



Priority rule for resource constrained project planning problem with predetermined work package durations

Pelin Çetin*^{ID}, Ö. Özgür Tanrıöver^{ID}

Ankara University, Faculty of Engineering, Computer Engineering Department, Ankara, 06830, Turkey

Highlights:

- Comparison of priority rules for the RCPSP
- Examination of hybrid rules for the RCPSP
- Proposition of a new priority rule for the RCPSP

Keywords:

- Resource Constrained Project Scheduling
- Project Management
- Project Planning
- Priority Rules
- Scheduling

Article Info:

Research Article
Received: 28.03.2019
Accepted: 22.02.2020

DOI:

10.17341/gazimmfd.545873

Correspondence:

Author: Pelin Çetin
e-mail: pelincetin.cs@gmail.com
phone: +90 312 298 1225

Graphical/Tabular Abstract

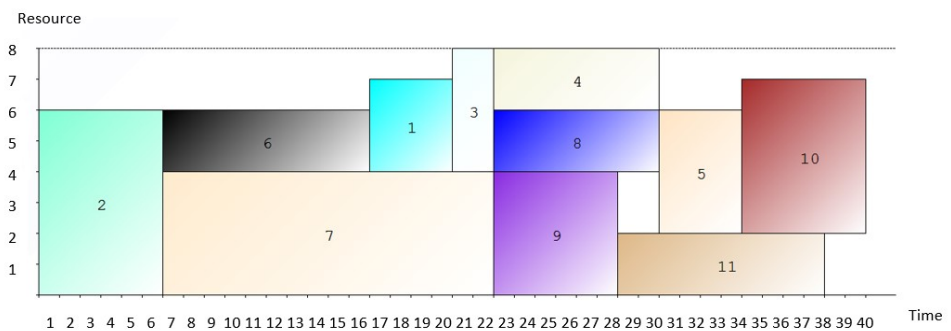


Figure A. A project schedule example for a single resource

Purpose: Project planning has a significant role in project management to progress regularly and complete a project on time. The studies for Resource Constrained Project Scheduling Problem, where the available resources are limited, indicate that the choice of priority rule affects the success of planning and there is not a single priority rule for reaching the optimum solution for all problems. In this study, a priority rule that can be used to solve Resource Constrained Project Scheduling Problem with predetermined work packages is proposed.

Theory and Methods:

In order to explore the most common 8 priority rules PSPLib J30, J60, and J120 datasets are used. Since the success of the rules changed according to the dataset, *Maximum Area* and *Longest Progressing Time* rules are selected from the priority rules, which give the most successful results, and examined. Two priority rules, which are *Area and Resource Usage* and *Area and Process Time*, has been proposed and tested on the PSPLib J30, J60, J120 dataset. The success of the rules is measured with the number of shortest schedule generation and average deviation for both serial and parallel scheduling.

Results:

In general, it is observed that the proposed *Area and Resource Usage* rule gives successful results in both serial and parallel scheduling, and it produces better results than the rules; *Maximum Area*, *Highest Resource Usage*, and *Longest Progressing Time* which are used separately. Also, the proposed rule was one of the two most successful one among the examined rules. It gave 57% better results than the *First Eligible Activity* rule, and produced the best results in the j120 dataset, where both the sample and the number of activities were higher. The proposed rule benefits from the shortest time generation feature of *Maximum Area* rule and the advantage of *Highest Resource Usage* rule that produces the closest to the optimum. In this way, it is able to produce more numbers of schedule which are close to optimum. In addition, the proposed *Area and Resource Usage* rule produces solutions below the average execution time of the other tested rules.

Conclusion:

The increase in the number of activities and samples leads to a decrease in the number of shortest-time schedule production for all priority rules and to an increase in the mean deviation. However, it was found that *Maximum Area* rule was successful in producing the shortest schedule and *Highest Resource Usage* rule was successful resulting in the lowest average deviation value. Therefore, a hybrid priority rule that takes these two rules into consideration is proposed. The proposed rule is successful when both the number of shortest-term schedule generation and average deviation are taken into consideration. It also performed in a time period below average execution time when compared with others.



İş paketi sürelerinin belirli olduğu kaynak kısıtlı proje planlama problemi için öncelik kuralı

Pelin Çetin*^{ID}, Ö. Özgür Tanrıöver^{ID}

Ankara Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Ankara, 06830, Türkiye

Ö N E Ç I K A N L A R

- RCPSP için öncelik kurallarının karşılaştırması
- RCPSP için melez kuralların incelenmesi
- RCPSP için yeni bir öncelik kuralı önerisi

Makale Bilgileri

Araştırma Makalesi
Geliş: 28.03.2019
Kabul: 22.02.2020

DOI:

10.17341/gazimmfd.545873

Anahtar Kelimeler:

Kaynak kısıtlı proje planlama,
proje yönetimi,
proje planlama,
öncelik kuralları,
çizelgeleme

ÖZET

Proje planlaması, projenin zamanında tamamlanabilmesi ve düzenli bir şekilde ilerleyebilmesi için proje yönetiminde büyük önem taşımaktadır. Kullanılabilecek kaynağın sınırlı olduğu Kaynak Kısıtlı Proje Planlama Problemi için yapılan çalışmalar, bütün problemler için optimum çözümü bulmada kullanılabilecek bir öncelik kuralı bulunmadığını ve öncelik kuralı seçiminin önemli olduğunu göstermektedir. Bu çalışmada iş paketleri sürelerinin ve talep edilen kaynakların belirli olduğu Kaynak Kısıtlı Proje Planlama Problemi probleminin çözümü için kullanılan öncelik kuralları incelenmiştir. En kısa süreli çizelge üretme sayıları kıyaslandığında, iş paketinin süresi ve talep ettiği kaynak miktarının çarpımıyla elde edilen “alan” değerinden en büyüğünün önceliklendirildiği En Büyük Alan kuralı ile en fazla işlem süresine sahip iş paketinin önceliklendirildiği En Uzun İşlem Süresi kuralının başarılı olduğu görülmüştür. Ortalama sapma değerlerine göre ise en çok kaynak kullanan iş paketinin önceliklendirildiği En Çok Kaynak Kullanımı kuralı başarılı sonuçlar vermiştir. Bu nedenle iş paketlerinin nasıl önceliklendirileceği konusunda Alan ve İşlem Süresi ile Alan ve Kaynak Kullanımı olmak üzere iki farklı sezgisel yaklaşım önerilmiştir. Önerilen yaklaşımların başarısını ölçmek için PSPLib j30, j60 ve j120 veri kümeleri kullanılmıştır. Elde edilen sonuçlar, Alan ve Kaynak Kullanımı kuralının hem en kısa süreli çizelge üretme sayısında hem de ortalama sapma değerinde başarılı sonuçlar verdiğini göstermektedir.

Priority rule for resource constrained project planning problem with predetermined work package durations

H I G H L I G H T S

- Comparison of priority rules for the RCPSP
- Examination of hybrid rules for the RCPSP
- Proposition of a new priority rule for the RCPSP

Article Info

Research Article
Received: 28.03.2019
Accepted: 22.02.2020

DOI:

10.17341/gazimmfd.545873

Keywords:

Resource constrained project scheduling,
project management,
project planning,
priority rules,
scheduling

ABSTRACT

Project planning has a significant role in project management to progress regularly and complete the project on time. The studies for Resource Constrained Project Scheduling problem, where the available resource is limited, indicate that since there is no priority for finding the optimum solution for all problems, the choice of priority rule affects the success of planning. In this study, priority rules used for the solution of Resource Constrained Project Scheduling problem, where the duration of the work packages and the requested resources are determined, are examined. When the shortest schedule generation numbers are compared, it was seen that Maximum Area rule, which prioritizes the work package with the highest value obtained by multiplying the duration and amount of resource usage of the package, and Longest Processing Time rule, which prioritizes the work package with the highest process time, were successful. According to the average deviations, Highest Resource Usage rule, which prioritizes the most resource-consuming work package yielded successful results. Hence, two different approaches, “Area and Processing Time” and “Area and Resource Usage”, have been proposed. The success of the proposed approach was measured by using PSPLib j30, j60, and j120 datasets. The results indicate that hybrid “Area and Resource Usage” rule produces good results for both in the number of shortest schedule generation and average deviations.

1. GİRİŞ (INTRODUCTION)

Çizelgeleme problemi; görev çizelgeleme [1], ders çizelgeleme [2, 3], üretim çizelgeleme [4], kaynak kısıtlı makine planlama [5] gibi birçok alanda çözülmesi gereken bir konudur. Çizelgeleme probleminin en çok kullanıldığı bir diğer alan ise proje planlamadır. Kurumlar, fırsatları değerlendirmek, bir problemi çözmek veya değişikliklere uyum sağlayabilmek için proje üretirler ve riskleri göz önünde bulundurarak, maksimum kârı elde etmeyi amaçlarlar. Proje fikri belirlendikten sonra yapılacak ilk adım proje planlamadır. Projenin hedeflerine ulaşabilmesi ve maksimum kârın elde edilebilmesi için proje zamanlamasının doğru bir şekilde yapılması büyük önem taşımaktadır. Bu çalışmada, kaynak kullanımının kısıtlı olduğu proje planlama problemi incelenmiştir. Kaynak kısıtlı proje planlama (Resource Constrained Project Scheduling, RCPS) problemi NP-zor bir problemdir [6, 7]. Bu konuda yapılan literatür çalışmaları, problem için yeni ve daha verimli çözümlere ihtiyaç olduğunu göstermektedir [8, 9]. Literatürde kullanılan optimal yöntemler, problemin çözümünü için kullanılan arama alanına göre sezgisel, meta-sezgisel ve hiper-sezgisel yöntemler olmak üzere üçe ayrılmaktadır. Sezgisel yöntemler probleme ait alan bilgisini kullanarak en iyi çözümü aramaktadır, ancak yerel optimaya takılabilmekte ve sadece tanımlı probleme özgü çözüm üretmektedirler. Meta-sezgisel yöntemler problemden bağımsız olarak problem alanının tamamında çözüm ararken, hiper-sezgisel yöntemler problemin çözümünde hangi sezgisel yöntemin kullanılacağı ile ilgili arama yapmaktadır. Literatürde, RCPS problemin çözümü için Geliştirilmiş Ayrık Guguk Kuşu Arama (Improved Discrete Cockoo Search, IDCS) [10], Faktörlü Gri Kurt Optimizasyon (Factored Grey Wolf Optimization, FGWO) algoritması [11] gibi doğadan esinlenen optimizasyon algoritmaları kullanılmaktadır. Bu algoritmalar arasında en çok kullanılanı Karnca Kolonisi Optimizasyonu (Ant Colony Optimization, ACO) algoritmasıdır. ACO algoritmasının RCPS probleminin çözümünde kullanılması için verinin tam bağlı bir çizge ile modellenmesi gerekmektedir. Bu durum problemin karmaşıklık boyutunu arttırmaktadır. Gonzalez-Pardo vd. [12] karmaşıklığı azaltmak için sadece en önemli feromonları tutup diğer feromonları (feromonların yaklaşık %79'unu) eleyen sezgisel bir yaklaşım kullanmışlardır. Afshar-Nadjafi vd. [13] ise her aktivitenin tamamlanma tarihinin verildiği problemin çözümü için ACO algoritmasına ve genetik algoritmaya dayanan iki meta-sezgisel algoritma geliştirilmiştir. Kaynaklar geçici olarak aktiviteler için kiralanmış ve aktivitenin süresine bağlı olarak kira ödemesi belirlenmiştir. Zamanında tamamlanmayan aktivitelere kiranın zamanında verilmemesi nedeniyle ceza uygulanmıştır. Çalışma sonucunda ACO algoritmasının, genetik algoritmadan daha başarılı olduğu belirtilmiştir. Đumić vd. [14] genetik programlama kullanarak öncelik kurallarını belirlemiş ve bu kuralları PSPLIB 30, 60, 90 ve 120 verileri üzerinde test etmiştir. Bu çalışma sonucunda, örnek sayısı arttığında genetik algoritmanın her zaman optimum çözümü bulamadığı, ancak elde edilen sonuçların

optimum çözüme yakın olduğu belirtilmiştir. Genetik algoritmanın kullanılması için bütün aktivitelerin ve özelliklerinin bilinmesi gerekmektedir. Bu nedenle genetik algoritmanın dinamik problem üzerinde kullanımının kolay olmadığı ifade edilmiştir.

RCPS problemi için sezgisel yaklaşım kullanan algoritmalarda arama stratejisi için çok ajanlı sistemler de kullanılabilmektedir. Jedrzejowic vd. [15] RCPS probleminin çözümü için Pekiştirmeli Öğrenme, (Reinforcement Learning, RL) yöntemini ve çok ajanlı JABAT sistemini kullanmışlardır. Önerilen sistemde, ajanlar eş zamanlı çalışarak alt problemlere çözüm ararken; etkileşim stratejisini öğrenmek için pekiştirmeli öğrenme kullanılmaktadır. Sonuçlar A-Team mimarisinin etkileşim stratejisinin kontrol edebilmesi için RL ile desteklenmenin olumlu etkisinin olduğunu göstermektedir. Zheng ve Wang [16] ise çok ajanlı bir optimizasyon algoritması önerilmiştir. Ajanlar çalıştığı çevreye göre sosyal davranışla, otonom davranışla, kendi kendine ve çevresel uyumla eğitilmiştir.

Coelho vd. [17] ise Dalandırma ve Sınırlandırma yaklaşımını uygulamak için literatürde bulunan en iyi alt sınır stratejilerini hem hız hem de kalite kriterlerine göre dinamik olarak arama ağacından seçerek bileşimlerini kullanmışlardır. Geliştirilen yöntem dört farklı veri kümesi üzerinde denenmiş ve başarılı olduğu gözlemlenmiştir. Ancak yine de problemin karmaşıklığı nedeniyle birçok örneğin optimal olarak çözülmediği belirtilmiştir. Elsayed vd. [18] RCPS problemi için genel bir çözüm bulunamaması ve tek bir algoritmanın bütün örnekler için başarılı olamaması nedenleriyle farklı sezgisel yöntemlerin kullanıldığı genel bir çerçeve geliştirmişlerdir.

Literatürde sıklıkla kullanılan optimizasyon algoritmalarından biri de Parçacık Sürüsü Optimizasyonu (PSO) algoritmasıdır. Koulinas vd. [19] hiper-sezgisel algoritma tabanlı parçacık sürüsü optimizasyonu (PSO) yöntemini kullanmışlardır. Parçacıklar her sayının düşük seviyeli bir sezgisel algoritmayı temsil ettiği sekiz tamsayıdan oluşan bir vektörle ifade edilmiştir. Bu vektördeki ilk iki sayı, sezgisel yaklaşımda yer değiştirme; sonraki üç sayı oluşan küme öncelikleri değiştirme; son üç sayıdan ise sezgisel yaklaşımlı çaprazlamalar için kullanılmaktadır. Kumar ve Vidyarthi [20] ise Konvansiyonel Kanonik PSO yöntemini kullanmıştır. Parçacık pozisyonu ve hız yükselmesi sırasında, RCPS problemindeki bağımlı faaliyetler nedeniyle partikülün geçersiz hale gelmesini önlemek için, geçerli parçacık jeneratörünü (VPG) önermiş ve PSO'nun hızlı yakınsamasında önemli bir rol oynayan atalet ağırlığı (ω), önceki değeri, fitness değeri ve iterasyon sayısının etkilerine göre hesaplanmıştır. Sezgisel yöntemlerin olası arama uzayını daraltabilmek için öncelik kuralları tanımlanmaktadır. Bu nedenle öncelik kuralları, problemin çözümü için önem taşımaktadır. Wang vd. [21] yaptığı çalışmada literatürdeki ölçütlere ek olarak deterministik ve stokastik ortamlar için yeni kriterler tanımlayarak öncelik kurallarının başarısını ölçmüştür. Elde

edilen sonuçlar kullanım amacına göre öncelik kurallarının başarısının değiştiğini göstermektedir. Örneğin; proje yöneticileri için en erken teslim tarihi en iyi sonucu verirken, portföy yöneticileri için iş ağacında en fazla çocuğa sahip işin en iyi sonucu verdiği belirtilmiştir. Chen vd. [22] ise verimli öncelik kuralı ile ilgili çalışma yapmış ve bu çalışmada 17 öncelik kuralını kullanmıştır. Öncelik kuralı seçiminin, ağır karmaşıklığı ve kaynaktan çok; kaynağın gücünden, yani kaynağın ortalama kullanım değerinin üst kullanım sınırına oranından, etkilendiği belirtilmiştir. Eren [23], zamana bağımlı öğrenme etkili tek makineli çizelgeleme probleminde minimum gecikmeyi hedeflemiş olup, en erken teslim tarihi ve en kısa işlem zamanına dayanan bir çözüm önermiştir. Önerilen çözümde teslim tarihi aynı olan işler için en kısa işlem zamanı, işlem zamanları aynı olan işlerde ise en erken teslim tarihine öncelik verilmiştir. Geliştirilen yöntemle 14 işe kadar optimal çözümler bulunduğu belirtilmiştir. Chand vd. [24] ise verimli öncelik kuralları üretmek için hiper-sezgisel genetik programlama üzerinde çalışmıştır. Kuralların oluşturulmasında çeşitli niteliklerin kullanımı, aritmetik ve karar tabanlı kuralların kullanımı, seri ve paralel çizelge üretimi, eğitim verisinin optimizasyonu için fazladan bir çalışma yapılmasının önlenmesi, özelliklerin normalleştirilmesi, eğitim verisinin küçültülmesi üzerine deneyler yapmışlardır. Ebeseke ve Yaman [25] önerdiği melez paralel genetik algoritma ile j120_35_5 verisi için mevcut yöntemlerden daha başarılı sonuç elde ettiğini ancak aşırı optimizasyon yerine basit ama etkili kuralların elde edilmesi için çalışılması gerektiğini belirtmiştir.

Yapılan çalışmalar [21-24], öncelik kurallarının problem çözümünde etkili olduğunu ve optimum çözüme yakın sonuç veren basit öncelik kuralları üzerinde çalışılması gerektiğini [25] göstermektedir. Bu nedenle diğer çalışmalardan farklı olarak sadece öncelik kurallarına odaklanılmıştır. Yapılan çalışma Chand vd.nin çalışmasında yer alan kural oluşturulması ile benzerlik göstermekle beraber genetik programlama kullanmadan, istatistiksel sonuçlardan yola çıkarak bir kural belirleme hedeflenmiştir. Bu hedef doğrultusunda, proje planlamada kullanılan öncelik kuralları, kaynağın kısıtlı ve iş paketi sürelerinin belirli olduğu, literatürde de sıklıkla kullanılan PSPLib j30, j60, j120 veri kümeleri üzerinde karşılaştırılmıştır. Karşılaştırma sonucunda optimum çözüme en yakın sonuçların üretilmesinde daha fazla etkisi olan kaynak kullanımı ile en kısa süreli çizelge üretmede başarılı olan alan kuralının beraber dikkate alındığı bir öncelik kuralı önerilmiştir. Önerilen kural, en fazla sayıda en kısa süreli çizelge üretme kriterine göre, seri çizelgelemede j30 ve j120, paralel çizelgelemede j120 veri kümeleri için; ortalama sapma değerine göre ise seri çizelgelemede j120 veri kümesi için test edilen diğer kurallardan daha başarılı sonuçlar vermiştir.

2. YÖNTEM (METHOD)

RCPS probleminde belirli sayı ve limitteki kaynakla toplam proje süresinin en aza indirilmesi amaçlanmaktadır. Toplam $D+2$ aktiviteden oluşan proje planında $\{1, 2, \dots, j, \dots, D+2\}$ iş paketlerindeki ilk ve son aktivite sahte aktiviteler olup

süreleri d_j olsun. $R = \{R_1, \dots, R_k, \dots, R_K\}$ farklı kaynakları için her aktivite k 'nci kaynaktan $r_{k,j}$ birim kaynak talep etsin. En baştaki sahte aktivitenin süresi $d_1=0$ ve en sondaki sahte aktivitenin süresi $d_{D+2}=0$ 'dır. $\forall k = \{1, 2, \dots, K\}$ için en baştaki sahte aktivitenin talep ettiği kaynak $r_{k,1}=0$ ve en sondaki sahte aktivitenin talep ettiği kaynak $r_{k,D+2}=0$ 'dır. İş paketlerinin kesintiye uğratılmayacağını, j 'nci aktivitenin öncelikli aktiviteler tamamlanmadan başlatılmayacağını ve talep ettiği kaynak $r_{k,j}$ 'nin her zaman mevcut R_k değerinden küçük veya eşit olduğunu varsayalım. Bu durumda RCPS problemi, $D+2$ 'nin bitiş süresi olan FT_{D+2} 'nin minimizasyonudur [18].

2.1. Çizelgeleme Yöntemleri (Scheduling Methods)

RCPS probleminde kullanılan sezgisel çözüm prosedürünün temelinde çizelgeleme oluşturma şemaları yer almaktadır. Bu şemalar seri ve paralel çizelgeleme olarak ikiye ayrılmaktadır. Seri çizelgelemede aktiviteler sırayla seçilir. Paralel çizelgelemede ise belirli bir zamanda kullanılabilen aktiviteler arasından seçim yapılır. Bu seçimler yapılırken kullanılan öncelik kuralı, oluşturulacak çizelgeyi ve buna bağlı olarak elde edilecek minimum süreyi etkilemektedir.

2.1.1 Seri Çizelgeleme (Serial Scheduling)

Seri çizelgelemede aktiviteler; **tamamlanmış**, önce yapılması gereken iş paketleri tamamlanmış yani **seçime hazır** ve **diğer** iş paketleri olmak üzere üç kümeye ayrılır. Seri çizelgeleme, önceliği en yüksek ve seçime hazır ilk aktivitenin seçildiği çizelgeleme türüdür.

Algoritma 1. Seri Çizelgeleme (Serial Scheduling)

```

 $F_0 = 0, S_i = \{0\}$ 
for  $i = 1$ 'den  $n$ 'ye kadar
    ( $k \in K$ );  $t \in F_i$  için  $E_i, F_i, R_k(t)$ 'yi hesapla
     $j \in E_i$ 'yi seç
     $EF_j = \max_{h \in P_j} \{F_h\} + p_j$ 
     $F_j = \min \{t \in [EF_j - p_j, LF_j - p_j] \cap F_i \mid r_{j,k} \leq R_k(T),$ 
         $k \in K, [t, t + p_j] \cap F_i\} + p_j$ 
     $S_i = S_{i-1} \cup j$ 
end

```

Seri çizelgeleme algoritması n aktivite için $j=1$ 'den n 'ye kadar tekrarlanır. Algoritma 1'de verilen S_j , j adımıdaki tamamlanmış aktiviteler kümesini; E_j , j adımıdaki seçime hazır aktiviteler kümesini; F_j , bitiş süreleri kümesini; P_j , j adımıdan önce tamamlanması gereken aktivitelerin kümesini; p_j , aktivitenin işlem süresini; $R_k(T)$, t anında kalan kaynak sayısını; F_j , j 'nci aktivitenin bitiş süresini; EF_j en erken, LF_j en geç bitiş süresini temsil etmektedir [26].

2.1.2. Paralel çizelgeleme (Parallel scheduling)

Paralel çizelgelemede aktiviteler; **tamamlanmış**, **işlemleri devam eden**, **seçime hazır** ve **diğer** olmak üzere dört kümeye ayrılır. Her T anında seçime hazır aktiviteler arasından önceliği en yüksek aktivite seçilir. Algoritma 2'de

verilen C_j tamamlanmış, A_i ise devam eden aktiviteler kümesini temsil etmektedir.

Algoritma 2. Paralel Çizelgeleme (Parallel Scheduling)

```

j = 0, t_j = 0, A_0 = {0}, C_0 = {0}, R_k(0) = R_k
while |A_j ∩ C_j| ≤ n
  j = j + 1
  t_j = min_{i ∈ A_j} {F_i}
  C_j, A_j, R_k(t_j) ve E_j'yi hesapla
  while |E_j| > 0
    i ∈ E_j'yi seç
    F_i = t_j + p_j
    A_j, R_k(t_j) ve E_j'yi hesapla
  end
end
F_{n+1} = max_{h ∈ P_{n+1}} {F_h}
    
```

Tablo 1’de verilen örnekte 1 ve 2 numaralı aktivitelerin süreleri 2 birim, kullandıkları kaynak sayıları 1 olup öncesinde tamamlanması gereken aktiviteler bulunmamaktadır. 1 numaralı aktivite, 2 numaralı aktiviteden daha önceliklidir. 3 numaralı aktivite; 3 birim sürede tamamlanmakta, 2 kaynak kullanmakta ve 1 numaralı aktivite tamamlanmadan başlayamamaktadır. 3 numaralı aktivitenin önceliği ise 2’dir.

Tablo 1’de verilen örnek, seri programlamayla çözüldüğünde, önce tamamlanması gereken aktivitelerin tamamlandığı **karar kümesi**, 1 ve 2 numaralı aktivitelerden oluşmaktadır. İlk adımda, karar kümesindeki aktivitelere bakıldığında 1 numaralı aktivitenin öncelik derecesi yüksek olduğu için 1 numaralı aktivite seçilir. İkinci adımda, 1 numaralı aktivitenin tamamlanmasıyla 3 numaralı aktivite de karar kümesine dâhil olur. 1 numaralı aktivite ise bitmiş aktiviteler kümesinde yer alır. Karar kümesinde yer alan 2 ve 3 numaralı aktivitelerin öncelik dereceleri kıyaslandığında 3 numaralı aktivite seçilir. Son olarak 2 numaralı aktivite seçilerek çizelgeleme Şekil 1’deki gibi tamamlanır.

2			3				
1	1		3			2	
Kaynak Süre	1	2	3	4	5	6	7

Şekil 1. Seri Çizelgeleme (Serial Scheduling)

Aynı örnek paralel programlamayla çözüldüğünde, $T=0$ anında karar kümesinde 1 ve 2 numaralı aktiviteler bulunmaktadır. İlk adımda, 1 numaralı aktivitenin önceliği daha yüksek olduğu için yine 1 numaralı aktivite seçilir. İkinci adımda $T=1$ anında; 1 devam eden, 2 seçime hazır, 3 diğer aktiviteler kümesinde yer almaktadır ve sadece 2 için yeterli kaynak bulunduğundan 2 numaralı aktivite seçilir. Son olarak 3 numaralı aktivite seçilerek Şekil 2’deki çizelge elde edilir. Şekil 1 ve Şekil 2’de de görüldüğü gibi seri çizelgeleme aktif çizelgeler üretirken, paralel çizelgeleme ertelemesiz çizelgeler üretmektedir.

2	2		3				
1	1		3				
Kaynak Süre	1	2	3	4	5	6	7

Şekil 2. Paralel Çizelgeleme (Parallel Scheduling)

2.2. Veri Kümesi (Dataset)

Çalışmada kaynağın kısıtlı ve iş paketi sürelerinin belli olduğu, literatürde de sıklıkla kullanılan PSPLib j30, j60, j120 veri kümeleri [27] kullanılmıştır. Üç veri kümesinde de 4 kaynak bulunmaktadır. Veri kümeleri, isimlerini aktivite sayılarından almaktadır. j30 ve j60 480, j120 600 örnekten oluşmaktadır. Verilerin görselleştirilmesi için Graphviz [28] kütüphanesi kullanılmıştır. Şekil 3’te, Graphviz ile görselleştirilen j30 veri kümesinden örnek bir aktivite ağı verilmiştir. Her kaynak kullanımı Şekil 4’te verildiği gibi ayrı çizelge ile görselleştirilmiştir.

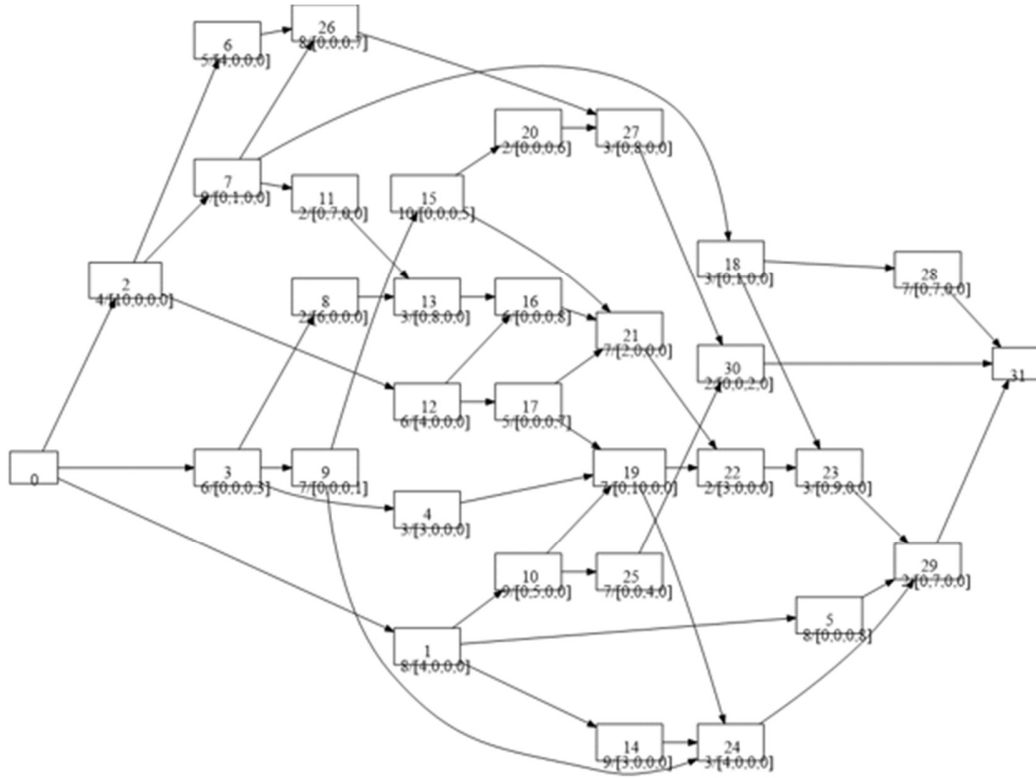
2.3. Öncelik Kuralları (Priority Rules)

Öncelik kuralları, çizelgeleme için oluşturulan karar kümesinden yapılacak seçimi belirleyen kurallardır. Temel öncelik kurallarının tanımlı olması, bu kuralların seri veya paralel çizelgelemeyle gösterilmesine olanak sağlaması, açık kaynak ve nesne yönelimli bir dil olan C#’da yazılmış olması ve sezgisel yaklaşımlar için kullanılabilir kural tanımlama desteği içermesi nedenleriyle bu çalışmada PSPSolver [29] kütüphanesi kullanılmıştır. Bu kütüphanede bulunan öncelik kuralları ise aşağıda kısaca tanımlanmıştır.

- **İlk seçkin aktivite:** Karar kümesinde bulunan ilk aktivite seçilir. First-come, first-served kuralı ile çalışır.
- **Son seçkin aktivite:** Karar kümesinde bulunan son aktivite seçilir. Last-come, first-served kuralı ile çalışır.

Tablo 1. Aktivite Verisi Örneği (Activity Data Example)

Aktivite	Süresi	Önce Tamamlanması Gereken Aktivite	Kullandığı Kaynak Miktarı	Öncelik Derecesi
1	2	-	1	2
2	2	-	1	1
3	3	1	2	2



Şekil 3. Örnek Aktivite Ağı (Example Activity Network)

- **Rastgele seçkin aktivite:** Karar kümesinde bulunan aktiviteler arasından rastgele seçim yapılır.
- **En kısa işlem süresi:** İşlem süresi en kısa olan aktivite seçilir.
- **En uzun işlem süresi:** İşlem süresi en uzun olan aktivite seçilir.
- **En küçük alan:** İşlem süresi ve kaynak kullanımı en az olan aktivite seçilir.
- **En büyük alan:** İşlem süresi ve kaynak kullanımı en fazla olan aktivite seçilir.
- **Tek örnekleme:** Rastgele seçilen n sayıda aktivitenin her biri için uygunluk hesaplanır ve bir kabul değeri belirlenir. Uygun olmayan örneklerin sayısı kabul sayısından büyükse n aktivitenin dışında kalan aktiviteler reddedilir, küçükse kabul edilir.

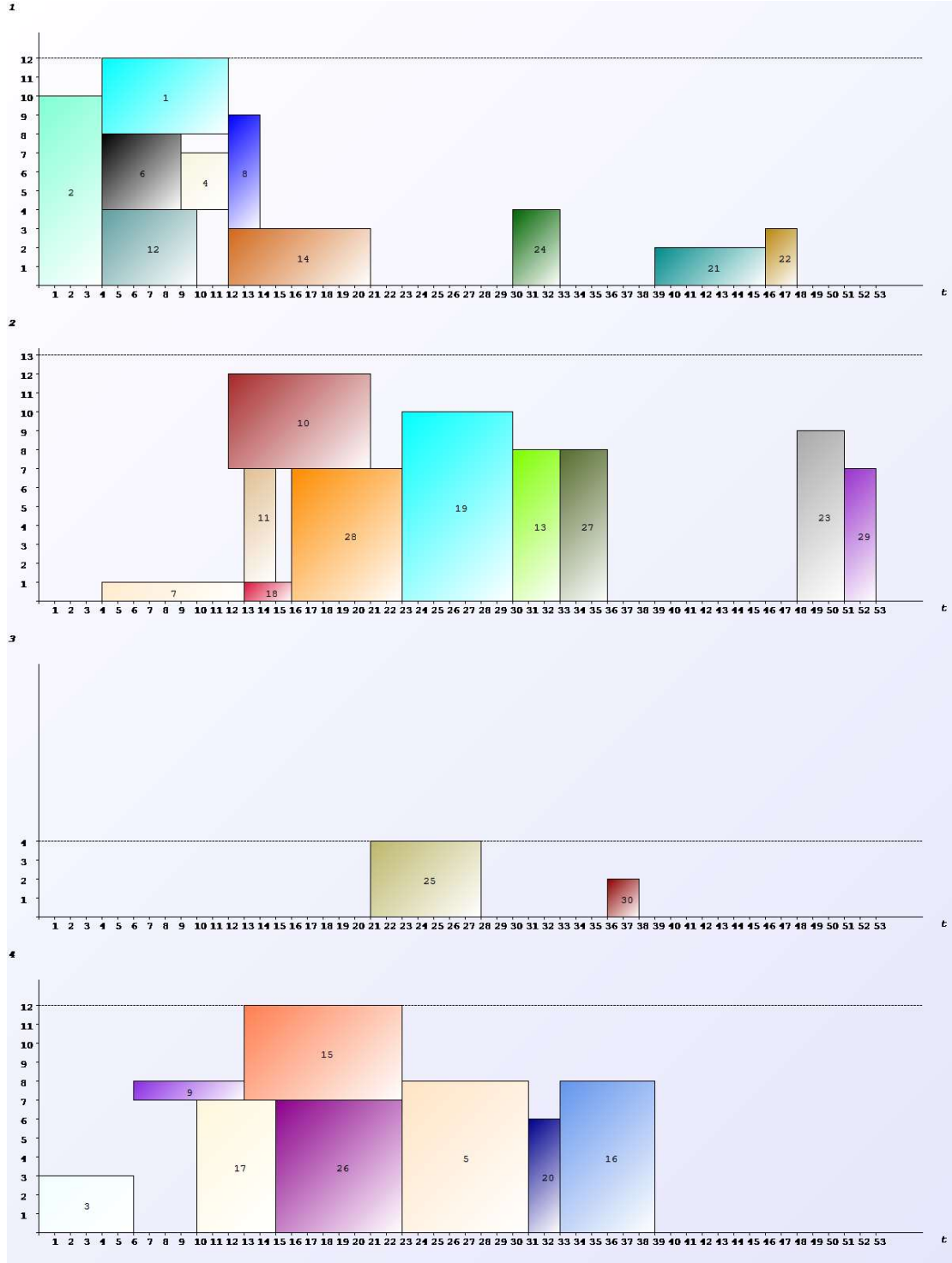
Çalışmamızda ilk olarak seri ve paralel çizelgeleme yöntemlerinde hangi öncelik kuralının daha başarılı olduğunu ölçmek için algoritmaların en kısa süreli çizelgeyi üretme sayıları kullanılmıştır. Belirlenen öncelik kuralları j30, j60 ve j120 veri setleri üzerinde test edilmiştir. Elde edilen sonuçlar (Şekil 5), En Büyük Alan kuralının her iki çizelgeleme tipi ve üç veri kümesi için başarılı sonuçlar verdiğini göstermektedir. En Büyük Alan kuralı seri çizelgelemede, paralel çizelgelemeye göre daha başarılı olmaktadır. Öncelik kurallarının başarısı genel olarak incelendiğinde aktivite sayısındaki artışın bütün kuralları olumsuz etkilediği ancak bu durumdan en az etkilenen iki kuralın en uzun işlem süresi ve en büyük alan olduğu

görülmektedir. Ayrıca aktivite sayısındaki artış, paralel çizelgelemede elde edilen en kısa süreli çizelge sayılarında daha fazla düşüşe neden olmaktadır.

2.4. Problem Çözüm Modeli (Problem Solving Model)

Kaynak-Süre çizelgesinin; genişliğinin kaynağı, yüksekliğinin süreyi temsil ettiği Şekil 6'daki gibi 2 boyutlu bir kutu olduğunu varsayalım. İş paketlerini ise bu kutuya yerleştirilmesi gereken paketler olarak düşünelim. Bu kutunun taban genişliği sabit olsun, ancak yüksekliği yerleştirdiğimiz paketlerin yüksekliğine bağlı olarak değişebilsin. Yani kutuya istediğimiz kadar paket sığdırabiliyor olabilelim. Böylece RCPS problemini; kutunun yüksekliğini mümkün olduğunca az tutacak şekilde bütün paketlerin yerleştirildiği bir optimizasyon problemine dönüştürebiliriz.

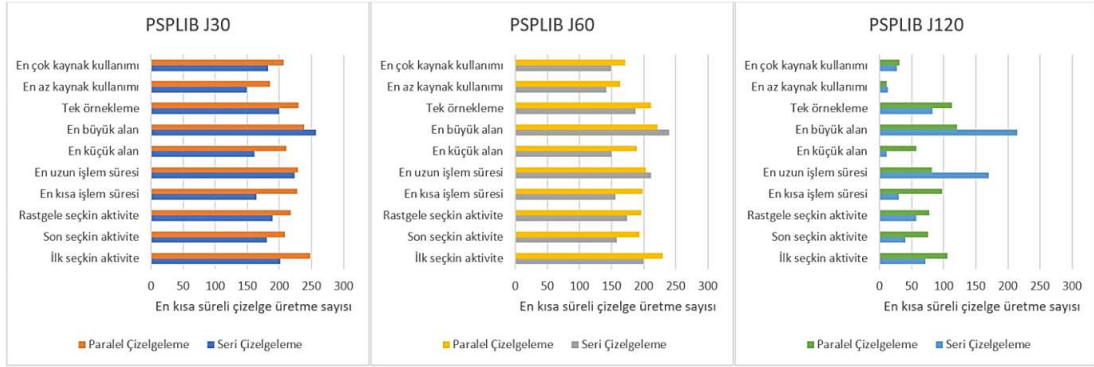
Problemi verimli bir şekilde çözmek için kutunun yüksekliğinin ve kutular arasındaki boşluğun en aza indirilmesi gerekmektedir. Bunun için de en büyük alana sahip paketlerin kutuya önce yerleştirilmesi gerekmektedir. Bu durumda paketleri kutuya yerleştirirken paketin enine veya boyuna göre bir öncelik vermenin etkisini incelemek amacıyla "Alan ve İşlem Süresi" ve "Alan ve Kaynak Kullanımı" olmak üzere iki kural önerilmiştir. Şekil 7'de belirtilen "İşlem Süresi", En Uzun İşlem Süresi, "Kaynak Kullanımı" En Çok Kaynak Kullanımı, "Alan" ise En Büyük Alan kuralını temsil etmektedir.



Şekil 4. Kaynak Kullanımı (Resource Usage)

Şekil 7’de 10 örnek durum için öncelik kurallarının *İş Paketi a* ve *İş Paketi b* arasından hangisini önceliklendirdiği verilmiştir. Kurallar incelendiğinde *İşlem Süresi*’nin 2 ve 5, *Kaynak Kullanımı*’nın ise 3 ve 4 numaralı durumlarda önceliklendirme yapmadığı görülmektedir. Bu iki kuralın çarpımıyla elde edilen *Alan* kuralı ise 3, 4, 7 ve 8 numaralı durumlarda *İşlem Süresi*; 2, 5, 6 ve 9 numaralı durumlarda

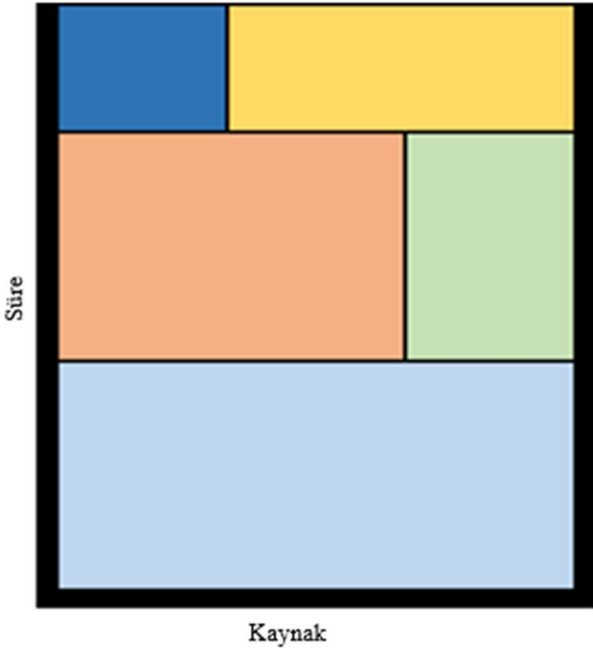
Kaynak Kullanımı kuralıyla aynı önceliklendirmeyi yapmaktadır. 1 numaralı durumda ise *İşlem Süresi* kuralı *İş Paketi a*’yı önceliklendirirken, *Kaynak Kullanımı* kuralı *İş Paketi b*’yi önceliklendirmekte, ancak *Alan* kuralı iki paket arasında önceliklendirme yapmamaktadır. Bu nedenle *Alan* kuralının önceliklendirme yapmadığı 1 numaralı durum için de önceliklendirme yapabilecek bir kuralın daha başarılı



Şekil 5. Mevcut Öncelik Kurallarının Karşılaştırması (Comparison of Existing Priority Rules)

olabileceği değerlendirilmiştir. 1 numaralı durumda *İş Paketi a* ve *İş Paketi b* aynı alana sahip olup *İş Paketi a* enine, *İş Paketi b* ise boyuna yer kaplamaktadır. Problemi paketlerini kutuya yerleştirme problemi olarak düşündüğümüzde, kutuya önce uzun olan paketleri mi yoksa geniş olan paketleri mi yerleştirmek daha verimli olur sorusuna cevap aranmaktadır.

Alan ve Kaynak Kullanımı kuralı 1, 7 ve 9 numaralı durumlarda *Kaynak Kullanımı* kuralı ile aynı çalışmaktadır. Önerilen kuralın *İşlem Süresi* ve *Kaynak Kullanımı* kurallarından farkı ise bu kuralların önceliklendirme yapmadığı durumlarda da bir önceliklendirme yapmasıdır. Böylece literatürde var olan En Büyük Alan ve En Uzun İşlem Süresi kurallarının beraber dikkate alındığı melez bir yöntem önerilmiştir. *Alan ve İşlem Süresi* kuralı, kutuya uzun paketleri yerleştirerek başlama durumunu temsil etmektedir. *Alan ve Kaynak Kullanımı* kuralı ise kutuya yerleştirme işleminde geniş paketlerden başlama durumu temsil etmektedir. Önerilen kurallar Algoritma 3'te verildiği şekliyle probleme uygulanmıştır.



Şekil 6. Problem Çözüm Modeli (Problem Solving Model)

Alan kuralının, *İşlem Süresi* ve *Kaynak Kullanımı* kurallarının önceliklendirme yapmadığı durumlarda da çarpma işlemi sayesinde önceliklendirme yapabildiği ve iki kuralın ayrı ayrı verdiği sonuçlardan daha başarılı olduğu Şekil 5'te görülmüştür. Bu nedenle *Alan* kuralı için işlem süresi ve kaynak kullanımı sayılarıyla ikinci bir çarpma işlemi yaparak, alanla beraber işlem süresini ($\text{İşlem Süresi}^2 \times \text{Kaynak Kullanımı}$) ve alanla beraber kaynak kullanımını ($\text{Kaynak Kullanımı}^2 \times \text{İşlem Süresi}$) dikkate alan iki kural oluşturulmuştur. *Alan* kuralından farklı olarak *Alan ve İşlem Süresi* kuralı, 1, 6 ve 7 numaralı durumlarda *İşlem Süresi* kuralı ile aynı çalışırken; 1544

Algoritma 3. Önerilen Algoritma (Proposed Algorithm)

//Alan ve İşlem Süresi

$$\text{değerA} = (a\text{'nın işlem süresi})^2 * a\text{'nın kaynak sayısı}$$

$$\text{değerB} = (b\text{'nin işlem süresi})^2 * b\text{'nin kaynak sayısı}$$

eğer (değerA = değerB)
ise öncelikte değişiklik yok;
eğer (değerA > değerB)
ise a'nın önceliğini artırır;
eğer (değerA < değerB)
ise b'nin önceliğini artırır;

//Alan ve Kaynak Kullanımı


































































$$\text{değerA} = a\text{'nın işlem süresi} * (a\text{'nın kaynak sayısı})^2$$

$$\text{değerB} = b\text{'nin işlem süresi} * (b\text{'nin kaynak sayısı})^2$$

eğer (değerA = değerB)
ise öncelikte değişiklik yok;
eğer (değerA > değerB)
ise a'nın önceliğini artırır;
eğer (değerA < değerB)
ise b'nin önceliğini artırır;

2.5. Bulgular (Results)

En Büyük Alan kuralının iki çizelgeleme yöntemi için de başarılı sonuçlar üretmesi ve aktivite sayısındaki artış durumunda en kısa süreli çizelge üretme sayısının diğer kurallardan daha iyi olması nedenleriyle bu çalışmada

No	İş Paketi a	İş Paketi b	İşlem Süresi	Kaynak Kullanımı	Alan	Alan ve İşlem Süresi	Alan ve Kaynak Kullanımı
1	1  2	2  1	2  1	1  2	-	2  1	1  2
2	2  2	2  1	-	2  2	2  2	2  2	2  2
3	2  2	1  2	2  2	-	2  2	2  2	2  2
4	2  2	3  2	3  2	-	3  2	3  2	3  2
5	2  2	2  3	-	2  3	2  3	2  3	2  3
6	2  2	3  1	3  1	2  2	2  2	3  1	2  2
7	2  2	1  3	2  2	1  3	2  2	2  2	1  3
8	2  2	5  1	5  1	2  2	5  1	5  1	2  2
9	2  2	1  5	2  2	1  5	1  5	2  2	1  5
10	2  2	1  1	2  2	2  2	2  2	2  2	2  2

Şekil 7. Örnek İş Paketi Önceliklendirmeleri (Sample Work Package Prioritizations)

kaynak ve işlem süresine odaklanılmıştır. Bu nedenle *Alan ve İşlem Süresi* ile *Alan ve Kaynak Kullanımı* olmak üzere iki kural önerilmiştir. Önerilen bu kuralların ve diğer kuralların, j30, j60 ve j120 veri kümeleri için en başarılı çizelge üretme sayıları Şekil 8’de verilmiştir. Genel olarak *Alan ve Kaynak Kullanımı* kuralının hem seri hem de paralel çizelgelemede başarılı sonuçlar verdiği, alan ve kaynağın

ayrı ayrı kullanıldığı kurallardan daha başarılı sonuçlar ürettiği ve *Alan ve İşlem Süresi* kuralından daha başarılı olduğu görülmektedir.

Öncelik kurallarının en kısa süreli çizelge üretme sayılarının detayı Tablo 2’de verilmiştir. Bu sayılardan en düşüğü koyu kırmızı, en yükseği koyu yeşil ile renklendirilmiştir.

Sonuçlar incelendiğinde *Alan ve Kaynak Kullanımı* kuralının seri çizelgelemede j30 ve j120, paralel çizelgelemede j120 için en başarılı kural olduğu görülmektedir. Önerilen kural seri çizelgelemede genel olarak en başarılı sonuçları vermiştir; ancak j60 veri kümesinde En Büyük Alan kuralına göre 3 örnek için daha uzun süreli çizelge üretmiştir. Bununla birlikte diğer tüm veri kümelerinde hem seri hem de paralel çizelgelemede En Büyük Alan, En Uzun İşlem Süresi ve En Çok Kaynak Kullanımı kurallarından daha başarılı sonuçlar vermiştir. Önerilen bu kural, incelenen kurallar içerisinde en başarılı sonuç veren 2 algoritmadan biri olmuştur. Ayrıca hem örnek hem de aktivite sayısının daha fazla olduğu j120 veri kümesinde en iyi sonuç veren İlk Seçkin Aktivite kuralından %57 daha iyi sonuç vermiştir.

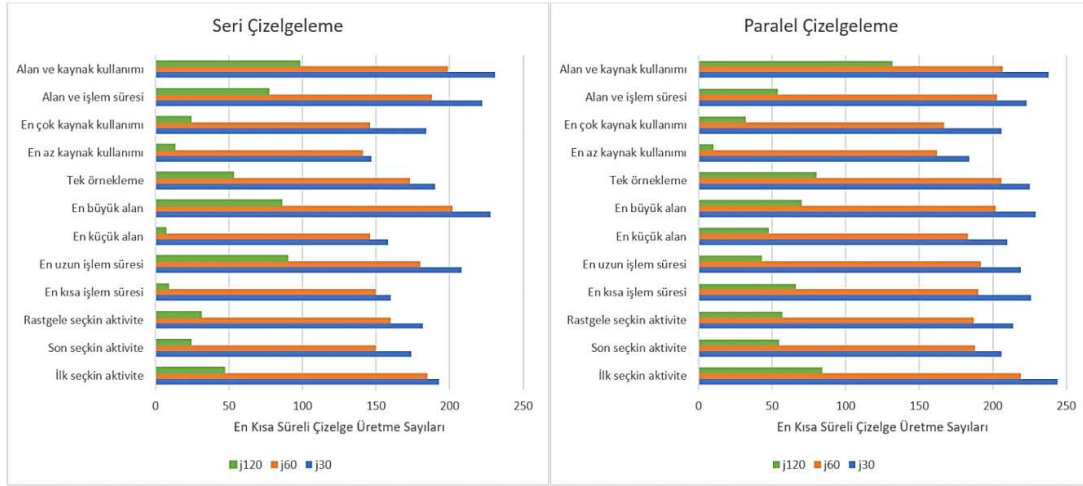
Alan ve Kaynak Kullanımı kuralının *Alan ve İşlem Süresi* kuralına göre daha başarılı olması, test edilen veri kümesinde kutuya paket yerleştirilmesi sırasında geniş paketleri tercih etmenin daha başarılı sonuç verdiğini göstermiştir. *Alan ve Kaynak Kullanımı* kuralında kaynak kullanımı değerinin karesini almak, kaynakların daha yoğun kullanıldığı çizelgeler üretmeyi ve işlerin tamamlanma süresini

azaltmayı sağlamıştır. Bu sonuçlar, öncelik kuralının başarısının veri kümesine bağlı olarak değiştiğini doğrulamakla birlikte, melez kuralların daha başarılı sonuçlar verebileceğini göstermektedir. Algoritmanın başarısını ölçmek için Kumar ve Vidyarthi'nin [20] çalışmasında olduğu gibi Eş. 1'de verilen formülle ortalama sapma değerleri hesaplanmıştır.

$$Ortalama\ Sapma = \frac{\sum \text{Örnek Sayısı} \left(\frac{\text{Elde Edilen Sonuç} - \text{En İyi Sonuç}}{\text{En İyi Sonuç}} \times 100 \right)}{\text{Örnek Sayısı}} \quad (1)$$

Kullanılan örnek sayıları j30 ve j60 için 480, j120 600'dür. En iyi sonuç değerleri ise Kolisch ve Sprecher'in çalışmasından [30] alınmıştır. Elde edilen ortalama sapma oranları karşılaştırmalı olarak Tablo 3'te verilmiştir.

Ortalama sapma değerleri incelendiğinde, En Çok Kaynak Kullanımı kuralının en kısa süreli çizelge üretme sayısında başarısız olmasına rağmen optimum çizelgelere daha yakın sonuçlar verdiği görülmektedir (Şekil 9). Bu da *Alan ve Kaynak Kullanımı* kuralının *Alan ve İşlem Süresi* kuralından



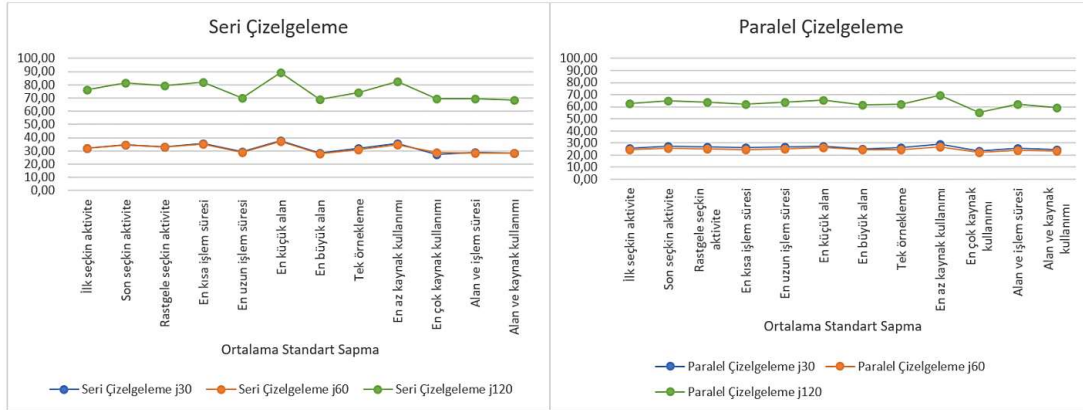
Şekil 8. Öncelik kurallarının en iyi sonuç verdiği örnek sayıları
(The number of Samples which the Priority Rules give the best result)

Tablo 2. En Kısa Süreli Çizelge Üretme Sayıları (Shortest Schedule Production Numbers)

Kural	Seri Çizelgeleme			Paralel Çizelgeleme		
	j30	j60	j120	j30	j60	j120
İlk seçkin aktivite	193	185	47	244	219	84
Son seçkin aktivite	174	150	24	206	188	55
Rastgele seçkin aktivite	182	160	31	214	187	57
En kısa işlem süresi	160	150	9	226	190	66
En uzun işlem süresi	208	180	90	219	192	43
En küçük alan	158	146	7	210	183	48
En büyük alan	228	202	86	229	202	70
Tek örnekleme	190	173	53	225	206	80
En az kaynak kullanımı	147	141	13	184	162	10
En çok kaynak kullanımı	184	146	24	206	167	32
Alan ve işlem süresi	222	188	77	223	203	54
Alan ve kaynak kullanımı	231	199	98	238	207	132

Tablo 3. Ortalama Sapma Değerleri (%) (Average Deviations (%))

Öncelik Kuralı	Seri Çizelgeleme			Paralel Çizelgeleme		
	j30	j60	j120	j30	j60	j120
İlk seçkin aktivite	32,11	31,81	76,30	25,42	24,54	62,48
Son seçkin aktivite	34,46	34,63	81,62	27,19	25,61	65,10
Rastgele seçkin aktivite	33,11	33,05	79,38	26,54	25,17	63,53
En kısa işlem süresi	35,39	35,19	82,16	26,03	24,37	62,02
En uzun işlem süresi	29,30	28,68	70,05	26,49	24,95	63,69
En küçük alan	37,67	37,29	89,02	27,34	25,96	65,58
En büyük alan	28,28	27,91	68,80	24,92	24,22	61,59
Tek örnekleme	31,79	31,12	74,22	26,02	24,27	61,89
En az kaynak kullanımı	35,63	34,61	82,51	28,94	26,65	69,57
En çok kaynak kullanımı	27,50	28,63	69,62	23,09	22,13	55,28
Alan ve işlem süresi	28,76	28,38	69,38	25,71	24,13	62,08
Alan ve kaynak kullanımı	28,22	28,15	68,62	24,42	23,35	59,02

**Şekil 9.** Ortalama Standart Sapma Grafiği (Average Standart Deviation Graph)

daha başarılı sonuç vermesini açıklamaktadır. Özellikle aktivite ve çizelge sayısının en fazla olduğu j120 veri kümesinde *Alan ve Kaynak Kullanımı* kuralı ile elde edilen standart sapma değerlerinin seri çizelgelemede en iyi, paralel çizelgelemede ikinci en iyi kural olduğu görülmektedir. Önerilen bu kural, En Büyük Alan kuralının en kısa süreli çizelge üretme özelliği ile En Çok Kaynak Kullanımı kuralının optimuma en yakın çizelge üretme avantajlarından faydalanmaktadır. Bu sayede daha fazla sayıda optimuma yakın çizelge üretebilmektedir.

Bu sonuçlar, daha önce kural çıkarımı için elde edilen sonuçları [23, 24] desteklemekte ve melez kuralların daha başarılı olabileceğini göstermektedir. Genetik programlama gibi meta-sezgisel yöntemler ile de daha kısa süreli çizelge veren öncelik kurallarını belirlemek mümkündür. Ancak tek

değerlendirme kriteri, en kısa süreli çizelge üretmek değildir. Başarılı sonucun kısa sürede üretilmesi de önem taşımaktadır ve genetik programlama ile elde edilen kurallar bu durumu göz ardı edebilmektedir. Bu nedenlerle, Chand vd.nin [24] çalışmasından farklı olarak öncelik kuralları belirlenirken genetik algoritma yerine istatistiksel sonuçlar kullanılmıştır. Eren'in [23] çalışmasından farklı olarak en erken teslim tarihi ve en kısa işlem süresi yerine, iş paketlerinin süre ve kaynak kullanımına odaklanılmıştır.

Önerilen kuralın işlem zamanını ölçmek için, Intel(R) Core (TM) i7-8565 CPU, 1.80 GHz, 8,00 GB RAM (LPDDR3), 4 çekirdek, 8 mantıksal işlemciye sahip 64 bit bilgisayar kullanılmıştır. Algoritmaların ortalama çalışma süreleri ms cinsinden Tablo 4'te karşılaştırılmıştır. Melez olmayan bazı kurallar daha kısa sürede çözüm üretmesine rağmen,

Tablo 4. Çalışma Zamanları (ms) (Process Time (ms))

Kural	Seri Çizelgeleme			Paralel Çizelgeleme		
	j30	j60	j120	j30	j60	j120
İlk seçkin aktivite	3,56	7,70	15,40	4,66	10,90	22,91
Son seçkin aktivite	3,66	7,01	14,21	4,56	9,65	23,23
Rastgele seçkin aktivite	3,66	6,90	14,86	4,49	9,74	22,36
En kısa işlem süresi	1,90	3,21	6,76	2,41	5,33	12,90
En uzun işlem süresi	1,87	3,06	6,39	2,40	5,30	12,69
En küçük alan	1,99	3,65	8,58	2,57	5,51	13,60
En büyük alan	2,04	3,88	8,25	2,52	5,74	13,51
Tek örnekleme	1,90	3,17	6,58	2,41	5,41	13,12
En az kaynak kullanımı	2,72	6,09	11,83	3,62	7,32	13,77
En çok kaynak kullanımı	2,86	4,29	9,16	3,59	7,22	15,12
Alan ve işlem süresi	2,06	3,68	8,47	2,61	5,47	13,33
Alan ve kaynak kullanımı	2,01	3,86	8,26	2,59	5,51	13,56
Ortalama	2,52	4,71	9,89	3,20	6,92	15,84

önerilen *Alan ve Kaynak Kullanımı* kuralı test edilen kuralların ortalama çalışma sürelerinin altında çözüm üretmektedir.

3. SONUÇLAR (CONCLUSIONS)

Aktivite ve örnek sayılarındaki artış, bütün öncelik kurallarının en kısa süreli çizelge üretme sayılarında düşüş ve ortalama sapma değerinde artışa neden olmaktadır. Ancak sonuçlar genel olarak değerlendirildiğinde En Büyük Alan kuralının en kısa süreli çizelge üretmede, En Çok Kaynak Kullanımı kuralının ise en düşük ortalama sapma değerini elde etmede başarılı olduğu görülmüştür. Bu nedenle bu iki kuralı dikkate alan melez bir öncelik kuralı önerilmiştir. Önerilen kural, hem en kısa süreli çizelge üretme sayısı hem de ortalama sapma değeri dikkate alındığında başarılı sonuç vermektedir. Ayrıca ortalama çalışma zamanının altında bir sürede çalışmaktadır.

KAYNAKLAR (REFERENCES)

1. Tapkan P., Özbakır L., Kulluk S., Telcioğlu B., Modelling and solving railway crew rostering problem, Journal of the Faculty of Engineering and Architecture of Gazi University, 33 (3), 953–965, 2018.
2. Uçar U. Ü., İşleyen S. K., Solving makeup course timetabling problem (MCTP) with mathematical modelling: a real case application, Journal of the Faculty of Engineering and Architecture of Gazi University, 31 (2), 331–346, 2016.
3. Demir Y., Çelik C., An integer programming approach for curriculum based timetabling problem solution, Journal of the Faculty of Engineering and Architecture of Gazi University, 31 (1), 145–160, 2016.
4. Ertogral K., Erkoç M., Ülker D. H., A production scheduling model and analysis for the maintenance repair and overhaul service providers, Journal of the Faculty of Engineering and Architecture of Gazi University, 30 (3), 513–522, 2015.
5. Bektur G., Saraç T., Two parallel injection machine scheduling under crane constraint, Journal of the Faculty of Engineering and Architecture of Gazi University, 31(4), 903–911, 2016.
6. Blazewicz J., Lenstra J. K., Rinnooy Kan A. H. G., Scheduling Subject to Resource Constraints: Classification and Complexity, Discrete Applied Mathematics, 5, 11–24, 1983.
7. Brucker P., Drexel A., Möhring R., Neumann K., Pesch E., Resource-constrained project scheduling: Notation, classification, models, and methods, European Journal of Operational Research, 112 (1), 3–41, 1999.
8. Özdamar L., Ulusoy G., A survey on the resource-constrained project scheduling problem, IIE Transactions (Institute of Industrial Engineers Transactions), 27 (5), 574–586, 1995.
9. Habibi F., Barzinpour F., Sadjadi S. J., Resource-constrained project scheduling problem: review of past and recent developments, Journal of Project Management, 3, 55–88, 2018.
10. Bibiks K., Hu Y. F., Li J. P., Pillai P., Smith A., Improved discrete cuckoo search for the resource-constrained project scheduling problem, Applied Soft Computing, 69, 493–503, 2018.
11. Xiao W., Deng H., Sheng Y., Hu L., Factored Grey Wolf Optimizer with Application to Resource-Constrained Project Scheduling, International Journal of Innovative Computing, Information and Control, 14 (3), 881–897, 2018.
12. Gonzalez-Pardo A., Del Ser J., Camacho D., Comparative study of pheromone control heuristics in ACO algorithms for solving RCPSP problems, Applied Soft Computing, 60, 241–255, 2017.
13. Afshar-Nadjafi B., Basati M., Maghsoudlou H., Project scheduling for minimizing temporary availability cost of rental resources and tardiness penalty of activities, Applied Soft Computing, 61, 536–548, 2017.
14. Đumić M., Šišeković D., Čorić R., Jakobović D., Evolving priority rules for resource constrained project scheduling problem with genetic programming, Future Generation Computer Systems, 86, 211–221, 2018.
15. Jedrzejowicz P., Ratajczak-Ropel E., Reinforcement Learning strategies for A-Team solving the Resource-

- Constrained Project Scheduling Problem, *Neurocomputing*, 146, 301–307, 2014.
16. Zheng X. L., Wang L., A multi-agent optimization algorithm for resource constrained project scheduling problem, *Expert Systems with Applications*, 42 (15), 6039–6049, 2015.
 17. Coelho J., Vanhoucke M., An exact composite lower bound strategy for the resource-constrained project scheduling problem, *Computers & Operations Research*, 93, 135–150, 2018.
 18. Elsayed S., Sarker R., Ray T., Coello C. C., Consolidated optimization algorithm for resource-constrained project scheduling problems, *Information Sciences*, 418–419, 346–362, 2017.
 19. Koulinas G., Kotsikas L., Anagnostopoulos K., A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem, *Information Sciences*, 277, 680–693, 2014.
 20. Kumar N., Vidyarthi D. P., A model for resource-constrained project scheduling using adaptive PSO, *Soft Computing*, 20 (4), 1565–1580, 2016.
 21. Wang Y., He Z., Kerkhove L. P., Vanhoucke M., On the performance of priority rules for the stochastic resource constrained multi-project scheduling problem, *Computers & Industrial Engineering*, 114, 223–234, 2017.
 22. Chen Z., Demeulemeester E., Bai S., Guo Y., Efficient priority rules for the stochastic resource-constrained project scheduling problem, *European Journal of Operational Research*, 270 (3), 957–967, 2018.
 23. Eren T., Minimizing the maximum lateness in a scheduling problem with a time-dependent learning effect: A non-linear programming model, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 23 (2), 459–465, 2008.
 24. Chand S., Huynh Q., Singh H., Ray T., Wagner M., On the use of genetic programming to evolve priority rules for resource constrained project scheduling problems, *Information Sciences*, 432, 146–163, 2018.
 25. Ebesk Ş., Yaman H., Kaynak Kısıtlı Proje Çizelgeleme Probleminde Tekrarsız Kromozom Destekli Paralel Genetik Algoritma Uygulaması, *Uluslararası Sosyal Araştırmalar Dergisi*, 12 (62), 519–529, 2019.
 26. Šišeković D., Evolution Of Scheduling Heuristics For The Resource Constrained Scheduling Problem, Yüksek Lisans Tezi, University of Zagreb, Faculty Of Electrical Engineering and Computing, Zagreb, 2016.
 27. Project Scheduling Library PSPLIB, <http://www.omdb.wi.tum.de/psplib/getdata.cgi?mode=sm>. Yayın tarihi Haziran 3, 2016. Erişim tarihi Şubat 24, 2020.
 28. Graphviz - Graph Visualization Software , <http://www.graphviz.org>. Erişim tarihi Şubat 24, 2020.
 29. PSPSolver -Project Scheduling Problem Solver. <https://pspsolver.soft112.com/>. Yayın tarihi Mart 27, 2009. Erişim tarihi Şubat 24, 2020.
 30. Kolisch R., Sprecher, A., 1996, "PSPLIB - A project scheduling library", *European Journal of Operational Research*, 96, 205-216.

