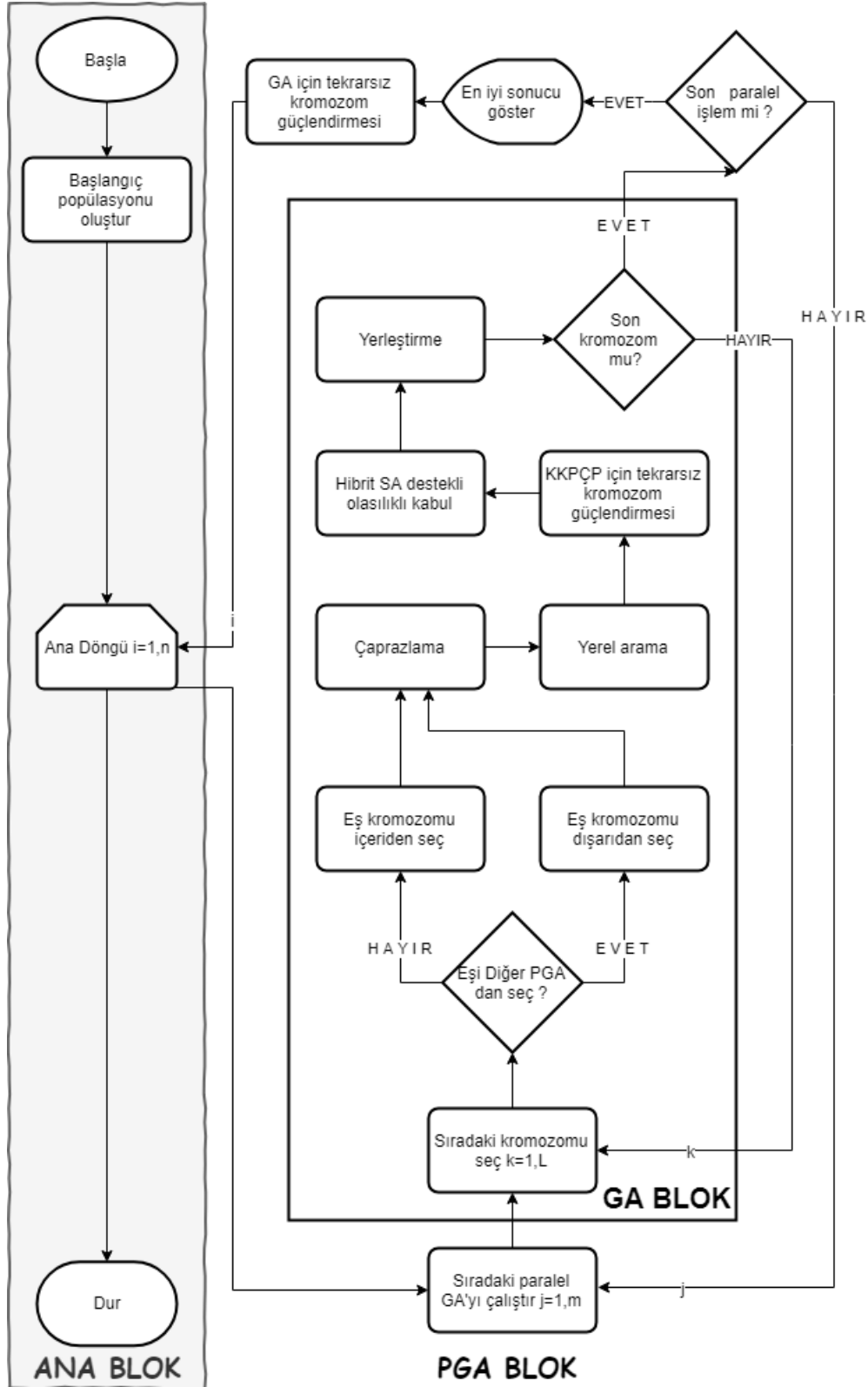
GA bloğu, kromozom seçimi işlemiyle başlar. Seçilen kromozomlara çaprazlama işlemi uygulanır. Çaprazlama işleminden elde edilen yeni kromozom üzerinde –daha iyi bir sonuç elde etme umuduyla– yerel arama gerçekleştirilir. Elde edilen en iyi sonucu taşıyan kromozom üzerinde tekrarsız kromozom güçlendirmesi işlemi uygulanır. Çözüm, hibrit SA destekli kromozom kabulü işlemi tarafından onaylanırsa, yerleştirme işlemi tarafından popülasyonda uygun pozisyona konumlanır.

Başlangıç popülasyonun oluşturulması:

GA temel kavramları ve popülasyon konusu Bölüm 3.3'te ele alınmıştır. Burada başlangıç popülasyonu oluşturulması sırasında kullanılan tekrarsızlık kontrolü işlemi açıklanmaktadır.

Başlangıç popülasyonu rastgele sayılarla hazırlanır. Elde edilen kromozomların birbirlerine yakın olması durumu –optimizasyon başlamadan– çeşitliliğin düşük olmasına yol açar. Başlangıç popülasyonu çeşitliliğini korumak için, kromozomların benzerlik durumu ölçülmeli ve eğer benzerlik yüksekse o kromozom popülasyona eklenememelidir. Bu uygulamada, benzerlik ölçümü için bir kromozom üzerindeki genin içeriği, diğer kromozomda aynı pozisyondaki genin içeriği ile karşılaştırılır. Eğer karşılıklı olarak iki genin içeriği aynı ise sayaç bir artırılır. Bu işlem kromozomdaki tüm genler için yapılır. İşlem sonunda elde edilen sayaç değeri,

kromozom uzunluğuna bölünerek yüzde cinsinden benzerlik değeri olarak kullanılır. Benzerlik fonksiyonu pseudo kodu Algoritma 1’de verilmiştir.



Şekil 4.3 : Uygulama akış şeması.

Algoritma 1: Benzerlik

function Benzerlik = (crmz1, crmz2, kromozom_uzunluđu)

sayac = 0 ;

for i = 1: kromozom_uzunluđu

if crmz1(i) == crmz2(i)

 sayac = sayac + 1 ;

end

end

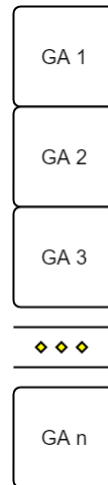
Benzerlik = sayac / kromozom_uzunluđu

end

Algoritma, popölasyona eklenmek üzere olan her yeni kromozomu, popölasyondaki diđer kromozomlarla karşılaştırmak için kullanılır. Eğer benzerlik seçilen eşik değeriinden düşükse ekleme gerçekleşir, değil ise, rastgele yeni bir kromozom üretilir ve benzerlik kontrolü yapılır. Eşik değeriinin çok küçük seçilmesi benzer kromozomların oluşma olasılıđını artırırken, çok büyük seçilmesi benzemezliğe zorlar. Eğer başlangıç popölasyon büyükse ve aşırı benzemezlik zorlaması yapılırsa, popölasyonu doldurmak olanaksız hale gelebilir. Popölasyonda yer alan kromozom sayısı artıkça benzerlik zorlaması azaltılmalıdır.

PGA Yöneticisi:

PGA'yı oluşturan GA'ların bellekteki pozisyonlanması Şekil 4.4'de verilmiştir. PGA yöneticisi paralel çalışma kuyruğunda bekleyen GA'ları sırayla çalıştırır. Her GA bağımsız olarak yürütölür ve kendi arama uzayı içerisinde etkilidir. PGA yöneticisi sırası gelen GA'nın bellek pozisyonlamasını yapar ve çalıştırır. GA'ın bir çevrimlik çalışması tamamlandıktan sonra sıradaki GA aktif hale gelir.



Şekil 4.4 : PGA– GA bellek pozisyonları.

Son GA tamamlanınca PGA yöneticisinin o çevrimdeki görevi biter ve PGA bloğunda bir sonraki adımda yer alan işleme –en iyi sonucun gösterimi– sıra gelir.

En iyi sonucun gösterimi:

Uzun süren optimizasyon işlemlerinde, elde edilen sonuçların izlenmesi için, her döngü sonunda ele edilen en iyi değerin, kullanıcı ara yüzüne gösterilmesi, ilerlemenin izlenmesini kolaylaştırmaktadır. Henüz tüm döngüler tamamlanmamış olsa bile, en iyi değere ulaşılmış olabilir. Bu durumda uygulama beklemeksizin durdurulabilir. En iyi sonucun kullanıcı ara yüzünde gösterimi Şekil 4.1’de verilmiştir.

GA’ya özel tekrarsız kromozom güçlendirme:

GA’ya özel tekrarsız kromozom güçlendirmesi Bölüm 4.2.1.1’de kavramsal olarak açıklanmıştır. Bu bölümde çeşitliliğin ölçülmesi, tekrar eden kromozomların bulunması ve yeni kromozomların yerine konması açıklanmaktadır. Popülasyonun tekillik açısından kontrol edilmesi işlemi, çeşitliliğin belli bir eşiğin altına inmesiyle veya rastgele olarak tetiklenebilir. Çeşitlilik, popülasyonun amaç fonksiyon sonuçlarının ortalamasının, mevcut en iyi çözüme yaklaşmasını izleyerek ölçülebilir. Çeşitlilik başlatma fonksiyonu pseudo kodu Algoritma 2’de verilmektedir. Eğer çeşitlik düşmüşse fonksiyon 1 değerini döndürür. Bu durumda popülasyonda yer alan tüm kromozomlar için tekillik kontrolü yapılır.

Algoritma 2: Çeşitlilik başlatma fonksiyonu

function Çeşitlilik_baslat = (rand_esik, z_ort, z_min)

```
çeşitlilik = 0 ;  
if rand < rand_esik  
    Çeşitlilik_baslat = 1 ;  
    return;  
end  
if z_ort == z_min  
    Çeşitlilik_baslat = 1 ;  
    return;  
end  
end
```

Elit kromozomların kaybolmaması için tüm popülasyon en iyi başarımlar değerinden en kötü başarımlar değerine göre sıralanır. En iyi başarımlar değerine sahip kromozomdan başlanarak tekillik kontrolü yürütülür. Her durumda tekillik kontrolü için ilk seçilen kromozom, ikinci kromozomdan daha yüksek (ya da eşit) başarımlar değerine sahiptir.

Eğer ikinci kromozom birinci kromozomun aynısı (benzerlik=1) ise veya birinci kromozoma verilen eşik değerinden daha fazla benziyorsa (örneğin: benzerlik>0.9) popülasyondan uzaklaştırılır. Uzaklaştırılan kromozomun yerine çeşitlendirmeyi sağlayacak yeni bir kromozom konur. Pseudo kodu Algoritma 3'te verilen yeni kromozom üretim mekanizması için olası seçenekler, Bölüm 4.2.1'de kavramsal olarak açıklanmış ve sıralanmıştır.

Algoritma 3: Kromozom_degistir

function *Kromozom_degistir*(pop, j)

```
v1 = rand ;
if    v1 < 0.25
    pop(j)= Yontem1(pop(j)) ;
elseif v1 < 0.50
    pop(j)= Yontem2(pop(j)) ;
elseif v1 < 0.75
    pop(j)= Yontem3(pop(j)) ;
elseif v1 < 1.00
    pop(j)= Yontem4(pop(j)) ;
end
end
```

GA'ya özel tekrarsız kromozom güçlendirmesi işlemlerin pseudo kodu Algoritma 4'te verilmiştir.

GA bloğunda yer alan işlemler aşağıda sırasıyla incelenmektedir

Kromozom seçimi:

GA havuzundaki her kromozom sırayla taranır. Sırası gelen kromozom için, öncelikle kendi popülasyon havuzundan veya, –düşük bir olasılıkla– paralel çalışan diğer GA kromozom havuzlarından bir eş kromozom seçilir. Eğer eş kromozomun paralel çalışan GA'lardan seçim olasılığı sıfıra yaklaşırsa, diğer GA'larla haberleşme zayıflar ve yürütülmekte olan GA kendi kromozom havuzundan beslenir. Diğer taraftan olasılık bire yaklaşırsa, eş kromozom çoğunlukla tüm GA'ların oluşturduğu kromozom havuzundan seçilir. Olasılığın sıfıra yaklaşması, üzerinde çalışılan GA'yı yalnızlaştırırken, bire yaklaşması kimliksizleştirir. Kromozom seçimi pseudo kodu Algoritma 5'te verilmiştir.

Algoritma 4: GA'ya özel tekrarsız kromozom güçlendirmesi

function GaTgk = (rand_esik, z_ort, z_min, pop)

if *Ceşitlilik_baslat()* == 0 /* Algoritma 2

return

end

populasyon_adet = **size**(pop,1) ;

kromozom_uzunluğu = **size**(pop,2) ;

for i = 1, populasyon_adet -1

 crmz1 = pop(i) ;

for j = i+1 to populasyon_adet

 crmz2 = pop(j) ;

 v1= *Benzerlik* (crmz1, crmz2, kromozom_uzunluğu) ; /* Algoritma 1

if v1> benzerlik_esik

Kromozom_degistir(pop,j) ; /* Algoritma 3

end

end

end

end

Algoritma 5: Kromozom Seçimi

function Kromozom_Secimi(crmz1, crmz2) = (pop_GA, pop_Out, rand_esik)

pop_GA_adet = **size**(pop_GA,1) ;

for i = 1, pop_GA_adet -1

 crmz1 = pop_GA(i) ;

if rand < rand_esik

 crmz2 ← pop_Out ;

else

 crmz2 ← pop_GA ;

end

end

end

Çaprazlama:

Belirlenen çaprazlama olasılığı üstünde kalan genler birinci kromozomdan, altında kalan genler ikinci kromozomdan alınarak yeni bir kromozom oluşturulur. Çaprazlama oranı 0,50'nin altında kalacak şekilde rastgele seçilmektedir. 0,5 durumunda her iki kromozom, yeni üretilen kromozoma eşit ağırlıklı olarak yansır. 0,5'ten küçük değerlerde birinci kromozom baskınlık kazanır. 0,5'ten büyük değerler

ikinci kromozomun baskınlığı artıracaktır. Ancak sıra birinci kromozomdadır. Sırası geldiğinde –şimdi ikinci durumda olan kromozom– birinci durumda olacak ve kendi baskınlığını yansıtacaktır. Böylece üst sınır 0,5 olarak belirir. Örneğin 0,22 olarak seçilen olasılık değeri için çaprazlama işlemi Şekil 4.5’te verilmiştir.

0,22	1	2	3	4	5	6	7	8	9	10	11
Olasılık	0,28	0,13	0,45	0,39	0,46	0,21	0,87	0,18	0,97	0,76	0,97
crmz1	0,22	0,17	0,22	0,92	0,26	0,11	0,19	0,30	0,28	0,74	0,36
crmz2	0,29	0,14	0,24	0,46	0,74	0,90	0,33	0,50	0,69	0,43	0,14
crmzYeni	0,22	0,14	0,22	0,92	0,26	0,90	0,19	0,50	0,28	0,74	0,36

Şekil 4.5 : Çaprazlama.

Çaprazlama işlemi pseudo kodu Algoritma 6’da verilmiştir.

Algoritma 6: Çaprazlama

```

function Caprazlama(crmzYeni) = (crmz1, crmz2, rand_esik )
kromozom_uzunluğu = size(crmz1,2) ;

for i = 1, kromozom_uzunluğu
    if rand < rand_esik
        crmzYeni(i) = crmz2(i) ;
    else
        crmzYeni(i) = crmz1(i) ;
    end
end
end

```

Yerel arama:

Yerel arama: ilerleyen bir optimizasyon sürecinde elde edilen son olurlu iyi çözümünün yakın çevresinin, daha iyi bir çözüm için araştırılmasıdır. Optimizasyon algoritması arama uzayında başka bir bölgeye sıçramadan önce yapılan yerel arama işlemi, çok yakında bulunabilecek daha iyi bir çözümün yakalanmasını sağlayabilir. Arama, genellikle mevcut komşuluk yapısı üzerinde küçük hareketlerle gerçekleştirilir Kromozom üzerinde yapılan değişiklikten sonra, değişikliğin ortaya çıkarttığı farklılığın etkisini ölçmek için, amaç fonksiyon hesaplaması yapılır. Amaç fonksiyon hesaplaması yüksek maliyetlidir. Maliyeti azaltmak için yerel arama yoğunluğu düşürülebilir. Yerel arama çoğunlukla probleme ve komşuluk yapısına özel olarak geliştirilir. Kromozom üzerinden seçilen bir veya birkaç gen üzerinde

değişiklikler yapmak için kullanılabilecek yaklaşımlardan bazıları aşağıda verilmektedir:

- Başka pozisyona yerleştirme,
- Bir başka genle karşılıklı olarak yer değiştirme
- Rastgele bir değer atama
- Aşırı değer atama.

Başka pozisyona yerleştirme: Bir pozisyonda bulunan değer, başka bir pozisyona yerleştirilir. Şekil 4.6'da 3. pozisyonda yer alan değer, 9. pozisyona yerleştirilmiştir.

	1	2	3	4	5	6	7	8	9	10	11
crmz1	0,22	0,17	0,46	0,92	0,26	0,11	0,19	0,30	0,28	0,74	0,36
crmzYeni	0,22	0,17	0,92	0,26	0,11	0,19	0,30	0,46	0,28	0,74	0,36

Şekil 4.6 : Yerel arama: Başka pozisyona yerleştirme.

Bir başka genle karşılıklı olarak yer değiştirme: İki farklı pozisyonda bulunan değer, karşılıklı olarak yer değiştirirler. Şekil 4.7'de 3. pozisyonda ve 8. pozisyonda yer alan değerler karşılıklı olarak değiştirilmiştir.

	1	2	3	4	5	6	7	8	9	10	11
crmz1	0,22	0,17	0,46	0,92	0,26	0,11	0,19	0,30	0,28	0,74	0,36
crmzYeni	0,22	0,17	0,30	0,92	0,26	0,11	0,19	0,46	0,28	0,74	0,36

Şekil 4.7 : Yerel arama: Bir başka genle karşılıklı olarak yer değiştirme.

Rastgele bir değer atama: Bir pozisyonda bulunan değer silinir ve yerine rastgele bir değer atanır. Yeni atanan değer, mevcut değerlerin alt ve üst sınırlarının içinde olmalıdır Şekil 4.8'de 3. pozisyona rastgele bir değer atanmıştır.

	1	2	3	4	5	6	7	8	9	10	11
crmz1	0,22	0,17	0,46	0,92	0,26	0,11	0,19	0,30	0,28	0,74	0,36
crmzYeni	0,22	0,17	0,71	0,92	0,26	0,11	0,19	0,30	0,28	0,74	0,36

Şekil 4.8 : Yerel arama: Rastgele bir değer atama.

Aşırı değer atama: Bir pozisyonda bulunan değer silinir ve yerine aşırı değer atanır. Yeni atanan değer, mevcut değerlerin alt ve üst sınırlarının dışında olmalıdır. Şekil 4.9'da 3. pozisyona aşırı değer atanmıştır.

	1	2	3	4	5	6	7	8	9	10	11
crmz1	0,22	0,17	0,46	0,92	0,26	0,11	0,19	0,30	0,28	0,74	0,36
crmzYeni	0,22	0,17	-10	0,92	0,26	0,11	0,19	0,30	0,28	0,74	0,36

Şekil 4.9 : Yerel arama: Aşırı değer atama.

Eğer gösterim biçimi gerektiriyorsa, öncelik–sonralık kurallarına uygun bir yerleştirme yapılmalıdır. Yukarıdaki işlemler yinelemeli olarak yapılabilirler. Aşırı yineleme durumunda yerel arama yaklaşımına aykırı olarak, orijinal kromozomdan çok uzaklaşılacağına dikkat edilmelidir. Yerel arama işlemi amaç fonksiyonun çözüm değeri iyileştikçe sürdürülmelidir. Yoğunluğu ayarlanabilen yerel arama pseudo kodu Algoritma 7’de verilmektedir.

Algoritma 7: Yerel Arama

/ Minimizasyon işlemine yönelik olarak kodlanmıştır.

function Yerel_Arama(crmz1, z_onceki) = (crmz1, max_ls,)

```

for i = 1,max_ls
    while true
        v1 = rand ;
        if v1 < 0.25
            crmz_Yeni = YA_Yontem1(crmz1) ;
        elseif v1 < 0.50
            crmz_Yeni = YA_Yontem2(crmz1) ;
        elseif v1 < 0.75
            crmz_Yeni = YA_Yontem3(crmz1) ;
        elseif v1 < 1.00
            crmz_Yeni = YA_Yontem4(crmz1) ;
        end

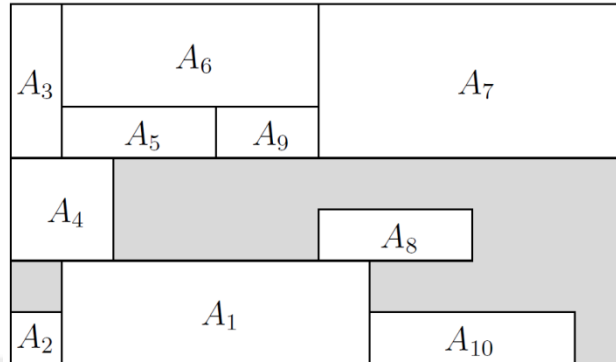
        z_son = Amac_fonksiyon(crmz_Yeni) ;

        if z_son < z_onceki
            z_onceki = z_son ;
            crmz1 = crmz_Yeni ;
        else
            break
        end
    end

end
end

```

Probleme özel tekrarsız kromozom güçlendirme Bölüm 4.2.1.2’de ayrıntılı olarak ele alınmıştır. Şekil 2.6’da verilen AON çizgenin kaynak kısıtları altında çözümü Şekil 4.10’da görülmektedir.



Şekil 4.10 : AON çizgenin kaynak kısıtları altında çözümü (Artigues ve diğ., 2010).

Çözümde yer alan aktivitelerin tetiklenme sırası birçok şekilde yazılabilir. Aşağıda aynı çözümü gösteren aktivite tetiklenme sırası listelerinden örnekler verilmiştir.

L=[2 4 3 1 5 6 9 8 7 10]

L=[3 4 2 6 5 1 9 7 8 10]

$$L=[4\ 3\ 2\ 5\ 1\ 6\ 9\ 8\ 7\ 10]$$

$$L=[4\ 2\ 3\ 5\ 6\ 1\ 9\ 8\ 7\ 10]$$

Bu tetiklenme listelerini barındıran kromozomlar, farklı görünseler de tamamen aynı çözümü işaret etmektedirler. Bu tip kromozomların popülasyonda yer alması; çeşitliliğin düşmesini, erken yakınsamanın ortaya çıkmasını ve yerel iyiye takılmayı hızlandırır. Bir çözüm, popülasyonda sadece bir kromozomla yer alırsa, geri kalan diğer kromozomlar arama uzayının farklı yerlerinin zenginliğini taşıyabilirler. Olurlu tetiklenme listesinin ES'a göre sıralanması tekil kromozom gösterim amacını sağlamaktadır. Tetikleme listesinde ES'ları aynı olan aktiviteler, kendi aralarında aktivite numarasına göre sıralanırlar. Bir kromozomun amaç fonksiyon değerinin hesaplanmasından hemen sonra yapılacak bu sıralama, tekil kromozom gösterimi için yeterlidir. İlerleyen işlemler sırasında, farklı görünen ama aynı çözümü gösteren bir tetiklenme listesi ortaya çıkarsa, yapılan düzeltme ve yerleştirme sonrasında popülasyonda birbirinin aynı kromozomlar ortaya çıkacaktır. Artık bunları tespit etmek, içlerinden birinin kalmasına izin vermek ve diğerlerinin yerine popülasyonu

çeşitlendirici yeni kromozomlar koymak çok kolaylaşmaktadır. Bu işlem kromozom üretimi sırasında yapılabileceği gibi, her çevrim sonunda tüm popülayon üzerinde de yapılabilir.

Hibrit SA destekli kromozom kabulü:

Hibrit SA destekli kromozom kabulü Bölüm 4.2.3'te ayrıntılı olarak incelenmiş ve Algoritma 8'de pseudo kod olarak verilmiştir.

Yerleştirme:

Kromozomun amaç fonksiyon değeri, birinci veya ikinci üretici kromozomdan iyi ise, en kötü değere sahip üretici kromozom uzaklaştırılır ve yerine yeni üretilen kromozom yerleştirilir. Bu yaklaşım, iyi değere sahip kromozomun korunmasını ve kötü değere sahip kromozomun iyileştirilmesini sağlar. Eğer yeni kromozomun değeri, kendisini üreten iki kromozomdan kötüyse, SA yaklaşımı uygulanır. Yeni kromozom küçük bir olasılıkla kötü değere sahip üretici kromozomun yerine yerleştirilir. Kötü kromozomun küçük bir olasılıkla popülasyona alınması, yerel optimumdan kaçış anlamına gelir. Bu kaçış hem çeşitliliği artırır hem de yeni bir patikaya geçme olanağı hazırlar. Yerleştirme işlemi pseudo kodu Algoritma 8'de verilmiştir.

Algoritma 8: Yerleştirme

/ Minimizasyon işlemine yönelik olarak kodlanmıştır.

/ z_crmz1 < z_crmz2 sırasında verilmelidir.

```
function Yerlestirme() = ( z_crmz1, z_crmz2, loc_crmz2, ...  
                           crmzYeni, z_crmzYeni, ...  
                           pop, rand_esik )
```

```
if z_crmzYeni < z_crmz2  
    pop(loc_crmz2) = crmzYeni ;  
else  
    if rand < rand_esik /* SA  
        pop(loc_crmz2) = crmzYeni ;  
    end  
end  
end
```

4.3.4 Uygulama algoritmasının sınanması

Uygulama HP Z420 iş istasyonu üzerinde hazırlanmış ve çalıştırılmıştır. İş istasyonları diğer masaüstü bilgisayarlara oranla yüksek işlem yükleri altında uzun

süreler hatasız ve kesintisiz çalışabilmektedir. İş istasyonu konfigürasyonu Çizelge 4.1’de görülmektedir.

Yazılım MATLAB uygulama ortamında geliştirilmiştir (Mathworks, 2019). MATLAB, araştırma ve mühendislik alanlarında karşılaşılan problemlere pratik, tutarlı, güvenilir, kolay kontrol edilebilir ve hızlı sonuçlar sunmakta ve tüm dünyada endüstri, devlet ve akademik kurumlarda yaygın olarak kullanılmaktadır.

İlk adımda, seçilen bir problemin çözümü için: giriş, optimizasyon ve çıkış kodları geliştirilmiştir. Giriş kodlaması, PSPLIB setinden seçilen bir problemin bilgisayar belleğine okunması işlevini yerine getirmektedir. Optimizasyon kodlaması, okunan bilgiler üzerinde çözüm yapılması için geliştirilmiştir. Çıkış kodlaması ise elde edilen değerleri görselleştirmek veya saklamak üzere uygun bir ortama aktarma işlevini üstlenmektedir. Tüm kodların geliştirilmesinde: tasarım, kodlama, sına, iyileştirme döngüsü izlenmiştir.

İkinci adımda, bir problemin çözümü için geliştirilen kodlama, tüm problemleri çözebilecek şekilde bir döngü içine alınmıştır. Döngü içerisinde çözülen her problemin sonucu, daha sonra topluca değerlendirilmek amacıyla, bir tablo yapısı halinde kullanılabilir olarak saklanmaktadır.

Problemlerin zorluk derecesine göre farklı çözüm zamanları ortaya çıkmaktadır. Çabuk çözüm elde edilen problemleri önce çözmek, daha uzun süreye ihtiyaç duyan problemleri yeniden sınamak amacı ile, ikinci adımda gerçekleştirilen döngü içerisine bir zaman sınırlayıcısı yerleştirilmiştir. Zaman sınırlayıcısına küçük bir değer verilerek tüm problemler sınanmakta, çözümü alınabilen problemlerin çözümü saklanmakta ve çözümü alınamayan problemler için bir önceki adımda verilen zaman sınırlaması iki katına çıkartılarak devam edilmektedir. Bir problemin çözümü için izin verilen üst sınır bir gün olarak kullanılmıştır.

Problemlerin çözümü tamamlandıktan sonra ortalama sapma değerleri hesaplanmış, elde edilen değerler, mevcut çalışmalardan elde edilmiş en iyi değerlerle karşılaştırılmıştır.

4.4 Uygulama Sonuçları ve Değerlendirilmesi

Dördüncü bölüm boyunca aktarılan yaklaşımların yansıtıldığı bir yazılım hazırlanmış ve PSPLIB de yer alan 2040 problemin çözümü yapılmıştır. Problemler hem baştan

başlamalı ileri yönde, hem de sondan başlamalı geri yönde işlenmiş, böylece çözülen problem sayısı 4080 olarak gerçekleşmiştir. Her problem için elde edilen tamamlanma süresi, kaynak kısıtlaması olmadan elde edilen en kısa tamamlanma süresine bölünmekte ve problem yüzdesel sapma miktarı bulunmaktadır. Sonraki adımda tüm problemlerin sapma miktarlarının ortalaması alınmaktadır. PSPLIB problemlerinin çözümü sonucunda elde edilen ortalama sapma değerleri topluca Çizelge 4.2’de görülmektedir.

Çizelge 4.2 : Çözüm sonucunda elde edilen ortalama sapma değerleri.

Problem Seti	Aktivite Sayısı	Problem Adedi	Ortalama Sapma %
J30	30	480	0,00
J60	60	480	10,52
J90	90	480	9,85
J120	120	600	31,06

Diğer çalışmalardan elde edilen sonuçlar yazarlar tarafından duyurulmakta ve genellikle beş yıl aralıklarla hazırlanan derleme çalışmalarında karşılaştırmalı olarak ele alınmaktadır. Doktora tezinde elde edilen çözüm sonuçları, rastlanan son derleme çalışmalarında verilen çözüm değerleri ile karşılaştırılmıştır. Bu çalışmalar 2014 yılına kadar olan çözüm sonuçlarını içermektedir.

Erişilen sonuçlar, J30:Çizelge 4.3, J60:Çizelge 4.4 ve J120:Çizelge 4.5’te verilmiştir (Berthaut ve diğ., 2014). Geliştirilen yazılım karşılaştırma tablolarında NRCPGA (*Tekrarsız Kromozom Destekli Paralel Genetik Algoritma, (Nonrepetitive Chromosome Improvement for Parallel Genetic Algorithm)*) olarak yer almaktadır.

PSPLIB problemlerinin çözümü sonucunda elde edilen ortalama sapma değerleri: J30 ve J60 problemlerinde aynı çözüm değerlerini veren diğer çalışmalarla birlikte en üst bantta, J120 problemlerinde altıncı sırada yer almaktadır.

Çözümler sırasında J120_35_5 numaralı problemde, –daha önceki çalışmalarda elde edilmemiş– en iyi değere erişilmiştir. Problemin çözümü onaylanmış ve PSPLIB

problemlerini barındıran Münih Teknik Üniversitesi resmi web sitesinde yer almıştır (PSPLIB TUM, 2017).

Daha önce de belirtildiği gibi, GA rastgelelik barındıran yapısı nedeniyle her çalıştırmada farklı patikalarda ilerleyebilir. Bir çalıştırmada çözümden çok uzaktayken takılıp kalabilir, başka bir çalıştırmada hızlı bir şekilde en iyi değere yaklaşabilir. Problemin yeniden test edilmesi sırasında daha iyi çözümlere ulaşılabilceği gibi, bazı problem sonuçlarına da kolaylıkla erişilemeyebilir. J120_35_5 numaralı problemde elde edilen en iyi sonuç, rastgele sayıların gülümsemesi olarak değerlendirilmektedir.

Mevcut çalışmaların başarımlar değeri, sınama problemleri üzerinden ölçülmektedir. Elde edilen sonuçların ışığında, uygulanan güçlendirmelerin ve desteklerin, bundan sonraki çalışmalarda aşağıdaki amaçlarla kullanılması faydalı olabilir:

- Farklı görünen ama aynı çözümü temsil eden kromozomların popülasyondaki varlıklarını engellemek,
- Popülasyonda daha yüksek çeşitlilik sağlamak,
- Daha küçük popülasyonla çözüme ulaşmak,
- Çözüme ulaşma süresini kısaltmak
- Çözüm kalitesini arttırmak.

Doktora tezinde geliştirilen uygulamanın başarımlar değeri, kara-kutu yaklaşımıyla, uygulamanın çıktıları üzerinden değerlendirilmektedir. Geliştirilen uygulama ve desteklerin her birinin çözüm başarımlar üzerindeki katkısı, ilerleyen çalışmalarda, farklı sınama problemleri üzerinden elde edilecek başarımlar değeri ile birlikte ele alınarak .beyaz-kutu'ya dönüştürülebilir ve açıklanabilir.

Çizelge 4.3 : PSPLIB J30 Ortalama sapma değerleri.

Algoritma	SGS	Geliştiren	1000	5000	50000
SAILS	Serial	Paraskevopoulos ve diğ., 2012	0,03	0,01	0,00
SS FBI	Serial	Berthaut ve diğ., 2014	0,10	0,02	0,00
Artificial Immune Algo FBI	Serial	Mobini ve diğ., 2011	0,05	0,03	0,00
SS FBI	Serial	Ranjbar ve diğ., 2009	0,10	0,03	0,00
GA,TS PR	Both	Kochetove Stolyar, 2003	0,10	0,04	0,00
GA FBI	Both	Wang ve diğ., 2010	0,14	0,04	0,00
GA	Both	Zamani, 2013	0,14	0,04	0,00
Bees Algo	–	Sadeghi ve diğ., 2011	0,15	0,09	0,00
NRCPGA		Bu Çalışma			0,00
ESS	Both	Mobini ve diğ., 2009	0,05	0,02	0,01
GA	Both	Mendes ve diğ., 2009	0,06	0,02	0,01
GA FBI	Serial	Gonçaves ve diğ., 2011	0,32	0,02	0,01
GA DP	Both	Cervantes ve diğ., 2008	0,16	0,04	0,01
PSO HH	Serial	Koulinas ve diğ., 2014	0,26	0,04	0,01
ACOSS	Both	Chen ve diğ., 2010	0,14	0,06	0,01
Cooling Process GA FBI	Serial	Lim ve diğ., 2013	0,21	0,07	0,01
SS FBI	Serial	Debels ve diğ., 2006	0,27	0,11	0,01
GA	Serial	Debels ve Vanhoucke, 2007	0,15	0,04	0,02
GA hybrid FBI	Serial	Valls ve diğ., 2008	0,27	0,06	0,02
Memetic Algo FBI	Serial	Carlier ve Moukrim, 2009	0,32	0,11	0,02
GA FBI	Serial	Valls ve diğ., 2005	0,34	0,20	0,02
GA FBI	Both	Alcaraz ve diğ., 2004	0,25	0,06	0,03
DABC	Serial	Nouri ve diğ., 2013	0,21	0,05	
SFLA FBI	Serial	Fang ve Wang, 2012	0,36	0,21	0,18
GANS	Serial	Proon ve Jin, 2011	1,83	1,27	0,71

Çizelge 4.4 : PSPLIB J60 Ortalama sapma değerleri.

Algoritma	SGS	Geliştiren	1000	5000	50000
GANS	Serial	Proon ve Jin, 2011	11,35	10,53	10,52
NRCPGA		Bu Çalışma			10,52
SAILS	Serial	Paraskevopoulos ve diğ., 2012	11,05	10,72	10,54
Artificial Immune Algo	Serial	Mobini ve diğ., 2011	11,17	10,80	10,55
FBI					
ESS	Both	Mobini ve diğ., 2009	11,12	10,74	10,57
GA FBI	Both	Wang ve diğ., 2010	11,55	10,96	10,57
GA FBI	Serial	Gonçalves ve diğ., 2011		11,56	10,57
SS FBI	Serial	Berthaut ve diğ., 2014	11,38	10,93	10,58
Cooling Process GA FBI	Serial	Lim ve diğ., 2013	11,73	11,14	10,63
SS FBI	Serial	Ranjbar ve diğ., 2009	11,59	11,07	10,64
GA	Both	Zamani, 2013	11,33	10,94	10,65
SFLA FBI	Serial	Fang ve Wang, 2012	11,44	10,87	10,66
ACOSS	Both	Chen ve diğ., 2010	11,75	10,98	10,67
GA	Both	Mendes ve diğ., 2009	11,72	11,04	10,67
GA	Serial	Debels ve Vanhoucke, 2007	11,45	10,95	10,68
PSO HH	Serial	Koulinas ve diğ., 2014	11,74	11,13	10,68
Memetic Algo FBI	Serial	Carlier ve Moukrim, 2009	11,62	11,09	10,70
SS FBI	Serial	Debels ve diğ., 2006	11,73	11,10	10,71
GA hybrid FBI	Serial	Valls ve diğ., 2008	11,56	11,10	10,73
GA,TS PR	Both	Kochetov ve Stolyar, 2003	11,71	11,17	10,74
GA FBI	Serial	Valls ve diğ., 2005	12,21	11,27	10,74
Bees Algo	–	Sadeghi ve diğ., 2011	11,93	11,48	10,74
GA DP	Both	Cervantes ve diğ., 2008	11,43	10,96	10,81
GA FBI	Both	Alcaraz ve diğ., 2004	11,89	11,19	10,84
DABC	Serial	Nouri ve diğ., 2013	11,74	11,16	

Çizelge 4.5 : PSPLIB J120 Ortalama sapma değerleri.

Algoritma	SGS	Geliştiren	1000	5000	50000
GANS	Serial	Proon ve Jin, 2011	33,45	31,51	30,45
ACOSS	Both	Chen ve diğ., 2010	35,19	32,48	30,56
Cooling Process GA FBI	Serial	Lim ve diğ., 2013	34,95	32,75	30,66
SAILS	Serial	Paraskevopoulos ve diğ., 2012	33,32	32,12	30,78
GA	Serial	Debels ve Vanhoucke, 2007	34,19	32,34	30,82
NRCPGA		Bu Çalışma			31,06
SFLA FBI	Serial	Fang ve Wang, 2012	34,83	33,20	31,11
SS FBI	Serial	Berthaut ve diğ., 2014	34,13	32,52	31,16
PSO HH	Serial	Koulinas ve diğ., 2014	35,20	32,59	31,23
GA hybrid FBI	Serial	Valls ve diğ., 2008	34,07	32,54	31,24
GA FBI	Both	Wang ve diğ., 2010	35,18	33,11	31,28
GA	Both	Zamani, 2013	34,02	32,89	31,30
ESS	Both	Mobini ve diğ., 2009	34,51	32,61	31,37
GA	Both	Mendes ve diğ., 2009	35,87	33,03	31,44
Artificial Immune Algo FBI	Serial	Mobini ve diğ., 2011	34,01	32,57	31,48
SS FBI	Serial	Ranjbar ve diğ., 2009	35,08	33,24	31,49
DABC	Serial	Nouri ve diğ., 2013	36,40	33,72	31,49
GA FBI	Both	Alcaraz ve diğ., 2004	36,53	33,91	31,49
Bees Algo	—	Sadeghi ve diğ., 2011	35,80	33,33	31,55
SS FBI	Serial	Debels ve diğ., 2006	35,22	33,10	31,57
GA FBI	Serial	Valls ve diğ., 2005	35,39	33,24	31,58
GA DP	Both	Cervantes ve diğ., 2008	33,71	32,57	31,65
GA,TS PR	Both	Kochetov ve Stolyar, 2003	34,74	33,36	32,06
GA FBI	Serial	Gonçalves ve diğ., 2011		35,94	32,76
Memetic Algo FBI	Serial	Carlier ve Moukrim, 2009	34,89	33,18	

5. SONUÇ VE ÖNERİLER

Kaynak kısıtlı proje çizelgeleme, pratikte birçok uygulamayı modellemek için kullanılabilen genel bir çizelgeleme problemidir. Amaç, kaynak sınırları olan bir projenin aktivitelerini, seçilen amaç doğrultusunda optimize etmektir. Kaynak kısıtlı proje çizelgeleme bir kombinatorial optimizasyon problemidir ve kaynak optimizasyonu için Proje ve Yapım Yönetmi alanında kullanılan önemli bir araçtır

Standart KKPÇP, karar değişkenleri, kısıtlamalar ve amaç fonksiyonlar ile ifade edilmektedir. Projeyi oluşturan aktivitelerin süreleri, kaynak ihtiyaçları, öncelik–sonralık ilişkileri ve kaynakların sınırları belirlenmiştir. Proje tamamlanma süresinin minimize edilmesi hedeflenmektedir. Tanımı basitçe yapılsa da, KKPÇP, NP–Hard optimizasyon problemlerinin sınıfına girer ve aslında pratikte en zorlu klasik problemlerden biridir.

Çizelgeleme problemlerinin çözümü için geliştirilen kesin yöntemler, özellikle büyük problemlerin çözümünde –pratikte kullanılamayacak kadar çok– zaman alıcıdır. Meta–sezgisel algoritmalar ise büyük boyutlara ve çok karmaşık kısıtlamalara sahip problemleri başarıyla çözen güçlü ve esnek arama mekanizmalarıdır ve makul hesaplama zamanlarında yeteri kadar iyi kalitede çözümler üretebilirler.

KKPÇP’nin meta–sezgisel algoritmalarla çözümü yoğun çalışılan bir alandır, Geliştirilen algoritmalar standart test problemleri kullanılarak sınanmakta ve elde edilen değerler önceki çalışmalarda elde edilmiş değerlerle karşılaştırılarak sunulmaktadır.

Yüksek başarımla ulaşabilen çözümlere geliştirmek motivasyonu ile tetiklenen literatür taraması ve boşluk analizi sırasında, GA’ya yönelik güçlendirme ve desteklerin etkili olabileceği gözlemlenmiş bulunmaktadır. Doktora tezinde uygulanan güçlendirme ve desteklerin ardındaki ana fikir, tekrarsızlığın sağlanması ve çeşitlendirilmenin artırılmasıdır.

GA’ ya yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi, düşen çeşitlilik ve hızlı yakınsama problemlerine karşı çözüm olarak kullanılabilecek olan bir

mekanizmadır. Tekrarlanan kromozomların uzaklaştırılması ve yerlerine –çeşitliliği sağlamak üzere– yenilerinin konması esasına dayanmaktadır.

KKPÇP’ne yönelik olarak aktarılan tekrarsız kromozom güçlendirmesi ise farklı görünen ama aynı kaynak profilini üreten kromozomların elenmesi ve yerlerine yenilerinin konulmasını sağlamaktadır.

Hibrit SA desteği, yerel iyiden kaçmak ve daha iyi çözümler içerebilecek patikalara sıçramak için kullanılmaktadır.

PGA desteği, GA’ya göre daha hızlı sonuca ulaşmak, daha küçük popülasyonla çalışmak, daha yüksek çeşitlilik sağlamak ve daha yüksek çözüm kalitesi elde etmek amaçlarına ulaşmak için kullanılmaktadır.

Doktora tezinde geliştirilen uygulama, KKPÇP’lerinin başarımlarının ölçülmesinde standart olarak kullanılan PSPLIB problemleri üzerinden sınanmıştır. Test problemlerinin çözümü için kullanılan yazılım, önemli bir prototip geliştirme ve sınama aracı olan MATLAB uygulama ortamında hazırlanmıştır. Elde edilen sonuçlar, literatürde yer alan en başarılı çalışmalarla karşılaştırılarak sunulmuştur.

5.1 Tezin Bütünlüğü

Birinci bölümde, arka plan, motivasyon, amaç, bilime katkı, yöntem ve araçlar ve doktora tezi organizasyonu ele alınmıştır.

İkinci bölümde, KKPÇP hakkında güncel literatür araştırması sunulmuştur. Literatür araştırması sınıflandırma, çözüm yöntemleri, çizelge gösterim biçimleri ve KKPÇP çözümüne yönelik çalışmalar başlıkları altında ele alınmıştır. Bu bölümde, KKPÇP ayrıntılı olarak incelenmiş ve dördüncü bölümde yer alan, hibrit paralel GA uygulaması konusuna hazırlık yapılmıştır.

Üçüncü bölümde GA incelenmiştir. GA incelemesi meta–sezgisel optimizasyon, No Free Lunch teorisi, GA ve PGA başlıkları altında ele alınmıştır. Bu bölümde, GA ayrıntılı olarak incelenmiş ve dördüncü bölümde yer alan, hibrit paralel GA uygulaması konusuna hazırlık yapılmıştır.

Dördüncü bölümde KKPÇP’ne yönelik olarak hazırlanan hibrit paralel GA uygulaması tanıtılmıştır. Uygulamanın ardındaki kavramsal esaslar, kullanılan algoritmalar ve elde edilen sonuçlar ele alınmıştır. Biri genel olarak GA’ya, diğeri

özel olarak KKPÇP'ne yönelik iki güçlendirme yaklaşımı verilmiş, ardından PGA ve hibrit SA destekleri açıklanmıştır. Önceki bölümlerde yapılan hazırlıkla ulaşılan ve uygulamanın ele alındığı dördüncü bölümün sonunda, uygulamadan elde edilen sonuçlar, literatürde yer alan çalışmalarla karşılaştırılarak verilmiştir.

5.2 Değerlendirme

Doktora tezinde geliştirilen uygulama, KKPÇP'lerinin başarımlarının değerlendirilmesinde standart olarak kullanılan PSPLIB problemleri üzerinden sınanmıştır. Problemlerinin çözümü sonucunda elde edilen ortalama sapma değerleri J30 ve J60 problemlerinde aynı çözüm değerlerini veren diğer çalışmalarla birlikte en üst bantta, J120 problemlerinde altıncı sırada yer almaktadır.

Çözümler sırasında J120_35_5 numaralı problemde, –daha önceki çalışmalarda elde edilmemiş– en iyi değere erişilmiştir. Problemin çözümü onaylanmış ve PSPLIB problemlerini barındıran Münih Teknik Üniversitesi resmi web sitesinde yer almıştır (PSPLIB TUM, 2017).

Elde edilen sonuçların ışığında, doktora tezinde uygulanan güçlendirmeler ve destekler başarılı olmuş, sınama problemlerinin çözümünde elde edilen sonuçlar, güncel en üst çözümler arasında yer almıştır. Doktora tezinde uygulanan güçlendirmeler ve destekler, KKPÇP'lerine yönelik olarak yapılacak yeni çalışmalarda yol gösterici olarak kullanılabilir.

5.3 Öneriler

Saf meta-sezgisel algoritmalar sınırlı başarımlar göstermektedirler. Daha iyi sonuçlar elde edebilmek için probleme özgü çözüm ve komşuluk yapıları geliştirilmelidir. Aşırı optimizasyon yerine basit ama etkili, yol gösterici kuralların elde edilmesine çalışılmalıdır. Genel olarak herhangi bir meta-sezgisel algoritmanın diğerine üstünlüğü saptanamamıştır. Bazı özel problem yapıları için bazı algoritmalar uygun gibi görünmekte fakat uzun dönemli ortalamalarda, baskınlığı gösterecek önemli bir fark ortaya çıkmamaktadır. No Free Lunch teoremi üzerinde yeni yapılan çalışmalar, probleme özgü meta-sezgisel algoritmaların genel meta-sezgisel algoritmalarla göre daha başarılı olabileceklerini öne sürmektedir. Son yıllarda ortaya çıkan hibrit algoritma yaklaşımı, daha iyi sonuçlara ulaşılmasında fayda sağlayabilir. Mümkünse

farklı algoritmalarda yer alan öz yaklaşımlar bir araya getirilmelidir. Problem çözümünde kullanılacak algoritmaları seçme veya ağırlıklarını belirleme amaçlı olarak ortaya atılan hiper meta-sezgisel algoritmalar önemli bir çalışma alanı olabilir. En iyi sonuçlara ulaşma konusunda iddialı olmayan fakat makul sonuçlara hızlıca yaklaşım gösteren algoritmalar göz ardı edilememelidir. Birçok durumda bir birim iyileşmenin maliyeti uygulamada katlanılamayacak kadar aşırı olabilir.

Tekrarlanan aktiviteler veya tekrarlanan alt projeler içeren KKPÇP ilginç bir çalışma alanı olabilir. Bu tip projeler için yeni gösterim ve hesaplama yolları geliştirilebilir. Aktivite sayısı çok artsa da, aktivite blokları aynı kalır. Aktivite bloklarının aynılığı üzerinden geliştirilecek algoritmalar arama uzayını önemli ölçüde daraltabilir.

Proje ve Yapım Yönetimi alanında KKPÇP üzerinde çalışılırken eldeki varsayımlara sahte bir kesinlikle tutunma konusuna dikkat edilmelidir. Aktivite süreleri yoğunlukla önceden belirlenmiştir ve önemli ölçüde varsayımlara dayanır. Kaynak kısıtları proje boyunca değişebilir. Kaynak kısıtlarında küçük bir iyileşme, proje tamamlanma süresini önemli ölçüde azaltabilir. Benzer şekilde aktivitelerin bölünebilirliği incelenmelidir. Aktiviteyi seri olarak bağlı birkaç aktiviteye bölerek planlamak, hesap ve modelleme kolaylıkları sağlayabilir. Eğer aktivitenin kesintisiz olarak devam etme zorunluluğu yoksa veya kesintili yürütmenin bir maliyeti bulunmuyorsa, kaynak dengeleme işlemi seçenekleri artar, daha kısa toplam proje tamamlanma süresi elde edilebilir. Kaynak ihtiyacı aktivite öncesinde veya aktivite başında yoğunlaşabilir, herhangi bir anda tekil yük olarak belirebilir. Bu özellikler iyi incelenmeli, aşırı kaba bir yaklaşımla düzgün yayılı yük olarak varsayılmamalıdır. Kaynak-Süre ilişkisi için duyarlılık analizleri yol gösterici olabilir. Proje ve Yapım Yönetimi alanında KKPÇP uygulamasında aşırı detaycılıkla aşırı kabalık arasında makul bir nokta bulunmalı, sistemin davranışında kaldıraç etkisi yaratan yerlere yoğunlaşmalıdır. Mümkünse öncelikle kaynak sınırı yükseltilmeli, süre ve kaynak tamponları kullanılarak, beklenen kaynak temin profili üzerindeki riskler düşürülmelidir. Gerek aktivite sürelerinin gerekse kaynakların olasılık dağılımlarına göre sisteme girdiği simülasyon yaklaşımları, olası gerçeğe yönelik daha tutarlı kestirimlere yardımcı olabilir.

KAYNAKLAR

- Abdolshah, M.** (2014). A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions. *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies*.
- Agarwal, A., Colak, S., & Eryarsoy, E.** (2006). Improvement heuristic for the flow-shop scheduling problem: An adaptive-learning approach. *European Journal of Operational Research*, 169(3), 801–815.
- Alcaraz, J., & Maroto, C.** (2001). A Robust Genetic Algorithm for Resource Allocation in Project Scheduling. *Annals of Operations Research*, 102(1), 83–109.
- Alcaraz, J., Maroto, C., & Ruiz, R.** (2004). Improving the performance of genetic algorithms for the RCPS problem. İçinde *Proceedings of the ninth international workshop on project management and scheduling* (C. 40, s. 43).
- Artigues, C., Demasse, S., Néon, E., & Sourd, F.** (2010). *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications*. Wiley. <https://doi.org/10.1002/9780470611227>
- Atashpaz-Gargari, E., & Lucas, C.** (2007). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. İçinde *Evolutionary computation, 2007. CEC 2007. IEEE Congress on* (ss. 4661–4667).
- Azencott, R.** (1992). *Simulated annealing: parallelization techniques* (C. 27). Wiley-Interscience.
- Balkaya, H.** (2011). *Kaynak Kısıtlı Çizelgeleme Problemlerinin Genetik Algoritma Yaklaşımıyla Optimizasyonu*. Dokuz Eylül Üniversitesi, Sosyal Bilimler Enstitüsü.
- Barricelli, N.** (1954). Esempi numerici di processi di evoluzione, *Methodos* (ss. 45–68).
- Battiti, R., & Brunato, M.** (2010). Reactive Search Optimization: Learning While Optimizing. İçinde *Springer* (ss. 543–571). https://doi.org/10.1007/978-1-4419-1665-5_18
- Bean, J. C.** (1994). Genetic algorithms and random keys for sequencing and optimization. *ORSA journal on computing*, 6(2), 154–160.
- Beheshti, Z., & Shamsuddin, S. M. H.** (2013). A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl*, 5(1), 1–35.

- Berthaut, F., Pellerin, R., Hajji, A., & Perrier, N.** (2014). A path relinking-based scatter search for the resource-constrained project scheduling problem. *International Journal of Project Organisation and Management*, 10(1). <https://doi.org/10.1504/IJPOM.2018.090372>
- Berthold, T., Heinz, S., Lübbecke, M. E., Möhring, R. H., & Schulz, J.** (2010). A constraint integer programming approach for resource-constrained project scheduling. İçinde *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming* (ss. 313–317).
- Bettemir, Ö. H.** (2009). *Optimization Of Time-Cost-Resource Trade-Off Problems In Project Scheduling Using Meta-Heuristic Algorithms*. Middle East Technical University, The Graduate School Of Natural And Applied Sciences.
- Bettemir, Ö. H., & Sonmez, R.** (2014). Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling. *Journal of Management in Engineering*, 31(5), 49–53. [https://doi.org/10.1061/\(ASCE\)ME.1943-5479.0000323](https://doi.org/10.1061/(ASCE)ME.1943-5479.0000323)
- Blazewicz, J., Lenstra, J. K., & Kan, A. H. G. R.** (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete applied mathematics*, 5(1), 11–24.
- Bouleimen, K., & Lecocq, H.** (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, 149(2), 268–281. [https://doi.org/10.1016/S0377-2217\(02\)00761-0](https://doi.org/10.1016/S0377-2217(02)00761-0)
- Brownlee, J.** (2011). *Clever Algorithms: Nature-Inspired Programming Recipes*. Search.
- Brucker, P., & Knust, S.** (2012). *Complex Scheduling, Second edition*. *Complex Scheduling, Second Edition*. <https://doi.org/10.1007/978-3-642-23929-8>
- Carlier, J., Moukrim, A.** (2009). A Memetic Algorithm for the Resource Constrained Project Scheduling Problem. İçinde *International Conference on Industrial Engineering and Systems Management IESM 2009*.
- Cervantes, M., Lova, A., Tormos, P., & Barber, F.** (2008). A Dynamic Population Steady-State Genetic Algorithm for the Resource-Constrained Project Scheduling Problem. İçinde *New Frontiers in Applied Artificial Intelligence* (ss. 611–620). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-69052-8_64
- Chen, W., Shi, Y., Teng, H., Lan, X., & Hu, L.** (2010). An efficient hybrid algorithm for resource-constrained project scheduling. *Information Sciences*, 180(6), 1031–1039. <https://doi.org/10.1016/j.ins.2009.11.044>
- Cheng, X., & Wu, C.** (2006). Hybrid algorithm for complex project scheduling. *Computer Integrated Manufacturing Systems*, 4, 18.

- Christofides, N., Alvarez-Valdés, R., & Tamarit, J. M.** (1987). Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, 29(3), 262–273.
- Çınar, G.** (2011). *Meta-Sezgisel Algoritmalar Kullanılarak Kaynak Kısıtlı Bilgi Teknolojisi Projelerinin Çizelgelenmesi*. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Çubukçu, R.** (2008). *Proje Yönetiminde Zaman Ve Maliyet Risklerinin Çizelgeleme Yöntemiyle Minimize Edilmesi*. Çukurova Üniversitesi, Fen Bilimleri Enstitüsü.
- Damay, J., Quilliot, A., & Sanlaville, E.** (2007). Linear programming based algorithms for preemptive and non-preemptive RCPSP. *European Journal of Operational Research*, 182(3), 1012–1022.
- Damcı, A.** (2012). *Denge Diyagramı Yönteminde Kaynak Dengelemesi İçin Bir Model Önerisi*. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Danielsson, J., & Fan, J. R.** (2018). Which numerical computing language is best: Julia, MATLAB, Python or R? Tarihinde 29 Aralık 2018, adresinden erişildi <https://voxeu.org/content/which-numerical-computing-language-best-julia-matlab-python-or-r>
- Darwin, C.** (1859). On the origins of species by means of natural selection. *London: Murray*, 247, 1859.
- Das, P. P., & Acharyya, S.** (2011). Meta-heuristic approaches for solving Resource Constrained Project Scheduling Problem: A Comparative study. *İçinde Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on (C. 2, ss. 474–478)*. <https://doi.org/10.1109/CSAE.2011.5952511>
- Dawkins, R.** (1976). *The selfish gene*. New York. Oxford university press.
- De Jong, K. A.** (2006). *Evolutionary computation: a unified approach*. MIT press.
- Debels, D., De Reyck, B., Leus, R., & Vanhoucke, M.** (2006). A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research*, 169(2), 638–653. <https://doi.org/10.1016/j.ejor.2004.08.020>
- Debels, D., & Vanhoucke, M.** (2007). A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem. *Operations Research*, 55(3), 457–469.
- Demeulemeester, E., & Herroelen, W.** (1992). A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management science*, 38(12), 1803–1818.
- Dennett, D. C.** (1995). *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. Simon & Schuster. <https://doi.org/10.1002/adom.201400043>
- Dorigo, M., & Blum, C.** (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2–3), 243–278. <https://doi.org/10.1016/j.tcs.2005.05.020>

- Eberhart, R., Shi, Y., & Others.** (2001). Particle swarm optimization: developments, applications and resources. İçinde *evolutionary computation, 2001. Proceedings of the 2001 Congress on* (C. 1, ss. 81–86).
- Elen, A.** (2011). *Çizelgeleme Probleminin Sezgisel Optimizasyon Yaklaşımıyla Çözümü*. Karabük Üniversitesi, Fen Bilimleri Enstitüsü.
- Erdal, M.** (2007). *Kısıtlı Kaynak Koşullarında Yapı Projelerinin Genetik Algoritma İle Programlanması*. Gazi Üniversitesi, Fen Bilimleri Enstitüsü.
- Fang, C., & Wang, L.** (2012). An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. *Computers & Operations Research*, 39(5), 890–901.
- Fogel, D. B.** (1993). Applying evolutionary programming to selected traveling salesman problems. *Cybernetics and systems*, 24(1), 27–36.
- Fonseca, C., & Fleming, P.** (1993). Multiobjective genetic algorithms in IEE Colloquium on Genetic Algorithms for Control Systems Engineering.
- Friedman, M.** (1975). *There's no such thing as a free lunch*. Open Court Pub Co.
- Futuyma, D., & Kirkpatrick, M.** (2017). *Evolution*. Sinauer, 2017 (Fourth Ed.).
- Gandomi, A.** (2012). Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 17(12), 4831–4845.
- Gandomi, A.** (2014). Interior search algorithm (ISA): a novel approach for global optimization. *ISA transactions*, 53(4), 1168–1183.
- Garey, M. R., & Johnson, D. S.** (1979). Computers and intractability: A guide to the theory of npcompleteness (series of books in the mathematical sciences), ed. *Computers and Intractability*, 340.
- Garfinkel, R. S., & Nemhauser, G. L.** (1972). *Integer programming*. Wiley, New York.
- Gavrilas, M.** (2010). Heuristic and metaheuristic optimization techniques with application to power systems. İçinde *Proceedings of the 12th WSEAS international conference on Mathematical methods and computational techniques in electrical engineering, Romania* (C. 9).
- Geem, Z. W., Kim, J. H., & Loganathan, G. V.** (2001). A new heuristic optimization algorithm: harmony search. *simulation*, 76(2), 60–68.
- Giffler, B., & Thompson, G. L.** (1960). Algorithms for solving production-scheduling problems. *Operations research*, 8(4), 487–503.
- Gigerenzer, G., & Gaissmaier, W.** (2011). Heuristic decision making. *Annual review of psychology*, 62, 451–482.
- Glover, F.** (1977). Heuristics for integer programming using surrogate constraints. *Decision sciences*, 8(1), 156–166.
- Glover, F.** (1990). Tabu search: A tutorial. *Interfaces*, 20(4), 74–94.

- Goldberg, D. E.** (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (1st baskı). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Gonçalves, J. F., Resende, M. G. C., & Mendes, J. J. M.** (2011). A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem. *Journal of Heuristics*, 17(5), 467–486. <https://doi.org/10.1007/s10732-010-9142-2>
- Granville, V., Krivánek, M., & Rasson, J.-P.** (1994). Simulated annealing: A proof of convergence. *IEEE transactions on pattern analysis and machine intelligence*, 16(6), 652–656.
- Habibi, F., Barzinpour, F., & Sadjadi, S. J.** (2018). Resource-constrained project scheduling problem: review of past and recent developments. *Journal of Project Management*. <https://doi.org/10.5267/j.jpm.2018.1.005>
- Haddad, O. B., Afshar, A., & Mariño, M. A.** (2006). Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization. *water resources management*, 20(5), 661–680.
- Hartmann, S.** (1999). *Project Scheduling under Limited Resources* (C. 478). Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-58627-9>
- Heinlein, R. A.** (1966). *The moon is a harsh mistress*. Ace.
- Held, M., & Karp, R. M.** (1962). A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics*, 10(1), 196–210.
- Helzer, D. M.** (2001). *An empirical study of parallel genetic algorithms for the traveling salesman and job-shop scheduling problems*. Montana State University-Bozeman, College of Engineering.
- Holland, J.** (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press.
- Hosseini, H. S.** (2007). Problem solving by intelligent water drops. İçinde *evolutionary computation, 2007. CEC 2007. IEEE congress on* (ss. 3226–3231).
- Huxley, J.** (1942). *Evolution the modern synthesis*. George Allen and Unwin.
- Igel, C.** (2014). No free lunch theorems: Limitations and perspectives of metaheuristics. İçinde *Theory and principled methods for the design of metaheuristics* (ss. 1–23). Springer.
- Jalilvand, A., Khanmohammadi, S., & Shabaninia, F.** (2005). Scheduling of sequence-dependant jobs on parallel multiprocessor systems using a branch and bound-based Petri net. İçinde *Proceedings of the IEEE Symposium on Emerging Technologies, 2005*. (ss. 334–339).
- Joyce, T., & Herrmann, J. M.** (2018). A Review of No Free Lunch Theorems, and Their Implications for Metaheuristic Optimisation. İçinde *Nature-Inspired Algorithms and Applied Optimization* (ss. 27–51). Springer.

- Kandil, A., & El-Rayes, K.** (2006). Parallel genetic algorithms for optimizing resource utilization in large-scale construction projects. *Journal of Construction engineering and Management*, 132(5), 491–498.
- Karaboga, D., & Basturk, B.** (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3), 459–471.
- Karakaya, A.** (2007). *Genetik Algoritmaların Yapı Proje Yönetiminde Kaynak Dengelemesi Amacıyla Kullanımının Modellenmesi*. Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü.
- Karnopp, D. C.** (1963). Random search techniques for optimization problems. *Automatica*, 1(2–3), 111–121.
- Kashan, A. H.** (2009). League championship algorithm: a new algorithm for numerical function optimization. İçinde *2009 International Conference of Soft Computing and Pattern Recognition* (ss. 43–48).
- Kashan, A. H.** (2015). A new metaheuristic for optimization: optics inspired optimization (OIO). *Computers & Operations Research*, 55, 99–125.
- Kelley Jr, J. E., & Walker, M. R.** (1959). Critical-path planning and scheduling. İçinde *Papers presented at the December 1-3, 1959, eastern joint IRE-AIEE-ACM computer conference* (ss. 160–173).
- Kernighan, B. W., & Lin, S.** (1970). An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2), 291–307.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P.** (1983). Optimization by simulated annealing. *science*, 220(4598), 671–680.
- Kochetov, Y., & Stolyar, A.** (2003). Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem. İçinde *Proceedings of the 3rd international workshop of computer science and information technologies CSIT'2003* (C. 132, ss. 1–4).
- Kohlmorgen, U., Schmeck, H., & Haase, K.** (1999). Experiences with fine-grained parallel genetic algorithms. *Annals of Operations Research*, 90, 203–219.
- Kolisch, R.** (1996). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, 14(3), 179–192.
- Kolisch, R., & Hartmann, S.** (1999). *Heuristic Algorithms for Solving the Resource-Constrained Project Scheduling Problem. Project scheduling: Recent models, algorithms, and applications* (C. 14). Springer Netherlands.
- Kolisch, R., & Hartmann, S.** (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal Of Operational Research*, 174(1), 23–37. <https://doi.org/10.1016/j.ejor.2005.01.065>

- Kolisch, R., & Sprecher, A.** (1996). PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, 96(1), 205–216. [https://doi.org/10.1016/S0377-2217\(96\)00170-1](https://doi.org/10.1016/S0377-2217(96)00170-1)
- Koné, O., Artigues, C., Lopez, P., & Mongeau, M.** (2013). Comparison of mixed integer linear programming models for the resource-constrained project scheduling problem with consumption and production of resources. *Flexible Services and Manufacturing Journal*, 25(1–2), 25–47.
- Koulinas, G., Kotsikas, L., & Anagnostopoulos, K.** (2014). A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. *Information Sciences*, 277, 680–693.
- Koza, J. R., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., & Lanza, G.** (2005). *Genetic Programming IV* (C. 5). New York: Springer-Verlag. <https://doi.org/10.1007/b137549>
- Krichen, S., & Chaouachi, J.** (2014). The Resource Constrained Project Scheduling Problem. *Graph-Related Optimization and Decision Support Systems*, 69–82.
- Krishnanand, K. N., & Ghose, D.** (2009). Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm intelligence*, 3(2), 87–124.
- Kuday, G.** (2014). *Yazılım Mühendisliği Yöntemleriyle Yazılım Test Süreci*. Haliç Üniversitesi, Fen Bilimleri Enstitüsü.
- Kuo, R. J., & Zulvia, F. E.** (2015). The gradient evolution algorithm: A new metaheuristic. *Information Sciences*, 316, 246–265.
- Laarhoven, P. J., & Aarts, E. H. L.** (1987). *Simulated Annealing: Theory and Applications*. Dordrecht: Springer Netherlands. <https://doi.org/10.1007/978-94-015-7744-1>
- Li, C., Bettati, R., & Zhao, W.** (1997). Static priority scheduling for ATM networks. İçinde *Real-Time Systems Symposium, 1997. Proceedings., The 18th IEEE* (ss. 264–273).
- Lim, A., Ma, H., Rodrigues, B., Tan, S. T., & Xiao, F.** (2013). New meta-heuristics for the resource-constrained project scheduling problem. *Flexible Services and Manufacturing Journal*, 25(1–2), 48–73.
- López, C. P.** (2014). *MATLAB Optimization Techniques*. Apress. <https://doi.org/10.1007/978-1-4842-0292-0>
- Lupetti, S., & Zagorodnov, D.** (2006). Data popularity and shortest-job-first scheduling of network transfers. İçinde *Digital Telecommunications,, 2006. ICDT'06. International Conference on* (s. 26).
- Madni, S. H. H., Latiff, M. S. A., Coulibaly, Y., & others.** (2016). An appraisal of meta-heuristic resource allocation techniques for IaaS cloud. *Indian Journal of Science and Technology*, 9(4).

- Malcolm, D. G., Roseboom, J. H., Clark, C. E., & Fazar, W.** (1959). Application of a technique for research and development program evaluation. *Operations research*, 7(5), 646–669.
- Mathworks.** (2019). Matlab. Tarihinde 20 Şubat 2019, adresinden erişildi <https://www.mathworks.com/products/matlab.html>
- Matyas, J.** (1965). Random optimization. *Automation and Remote control*, 26(2), 246–253.
- Mendeley.** (2019). Reference Management Software. Tarihinde 08 Ocak 2019, adresinden erişildi <https://www.mendeley.com>
- Mendes, J. J. M., Gonçalves, J. F., & Resende, M. G. C.** (2009). A random key based genetic algorithm for the resource constrained project scheduling problem. *Computers & Operations Research*, 36(1), 92–109. <https://doi.org/10.1016/j.cor.2007.07.001>
- Mercer, R. E., & Sampson, J. R.** (1978). Adaptive search using a reproductive meta-plan. *Kybernetes*, 7(3), 215–228.
- Michalewicz, Z.** (1996). Heuristic methods for evolutionary computation techniques. *Journal of Heuristics*, 1(2), 177–206.
- Mobini, M., Mobini, Z., & Rabbani, M.** (2011). An Artificial Immune Algorithm for the project scheduling problem under resource constraints. *Applied Soft Computing*, 11(2), 1975–1982.
- Mobini, M., Rabbani, M., Amalnik, M. S., Razmi, J., & Rahimi-Vahed, A. R.** (2009). Using an enhanced scatter search algorithm for a resource-constrained project scheduling problem. *Soft Computing*, 13(6), 597–610.
- Moore, L. J., & Clayton, E. R.** (1976). *GERT modeling and simulation: Fundamentals and applications*. Krieger Publishing Co., Inc.
- Mori, K., Tsukiyama, M., & Fukada, T.** (1993). Immune Algorithm with Searching Diversity and its Application to Resource Allocation Problem. *Trans. of the Institute of Electrical Engineers of Japan*, 113–C(10), 872–878.
- Moscato, P., & others.** (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826, 1989.
- Moumene, K., & Ferland, J. A.** (2008). New representation to reduce the search space for the resource-constrained project scheduling problem. *RAIRO - Operations Research*, 42(2), 215–228. <https://doi.org/10.1051/ro:2008010>
- Mühlenbein, H.** (1992). Parallel Genetic Algorithms in Combinatorial Optimization. *Computer Science and Operations Research*, 441–453. <https://doi.org/10.1016/B978-0-08-040806-4.50034-4>

- Mutalik, P. P., Knight, L. R., Blanton, J. L., & Wainwright, R. L.** (1992). Solving combinatorial optimization problems using parallel simulated annealing and parallel genetic algorithms. İçinde *Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing: technological challenges of the 1990's* (ss. 1031–1038).
- Neshat, M.** (2013). Swallow swarm optimization algorithm: a new method to optimization. *Neural Computing and Applications*, 23(2), 429–454.
- Nouri, N., Krichen, S., Ladhari, T., & Fatimah, P.** (2013). A discrete artificial bee colony algorithm for resource-constrained project scheduling problem. İçinde *2013 5th International Conference on Modeling, Simulation and Applied Optimization (ICMSAO)* (ss. 1–6).
- Oguz, O., & Bala, H.** (1994). A comparative study of computational procedures for the resource constrained project scheduling problem. *European Journal of Operational Research*, 72(2), 406–416.
- Özdemir, G.** (2006). *Kısıtlı Kaynaklarla Proje Çizelgelemesi Problemlerinde Kullanılan Genetik Algoritma Metodları Ve Bunların Karşılaştırılması*. Ankara Üniversitesi, Fen Bilimleri Enstitüsü.
- Özleyen, E.** (2011). *A Genetic Algorithm For The Resource Constrained Project Scheduling Problem*. Middle East Technical University, The Graduate School Of Natural And Applied Sciences.
- Paksoy, S.** (2007). *Genetik Algoritma İle Proje Çizelgeleme*. Çukurova Üniversitesi , Sosyal Bilimler Enstitüsü.
- Paraskevopoulos, D. C., Tarantilis, C. D., & Ioannou, G.** (2012). Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm. *Expert Systems with Applications*, 39(4), 3983–3994. <https://doi.org/10.1016/j.eswa.2011.09.062>
- Patterson, J. H.** (1984). A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. *Management science*, 30(7), 854–867.
- Pritsker, A. A. B., Waiters, L. J., & Wolfe, P. M.** (1969). Multiproject scheduling with limited resources: A zero-one programming approach. *Management science*, 16(1), 93–108.
- Proon, S., & Jin, M.** (2011). A genetic algorithm with neighborhood search for the resource-constrained project scheduling problem. *Naval Research Logistics (NRL)*, 58(2), 73–82. <https://doi.org/10.1002/nav.20439>
- PSPLIB TUM.** (2017). PSPLIB Single Mode Data Sets. Tarihinde 16 Ekim 2018, adresinden erişildi <http://www.om-db.wi.tum.de/psplib/files/j120hrs.sm>
- Publish_or_Perish.** (2019). Academic Citations Software. Tarihinde 08 Ocak 2019, adresinden erişildi <https://harzing.com/resources/publish-or-perish>
- Quoteinvestigator.com.** (2018). There Ain't No Such Thing as a Free Lunch. Tarihinde 08 Kasım 2018, adresinden erişildi <https://quoteinvestigator.com/2016/08/27/free-lunch/>

- Ranjbar, M., De Reyck, B., Kianfar, F., Reyck, B. De, & Kianfar, F.** (2009). A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling. *European Journal of Operational Research*, 193(1), 35–48. <https://doi.org/10.1016/j.ejor.2007.10.042>
- Rao, R. V, Savsani, V. J., & Vakharia, D. P.** (2011). Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303–315.
- Rashedi, E.** (2009). GSA: a gravitational search algorithm. *Information sciences*, 179(13), 2232–2248.
- Rezaie, K., Amalnik, M. S., Gereie, A., Ostadi, B., & Shakhseeniaee, M.** (2007). Using extended Monte Carlo simulation method for the improvement of risk management: Consideration of relationships between uncertainties. *Applied Mathematics and Computation*, 190(2), 1492–1501. <https://doi.org/10.1016/J.AMC.2007.02.038>
- Russell, S. J., & Norvig, P.** (2009). *Artificial Intelligence: A Modern Approach*, 3rd edition. Pearson. <https://doi.org/10.1017/S0269888900007724>
- Sadeghi, A., Kalanaki, A., Noktehdan, A., Samghabadi, A. S., & Barzinpour, F.** (2011). Using bees algorithm to solve the resource constrained project scheduling problem in PSPLIB. İçinde *Theoretical and Mathematical Foundations of Computer Science* (ss. 486–494). Springer.
- Salcedo-Sanz, S., & Del Ser, J.** (2013). The Coral Reefs Optimization Algorithm: A new metaheuristic algorithm for hard optimization problems. İçinde *Proc. of the 15th International Conference on Applied Stochastic Models and Data Analysis (ASMDA)*, Mataró, Barcelona.
- Sampson, S. E., & Weiss, E. N.** (1993). Local search techniques for the generalized resource constrained project scheduling problem. *Naval Research Logistics (NRL)*, 40(5), 665–675.
- Sari, T.** (2008). *Metasezgisel Yöntemlerle Proje Çizelgeleme Optimizasyonu*. Marmara Üniversitesi, Sosyal Bilimler Enstitüsü.
- Satıç, U.** (2014). *Çok Kaynak Kısıtlı Projelerin Sezgisel Yöntemlerle Çizelgelenmesi*. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Schwefel, H. P.** (1981). *Numerical optimization of computer models*. John Wiley & Sons, Inc.
- Shah-Hosseini, H.** (2011). Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering*, 6(1–2), 132–140.
- Sivanandam, S. N., & Deepa, S. N.** (2008). *Introduction to genetic algorithms* (1. baskı). Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73190-0_2
- Storn, R., & Price, K.** (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341–359.

- Sumathi, S., Hamsapriya, T., & Surekha, P.** (2008). *Evolutionary Intelligence An Introduction to Theory and Applications with Matlab*. Springer-Verlag Berlin Heidelberg.
- Talbi, E.-G.** (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Tamura, K., & Yasuda, K.** (2011). Spiral optimization-A new multipoint search method. *İçinde Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on* (ss. 1759–1764).
- Taylor, B.** (1978). Project management using GERT analysis. *Proj. Manag. Q*, 9(3), 15–20.
- Uslu, F.** (2016). *Melez Sezgisel Ve Meta-Sezgisel Algoritmalar Kullanarak Çizelgeleme Problemlerinin Çözümlemesi*. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Valls, V., Ballestin, F., & Quintanilla, S.** (2005). Justification and RCPSP: A technique that pays. *European Journal of Operational Research*, 165(2), 375–386.
- Valls, V., Ballestin, F., & Quintanilla, S.** (2008). A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 185(2), 495–508. <https://doi.org/10.1016/j.ejor.2006.12.033>
- Wang, H., Li, T., & Lin, D.** (2010). Efficient genetic algorithm for resource-constrained project scheduling problem. *Transactions of Tianjin University*, 16(5), 376–382. <https://doi.org/10.1007/s12209-010-1495-y>
- Wang, L., & Zheng, D.-Z.** (2001). An effective hybrid optimization strategy for job-shop scheduling problems. *Computers & Operations Research*, 28(6), 585–596. [https://doi.org/10.1016/S0305-0548\(99\)00137-9](https://doi.org/10.1016/S0305-0548(99)00137-9)
- Widmer, M., Hertz, A., & Costa, D.** (2008). Metaheuristics and Scheduling. *Production Scheduling*, 33–68.
- Wolpert, D. H., & Macready, W. G.** (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1), 67–82.
- Yang, B., Geunes, J., J. W., & O'Brien.** (2001). *Resource-constrained project scheduling: Past work and new directions*.
- Yang, X.-S.** (2009). Firefly algorithms for multimodal optimization. *İçinde International symposium on stochastic algorithms* (ss. 169–178).
- Yang, X.-S.** (2010). A new metaheuristic bat-inspired algorithm. *İçinde Nature inspired cooperative strategies for optimization (NICSO 2010)* (ss. 65–74). Springer.
- Yang, X.-S., & Deb, S.** (2009). Cuckoo search via Levy flights. *İçinde Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on* (ss. 210–214).

- Yazdani, M., & Jolai, F.** (2016). Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering*, 3(1), 24–36.
- Yücenur, G. N.** (2011). *Optimizasyon Problemlerinin Çözümünde Melez Metasezgisel Bir Algoritmanın Tasarımı*. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü.
- Zamani, R.** (2013). A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem. *European Journal of Operational Research*, 229(2), 552–559. <https://doi.org/10.1016/j.ejor.2013.03.005>
- Zheng, Y.-J.** (2015). Water wave optimization: a new nature-inspired metaheuristic. *Computers & Operations Research*, 55, 1–11.



EKLER

EK A: KKPÇP için diğer önemli kaynaklar



EK A: KKPÇP için diğer önemli kaynaklar

Çizelge A.1 : KKPÇP için PSPLIB çözümü bildiren önemli çalışmalar.

Çalışma	Yazar
A path relinking-based scatter search for the resource-constrained project scheduling problem	Berthaut ve diğ., 2014
A particle swarm optimization based hyper-heuristic algorithm for the classic resource	Koulinas ve diğ., 2014
A competitive magnet-based genetic algorithm for solving the resource-constrained project scheduling problem	Zamani, 2013
New meta-heuristics for the resource-constrained project scheduling problem	Lim ve diğ., 2013
A discrete artificial bee colony algorithm for resource-constrained project scheduling problem	Nouri ve diğ., 2013
Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm	Paraskevopoulos ve diğ., 2012
An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem	Fang ve Wang, 2012
Using an enhanced scatter search algorithm for a resource-constrained project scheduling problem	Mobini ve diğ., 2011
Using bees algorithm to solve the resource constrained project scheduling problem in PSPLIB	Sadeghi ve diğ., 2011
A biased random-key genetic algorithm with forward-backward improvement for the resource constrained project scheduling problem	Gonçalves ve diğ., 2011
A genetic algorithm with neighborhood search for the resource-constrained project scheduling problem	Proon ve Jin, 2011
Efficient genetic algorithm for resource-constrained project scheduling problem	Wang ve diğ., 2010
An efficient hybrid algorithm for resource-constrained project scheduling	Chen ve diğ., 2010
A hybrid scatter search for the discrete time/resource trade-off problem in project scheduling	Ranjbar ve diğ., 2009
Using an enhanced scatter search algorithm for a resource-constrained project scheduling problem	Mobini ve diğ., 2009
A random key based genetic algorithm for the resource constrained project scheduling problem	Mendes ve diğ., 2009
A Memetic Algorithm for the Resource Constrained Project Scheduling Problem	Carlier ve Moukrim, 2009
A dynamic population steady-state genetic algorithm for the resource-constrained project scheduling problem	Cervantes ve diğ., 2008
A hybrid genetic algorithm for the resource-constrained project scheduling problem	Valls ve diğ., 2008
A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem	Debels ve Vanhoucke, 2007
A hybrid scatter search/electromagnetism meta-heuristic for project scheduling	Debels ve diğ., 2006
Justification and RCPSP: A technique that pays	Valls ve diğ., 2005
Improving the performance of genetic algorithms for the RCPS problem	Alcaraz ve diğ., 2004
Evolutionary local search with variable neighborhood for the resource constrained project scheduling problem	Kochetov ve Stolyar, 2003

Çizelge A.2 : KKPÇP için Ulusal Tez Merkezi’nden erişilen yeni çalışmalar.

Tez	Yazar
Melez Sezgisel Ve Meta-Sezgisel Algoritmalar Kullanarak Çizelgeleme Problemlerinin Çözümlemesi	Uslu, 2016
Çok Kaynak Kısıtlı Projelerin Sezgisel Yöntemlerle Çizelgelenmesi	Satıç, 2014
Denge Diyagramı Yönteminde Kaynak Dengelemesi İçin Bir Model Önerisi	Damcı, 2012
Optimizasyon Problemlerinin Çözümünde Melez Meta-Sezgisel Bir Algoritmanın Tasarımı	Yücenur, 2011
Kaynak Kısıtlı Çizelgeleme Problemlerinin Genetik Algoritma Yaklaşımıyla Optimizasyonu	Balkaya, 2011
A Genetic Algorithm For The Resource Constrained Project Scheduling Problem	Özleyen, 2011
Meta-Sezgisel Algoritmalar Kullanılarak Kaynak Kısıtlı Bilgi Teknolojisi Projelerinin Çizelgelenmesi	Çınar, 2011
Çizelgeleme Probleminin Sezgisel Optimizasyon Yaklaşımıyla Çözümü	Elen, 2011
Optimization Of Time–Cost–Resource Trade–Off Problems In Project Scheduling Using Meta–Heuristic	Bettemir, 2009
Proje Yönetiminde Zaman Ve Maliyet Risklerinin Çizelgeleme Yöntemiyle Minimize Edilmesi	Çubukçu, 2008
Metasezgisel Yöntemlerle Proje Çizelgeleme Optimizasyonu	Sari, 2008
Kısıtlı Kaynak Koşullarında Yapı Projelerinin Genetik Algoritma İle Programlanması	Erdal, 2007
Genetik Algoritma İle Proje Çizelgeleme	Paksoy, 2007
Genetik Algoritmaların Yapı Proje Yönetiminde Kaynak Dengelemesi Amacıyla Kullanımının Modellenmesi	Karakaya, 2007
Kısıtlı Kaynaklarla Proje Çizelgelenmesi Problemlerinde Kullanılan Genetik Algoritma Metodları Ve Bunların Karşılaştırılması	Özdemir, 2006

ÖZGEÇMİŞ
(CURRICULUM VITAE)

Ad-Soyad : Şafak EBESEK
(Name–Surname)

Doğum Tarihi : 1960
(Date of Birth)

E-posta : safak@hotmail.com
(E-mail) safakebesek@yahoo.com



ÖĞRENİM DURUMU:
(EDUCATION)

Yükseklisans : 1988, İstanbul Teknik Üniversitesi, Yapı Bilimleri,
Endüstrileşmiş Yapım Sistemleri ve Yapım Yönetimi

(M.Sc.) (1988, Project and Construction Management, Graduate School
of Science Engineering and Technology, Istanbul Technical
University)

Lisans : 1984, İstanbul Teknik Üniversitesi, Mimarlık Fakültesi,
Mimarlık Bölümü

(B.Sc.) (1984, Architecture, Faculty of Architecture, Istanbul Technical
University)

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR:
(PUBLICATIONS/PRESENTATIONS ON THE THESIS)

Ebesek Ş. & Yaman H., Parallel Genetic Algorithm Application With Nonrepetitive Chromosome Improvement For Resource Constrained Project Scheduling Problem, *The Journal of International Social Research*, Vol. 12, No: 62, Şubat 2019, s. 519-529, ISSN: 1307-9581

Ebesek Ş. & Yaman H., Conceptual Perspective Via Meta-Heuristic Optimization Algorithms For Resource Leveling Problem In Project Networks, 3. *International Conference on Civil and Environmental Engineering*, İzmir/TÜRKİYE, Vol. 2, No. 1, 24 Nisan 2018, s. 648-657

Tüz, Ö. & **Ebesek, Ş.**, “A Glance at Artificial Intelligence Based Metaheuristic Algorithms in Optimizable Architectural Design Process” *1st International Symposium on Architecture and Design*, Istanbul Association of Science and Academicians, İstanbul, 01 – 02 December 2017

Ebesek Ş. & Yaman H, “Meta Heuristic Optimization Algorithms on Resource Leveling Problem in Project Networks”, *3. Project and Construction Management Congress*, Akdeniz University, 6-8 November 2014 Antalya, Turkey

BİLİMSEL TOLANTILAR VE SUNUMLAR:

(PUBLISHED PRESENTATIONS AT SCIENTIFIC MEETINGS)

Tüz, Ö. & **Ebesek Ş.**, “Overview Of Risk Management In Subcontractor Selection For Construction Projects” *3.th ICOCEE–Cesme (International Conference on Civil and Environmental Engineering)*, Sheraton Çeşme Hotel Resort & Spa in Çeşme, İzmir, 24 – 27 April 2018

Tüz, Ö. & **Ebesek Ş.**, “A General Glance at Risk Attitude of Construction Professionals” *3.th ICOCEE–Cesme (International Conference on Civil and Environmental Engineering)*, Sheraton Çeşme Hotel Resort & Spa in Çeşme, İzmir, 24 – 27 April 2018

Tüz, Ö. & **Ebesek, Ş.**, “Financial Risk Overview in Construction Projects” *7th Construction Management Congress*, TMMOB Chamber of Civil Engineers, Samsun, 06 – 07 October 2017

Tüz, Ö. & Ebesek, Ş., “Cash Flow Risk in Construction Projects” *7th Construction Management Congress*, TMMOB Chamber of Civil Engineers, Samsun, 06 – 07 October 2017

Tüz, Ö. & **Ebesek, Ş.**, “An Algorithmic Approach to Restaurant Design” *1st Futourism Congress (1st International Congress on Future of Tourism: Innovation, Entrepreneurship and Sustainability - 2017)*, Mersin University, Mersin, 28 – 30 September 2017

Tüz, Ö. & **Ebesek, Ş.**, “Overview of Design Criteria in Food and Beverage Business” *1st Futourism Congress (1st International Congress on Future of Tourism: Innovation, Entrepreneurship and Sustainability - 2017)*, Mersin University, Mersin, 28 – 30 September 2017

Tüz, Ö. & **Ebesek, Ş.**, “Overview of Financial Risks in Construction Projects” *2nd ICOCEE–Cappadocia (International Conference on Civil and Environmental Engineering)*, Nevşehir Hacı Bektaş Veli University, Nevşehir, 8 – 10 May 2017

Tüz, Ö. & **Ebesek, Ş.**, “Cash-Flow Risk Management for Construction Companies at Project Portfolio Level” *2nd ICOCEE–Cappadocia (International Conference on Civil and Environmental Engineering)*, Nevşehir Hacı Bektaş Veli University, Nevşehir, 8 – 10 May 2017

Tüz, Ö. & **Ebesek, Ş.**, “Monte Carlo Simulation in Project Cash Flow Distortion Risk Estimation” *4th Project and Construction Management Congress*, Anadolu University, Eskişehir, 3-5 November 2106

Tüz, Ö. & **Ebesek, Ş.**, “A Model Suggestion to Predict Leverage Ratio for Construction Projects” *ACE2014, 11th International Congress on Advances in Civil Engineering, International*, İstanbul Technical University, İstanbul, 21-25 October 2014

Tüz, Ö. & **Ebesek, Ş.**, “Construction Project Risks from the Perspective of Project Management” *The 23rd SRA (The Society for Risk Analysis)-Europe Conference*, İstanbul Technical University, İstanbul, 16-18 June 2014

Tüz, Ö. & **Ebesek, Ş.**, “A Model Suggestion to Measure the Restaurant Design Performance, *NWSA-Engineering Sciences*, Vol 9, Issue I (2014)

Tüz, Ö. & **Ebesek, Ş.**, “A Model Suggestion to Predict Leverage Ratio for Construction Projects,” *Journal of Business Research-Turk*, Vol. 5, Issue 4, 227-239 (2013)

Ebesek Ş. & Yaman H, “A Glance at Complexity and System Approach in Project Management”, *2nd Project and Construction Management Congress*, Izmir Institute of Technology, Gülbahçe, İzmir, 13-16 September 2012

Ebesek, Ş., ‘Risk Management Templates’ , *IPYD Dinamikler 2007 Project Management Congress*, Istanbul, , Organized by IPYD and Kadir Has University, Istanbul 13 April 2007-15 April 2007

Ebesek, Ş., ‘Complexity Chaos and Uncertainty in Project Management’, *IPYD Dinamikler 2002 Project Management Congress*, Istanbul, Organized by IPYD and Istanbul Kultur University, Istanbul, 4 April 2002 - 5 April 2002

TEZLER (*THESIS*)

Ebesek, Ş., “A Model for Determining Cost-Time Trade-offs in Building Construction”, M Arch Thesis, *İstanbul Technical University, Institute of Science and Technology*, Istanbul, June, 1988.

YAYINLAR: (*LIST OF JOURNAL PAPERS*)

Tüz, Ö. & **Ebesek, Ş.**, “A Model Suggestion to Measure the Restaurant Design Performance, *NWSA-Engineering Sciences*, Vol 9, Issue I ,2014)

Tüz, Ö. & **Ebesek, Ş.**, “A Model Suggestion to Predict Leverage Ratio for Construction Projects,” *Journal of Business Research-Turk*, Vol. 5, Issue 4, 227-239 ,2013)

YARIŞMA VE ÖDÜLLER: (*AWARDS & COMPETITIONS*)

Ebesek Ş. & Yaman H, "Best minimum solution of j12035_5 problem in PSPLIB problem set". Accepted and Registered at Technical University of Munich , TUM web site: <http://www.om-db.wi.tum.de/psplib/files/j120hrs.sm>, 2017

AKADEMİK POZİSYONLAR VE VERİLEN DERSLER: (*ACADEMIC POSITIONS & TEACHING EXPERIENCE*)

Visiting Scholar, Technische Universität Dresden, Institute of Construction Informatics, Department of Civil Engineering, 2015.08–2015.09

"CM 540 Risk Management in Construction ", Istanbul Bilgi University, Construction Management Graduate Program, 2013-2017

"CM 541 Cost Management in Construction ", Istanbul Bilgi University, Construction Management Graduate Program, 2013-2017

"IND 471 Professional Practice Management ", Istanbul Bilgi University, Faculty of Architecture, Department of Interior Design, 2013- 2017

"ARCH 472 Professional Practice Management ", Istanbul Bilgi University, Faculty of Architecture, Department of Architecture, 2015- 2017

"FCP 005 Computerized Project Planning Techniques ", Istanbul Kültür University, Building Science and Technology, 2007-2009

EĞİTİM FAALİYETLERİ: (TEACHING EXPERIENCE)

“Advanced Cost Management in Construction”, Certification Programs, 2008-2010

“Project and Risk Management in Construction”, Certification Programs, 2008-2010

“Computerized Solutions and Algorithms for Civil Engineering Problem”, Chamber of Civil Engineers İstanbul, 1986-1988

İLGİ VE ARAŞTIRMA ALANLARI (RESEARCH INTERESTS)

Artificial Intelligence, Optimization, Meta Heuristic Methods , Combinatorial Optimization, Evolutionary Algorithms, Graph Theory, Scheduling, Resource Leveling in Project Networks,

Project Management, Risk Management, Financial Risk Management,

Trading Systems, Quantitative Finance,

Drinking Water Networks Optimization, Simulation and Sizing.

Information System in Construction Management, Data Mining,

BİLGİ-İŞLEM BECERLERİ: (COMPUTER SKILLS)

Interest : Python, Grasshopper, Rhino, R, RapidMiner

Actual : MatLab, EPANET, Delphi, FireBirdSql, MySQL,

Formerly : Basic, Fortran77, Xbase, Clipper, Java

Construction: : MS Project, Primavera, ACAD, EPANET,

Web : Joomla, WampManager, Php, PHPMaker, MySql

Trading : TradeStation, MetaTrader 5

ÇALIŞMA GEÇMİŞİ VE GELİŞTİRLEN ÖNEMLİ YAZILIMLAR:
(MAJOR PROFESSIONAL EXPERIENCE AND SOFTWARE DESIGN & DEVELOPMENT)

Drinking Water Networks Optimization, Simulation and Sizing
SEDES INFRASTRUCTURE Ltd, Istanbul, Ankara, 2011-Ongoing,
More than 9000 km pipelines & 50 Country of drinking water network optimization
with MATLAB & EPANET.

Construction Management Information System
MESA-NUROL Joint Venture Bahcesehir Project, Istanbul, 1997-2009

Cost Management in Architectural Offices
SPACE Arch LTD, Istanbul, 2009

Risk Management and Auditing
for UBD A.S. (National Independent Auditors and Certified Accountants), Istanbul,
2006-2008

Computer Based Office Automation System
PİRAMİT Arch, Urban Plan & Eng Comp, Istanbul, 2008.

Partnership for San Francisco Project on JAVA, Istanbul, 1997-1999
IBM, Istanbul, 1997-1999

Project Management Information System
PROMET Project and Construction Management A.Ş, Istanbul, 1996-1998

Multi-Level Marketing System
YASAMNET A.Ş, Istanbul, 1997-2001

International Cost Management System
NISHO-IWAI A.Ş., Istanbul, 1997-2000

Information and Documentation Management System
TARİH VAKFI, Istanbul, 1997

Total Quality Control Project- Planned Maintenance and Spare Parts System
MARSHALL Boya A.Ş (Akzo-Nobel), Kocaeli, 1991-1996

Spare Parts Supply System Of Turkish Naval Forces
AYKOR A.Ş, Istanbul, 1991-1992

The Central Bank of Turkey, Risk Centralization Project
DISBANK A.Ş., Istanbul, 1990
TURKISHBANK A.Ş., Istanbul, 1990

Modules in Order Management System between Brokers and Borsa Istanbul
TEPUM A.Ş., Istanbul, 1987-1988

Address Bank and Bookstore Management
YEM A.Ş.(The Center of Building Information System), Istanbul, 1986-1993

Address Bank and Accounting System
FINANCE CLUB, Istanbul, 1987-1993

Construction Management and Accounting System
YILMAZ İNŞ.LTD, Istanbul, 1986-1993

YURTDIŐI ÇALIŐMA GEÇMİŐİ

(MAJOR PROFESSIONAL EXPERIENCE OUT OF TURKEY)

Spare Parts Supply and Trading for International Construction Companies

Diesel Engine Parts Handels-GmbH, Venekotenweg 104, 41372 Niederkrüchten,
Deutschland

Geschäftsführer (CEO), August 1994 - Oktober 1995

<https://www.companyhouse.de/Diesel-Engine-Parts-Handels-GmbH-Niederkruechten>

