

Kaynak Dengeleme Probleminin Arama Uzayını Paralel Programlama ile Tarayarak Kesin Çözümü

Tuğba ERZURUM¹
Önder Halis BETTEMİR²

ÖZ

Kaynak dengeleme problemi (KDP) sezgisel, modern sezgisel ve matematiksel yöntemlerle çözülmektedir. Fakat belirtilen yöntemler özellikle büyük boyutlu problemler için kesin çözümü garanti edememektedir. Bu çalışmada KDP'nin aktiviteler arasındaki bağımlılık ilişkilerini ihlal etmeden ve proje süresinde uzamaya neden olmayacak şekilde bolluğu olan aktivitelerin ertelenmesi ile elde edilebilecek birbirinden farklı kaç iş programı oluşturulabileceği hesaplanmıştır. Arama uzayı olarak tanımlanan tüm uygulanabilir iş programlarının tamamının denemesi ile garantili biçimde KDP'nin en iyi çözümü elde edilerek mevcut yöntemlerden farklı biçimde KDP'nin çözülmesi sağlanmıştır. Aktivite sayısı ile arama uzayı arasında seri bağlı aktiviteler için üstel bağıntı formülü türetilerek büyük projelerin tek işlemci ile çözümünün makul sürede gerçekleştirilemeyeceği belirlenmiştir. Problemin paralel programlama ile tüm işlemciler eşit sayıda şebeke çözümü düşecek şekilde paylaşılması sağlanmıştır. Bu çalışmada en büyüğü 36 aktiviteli olan 4 KDP arama uzayının tamamı taranıp makul sürede çözümlenerek geliştirilen yöntemin uygulanabilir olduğu kanıtlanmıştır. Bu yöntem ile daha küçük parçalara ayırmak sureti ile daha büyük kaynak dengeleme problemlerinin kesin çözümü elde edilebilecektir.

Anahtar Kelimeler: Kaynak dengeleme problemi, optimizasyon, kritik yol yöntemi, paralel programlama.

ABSTRACT

Exact Solution of Resource Leveling Problem by Exhaustive Enumeration with Parallel Programming

Resource Leveling Problem (RLP) is solved by heuristic, meta-heuristic, and mathematical methods. However, the aforementioned methods cannot guarantee the exact solution for large

Not: Bu yazı

- Yayın Kurulu'na 22 Temmuz 2019 günü ulaşmıştır. 23 Mayıs 2020 günü yayımlanmak üzere kabul edilmiştir.
- 31 Temmuz 2021 gününe kadar tartışmaya açıktır.

• <https://doi.org/10.18400/tekderg.595238>

1 İnönü Üniversitesi İnşaat Mühendisliği Bölümü, Malatya - tugbaerzrm@gmail.com
<https://orcid.org/0000-0003-4788-6999>

2 İnönü Üniversitesi İnşaat Mühendisliği Bölümü, Malatya - onder.bettemir@inonu.edu.tr
<https://orcid.org/0000-0002-5692-7708>

size problems. In this study, number of feasible schedules which can be obtained by delaying the non-critical activities without violating the precedence relationships and elongating the project completion time are computed. All of the feasible schedules which can be defined as the search domain are enumerated and the guaranteed optimum solution for the RLP is obtained by a different method from the existing methods. Exponential equation between the search domain and the number of activities on serial path is derived and the insolubility of large RLP in a reasonable time by one central processing unit is verified. Partitioning of the problem into equal sizes is provided by parallel programming so that each particle contains the same number of enumeration. In this study, four RLP in which the largest problem has 36 activities are solved by exhaustive enumeration within reasonable solution time and it is proved that the proposed method is applicable. Exact solutions of larger problems can also be obtained by the proposed method if the problem is partitioned into smaller sizes.

Keywords: Resource leveling problem, optimization, critical path method, parallel programming.

1. GİRİŞ

Kaynak dengeleme, bir projenin hedeflenen tamamlanma süresi uzatılmadan proje süresince oluşan kaynak kullanımındaki dalgalanmaları ve en yüksek günlük kaynak talebini düşürmeyi amaçlar. En yüksek günlük kaynak talebindeki düşüş, kaynak ihtiyacının çok yoğun olduğu dönemlerde bolluğu olan aktivitelerin geciktirilerek daha az kaynak talebinin görüldüğü günlerde yapılmasının sağlanması ile elde edilir. Kaynak kullanımının çok yoğun günlerden az yoğun günlere kaydırılması ile inşaatın kaynak dengelemesi yapılmamış iş programına göre daha düşük en yüksek kaynak talebi ile gerçekleştirilmesi mümkün olacaktır. Bunun sonucunda personel ihtiyacının fazla olduğu dönemlerde ilave personelin işe alınma ve personel talebinin azaldığı dönemlerde işten çıkarılması gerekliliği azalacaktır. Benzer şekilde kısa süreli iş makinesi kiralama ve iade etme ihtiyacı azaltılacaktır. Kısa süre içinde oluşan kaynak kullanımındaki dalgalanmalar azaltılacağı için proje süresince kaynakların atıl kaldığı süreler de düşürülecektir. Bu sayede inşaat projelerinin maliyetlerinde önemli tasarruflar elde edilebilecektir. Direk yararı yanı sıra personel hareketliliğinin azalması sonucu çalışanların öğrenme eğrisi ve iş alışkanlıkları süreklilik gösterecek ve proje yönetimi kolaylaşacaktır. Kötü proje yönetimi inşaat işlerinde süre ve maliyet aşımına yol açmaktadır [1]. Belirtilen doğrudan katkılara ek olarak kaynak dengelemenin oluşturacağı iş sürekliliği proje yönetimi konusunda sağlayacağı dolaylı katkı olacaktır.

1.1. KDP'nin Özellikleri

KDP'nin literatürde birkaç türü bulunmaktadır. Aktivitelerin ertelenmesinde bolluk sürelerine ek olarak minimum ve maksimum erteleme kısıtları konularak kaynak dengeleme problemi çözülebilir [2]. Erteleme üzerindeki kısıtların sektördeki karşılığına sıra işi tamamlandığında boyanın başlayabilmesi için sıvanın kuruması gerekliliği örnek olarak gösterilebilir. Minimum erteleme süre kısıtı sıvanın boya yapılabilir düzeyde prizini alıp kuruması olarak tanımlanabilir. Maksimum erteleme kısıtına ise donatı işinin tamamlanmasından sonra beton dökümünün başlaması için konulabilecek maksimum zaman

aralığı örnek verilebilir. Beton dökümünün başlama zamanı çok geciktirilirse donatı paslanabilir veya rüzgâr vb. çevresel etkilerden dolayı kolon donatıları şakul doğrusundan sapıp yana eğilebilir. Benzer sorunları önlemek için aktiviteler arasında maksimum erteleme süre kısıtı tanımlanabilir.

KDP aktivite sürelerinin rastgele değişken olarak tanımlanıp stokastik optimizasyon problemi olarak da çözülmüştür [3]. Ayrıca aktivitelere birden fazla tamamlanma sürelerinin atanabildiği çok modlu KDP de incelenmiştir [4,5]. Abadi vd. KDP'yi paranın zaman değerini dikkate alarak çözmüştür [6]. Kaynak dengelemenin başka bir türünde ise aktivitelerin yürütülmesi sırasında aktiviteler durdurulup bir süre sonra tekrar başlanabilir [7] ve buna ek olarak farklı tamamlanma sürelerine göre kaynak dengelemesi yapılabilir. İnşaat işlerinde ve endüstride genellikle aktivitelerin yarıda kesilip bir süre sonra devam edilmesi çok tercih edilir bir işleyiş değildir. Bu nedenle bu çalışmanın kapsamı kaynak dengeleme probleminin elde edilen ilk proje tamamlama süresini aşmadan ve aktivitelerin yarıda kesilmeden kaynak dalgalanmasını en aza indirme hali ile sınırlı tutulmuştur.

1.2. KDP'nin Çözümünde Uygulanan Yöntemler

Kaynak dengeleme probleminin çözümü için sezgisel, modern sezgisel ve matematiksel yöntemler uygulanmaktadır. Harris, kaynak dengeleme probleminin çözümü için 'pack metodu' adını verdiği sezgisel bir yöntem geliştirmiştir [8]. Bu yöntemde aktivitelerin günlük kaynak taleplerinin fazlalığına göre her aktiviteye öncelik atanmaktadır. Kaynak kullanımın fazla olduğu günlerde önceliği en düşük olan aktiviteden başlayarak aktiviteleri sırayla geciktirerek günlük kaynak histogramının karelerinin toplamını en aza indirmeyi amaçlamıştır. Hiyassat (2001), minimum moment kaynak dengeleme ölçeğini revize ederek modifiye edilmiş minimum moment ölçeğini geliştirmiştir [9]. Bu ölçeği kullanarak birden fazla kaynağı dengeleyen yöntem geliştirmiştir. Karşılaştırma amacıyla, örnek problemi iyileştirilmiş minimum moment ve geleneksel minimum moment yöntemiyle çözümlenmiştir. Harris ve Hiyassat tarafından önerilen yöntemler 11 aktiviteden oluşan küçük projeler üzerinde denenmiştir. Yöntemler daha büyük projelerde en iyi sonuçtan çok uzak sonuçlara yakınsamaktadır. Sezgisel yöntemler sadece küçük projelerde iyi sonuçlar vermekte, büyük projelerde modern sezgisel ve matematiksel yöntemlere göre yetersiz kalmaktadır. Bu nedenle sezgisel yaklaşımlar inşaat maliyetlerini düşürmek için uygun yöntemler değildir.

KDP NP-zor sınıfı bir problem türü olduğu için [10, 11] aktivite sayısı artınca problemin zorluk derecesi üstel biçimde artmakta ve sabit kurallar çözüm üretemez duruma gelmektedir. Sezgisel yöntemler yetersiz kaldığı için modern sezgisel yöntemler de kaynak dengeleme probleminin çözümünde uygulanmaktadır. Hegazy (1999) KDP için Genetik Algoritma (GA) tabanlı bir optimizasyon aracı geliştirmiştir [12]. Son ve Skibniewski (1999) Tavlama Benzetimi algoritması ile KDP'yi çözmüştür [13]. Çalışmasında 11 ve 13 aktiviteli bir projenin tam çözümünü elde etmiştir. Leu vd. (2000) çok kaynaklı KDP'nin çözümü için GA uygulamıştır [14]. GA, ayrıca KDP'nin çözümü için Zheng vd. (2003), El-Rayes ve Jun (2009), Christodoulou vd. (2009), Ponz-Tienda vd. (2013) ve Li vd. (2017) tarafından da kullanılmıştır [15-19]. Parçacık Sürü Optimizasyonu KDP'nin çözümü için Qi vd. (2007) tarafından uygulanmıştır [20]. Li vd. (2010) ve Geng vd. (2011) Karınca Koloni Algoritmasını KDP'nin çözümü için uygulamışlardır [21, 22]. Tran ve Hoang (2014) Diferansiyel Evrim algoritmasını 44 aktiviteli bir KDP'nin çözümünde kullanmıştır [23]. Xu

vd. yapay arı kolonisi algoritmasını, Prayogo ve Kusuma ile Prayogo vd. simbiyotik organizma arama yöntemini kullanarak KDP'yi çözmüştür [24-26]. Modern sezgisel yöntemlerle KDP çözümünde tam sonucu elde edilen en büyük proje 44 aktivitelidir ve Tran ve Hoang tarafından elde edilmiştir.

KDP'de aktivite sayısı arttıkça problemin arama uzayı üstel biçimde arttığı örnek problemler üzerinde gösterilmiştir [27]. Bu nedenle büyük projelerin kaynak dengeleme probleminin modern sezgisel yöntemlerle çözümünde popülasyon boyutunun yüksek tutulması ve yenileme sayısının artırılması gerekmektedir. Bu sorunu çözmek için öğretme-öğrenme tabanlı optimizasyon yaklaşımı ile popülasyon oluşturulması yoluna gidilmiştir [28]. Modern sezgisel yöntemlerde tekrar sayısı ve popülasyon sayısının artması bellek kullanımı açısından bir sorun oluşturmazken hesaplama süresini çok hızlı artırmaktadır. Bu nedenle büyük projelerin çözümünde belirli süre hesap yapıldıktan sonra yakınsama olmamasına rağmen hesaplamalar durdurulmaktadır. Ayrıca modern sezgisel yöntemlerle gerçekleştirilen optimizasyonda küresel optimuma erişilip erişilmediği kesin olarak bilinmemektedir.

Sezgisel ve modern sezgisel yöntemlerin tam sonucu garantileyememeleri nedeniyle KDP'nin çözümü için matematiksel programlamaya dayalı kesin çözüm elde edebilen algoritmalar geliştirilmiştir. Karaa ve Nasr (1986) en az iş makinesi kira bedelinin elde edilmesini sağlayacak şekilde doğrusal programlama ile kaynak dengeleme problemini formülleştirilmiştir [29]. Takamoto vd. (1995) kuadratik programlama ile KDP'yi çözmüştür [30]. Easa (1989), Hariga ve El-Sayegh (2010) tamsayı-doğrusal programlamayı uygulamışlardır [31, 32]. Karışık tamsayı programlama ise Gather vd. (2011) ile Rieck ve Zimmermann (2015) ile Mattila ve Abraham (1998) tarafından uygulanmıştır [33-35]. Matematiksel yöntemlerin ortak özelliği kesin çözümü sunabilmeleridir. Fakat aktivite sayısı arttıkça denklemdaki bilinmeyen sayısı çok hızlı artmaktadır. Bu durum problemin çözüm süresini çok hızlı artırmakta ve hesaplamalar kesin sonuç alınmadan durdurulmaktadır. Bu nedenle büyük KDP'nin çözümünde matematiksel yöntemler tam sonuç verememektedir. Çözülen şebekenin karmaşıklığına da bağlı olmakla birlikte kesin yöntemlerle çözülebilen en büyük proje 50 aktivitelidir [34].

1.3. Motivasyon

Kaynak dengeleme probleminin çözümü için uygulanan sezgisel, modern sezgisel ve matematiksel programlama yöntemleri büyük boyutlu problemlerin kesin çözümü için elverişli değildir. Bu çalışmada mevcut yöntemlere eklemeler yaparak performanslarının artırılması yerine kaynak dengeleme probleminin uygulanabilir çözümlerinin tamamını deneyerek en iyi çözümü elde etme yaklaşımı geliştirilmiştir.

Kaynak dengeleme probleminde uygulanabilir çözümler bolluğu olan aktivitelerin aktiviteler arasındaki öncelik ilişkilerini ve bolluk sürelerini ihlal etmeden geciktirilmesi ile sağlanır. Toplam bolluk süresi aktivitenin ertelenebileceği toplam süreyi, serbest bolluk ise aktiviteden sonra başlayacak olan aktivitelerin başlangıç zamanlarını etkilemeden gerçekleştirilebilecek erteleme süresini tarif eder. Elde edilen tüm uygulanabilir çözümler arama uzayını oluşturur. Kaynak dengeleme probleminin tüm arama uzayının taranması, uygulanabilir çözümlerin tamamının denenmesini kesinleştireceği için garantili biçimde en iyi çözümün elde edilmesini sağlar.

Arama uzayının hesaplanabilmesi için kritik yol yöntemi ile şebekenin çözülerek kritik ve kritik olmayan aktiviteler belirlenir. Arama uzayının belirlenmesi birbirleri ile paralel olan hatların bolluk sürelerine sıfır gün erteleme dâhil edilerek bolluk sürelerinin bir fazlası ile çarpılması, birbirlerine seri olan hat üzerindeki aktivitelerin ise önce başlayan aktivitelerin kendinden sonra başlayan aktivitenin ertelemesini aşmayacak şekilde kaç farklı biçimde geciktirilebileceğinin hesaplanması ile gerçekleştirilir. Şebekedeki bağımlılık ilişkilerini ihlal etmeden ve proje süresini uzatmadan kritik olmayan aktivitelerin kaç farklı şekilde ertelenebileceği hesaplanarak arama uzayının büyüklüğü belirlenir [27].

Bu çalışmanın öncesinde arama uzayını belirleyip tamamını tarayarak KDP çözümü 13-aktiviteye kadar projeler için Erzurum ve Bettemir (2018) ve Bettemir ve Erzurum (2019) tarafından gerçekleştirilmiştir [36, 37]. Ayrıca KDP'nin arama uzayının hesaplanması ve çözülmesi Erzurum (2019) tarafından detaylı biçimde incelenmiştir [38]. Problemin NP-zor olmasından dolayı büyük projelerin çözümünde tüm arama uzayının tek işlemci ile taranması oldukça zaman alacaktır. İhtiyaç duyulan hesaplamaların birden fazla işlemciye dağıtılarak gerçekleştirilmesi ile hesaplama sürelerinde önemli kısaltmalar elde edilebilmektedir. Fakat paralelleştirme yapabilmek için problemin paralel programlamaya uygun olması gereklidir [39].

Parallelleştirme ince tanecikli, orta tanecikli ve iri tanecikli olarak üç sınıfa ayrılabilir. İnce taneli paralelleştirmede problem çok sayıda küçük alt probleme bölünür ve hesaplama çok sayıda işlemciye dağıtılarak gerçekleştirilir. İnce tanecikli paralelleştirmede işlemcilere düşen yük az olur ve iş miktarları birbirine yakın boyuta getirilebilir. Fakat işlemci sayısı çok artınca işlemciler arasında veri akışı ihtiyacı çok yükselebilir ve senkronizasyon sorunları ortaya çıkabilir. Kaba taneli paralelleştirmede problem daha az sayıda parçaya bölünür. Problemin az sayıda parçaya bölünmesi her parçanın birbirine yakın hesap yükü içerecek şekilde paylaşılmasını garantileyemeyebilir. Bu nedenle bazı işlemciler hesaplamayı erken tamamlayıp boşa kalabilir. Kaba taneli paralelleştirmenin avantajı ise işlemciler arası veri alışveriş ihtiyacının az olmasıdır [40]. Tüm arama uzayının taranması ile gerçekleştirilen KDP çözümünde işlemciler arası veri akışı ihtiyacı sıfır olduğu ve problem eşit hesap yüküne sahip çok sayıda parçaya bölünebildiği için yüksek verimli biçimde ince taneli paralelleştirme yapılabilmesine elverişlidir.

Paralel programlama optimizasyon sürecinde yaygın olarak kullanılmaktadır, özellikle modern-sezgisel yöntemler paralel programlama ile çözülmeye yatkındırlar. Popülasyon tabanlı yöntemler oldukları için paralelleştirme bireylerin çözümü üzerinden gerçekleştirilir ve hesaplamalar birey sayısı kadar işlemciye dağıtılabilir. Bu sayede hesaplama süresinde önemli tasarruflar elde edilebilmektedir [41,42]. Kandil ve El-Rayes zaman-maliyet ödünleşim (*trade-off*) problemini Genetik Algoritma (GA) ile optimize etmiştir [43, 44]. 720 aktiviteli proje analiz edilirken hesaplamalar 50 işlemciye dağıtılmış ve hesaplama süresi sekizde bire düşürülmüştür. Modern sezgisel yöntemlerde yenilemenin tamamlanabilmesi için tüm bireylerin çözülmüş olması gerekir. Bu nedenle her yenilemenin tamamlanma hızı en yavaş bireyin hızına eşit olmaktadır. Bu durum hesaplama süresinden elde edilebilecek tasarrufu düşürmektedir. Kandil vd., (2010) GA ile çözdüğü optimizasyon probleminde kaba taneli (*coarse grained*) paralelleştirme yapmış ve 50 işlemci kullanarak gerçekleştirilen paralelleştirmede global paralelleştirmeye göre hesap süresini üçte bir oranında kısaltmıştır [44]. Hesaplama süresinin istenilen ölçüde kısaltmamasının nedeni optimizasyon sürecinin her yenilemesinde en yavaş işlemcinin beklenmesi zorunluluğudur. İşlemcilere dağıtılan

hesap yükü tek şebeke çözümünün yapılmasıdır. Her yenilemede tüm işlemcilerin hesaplamasını tamamlaması beklenmekte ve en yavaş işlemci çözüm süresini belirlemektedir.

Kaynak dengeleme probleminin doğrusal programlama veya dal-sınır algoritmaları ile çözümü verimli biçimde paralelleştirmeye çok elverişli değildir. Denklemler probleme bağlı olduğu için çözümü sistematik biçimde verimli olarak parçalanamamaktadır. Parallelleştirme ağırlıklı olarak matris tersinin alınması veya pivotlama sırasında yapılmaktadır. Bunun dışındaki süreçte seri hesaplama yapılmakta ve paralelleştirme düşük verimde olduğu için hesaplama süresinde kayda değer bir tasarruf sağlanamamaktadır.

Yapım yönetimi alanında paralel programlama kullanarak gerçekleştirilen optimizasyon çalışmaları ile ilgili literatür oldukça sınırlıdır. Kaynak dengeleme problemi genel olarak modern sezgisel yöntemlerle gerçekleştirilmekte ve orta büyüklükteki projelerde yakın-optimum sonuçlar elde edilmektedir. Problemin zorluk düzeyi arttıkça yakın-optimum'dan daha da uzaklaşmaktadır. Günümüzde yaygınlaşan çok çekirdekli işlemciler ele alındığında yakın-optimum sonuçlarla yetinmek ve inşaat maliyetlerini yeteri kadar düşürememek önemli bir eksikliktir. Bu nedenle tüm arama uzayını paralel-programlama ile tarayarak garantili biçimde KDP'nin en iyi çözümünü elde eden bir yöntem önerilmiştir. Makalenin 2. bölümünde arama uzayının nasıl belirlendiği ve tarandığı açıklanmakta, 3. bölümünde ise literatürden derlenen 3 hipotetik çalışma ile 2 katlı bina inşaatı projesinin paralel programlama ile çözümü ve elde edilen değerler sunulmakta, 4. bölümde yöntemin avantajı ve eksiklikleri tartışılmakta ve ileride yapılabilecek çalışmalara değinilmekte, 5. bölümde sonuç bölümü sunulmaktadır.

2. YÖNTEM

Bu çalışmada kaynak dengeleme probleminin kesin optimum sonucunun elde edilmesi için geliştirilen yöntem; kaynak dengeleme probleminin arama uzayının belirlenmesi, paralelleştirmenin yapılması ve arama uzayının taranması işlem adımlarından oluşmaktadır. Belirtilen işlem adımlarının detayları bu bölümde açıklanmıştır.

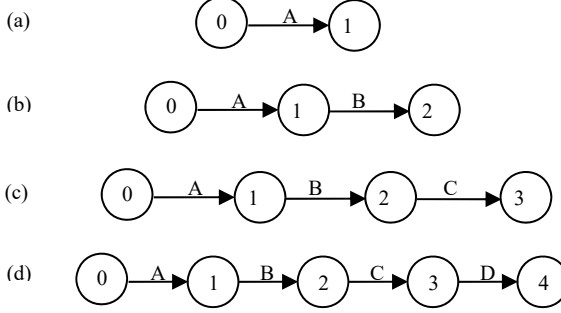
2.1. Arama Uzayının Belirlenmesi

Kaynak dengeleme probleminin tam sonucu tüm alternatifler denenerek elde edileceği için şebekenin bolluğa sahip aktivitelerinin geciktirilerek birbirinden farklı kaç iş programı oluşturulabileceğinin belirlenmesi gerekmektedir. Aktivitelere atanabilecek erteleme süresi kendinden önce ve sonra gerçekleşen aktivitelere atanan erteleme sürelerine de bağlı olduğu için toplam bolluk ve serbest bolluk hesabının daha sistematik biçimde yapılabildiği ok diyagramı ile şebeke analizleri gerçekleştirilmiştir. Bolluğu olan aktivitelerin kaç farklı şekilde ertelenebileceği kombinasyon hesabı ile belirlenir.

Şebekenin ok diyagramı ile oluşturulmasının ardından kritik yol yöntemi ile tüm aktivitelerin toplam bolluk ve serbest bolluk süreleri belirlenir. Bolluğu sıfır olan aktivitelerin oluşturduğu kritik hat dışında kalan hatlar üzerinde erteleme yapılacağı için şebekedeki tüm kritik olmayan hatlar belirlenir. Arama uzayının büyüklüğü aynı hat üzerinde yer alan birbirine seri aktiviteler ile paralel hatların kombinasyon sayılarının çarpılması ile hesaplanır.

2.1.1. Seri aktiviteler

Birbirine seri olan aktiviteler başlayınca biter türü ilişkinin tanımlanması ile oluşur. Öncel olan aktivitenin tamamlanması ile ardıl aktivite başlayabilir. Seri bağlı aktivitelere örnek Şekil 1'de gösterilmiştir.



Şekil 1 - Seri bağlı aktivite örnekleri

Tek aktiviteden oluşan hat Şekil 1.a'da gösterilmiştir. Tek aktiviteli hattın olması durumunda aktivitenin n gün toplam bolluğu olduğunda aktivite 0 gün erteleme dâhil olmak üzere n güne kadar ertelenebilecek ve toplamda $n+1$ adet erteleme durumu gerçekleşebilecektir. Birbirine seri bağlı iki aktivitenin n gün toplam bolluklarının olması durumunda gerçekleşebilecek erteleme sayısı öncel aktivitenin gecikme süresi göz önüne alınarak hesaplanır (Şekil 1.b). A aktivitesi öncel aktivite olduğu için kendinden sonra başlayacak olan aktivitelerin ertelenme süreleri mutlaka A aktivitesinin ertelenmesine eşit veya daha fazla olmalıdır. Bu nedenle aktivite bolluklarının kaç farklı şekilde gerçekleşebileceğinin hesaplanması sırasında öncel aktivitelerin ertelenme süreleri dikkate alınmalıdır. Öncel aktivitenin sıfır gün ertelenmesi durumunda ardıl aktivite 0 günden n güne kadar toplam $n+1$ adet farklı biçimde ertelenebilir. Öncel aktivitenin 1 gün ertelenmesi durumunda ise ardıl aktivite 1 günden n güne toplam n adet farklı biçimde ertelenebilir. Bu şekilde birbirine seri bağlı iki aktiviteden üretilebilecek erteleme seçenekleri Çizelge 1'de gösterilmiştir.

Çizelge 1 - Seri bağlı 2 aktivitenin ertelenme seçenekleri

A ertelenmesi	B ertelenme seçenek sayısı	B ertelenme seçenekleri
0	$n+1$	$0,1,\dots,n$
1	n	$1,2,\dots,n$
2	$n-1$	$2,3,\dots,n$
3	$n-2$	$3,4,\dots,n$
\vdots	\vdots	\vdots
$n-2$	3	$n-2,n-1,n$
$n-1$	2	$n-1,n$
n	1	n

Çizelge 1'in ikinci sütunundaki değerler toplandığında tüm erteleme seçeneklerinin toplam sayısı elde edilecektir. Toplam rakam, 1'den $n+1$ 'e kadar ardışık $n+1$ adet sayının toplamı olan $\frac{(n+1)(n+2)}{2}$ değerine eşit olacaktır.

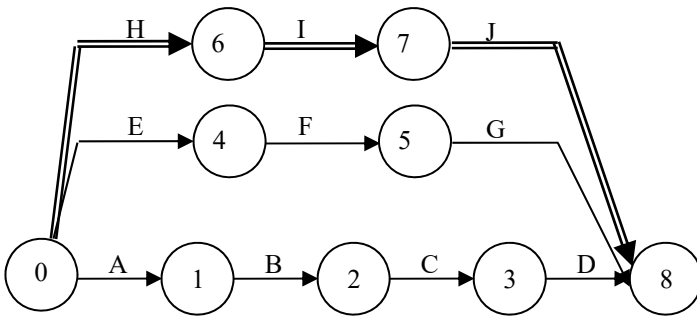
Birbirine seri bağlı aktivite sayısının 3 olması durumu Şekil 1.c'de gösterilmiştir. Aktivitelerin toplam bolluk süresi n gün olursa A aktivitesinin gecikmesi kademeli olarak 0 günden n güne arttırılırsa B ve C aktivitelerinin geciktirilmesi ile elde edilebilecek farklı iş programı sayısı EK-1'de gösterildiği gibi olacaktır. Seri bağlı 4 aktivite Şekil 1.d'de gösterilmiştir. İncelenen hattın n gün toplam bolluk süresine sahip olması durumu için A aktivitesinin gecikmesi kademeli olarak 0 günden n güne arttırıldığında B, C ve D aktivitelerinin geciktirilmesi ile elde edilebilecek farklı iş programı sayısı EK-2'de gösterildiği gibi olacaktır.

Seri hat üzerindeki aktivite sayısı arttıkça gerçekleştirilebilecek uygulanabilir erteleme sayısı sistematik bir şekilde artmaktadır. Birbirine seri bağlı m aktiviteden oluşan bir hat üzerinde toplam bolluğun n gün olduğu durumda şebekede tanımlanan bağımlılık ilişkilerini ihlal etmeden oluşturulabilecek erteleme sayısı Eşitlik 1'de genel haliyle ifade edilmiştir.

$$\frac{1}{m!} \prod_{i=1}^m (n+i) \quad (1)$$

2.1.2. Paralel Aktiviteler

Birbirine paralel olan aktivitelerin aynı anda yürütülmesinde şebeke kısıtları bir engel oluşturmaz. Bir aktivitenin başlayabilmesi için bir diğerinin tamamlanma zorunluluğu yoksa incelenen aktiviteler paralel olarak değerlendirilebilir. Şekil 2'de birbirine paralel olarak yürütülebilecek 3 hat gösterilmektedir.



Şekil 2 - Paralel hat örneği

Şekil 2'de gösterilen şebekede H-I-J, E-F-G ve A-B-C-D olmak üzere 3 paralel hat bulunmaktadır. H-I-J hattının kritik hat olduğu varsayılarak çift çizgi ile temsil edilmiştir. Kritik hattın bolluk süresi 0 olduğu için bu hattın tüm aktiviteleri 0 gün bollukla ertelenebilecektir. Bu nedenle H-I-J hattı kombinasyon hesaplamalarına dâhil edilmeyecek, kombinasyon hesabı kalan 2 hat üzerinden gerçekleştirilecektir. Birbirine paralel hatlar üzerinde bulunan E, F ve G aktiviteleri ile A, B, C ve D aktiviteleri birbirlerine seri bağlıdır. E-F-G hattı ile A-B-C-D hattının kaç farklı şekilde ertelenebilecekleri Eşitlik 1 kullanılarak hesaplanır. Hat1 (E-F-G) ile Hat2 (A-B-C-D) için hesaplanan erteleme seçenekleri birbirini etkilememektedir. Hat1 için tayin edilen ertelemeyen bağımsız biçimde Hat2 için erteleme süresi tayin edilebilmektedir. Bu nedenle Hat1 ve Hat2 birlikte düşünüldüğünde elde edilebilecek farklı erteleme seçeneği sayısı Hat1 ve Hat2 için elde edilen erteleme seçeneklerinin çarpımına eşit olacaktır. Paralel hatlardan elde edilebilecek erteleme sayısı, hatların erteleme sayılarının çarpımlarına eşit olacaktır.

2.1.3. Seri ve Paralel Hatların Birlikte Olduğu Şebekeler

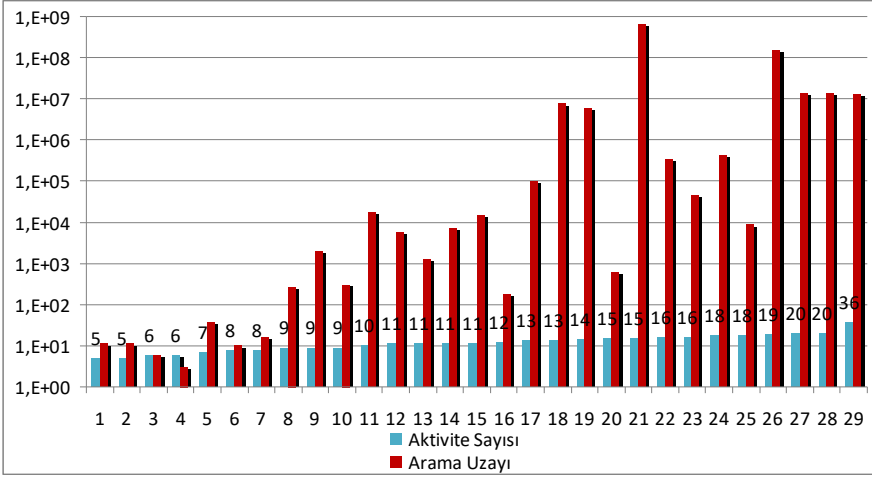
Aktiviteler arasındaki başlama ve tamamlama ilişkileri şebekede yer alan aktivite sayısı arttıkça karmaşıklaşmaktadır. Bu durumda seri ve paralel hatlar iç içe geçeceği için erteleme sayısının hesaplanmasında sadece seri ve sadece paralel hatlar için geliştirilen hesaplama yöntemi yetersiz kalacaktır. Bu nedenle seri ve paralel hatların yer aldığı şebekede toplam bolluk ve serbest bolluk süreleri göz önüne alınarak şebeke analiz edilir.

Karmaşık şebekelerde erteleme sayısının hesaplaması için öncelikle şebekede yer alan hatların belirlenmesi gereklidir. Kritik hatlar göz ardı edilir ve sadece bolluğu olan hatlar incelenir. Hatların kendi aralarındaki karmaşıklık düzeyine göre öncel aktivitelerin erteleme süreleri seri ve paralel hatların erteleme sürelerini etkileyebilmektedir. Bu durumda öncel aktivitelere tayin edilen erteleme süreleri için ayrı çözüm yapılması ve elde edilen erteleme sürelerinin toplanarak tüm şebekenin ertelenme süresinin hesaplanması gerekecektir.

2.2. Paralel Hesaplamanın Yapılması

KDP'nin arama uzayı belirlendikten sonra tüm arama uzayının taranması ile en iyi çözüm elde edilecektir. Literatürden derlenen 29 örnek problem için KDP'nin arama uzayı büyüklükleri Şekil 3'te yarı logaritmik çubuk diyagram biçiminde gösterilmektedir [38].

Şekil 3'te mavi çubuklar problemdeki aktivite sayısını, kırmızı çubuklar ise aktiviteler arasındaki öncelik ilişkilerini ve proje süresini ihlal etmeden elde edilebilecek farklı iş programı sayısını belirtmektedir. Grafiğin y ekseninin logaritmik ölçekte olmasına rağmen aktivite sayısı artınca arama uzayı büyüklüğünün hızlı biçimde arttığı görülmektedir. Aktivite sayısı dışında, arama uzayı aktivitelerin bolluk süreleri ve şebekenin karmaşıklığına da bağlıdır. En büyük arama uzayına sahip projede arama uzayı büyüklüğünün 1 milyara yaklaştığı görülmektedir. Kesin çözümün elde edilebilmesi için tüm uygulanabilir çözümlerin denemesi, 1 milyara yakın şebeke analizinin yapılması gerekmektedir. Bu kadar çok sayıda analizin tek işlemciyle gerçekleştirilmesi çok uzun süreceği için şebeke analizleri paralel programlama ile küçük parçalara ayrılıp birden fazla işlemciye dağıtılarak hesaplama süresi kısaltılmıştır.

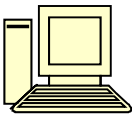


Şekil 3 - Aktivite sayısı arama uzayı arasındaki ilişki

Arama uzayının tamamının taranması paralelleştirmeye yatkın bir problem türüdür. Birbirine paralel olan hatlar üzerinden problem küçük parçalara ayrılabilirdiği için hesaplama sırasında işlemcilerin diğer hesaplamaların sonucunu beklemesine gerek kalmamaktadır. Bu sayede hesaplamalar kesintisiz biçimde yürütülebilmekte ve hesaplama süresinde önemli tasarruflar elde edilebilmektedir.

Tek bilgisayarda hesaplama

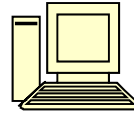
$p \cdot r$ adet hesaplama



Optimum Çözüm

p adet bilgisayarda hesaplama

Bilgisayar 1



r adet hesaplama
Çözüm r_1

Bilgisayar 2

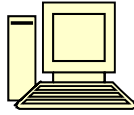


r adet hesaplama
Çözüm r_2

⋮

⋮

Bilgisayar p



r adet hesaplama
Çözüm r_p

Optimum Çözüm

Şekil 4 - Paralel hesaplama akış şeması

Şekil 2'de sunulan şebekede bolluğu olan 2 hat bulunmaktadır. Bu hatlar birbirinden bağımsız olduğu için Hat1 üzerinde tayin edilen erteleme süreleri Hat2'yi etkilememektedir. Hat1'in p adet, Hat2'nin ise r adet farklı erteleme seçeneğinin bulunduğu durumda Hat1 için tanımlanan i 'ninci ($i = 0, 1, \dots, p$) erteleme seçeneği tüm Hat2 erteleme kombinasyonları için i 'ninci bilgisayara (işlemciye) çözdürülerek problem p eşit parçaya bölünebilmektedir. Tüm işlemcilere Hat2 erteleme kombinasyonu sayısı kadar şebeke analizi yaptırılacağı için tek işlemciyle çözümden işlemci başına $p * r$ adet çözüm yapılacaktır, p adet bilgisayarda çözüm yaptırılarak her işlemciye r adet çözüm yaptırılarak daha kısa sürede çözüm elde edilecektir. Parallellendirme akış şeması Şekil 4'te gösterilmiştir. Şemada p eşit parçaya bölünebilecek bir hat olduğu varsayılarak paralellendirme yapılmıştır. Hattın p parçaya bölünememesi durumunda p 'den daha küçük olan ve eşit olarak parçalanabileceği en büyük sayıya bölünecek ve paralellendirme bölünen sayı üzerinden yapılacaktır.

2.3. Arama Uzayının Taranması

KDP'nin tüm arama uzayının taranması hesapçetveli uygulaması üzerinde çalışan bir makro ile gerçekleştirilmiştir. Problemin ardıl ilişkileri, aktivite süreleri ve kaynak talepleri kullanıcı tarafından hesapçetveli uygulamasına girilmektedir. Veri girişinin ardından aktivitelerin toplam ve serbest bolluk süreleri hesaplanmakta ve proje süresi belirlenmektedir. Başlangıç değeri olarak tüm aktivitelere 0 günlük erteleme süresi tayin edilmektedir. Aktivitelerin bolluk süresi sırayla arttırılacak şekilde döngüler oluşturulur. Atanan bolluk süresi dikkate alınarak tüm aktivitelerin başlangıç ve bitiş zamanları hesaplanıp kaynak talepleri dikkate alınarak proje süresince gerçekleşen günlük kaynak kullanım çizelgesi oluşturulur. Kaynak kullanım eğrisinin dalgalanması ve uygunluğu Eşitlik 2'de sunulan minimum moment (MM) olarak adlandırılan kaynak dağılım ölçüğü kullanılarak hesaplanmaktadır [8].

$$MM = \sum_{i=1}^S r_i^2 \quad (2)$$

Eşitlik 2'de S proje süresini, r_i i 'ninci günde ihtiyaç duyulan kaynak sayısını ifade etmektedir. Hesapçetveli üzerine yazılan makro ile bolluğu olan aktiviteler sistematik biçimde ertelenme süreleri 0'dan başlayarak toplam bolluk süresine kadar arttırılır. Her bolluk artışında şebekede tanımlı olan ardıl ilişkiler kontrol edilerek bu ilişkileri ihlal eden bir iş programının oluşturulması engellenir. Şebekede yer alan aktivitelerin başlangıç ve bitiş süreleri yeniden hesaplanıp kaynak kullanım eğrisi oluşturulur ve Eşitlik 2 kullanılarak kaynak dağılımının analizi yapılır. Elde edilen kaynak dağılım değeri o ana kadar elde edilen kaynak dağılım değerinden daha düşük ise dağılım değeri güncellenir ve en iyi çözüm olarak kaydedilir. Bu işlemler tüm arama uzayı taranana kadar tekrar edilir.

Eşitlik 2'de sunulan ve günlük kaynak talep değerlerinin karelerinin toplamını minimum yapmayı hedefleyen ölçek atalet momenti hesaplamasına benzetilmektedir. Alanın sabit olduğu durumda atalet momentinin en düşük olabilmesi kaynak kullanımının inşaat süresince her gün eşit miktarda olması ile gerçekleşir. Atalet momentinin teorik olarak en yüksek olduğu durum ise tüm işlerin bir günde gerçekleşmesi ile elde edilebilir. KDP'de projenin büyüklüğü ve karmaşıklığına bağlı olarak çok fazla yerel minimum noktası bulunabilir, hatta birden

fazla farklı iş programı en düşük moment değerini sunabilir. Bu durumda birbirinden farklı fakat birbirine denk çözümler elde edilmiş olur. Bu çalışmada problemin en iyi çözümü aranırken aktivitelerin erteleme süreleri sıfır günden başlayıp toplam bolluk süresine kadar ertelenmektedir. Arama yapılırken mevcut en iyi çözüme eş bir çözüm bulunursa mevcut çözüm güncellenmemektedir. Bunun nedeni denk çözümler arasından bolluğu olan aktivitelerin daha az ertelenmiş durumdaki hali ile oluşturulan iş programının seçilmesini sağlamak ve inşaat sırasında olası aksaklıklardan dolayı oluşabilecek süre uzamalarının proje süresinde en az gecikmeye yol açmasını sağlamaktır. Çünkü inşaat süresi boyunca dış etkenlerden dolayı ortaya çıkan çalışılmayan günler iş gecikmesinde önemli etkiye sahiptir [45].

Kaynak çeşidinin birden fazla olması durumunda problem çoklu kaynak dengeleme problemine dönüşmektedir. Bu durumda her kaynak çeşidi için günlük kaynak kullanım miktarlarının karelerinin toplamı hesaplanıp kaynak çeşidinin önemine göre bir ağırlıkla çarpılarak genel toplamı elde edilir. Belirtilen işlem Eşitlik 3'te tanımlanmıştır.

$$MM(\text{Çoklu kaynak}) = \sum_{i=1}^S \sum_{j=1}^k w_j r_{i,j}^2 \quad (3)$$

Eşitlik 3'te k , kaynak çeşidi sayısını, w_j ise j 'nci kaynak çeşidinin ağırlığını ifade etmektedir. Kaynak çeşitleri için tayin edilen ağırlıklar genellikle personel yevmiyesi veya kaynak iş makinesi ise günlük kira bedeli ile orantılı bir sayı olmaktadır. Kaynak maliyeti ne kadar fazla ise tayin edilen ağırlık da o kadar fazla olur. Bu sayede maliyeti yüksek kaynağa diğer kaynaklara göre daha fazla önem verilmiş olur. Çoklu kaynak dengeleme probleminin bir zorluğu bir kaynağı iyi bir şekilde dengelerken diğer kaynakların dağılımının önemli ölçüde bozulmasına neden olabilmesidir.

3. VAKA ANALİZLERİ

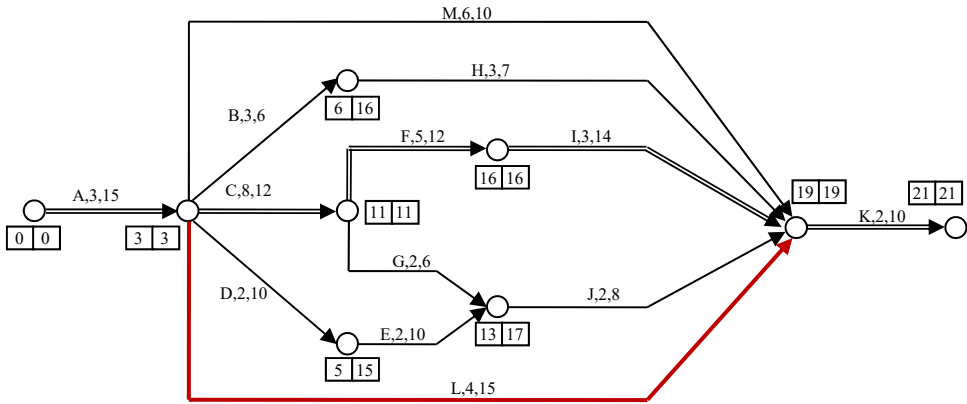
Bu çalışmada literatürden derlenen 3 proje ve yazarlar tarafından oluşturulan 2 katlı bir bina inşaatına ait 1 proje olmak üzere 4 orta büyüklükteki kaynak dengeleme probleminin optimum çözümü paralel programlama ile arama uzayının tamamı taranarak elde edilmiştir. Örnek problemlerin arama uzayı çok geniş olduğu için paralel programlama yapılarak hesaplama süresi düşürülmüştür. Hesaplamalar İnönü Üniversitesi İnşaat Mühendisliği Bölümüne ait bilgisayar laboratuvarında 3 GHz hızındaki Pentium D 925 model işlemciye sahip bilgisayarlar üzerinde yürütülmüştür. Geliştirilen yöntem hesapçetveli uygulaması üzerinde çalışan Visual Basic dilinde yazılan bir makro ile uygulanmıştır.

3.1. Problem 1

Örnek problem 1 Leu vd. (2000) tarafından oluşturulmuştur [14]. Şekil 5'te gösterildiği üzere proje 13 aktiviteden oluşmaktadır ve 5 aktivite kritiktir. Kritik hattın dışında şebekede birbirinden bağımsız 4 hat bulunmaktadır. Aktiviteler oklarla temsil edilmektedir. Okların üzerinde bulunan etiketteki ilk harf aktivite adını, ilk rakam aktivite süresini ve takip eden rakam aktivitenin yürütülmesi için gereken kaynak kullanımını belirtmektedir. Şebeke kritik

yol yöntemine göre çözümlenerek hesaplanan erken ve geç olay zamanları düğüm noktalarının üzerinde Şekil 5'te gösterildiği gibi belirtilmiştir.

Erken ve geç olay zamanları ile aktivite süreleri dikkate alınarak toplam bolluk ve serbest bolluk süreleri hesaplanır. Şebekede çift ok ile gösterilen A-C-F-I-K hattı kritik hat olduğu için bu hat üzerinde yer alan aktiviteler ertelenmediği için kritik hat arama uzayı hesabında yer alamaz. Arama uzayı M, B-H, J-D-E-G ve L hatlarının kaç farklı biçimde erteleneceği dikkate alınarak hesaplanır. M aktivitesinin toplam bolluk süresi 10 gündür. M aktivitesi için atanabilecek erteleme süreleri 0 gün ve 10 gün arasında olabileceği için toplam 11 farklı erteleme süresi tayin edilebilecektir. Benzer şekilde L aktivitesinin toplam bolluğu 12 gün olduğu için 13 farklı erteleme süresi tayin edilebilecektir.



Şekil 5 - Paralel programlama ile çözülen örnek problem-1 [14].

B-H hattı birbirine seri bağlı B ve H hatlarından oluşmaktadır. Bu hattın bolluk süresi 10 gündür. Hat üzerinde birbirine seri bağlı 2 aktivite olduğu için toplam farklı erteleme $\frac{(n+1)(n+2)}{2}$ adet olacaktır. Denklemde n yerine 10 girildiğinde 66 sonucu elde edilecektir.

J-D-E-G hattında aktivitelere atanabilecek erteleme süreleri G ve D-E dallarından dolayı tek formülle hesaplanamamaktadır. Bu nedenle oluşabilecek durumlar ayrı ayrı çözümlenerek uygulanabilir erteleme sayısı hesaplanacaktır. J aktivitesi hattın en sonunda yer aldığı için J aktivitesine atanan erteleme süresine göre çözüm yapılmıştır.

İlk durum analizinde 4 gün toplam bolluk süresine sahip olan J aktivitesinin ertelenme süresi 0 gün olarak tayin edilmiştir. Bu durumda G aktivitesinin ertelenme süresi 0 olmak zorundadır. Bununla birlikte D-E hattının 6 gün bolluğu vardır. İki aktiviteli bir hat olduğu için $n=6$, $m=2$ olacak ve $7*8/2 = 28$ adet birbirinden farklı erteleme seçeneği oluşacaktır.

İkinci durumda J aktivitesinin ertelenme süresinin 1 gün olarak tayin edilmesi incelenmiştir. Bu durumda G aktivitesinin ertelenmesi 0 veya 1 gün olabilirken D-E hattının bolluğu 7 gün olacaktır. D-E hattı üzerinde $n=7$, $m=2$ değerlerini alacak ve $8*9/2 = 36$ adet farklı erteleme oluşacaktır. G aktivitesi 0 ve 1 olmak üzere 2 farklı biçimde ertelenebilecektir. D-E ve G

üzerinde yapılan ertelemeler birbirinden bağımsız oldukları için J aktivitesinin 1 gün ertelendiği durum için $36*2 = 72$ farklı erteleme durumu oluşur.

Üçüncü durumda J aktivitesinin 2 gün ertelenmesi incelenmiştir. G aktivitesi için tayin edilebilecek erteleme süreleri 0, 1 ve 2 gün olurken D-E hattının bolluğu 8 gün olmuştur. İncelenen durum için birbirinden farklı erteleme sayısı $3*9*10/2 = 135$ olacaktır. J aktivitesinin ertelenmesi 1 gün artınca D-E ve G için tayin edilebilecek erteleme süreleri de 1 gün artmaktadır. J aktivitesinin 3 gün ertelenmesi durumunda oluşacak birbirinden farklı erteleme sayısı $4*10*11/2 = 220$, J aktivitesinin 4 gün ertelenmesi durumunda $5*11*12/2 = 330$ olacaktır. İncelenen tüm durumların toplamı $28+72+135+220+330 = 785$ yapmaktadır. Bu rakam D-E-G-J hattında birbirinden farklı ve şebeke kurallarını ihlal etmeyen kaç farklı erteleme yapılabileceğine eşittir.

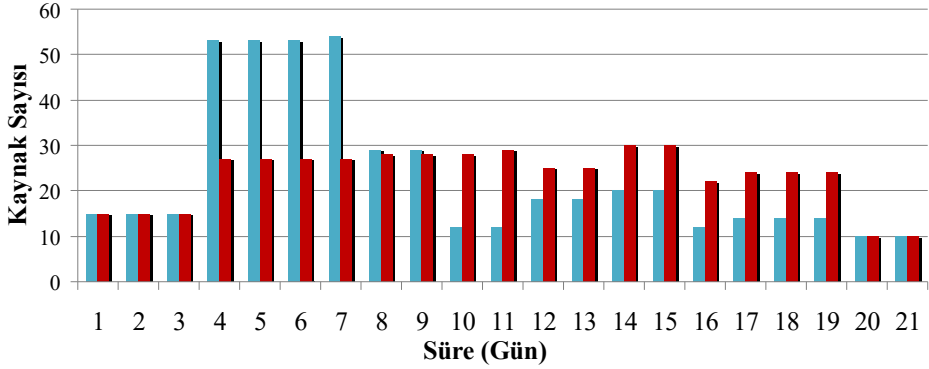
Çizelge 2 - Problem-1 için farklı erteleme sayısı hesaplaması

Hat	Ertelenme Sayısı
B-H	66
J-D-E-G	785
L	13
M	11
Arama Uzayı Büyüklüğü = $66*785*13*11 = 7.408.830$	

Projedeki birbirinden bağımsız kritik olmayan hatlar ve gerçekleştirilebilecek erteleme sayısı Çizelge 2'de gösterilmiştir. Hatlar birbirinden bağımsız olduğu için erteleme sayısı tüm hatların ertelenme sayılarının çarpımına eşit olacaktır. Örnek problem 1 için 7.408.830 adet birbirinden farklı erteleme yapılabileceği hesaplanmıştır. Bu proje için kaynak dengeleme probleminin en iyi çözümünü tüm arama uzayını tarayarak kesin olarak elde edebilmek için 7.408.830 adet iş programının çözülmesi gerekmektedir.

Bu kadar geniş bir arama uzayının tek işlemci ile çözülmesi çok zaman alacağı için problem parçalara ayrılarak çözülmüştür. L aktivitesi üzerinden 13 adet farklı erteleme gerçekleştirilebildiği için arama uzayının taranması L aktivitesi üzerinden 13 eşit parçaya bölünebilmektedir. Birinci parça L aktivitesine 0 gün, ikinci parça L aktivitesine 1 gün ve diğer parçalar içinde aynı kural uygulanarak son parça olan on üçüncü parça için L aktivitesinin sadece 12 gün ertelendiği durum çözdürülür. Diğer hatların tüm ertelemeleri çözdürüldüğünde 1 bilgisayarın çözmesi gereken iş programı sayısı $66*785*11 = 569.910$ olacaktır. Şekil 5'te paralelleştirmenin yapıldığı hat kırmızı ile işaretlenmiştir. Tek işlemci ile 15337 saniye süren çözüm, 13 bilgisayarda en uzununu 1195 saniyede elde edilmiştir. Hesaplama süresi yaklaşık 4 saat kısaltılarak 20 dakikada optimum çözüme garantili biçimde ulaşıldı. Maksimum günlük kaynak talebi 54'ten 30'a düşürülmüştür. Elde edilen kaynak dağılım grafiği Şekil 6'da sunulmuştur.

Problem1 Kaynak Histogramı



Şekil 6 - Problem 1'in başlangıç ve dengelenmiş kaynak kullanım grafiği

Çizelge 3 - Problem 1 için başlangıç ve optimum çözümlerin karşılaştırılması

Aktivite	Serbest Bolluk	Toplam Bolluk	Erken Başlangıç Zamanında Başlanırsa aktivitelerin başlama-bitiş zamanları			Optimizasyon sonrası aktivitelerin başlama-bitiş zamanları		
			Gecikme	Başlama	Bitiş	Gecikme	Başlama	Bitiş
A	0	0	0	0	3	0	0	3
B	0	10	0	3	6	4	7	10
C	0	0	0	3	11	0	3	11
D	0	10	0	3	5	4	7	9
E	6	10	0	5	7	4	9	11
F	0	0	0	11	16	0	11	16
G	0	4	0	11	13	0	11	13
H	10	10	0	6	9	4	10	13
I	0	0	0	16	19	0	16	19
J	4	4	0	13	15	0	13	15
K	0	0	0	19	21	0	19	21
L	12	12	0	3	7	0	3	7
M	10	10	0	3	9	10	13	19

Şekil 6'da mavi çubuklar tüm aktivitelerin erken başlangıç zamanında başlatılması ile elde edilen günlük kaynak kullanım grafiğini vermektedir. Kırmızı çubuklar ise optimizasyon sonrası elde edilen en iyi iş programıdır. Karşılaştırmanın daha kolay yapılabilmesi için başlangıç durumu ile en iyi çözüm aynı grafikte verilmiştir. Başlangıç durumu elde edilirken tüm aktivitelere erteleme süresi olarak 0 gün girilmiştir. Başlangıç koşulu olarak erken başlangıç zamanlarının alınma nedeni kritik yol yöntemi ile yapılan ilk çözümde tüm aktivitelere sıfır gün erteleme süresi tayin edilmesidir. Denklem 2'de verilen kaynak dağılım ölçeğinin en düşük değeri optimize edilmiş kaynak kullanım grafiği ile elde edilir. Grafiğin x eksenini gün biriminden zamanı, y eksenini ise günlük kullanılan kaynak miktarını belirtmektedir. Problem 1, 13 bilgisayarda 1195 saniyede çözülmüştür.

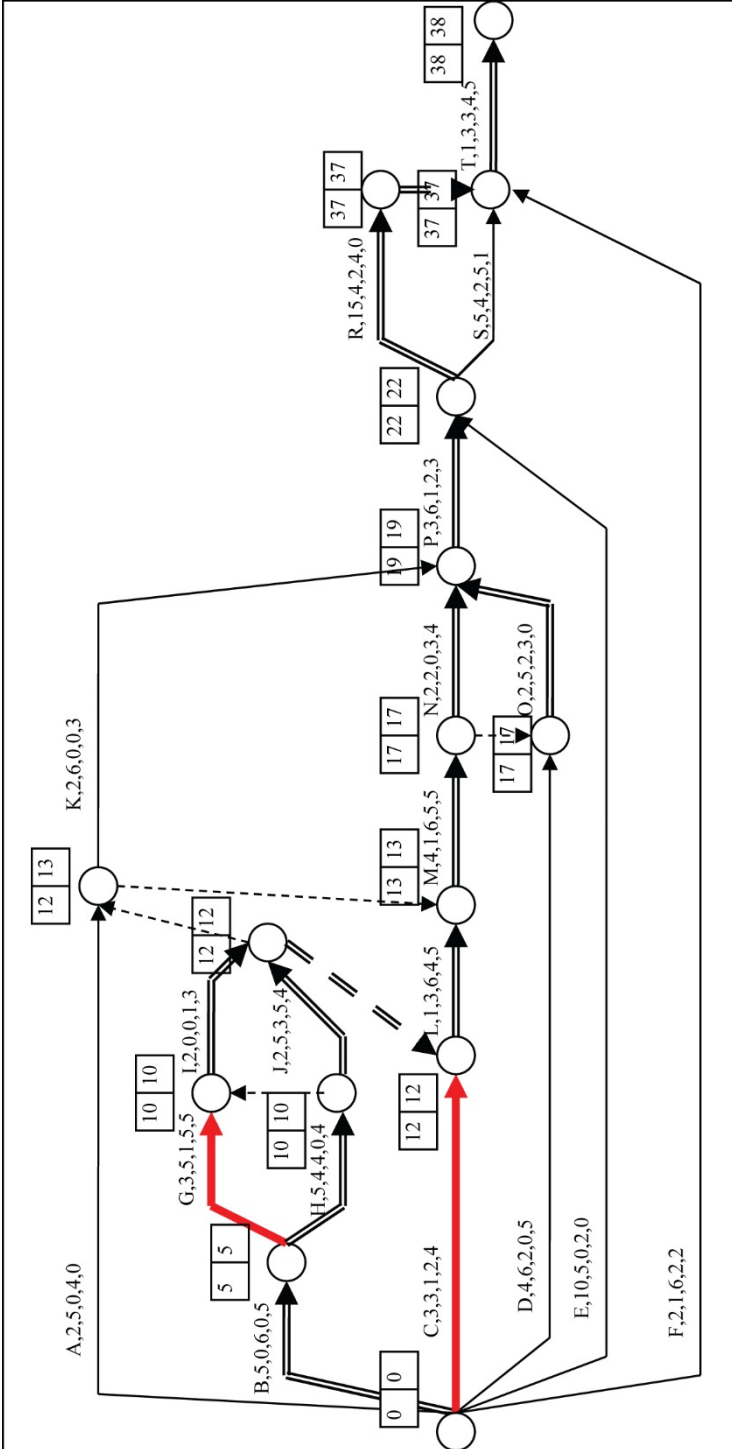
3.2. Problem 2

Literatürden elde edilen ikinci problem Stevens (1990) tarafından oluşturulmuştur [46]. Problem, 11'i kritik olmak üzere 19 aktiviteden oluşmaktadır. Problemin orijinal halinde aktivitelerin yürütülmesi için tek kaynak cinsi gereklidir. Bu problem Mutlu (2010) tarafından revize edilerek aktivitelerin gerçekleştirilebilmesi için her aktiviteye 4 farklı kaynak tayin ederek problemi çok-kaynaklı kaynak dengeleme problemine çevirmiştir [47]. Problemin şebeke diyagramı Şekil 7'de verilmiştir. Aktiviteler üzerinde yer alan etikette ilk harf aktivite adını, harfin ardından gelen ilk rakam gün biriminde aktivite süresini ve bu rakamı takip eden 4 rakam ise aktivitenin yürütülmesi için tayin edilmesi gereken kaynak miktarlarını belirtmektedir. Süreden sonra gelen ilk rakam ihtiyaç duyulan birinci kaynak miktarını, dördüncü rakam ise ihtiyaç duyulan dördüncü kaynak miktarını belirtmektedir.

A aktivitesinin 10 gün serbest, 11 gün toplam bolluğu bulunmaktadır. K aktivitesinin ise bolluk süresi A aktivitesi 10 güne kadar ertelendiğinde 5 gün, 11 gün ertelendiğinde ise 4 gündür. Sıfır günlük erteleme dikkate alındığında A aktivitesinin 10 güne kadar ertelenmesinde $11 \times 6 = 66$ farklı erteleme seçeneği ortaya çıkacaktır. A aktivitesinin 11 gün ertelendiği durumda ise oluşan 5 farklı erteleme eklendiğinde A-K hattı toplam 71 farklı erteleme seçeneğine sahip olur.

E aktivitesinin 12 gün bolluğu bulunmaktadır bu nedenle 13 farklı biçimde ertelenebilir. S aktivitesi 10 gün bolluğa, 11 farklı erteleme seçeneğine sahiptir. E ve S aktivitelerinin ertelemeleri birbirinden bağımsız olduğu için E-S hattı 143 farklı biçimde ertelenebilecektir. G, C, D ve F hatlarının ertelenme sayıları ise bolluk sürelerinin bir fazlasına eşit olacaktır. Projeyi oluşturan hatların erteleme sayıları Çizelge 4'te verilmiştir.

Problemin tüm arama uzayı taranarak tam çözümünün elde edilebilmesi için 153 milyondan daha fazla iş programının çözülmesi gerekmektedir. Hesaplama süresini kısaltmak için paralel hesaplama ile problem alt parçalara ayrılıp farklı bilgisayarlarda daha kısa sürede çözülmesi yoluna gidildi. Problem G hattı üzerinden parçalanırsa 3 parçaya, C hattı üzerinden parçalanırsa 10 parçaya ayrılabilir. Benzer şekilde D aktivitesi üzerinden 14 parçaya ayrılabilir. Bu şekilde yapılacak parçalamalar problemin yeterli kadar küçük parçaya ayrılmasını sağlamayacaktır. A-K hattı üzerinden yapılacak parçalamaya ise mevcut bilgisayar sayısından daha fazla olduğu için parçalamaya aşırı fazla sayıda olacaktır. En verimli biçimde parçalamaya yapabilmek için Şekil 7'de gösterildiği üzere G ve C hatları birleştirilerek problem 30 eşit parçaya ayrılmıştır.



Şekil 7 - Örnek problem 2 şebeke diyagramı [46, 47]

Çizelge 4 - Problem-2 için farklı ertelenme sayısı hesaplaması

Hat	Ertelenme Sayısı
A-K	71
E-S	143
G	3
C	10
D	14
F	36
Arama uzayı büyüklüğü: $71*143*3*10*14*36 = 153.513.360$	

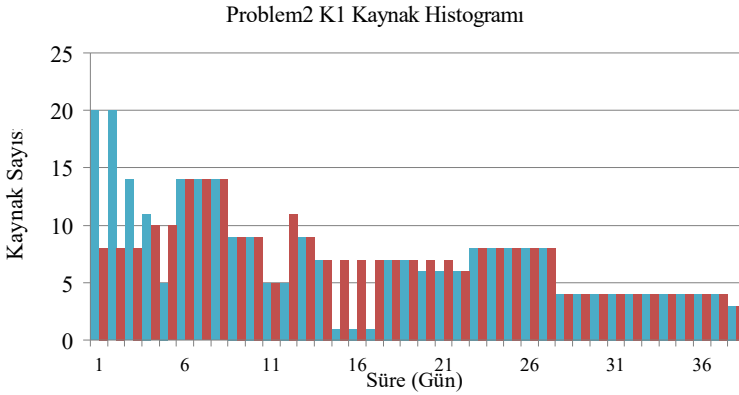
Çizelge 5 - Problem 2 için başlangıç ve optimum çözümlerin karşılaştırılması

Aktivite	Serbest Bolluk	Toplam Bolluk	Erken başlangıç zamanı aktivitelerin başlama-bitiş zamanları			Optimizasyon sonrası aktivitelerin başlama-bitiş zamanları		
			Gecikme	Başlama	Bitiş	Serbest Bolluk	Toplam Bolluk	Gecikme
A	0	11	0	0	2	3	3	5
B	0	0	0	0	5	0	0	5
C	9	9	0	0	3	0	0	3
D	13	13	0	0	4	11	11	15
E	12	12	0	0	10	0	0	10
F	35	35	0	0	2	19	19	21
G	2	2	0	5	8	0	5	8
H	0	0	0	5	10	0	5	10
I	0	0	0	10	12	0	10	12
J	0	0	0	10	12	0	10	12
K	5	5	0	12	14	3	15	17
L	0	0	0	12	13	0	12	13
M	0	0	0	13	17	0	13	17
N	0	0	0	17	19	0	17	19
O	0	0	0	17	19	0	17	19
P	0	0	0	19	22	0	19	22
R	0	0	0	22	37	0	22	37
S	10	10	0	22	27	0	22	27
T	0	0	0	37	38	0	37	38

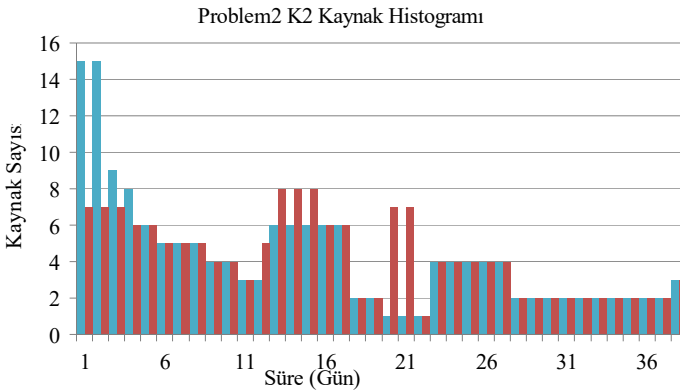
Parçalanmanın G ve C aktiviteleri üzerinden yapılabilmesi için C aktivitesinin ertelenmesi 0 gün iken G aktivitesinin 0, 1 ve 2 gün ertelemeleri sırası ile 1, 2 ve 3 numaralı bilgisayarlara çözdürülür. C aktivitesinin ertelenmesi 1 gün olarak sabit tutulduğu durum için G aktivitesinin 0, 1 ve 2 gün ertelemeleri sırası ile 4, 5 ve 6 numaralı bilgisayarlara çözdürülür. Bu şekilde paylaşım yapıldığında C aktivitesinin ertelenmesi 9 gün olduğunda C aktivitesinin 0, 1 ve 2 gün olan ertelemeleri sırası ile 28, 29 ve 30 numaralı bilgisayarlara

çözdürülür. Böylece problem her bilgisayara 5.117.112 iş programı çözümü düşecek şekilde 30 eşit parçaya ayrıldı. Hesaplamalar 30 bilgisayarda paralel bir şekilde yürütüldü ve 3 saatlik (10387 saniye) hesaplama sonrasında çözüm elde edildi. Aktivitelerin başlangıç ve optimizasyon sonrası başlama ve bitiş süreleri Çizelge 5'te sunulmaktadır.

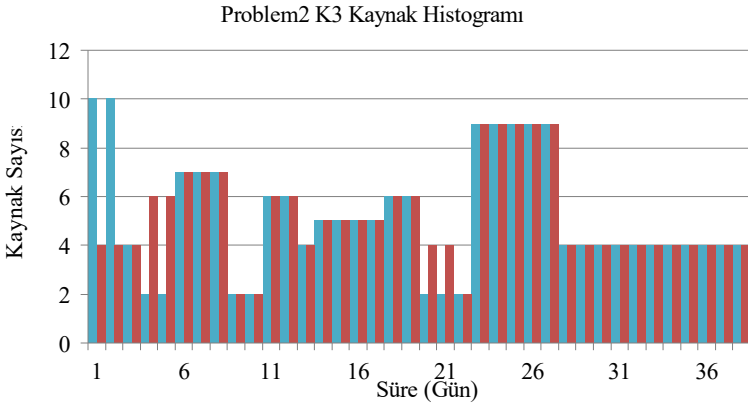
Çok kaynaklı kaynak dengeleme probleminin çözümünde eşitlik 3'te yer alan j indisi 1'den 4'e kadar saydırılarak amaç fonksiyonu tüm kaynak türleri için uygulanmış ve elde edilen değerlerin toplamının en düşüğü aranmıştır. Tüm kaynakların ağırlığı 1 alınmıştır. Kaynağın temin zorluğu veya temin maliyeti dikkate alınarak farklı ağırlıklar verilerek de problem çözülebilir. Elde edilen iş programı sonucu oluşan 4 farklı kaynağın günlük kullanım değerleri 1, 2, 3 ve 4 numaralı kaynaklar için sırası ile Şekil 8.a, 8.b, 8.c ve 8.d'de gösterilmiştir. Şekil 8'de mavi çubuklar tüm aktivitelerin 0 gün ertelendiği durumdaki iş programına göre, kırmızı çubuklar ise en iyi çözüme göre oluşan kaynak kullanım değerlerini vermektedir. Maksimum günlük kaynak talebi K1 kaynağı için 20'den 14'e, K2 kaynağı için 15'den 8'e, K3 kaynağı için 10'dan 9'a ve K4 kaynağı için 16'dan 12'ye düşürülmüştür.



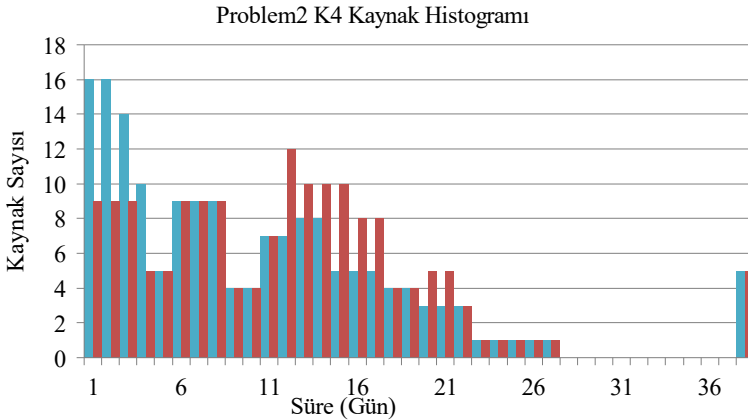
Şekil 8a - Örnek problem 2 Kaynak1 başlangıç ve dengelenmiş kaynak kullanım grafiği



Şekil 8b - Örnek problem 2 Kaynak2 başlangıç ve dengelenmiş kaynak kullanım grafiği



Şekil 8c - Örnek problem 2 Kaynak3 başlangıç ve dengelenmiş kaynak kullanım grafiği



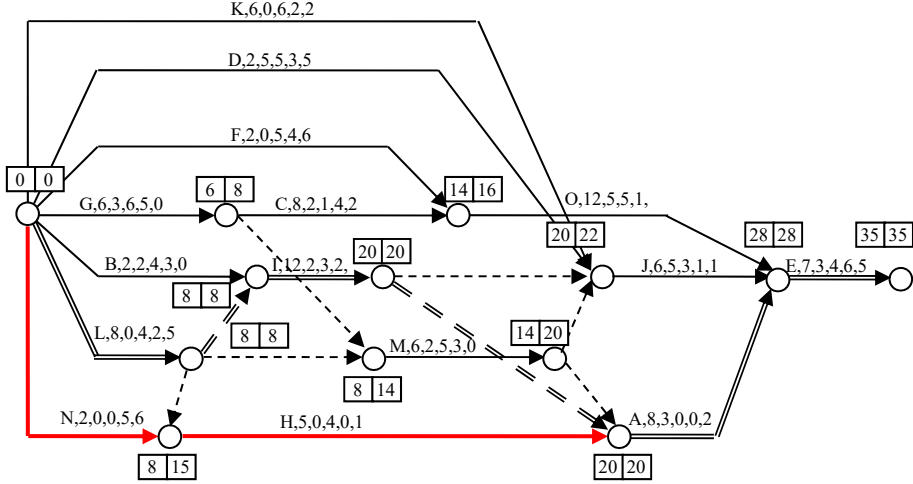
Şekil 8d - Örnek problem 2 Kaynak4 başlangıç ve dengelenmiş kaynak kullanım grafiği

3.3. Problem 3

Örnek problem 3 Stevens (1990) tarafından tek kaynaklı olarak üretilmiştir [46]. Bu problem Mutlu (2010) tarafından dört kaynaklı olacak şekilde revize edilmiştir [47]. Bu çalışmada problemin daha zor olması ve daha güncel hali olduğu için dört kaynaklı hali kullanılmıştır. Problemin şebeke diyagramı Şekil 9'da gösterilmektedir.

Şekil 9'da okların üzerindeki etiketin ilk harfi aktivite adını, harften sonra gelen ilk sayı aktivite süresini ve sonraki sayılar sırası ile kaynak 1'den kaynak 4'e kadar aktivitenin yürütülmesi sırasında görevlendirilmesi gereken kaynak miktarını belirtmektedir. Kaynaklar için 1 numaralı kaynaktan 4 numaralı kaynağa kadar sırası ile K1, K2, K3 ve K4 olarak kısaltmalar kullanılmıştır. Arama uzayının büyüklüğünün hesaplanabilmesi için birbirlerinin ertelenme sürelerinden etkilenen aktivitelerin belirlenmesi gereklidir. Şebeke diyagramı ve

bolluk süreleri birlikte değerlendirildiğinde J aktivitesinin D-K-M aktivitelere, O aktivitesinin C-F aktivitelere, C ve M aktivitelere G aktivitesine bağımlı olduğu görülmektedir. Kritik hat elendiğinde birbirinden bağımsız O-G-C-F, J-M-D-K-, N-H ve B olmak üzere toplam 4 hattın olduğu görülmektedir. Bu hatların her biri ayrı ayrı incelenerek kaç farklı biçimde ertelenebileceği tespit edilip bu hatlardan kaç farklı seçenek oluşturulabileceği hesaplanır.



Şekil 9 - Literatürden alınan 15 Aktiviteli örnek Kaynak Dengeleme Problemi [46, 47]

O-G-C-F hattının analizi çizelge 6'da sunulmuştur. Çizelgenin ilk sütunu O aktivitesine atanan gecikme süresini belirtmektedir. İkinci sütundan dördüncü sütuna kadar olan sütunların ilgili satırları ise O aktivitesinin gecikme sürelerine göre ilgili aktivitelere tayin edilebilecek erteleme sürelerini belirtmektedir. İkinci sütunda G aktivitesi için tayin edilebilecek gecikme süreleri yer almaktadır. Bu sütunda yer alan değerler O aktivitesinin gecikme süresi göz önüne alınarak tayin edilmiştir. Örneğin O aktivitesi 0 gün ertelendiğinde G aktivitesi sadece 0 gün ertelenebilmektedir. O aktivitesinin 1 gün ertelendiği durumda ise G aktivitesi 0 veya 1 gün ertelenebilmektedir. Üçüncü sütunda yer alan C aktivitesi ise O ve G aktivitelere gecikme sürelerine göre erteleme süresi tayin edilebilmektedir. Dördüncü sütunda ilk 3 sütunda belirtilen gecikme sürelerini ihlal etmeden F aktivitesine tayin edilebilecek gecikme sürelerini içermektedir. O, G ve C aktivitelere gecikmeleri 0 gün olduğu durumda F aktivitesinin gecikmesi 0 günden 12 güne kadar toplam 13 farklı biçimde olabilir. Çizelge 6'nın 5. sütununda belirtilen değerler toplandığında 145 değeri elde edilir.

J-M-D-K hattının ertelenme hesaplaması Çizelge 7'de sunulmuştur. Sütunlarda sırası ile J, M, D ve K aktivitelere ertelenme süreleri yer almaktadır. Bu hat üzerinde bulunan D aktivitesinin 18, K aktivitesinin 14 gün serbest bolluğu bulunmaktadır. M aktivitesinin serbest bolluğu kendisinden sonraki düğüm noktasının önündeki kukla aktivitelere nedeniyle 0 hesaplanmıştır. Fakat şebeke incelendiğinde M aktivitesinin de 6 gün serbest bolluğu olduğu görülmektedir. Bu durum dikkate alındığında J aktivitesi 0 gün geciktirildiğinde M aktivitesi 0 ile 6 gün, D aktivitesi 0 ile 18 gün, K aktivitesi 0 ile 14 gün arasında

ertelenebilmektedir. J aktivitesinin 0 gün ertelenmesi ile oluşabilecek birbirinden farklı erteleme sayısı tablonun beşinci sütununun ilk satırında gösterilmiştir. Hesaplamalar J aktivitesinin 1 ve 2 gün ertelenmesi için tekrar edildiğinde Çizelge 7'nin 2 ve 3. satırlarında belirtilen değerler elde edilir. J aktivitesinin ertelendiği her gün için M, D ve K aktiviteleri 1 gün daha uzun süre ertelenebilmektedir. Çizelgenin 5. sütununda belirtilen değerler toplandığında 6734 değeri elde edilir.

Çizelge 6 - O-G-C-F hattının için farklı erteleme sayısı hesaplaması

O ertelenme	G ertelenme	C ertelenme	F ertelenme	Ertelenme Sayısı
0	0	0	$0 \leq F \leq 12$	13
1	0	0	$0 \leq F \leq 13$	14
	0	1	$0 \leq F \leq 13$	14
	1	1	$0 \leq F \leq 13$	14
	2	1	$0 \leq F \leq 13$	14
2	0	0	$0 \leq F \leq 14$	15
	0	1	$0 \leq F \leq 14$	15
	1	1	$0 \leq F \leq 14$	15
	0	2	$0 \leq F \leq 14$	15
	1	2	$0 \leq F \leq 14$	15
	2	2	$0 \leq F \leq 14$	15

Çizelge 7 - J-M-D-K hattı için farklı erteleme sayısı hesaplaması

J erteleme	M erteleme	D erteleme	K erteleme	Ertelenme Sayısı
0	$0 \leq M \leq 6$	$0 \leq D \leq 18$	$0 \leq K \leq 14$	$7*19*15 = 1995$
1	$0 \leq M \leq 6$	$0 \leq D \leq 19$	$0 \leq K \leq 15$	$8 * 20 * 16 = 2240$
2	$0 \leq M \leq 6$	$0 \leq D \leq 20$	$0 \leq K \leq 16$	$9*21*17 = 2499$

N-H hattının ertelenme sayısının hesaplanması N aktivitesinin ertelenme süresine bağlıdır. N aktivitesi 0 gün ile 6 gün arasında ertelendiğinde H aktivitesi 0 gün ile 7 gün arasında ertelenebilir. Bu durumda N aktivitesi 7, H aktivitesi ise 8 farklı biçimde ertelenebileceği için iki aktiviteden $7*8 = 56$ farklı biçimde ertelenebilir. N aktivitesinin 7 gün ve daha uzun süre ertelendiği her gün için H aktivitesinin ertelenebileceği gün sayısı 1 gün azalmaktadır. N aktivitesi 7 gün ertelendiğinde H aktivitesinin 7 farklı ertelenme seçeneği bulunurken N aktivitesinin ertelenme süresi 13 gün olduğunda H aktivitesinin ertelenme seçeneği sadece 13 gün olan 1'e düşmektedir. N-H aktivitelerinin 7-13 gün arası farklı ertelenme seçeneği $7*8/2 = 28$, tüm farklı ertelenme seçeneği toplam $56 + 28 = 84$ olacaktır.

Projede kritik hat dışında 4 ayrı kritik olmayan hat bulunmaktadır. Bu hatlar birbirinden bağımsızdır. Bir hattın ertelenme süresi diğer hatların ertelenme süresini etkilemediği için tüm şebekenin toplam ertelenebilirlik sayısı Çizelge 8'de verilen hatların erteleme kombinasyon sayılarının çarpımına eşittir.

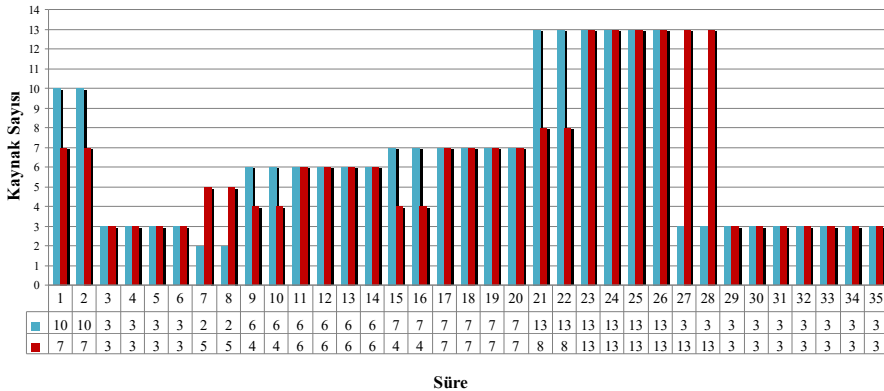
Çizelge 8 - Örnek proje 3'ün erteleme kombinasyon sayısı

Hat	Ertelme Sayısı
O-G-C-F	145
J-M-D-K	6734
N-H	84
B	7
Toplam	$145 * 6734 * 84 * 7 = 574.140.840$

Örnek problem 3'ün arama uzayı büyüklüğü 574.140.840 adet uygulanabilir iş programı olarak belirlenmiştir. Bu çalışmada önerilen yöntem tüm arama uzayının tamamının taranması ile en iyi çözümün elde edilmesini sağlamaktadır. Hesaplanan arama uzayının tek işlemci veya bilgisayar ile taranması çok uzun sürede gerçekleştirilebileceği için yöntemin uygulanması pratik olmayacaktır. Çizelge 8'de verilen hatlar üzerinden problem daha küçük parçalara ayrılabilir ve her parça farklı bir bilgisayar veya işlemci üzerinde çözdürülebilir. Bu sayede hesaplama süresi önemli ölçüde kısaltılabilecektir. Problemin parçalara ayrılması B hattı üzerinden yapılırsa problem toplam 7 eşit parçaya ayrılacak ve B aktivitesinin ertelenme süresi her parça için 0 ile 6 arasında sabit tutularak O-G-C-F, J-M-D-K ve N-H hatlarından oluşacak erteleme kombinasyonları çözülecektir. Bu şekilde her parça $145 * 6734 * 84 = 82.020.120$ iş programı hesaplamasından oluşacaktır. Bu parçalama çözüm süresini kısaltmakla birlikte problem daha küçük parçalara ayrılabilirliği için tercih edilmemiştir.

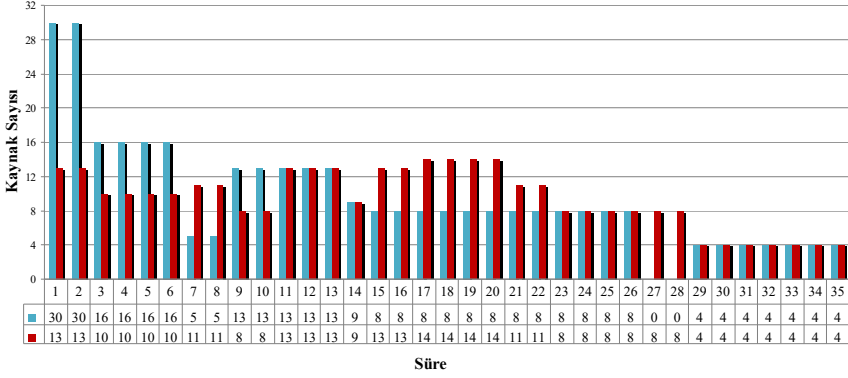
Paralel hesaplama için kullanılabilir 35 bilgisayar bulunmaktadır. Fakat Çizelge 8'de belirtilen hatlar (B hariç) bu rakamdan yüksektir. Parça sayısının eldeki bilgisayar sayısına yakın olması için hattın içindeki ertelenmeler kümelenecek parçalandı. N-H hattı 84 farklı ertelenmeye sahiptir bu hattın 35'den küçük fakat 35'e en yakın parçalanabilme sayısı 28'dir. N-H hattındaki ertelenme kombinasyonları 3'erli biçimde kümelenecek diğer hatlarla birlikte çözümlerse problem 28 eşit parçaya bölünmüş olacaktır.

Problem 3 K1 Histogramı



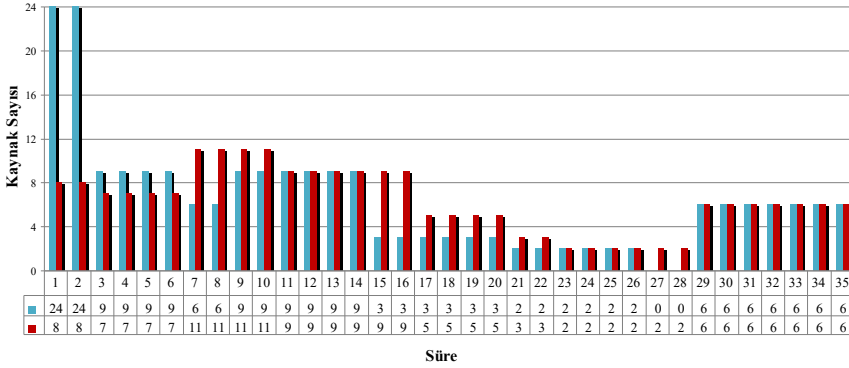
Şekil 10a - Örnek problem 3 Kaynaklı başlangıç ve dengelenmiş kaynak kullanım grafiği

Problem 3 K2 Histogramı



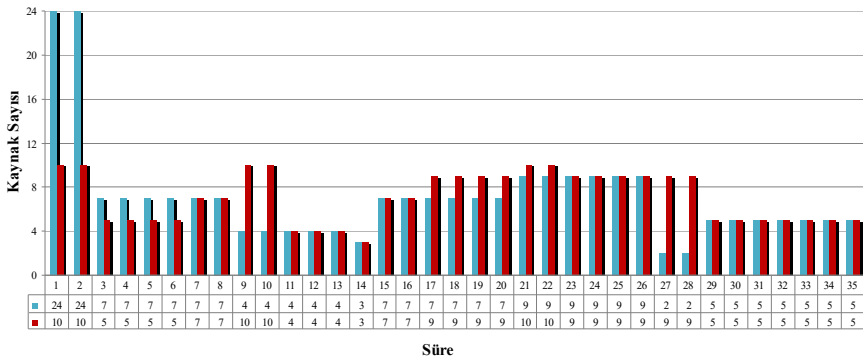
Şekil 10b - Örnek problem 3 Kaynak2 başlangıç ve dengelenmiş kaynak kullanım grafiği

Problem 3 K3 Histogramı



Şekil 10c - Örnek problem 3 Kaynak3 başlangıç ve dengelenmiş kaynak kullanım grafiği

Problem 3 K4 Histogramı



Şekil 10d - Örnek problem 3 Kaynak4 başlangıç ve dengelenmiş kaynak kullanım grafiği

Mevcut bilgisayar sayısına en yakın olacak şekilde problem $145 * 6734 * 3 * 7 = 20.505.030$ hesaplama olmadan oluşan parçalara ayrılacaktır. Hesaplamalar 28 bilgisayarda 7,7 saatte (27735 saniye) tamamlanmıştır. Hesaplama sonucu elde edilen en iyi kaynak kullanımı ile başlangıç değerleri K1, K2, K3 ve K4 için sırası ile Şekil 10a, 10b, 10c ve 10d'de verilmiştir.

Bilgisayarlar aynı model olsa dahi problemi aynı hızda çözememekte, çözüm süresi rastgelelik göstermektedir. Çözüm en yavaş olan bilgisayarın hesaplamasını tamamlamasının ardından elde edileceği için çözüm süresi en yavaş bilgisayarın hesaplama süresine eşit olacaktır. Bu nedenle parçalama sonrası elde edilen süre tasarrufu tek işlemci kullanılarak çözümlenmesine göre doğrusal biçimde azalmamaktadır. Şekil 10'da sunulan 4 farklı kaynağın başlangıç durumu ve optimize edilmiş durumları göz önüne alındığında önemli ölçüde iyileştirmenin elde edildiği görülecektir. Maksimum günlük kaynak talebi K1 için 13 olarak sabit kalmış, K2 için 30'dan 14'e, K3 için 24'ten 11'e, K4 için 24'ten 10'a düşürülmüştür. Ayrıca proje süresince kaynak kullanımındaki iniş çıkışların azaldığı görülmektedir. Bu sayede proje boyunca atıl durumda kalan kaynak miktarında önemli azalmalar sağlanmıştır.

Çizelge 9 - Problem 3 için başlangıç ve optimum çözümlerin karşılaştırılması

Aktivite	Serbest Bolluk	Toplam Bolluk	Başlangıç durumunda aktivite başlama-bitiş zamanları			Optimizasyon sonrası aktivite başlama-bitiş zamanları		
			Erteleme	Başlama	Bitiş	Erteleme	Başlama	Bitiş
A	0	0	0	20	28	0	20	28
B	6	6	0	0	2	0	0	2
C	0	2	0	6	14	0	6	14
D	18	20	0	0	2	0	0	2
E	0	0	0	28	35	0	28	35
F	12	14	0	0	2	14	14	16
G	0	2	0	0	6	2	2	8
H	7	7	0	8	13	0	8	13
I	0	0	0	8	20	0	8	20
J	2	2	0	20	26	2	22	28
K	14	16	0	0	6	16	16	22
L	0	0	0	0	8	0	0	8
M	0	6	0	8	14	2	10	16
N	6	13	0	0	2	8	8	10
O	2	2	0	14	26	2	16	28

3.4 Problem 4

Literatürden alınan örnek problemlerin genel yapısı birbirine paralel ilerleyebilen çok sayıda hattan oluşmalarıdır. Bu durum problemin çözümünü zorlaştırmakta ve optimizasyon yöntemlerinin daha iyi test edilmesini sağlamaktadır. Şekil 3'te gösterildiği gibi orta boy

problemlerde dahi arama uzayı 1 milyara yaklaşmaktadır. Fakat inşaat projelerinde paralel olarak yürütülebilecek aktivite sayısı örnek problemlerdeki kadar çok fazla değildir. Bu nedenle hipotetik projelerin çözümüne ek olarak 2 katlı bir inşaat projesi üzerinde kaynak dengeleme problemi çözülerek algoritmanın başarısı denenmiştir. Örnek problemin şebekesi Şekil 11'de, aktivitelerin tanımı ise Çizelge 11'de gösterilmiştir. Örnek problem 4'ün arama uzayı F, K-L-J, AG-AF-AH ve W-AB-AD-T-V-Z hatlarının kombinasyonundan oluşmaktadır. Bu hatların arama uzayı örnek projelerde gösterildiği gibi gerçekleştirilmektedir. Hatların erteleme sayıları Çizelge 10'da verilmiştir.

Çizelge 10 - Örnek Proje 4 arama uzayı hesaplaması

Hat	Erteleme Sayısı
K-L-J	84
F	7
AF-AG-AH	816
T-V-Z-W-AB-AD	10307
Arama Uzayı	4.945.381.056

Çizelge 11 - Örnek problem 4 (2 katlı inşaat)

İş Kalemi	Gösterim	Kaynak	Serbest Bolluk	Toplam Bolluk
Tesviye	A	2	0	0
Yerleşim	B	5	0	0
Temel Kazısı	C	3	0	0
Grobeton	D	3	0	0
Su Yalıtımı ve Yalıtım Betonu	E	7	0	0
Mekanik Ve Tesisat Bağlama	F	4	6	6
Kalıp-Donatı (Temel)	G	10	0	0
Beton Dökümü (Temel)	H	6	0	0
Kür (Temel)	I	1	0	0
Yağmur Drenajı	J	5	0	6
Atık Su Hattı Bağlama	K	5	0	6
Geri Dolgu	L	4	6	6
Kalıp-Donatı (Zemin)	M	10	0	0
Beton Dökümü (Zemin)	N	6	0	0
Kür (Zemin)	O	1	0	0
Kalıp-Donatı (Kat 1)	P	10	0	0
Beton Dökümü (Kat 1)	R	6	0	0
Kür (Kat 1)	S	1	0	0
Duvar (Zemin)	T	8	0	5
Duvar (Kat 1)	U	8	0	0
Tesisat (Zemin)	V	5	0	7
Tesisat (Kat 1)	Y	5	0	0

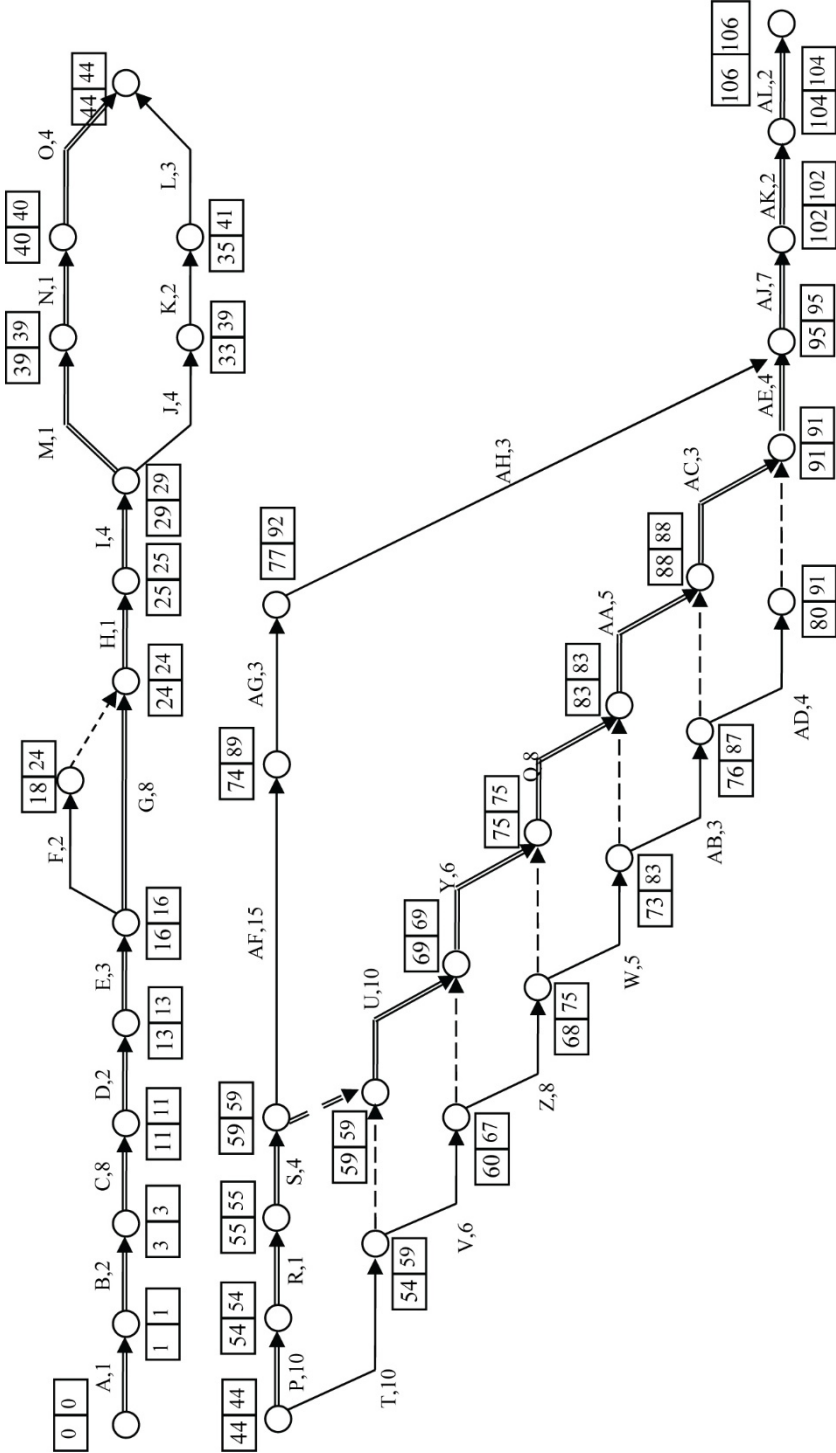
Çizelge 11 - Örnek problem 4 (2 katlı inşaat) (devam)

İş Kalemi	Gösterim	Kaynak	Serbest Bolluk	Toplam Bolluk
Sıva (Zemin)	Z	7	0	7
Sıva (Kat 1)	Q	7	0	0
Doğrama (Zemin)	W	4	0	10
Doğrama (Kat 1)	AA	4	0	0
Parke-Fayans (Zemin)	AB	5	0	11
Parke-Fayans (Kat 1)	AC	5	0	0
Boya (Zemin)	AD	4	11	11
Boya (Kat 1)	AE	4	0	0
Çatı	AF	9	0	15
Çatı İzolasyonu	AG	4	0	15
Yağmur Olukları	AH	3	15	15
Çevre Düzenleme	AJ	5	0	0
Kontrol ve Testler	AK	2	0	0
Temizlik	AL	5	0	0

Şebekeyi oluşturan kritik olmayan 4 hattın erteleme sayılarının çarpımı 5 milyara yakın bir arama uzayı büyüklüğünü ortaya çıkarmaktadır (Çizelge 10). Fakat Şekil 11'de sunulan şebeke diyagramı incelendiğinde F ve K-L-J hatlarının diğer hatlarla hiçbir şekilde etkileşim halinde olmadıkları görülmektedir. Bu nedenle belirtilen 2 hattın tek başına en iyi çözümü verdiği erteleme seçeneklerinin hesaplanıp diğer hatlar olan AF-AG-AH ile T-V-Z-W-AB-AD hatlarının erteleme seçenek sayılarına eklenmesi ile arama uzayının büyüklüğü hesaplanacaktır. Bu şekilde örnek problem 4'ün arama uzayı $816 * 10307 + 7 + 84 = 8.410.603$ erteleme seçeneğinden oluşmaktadır.

Belirtilen arama uzayı tek bilgisayarda çözdürülebileceği gibi paralel hesaplama ile çok sayıda küçük parçaya ayrılıp eş zamanlı biçimde farklı bilgisayarlarda da çözdürülebilir. Çözüm süresi tek bilgisayarda 7,1 saat olarak ölçülmüştür. Parallelleşme AF-AG-AH ve T-V-Z-W-AB-AD hatları üzerinden yapılabilmektedir. Üç aktiviteden oluştuğu için paralelleştirmesi daha kolay olan AF-AG-AH hattı tercih edilmiştir. Çizelge 10'da belirtildiği üzere bu hattın 816 erteleme seçeneği bulunmaktadır. Erteleme seçeneği 16'ya bölünebilmektedir böylece bu hat üzerinden her bilgisayara hattın 51 erteleme seçeneği çözdürülmek üzere 16 bilgisayara arama uzayı eşit biçimde taratılabilir. Arama uzayını 16 eşit parçaya ayırarak tarayan visual basic kodu Şekil 12'de gösterilmiştir. Optimizasyon öncesi ve sonrası aktivitelerin başlama ve bitiş zamanları çizelge 12'de sunulmuştur.

Şekil 12'de sunulan örnek kodlamada 1 numaralı bilgisayara dağıtım yapılmaktadır. Kodlamadaki 1 rakamı bilgisayar sayısı ile değiştirilerek tüm bilgisayarlar için genelleştirme yapılabilir. Paralel hesaplama ile bir bilgisayara $51 * 10307 = 525.657$ adet şebeke çözümü yaptırılmış ve örnek problem-4, 16 bilgisayar kullanılarak 27 dakikada çözülmüştür. 2 katlı bir binanın iş programı olan 36 aktiviteli projenin garanti edilmiş en iyi çözümü paralel programlama ile makul sürede elde edilebilmiştir. Örnek problem 4'ün optimizasyon öncesi ve sonrasındaki kaynak dağılımı Şekil 13'te sunulmuştur.



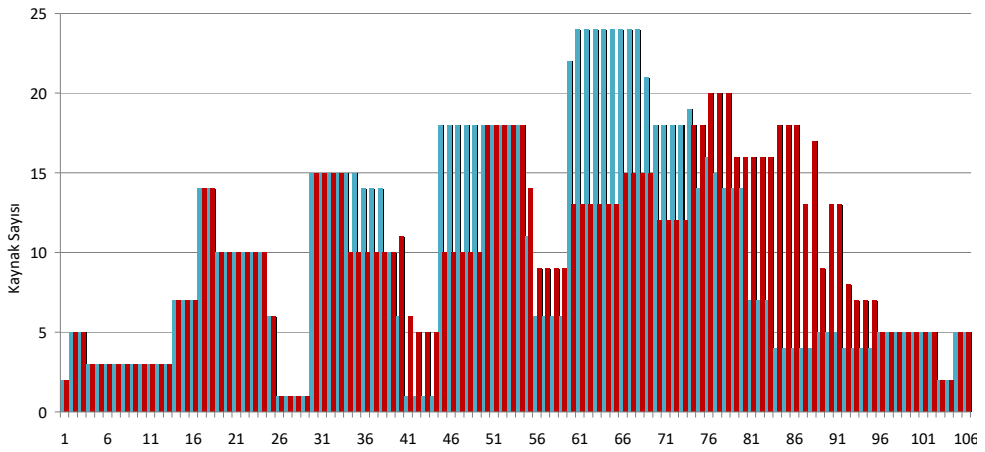
Şekil 11 - Örnek problem 4 şebeke diyagramı

```

For tGecikme = 0 To 5
For vGecikme = tGecikme To 7
For zGecikme = vGecikme To 7
For wGecikme = zGecikme To 10
For abGecikme = wGecikme To 11
For adGecikme = abGecikme To 11
paralelSayac = 0
For afGecikme = 0 To 15
For agGecikme = afGecikme To 15
For ahGecikme = agGecikme To 15
paralelSayac = paralelSayac + 1
bilgisayarNo = (paralelSayac - paralelSayac Mod 51) / 51 + 1
If bilgisayarNo = 1 Then

```

Şekil 12 - Örnek problem 4'ü alt parçalara ayıran kodlama



Şekil 13 - Problem 4 başlangıç ve dengelenmiş kaynak dağılımı

Çizelge 12 - Problem 3 için başlangıç ve optimum çözümlerin karşılaştırılması

	Serbest bolluk	Toplam bolluk	Başlangıç durumunda aktivitelerin başlama-tamamlanma zamanları			Optimizasyon sonrası aktivitelerin başlama-tamamlanma zamanları		
			Erteleme	Başlama	Bitiş	Erteleme	Başlama	Bitiş
A	0	0	0	0	1	0	0	1
B	0	0	0	1	3	0	1	3
C	0	0	0	3	11	0	3	11
D	0	0	0	11	13	0	11	13
E	0	0	0	13	16	0	13	16
F	6	6	0	16	18	0	16	18

Çizelge 12 - Problem 3 için başlangıç ve optimum çözümlerin karşılaştırılması (devam)

	Serbest bolluk	Toplam bolluk	Başlangıç durumunda aktivitelerin başlama-tamamlanma zamanları			Optimizasyon sonrası aktivitelerin başlama-tamamlanma zamanları		
			Erteleme	Başlama	Bitiş	Erteleme	Başlama	Bitiş
G	0	0	0	16	24	0	16	24
H	0	0	0	24	25	0	24	25
I	0	0	0	25	29	0	25	29
J	0	6	0	29	33	0	29	33
K	0	6	0	33	35	6	39	41
L	6	6	0	35	38	6	41	44
M	0	0	0	29	39	0	29	39
N	0	0	0	39	40	0	39	40
O	0	0	0	40	44	0	40	44
P	0	0	0	44	54	0	44	54
R	0	0	0	54	55	0	54	55
S	0	0	0	55	59	0	55	59
T	0	5	0	44	54	5	49	59
U	0	0	0	59	69	0	59	69
V	0	7	0	54	60	5	59	65
Y	0	0	0	69	75	0	69	75
Z	0	7	0	60	68	5	65	73
Q	0	0	0	75	83	0	75	83
W	0	10	0	68	73	5	73	78
AA	0	0	0	83	88	0	83	88
AB	0	11	0	73	76	10	83	86
AC	0	0	0	88	91	0	88	91
AD	11	11	0	76	80	11	87	91
AE	0	0	0	91	95	0	91	95
AF	0	15	0	59	74	14	73	88
AG	0	15	0	74	77	15	89	92
AH	15	15	0	77	80	15	92	95
AJ	0	0	0	95	102	0	95	102
AK	0	0	0	102	104	0	102	104
AL	0	0	0	104	106	0	104	106

Proje süresince ortaya çıkan günlük en yüksek kaynak kullanım miktarı 24'ten 20'ye düşürülmüştür. Ayrıca kaynak dağılımında proje süresince daha az dalgalanmanın olması sağlanmıştır. Proje süresinin 106 gün olduğu göz önüne alınırsa 27 dakika olan hesaplama süresinin makul olduğu değerlendirilebilir.

4. BULGULAR

Bu çalışmada inşaat sektörü için önemli bir araştırma alanı olan kaynak dengeleme probleminin tüm arama uzayını tarayarak kesin çözümünü elde eden yeni bir yöntem sunulmuştur. Bu yöntem aktiviteler arasındaki ardıl öncel ilişkisini ihlal etmeden bolluğu olan aktiviteler üzerinden elde edilebilecek tüm erteleme seçeneklerinin sistematik biçimde deneyerek en iyi çözümü elde etmektedir. Modern sezgisel yöntemlerin rastgelelik içeren arama biçimi nedeniyle aynı çözümün birkaç tekrardan sonra tekrar denenmesi mümkündür. Bu durum özellikle bireylerin mevcut en iyi çözüme yakınsamaları nedeniyle daha sık rastlanır hale gelmektedir. Tabu arama yöntemi geçmişte denenmiş ve iyi sonuç vermeyen çözümleri reddetmektedir. Benzeri yöntemler tekrar çözümlerin önüne geçebilir, fakat özellikle büyük problemlerde geçmiş denemelerin tutulması önemli bellek kullanımına ve karşılaştırma yapılmasına yol açar. Bu nedenle modern sezgisel yöntemlerde tekrar çözümlerin engellenmesi çok yüksek hesaplama maliyeti getirir. Küçük problemlerde ise rastgele arama nedeniyle tam sonucun elde edilmesi tüm arama uzayının taranmasından daha fazla hesap yükü gerektirmektedir. Büyük problemlerde ise modern sezgisel yöntemlere göre daha yüksek verimde paralel hesaplama yapılabilen arama uzayının taranması yöntemi bu avantajı ile daha fazla hesap yükü gerektirme sorununu gidermektedir.

Önerilen yöntem kaynak dengeleme probleminin çözümünde kullanılmakta olan modern sezgisel ve matematiksel yöntemlere göre daha kolay uygulanabilmektedir. Yöntemin uygulanabilmesi için optimizasyon üzerine kuramsal bilgiye ihtiyaç duyulmamaktadır. Sektörde çalışan bir inşaat mühendisi sadece hesapçetveli üzerinde makro yazarak kaynak dengeleme probleminin optimum çözümünü elde edebilir. Bu durumu sağlayabilmek için analizler hesaplama süresi C++ kodlamasına göre daha uzun olmasına rağmen hesapçetveli üzerinde yürütülmüş ve örnek kodlama makale içinde verilmiştir. Bu sayede kurumsal firmalara ek olarak küçük ve orta büyüklükteki tüm inşaat firmaları kaynak dengeleme yapıp önemli ölçüde maliyetlerini düşürebilmeleri amaçlanmıştır.

Geliştirilen yöntem 3'ü literatürden elde edilen 1'i ise 2 katlı bir inşaat işine ait olan 4 problemin kaynak dengeleme probleminin çözümünde uygulandı. Kaynak dengeleme problemi NP-Zor sınıfı bir problem olduğu için aktivite sayısının artması ile arama uzayı sayısı çok hızlı biçimde artmaktadır. Bu nedenle büyük problemlerde tüm arama uzayının tek işlemci ile taranması makul sürelerde olmadığı için paralel programlama ile problem eşit hesap yükü içeren küçük parçalara ayrılıp birden fazla bilgisayarda eş zamanlı çözdürülerek hesaplama süresi kısaltılmıştır.

Geliştirilen yöntem paralel programlamaya oldukça yatkındır. Problem kolaylıkla çok sayıda birbirine eşit alt parçalara ayrılabilir. Ayrıca hesaplama sırasında alt parçalar arasında veri alışverişine ihtiyaç duyulmamakta ve hesaplama süreci kesintisiz biçimde yürütülmektedir. Bu sayede paralelleştirme yüksek verimde gerçekleştirilerek hesaplama süresinde önemli tasarruflar sağlanmaktadır. Birinci problemde tek hat üzerinde paralelleştirme yapılmış, ikinci problemde birbirine paralel iki hat birleştirilerek oluşan

erteleme seçenekleri üzerinden paralelleştirme yapılmış, üçüncü örnek problemde ise seri hat üzerinden paralelleştirme yapılmıştır. Olabilecek tüm paralelleştirme senaryoları incelenerek makaleyi okuyan tüm araştırmacıların geliştirilen yöntemi uygulayabilmeleri amaçlanmıştır.

KDP'nin çözülmesi ile elde edilebilecek fayda ve tasarruflar Şekil 6'da sunulan örnek problem için başlangıç durumunda ve optimizasyon sonrası elde edilen kaynak kullanım eğrileri üzerinden örnekleyerek açıklanmıştır. Mavi ve kırmızı çubuklar sırası ile optimizasyon öncesi ve sonrası kaynak kullanımını göstermektedir. Kaynak dengelemesi yapılmıyorsa 50 ve üzeri personelin çalışması gereken sürelerde kısa süreli çalışacak işçiler istihdam edilecek ve bu işçilere iş güvenliği kuralları ve şantiye süreci öğretilip işe uyumları sağlanacak bir süre sonra bu işçilerin 20'si işten çıkarılacaktır. Bu durum önemli bir maliyet kaynağı ve emek israfıdır. Ayrıca 10 ve 11. zamanda kaynak dengelenmemiş durumda diğer günlere göre daha az personel çalışmaktadır. Bu süre zarfında personelin bir kısmına çalışmadığı halde ödeme yapılacak veya ücretli/ücretsiz izne çıkartılacaktır. Ücretsiz izne çıkartılması ise personelin iş verimini önemli ölçüde düşürecektir. Şantiyenin yerleşim merkezinden uzak bir konumda bulunduğu varsayıldığında kaynak dengelemesi yapılmadan önce 50 kişiye hizmet verebilecek yatakhane, yemekhane, lavabo-duş yerleri inşa etmek gerekirken kaynak dengeleme sonrası 30 kişilik tesis yeterli olacaktır.

Kaynak dengeleme problemi güncel çalışmalarda ağırlıklı olarak modern sezgisel yöntemlerle çözülmektedir. Belirtilen yöntemlerin en önemli sorunu yerel minima'ya yakınsaması ve popülasyonun baskın biçimde yerel minima'ya yakın bireylerden oluşmasıdır. Bu duruma gelen bir popülasyondan elde edilen yeni bireylerin daha önce denenmiş çözümlerle birebir aynı olma olasılığı çok yüksektir. Bu duruma örnek olarak Bettemir [48] tarafından çözülen 11 aktiviteye kadar olan KDP'de garantili biçimde kesin sonuca ulaşmak için yapılan deneme sayısı arama uzayından daha fazla olmaktadır. Bu nedenle bu çalışmada önerilen tüm arama uzayının taranması yöntemi küçük problemler için modern sezgisel yöntemlerden daha az hesap yükü ile tam sonuca ulaşabilmektedir. Önerilen yöntemin dezavantajı arama uzayının aktivite sayısı ile üstsel olarak artması ve hesap süresinin çok uzamasıdır. Bu çalışmada arama uzayının paralel programlama ile alt parçalara ayrılarak eş zamanlı olarak birden fazla bilgisayarda aynı anda taranması hesap süresini önemli ölçüde kısaltarak belirtilen olumsuzluk azaltılmıştır. Daha büyük problemlerin çözülebilmesi için 4000 ve üzeri sayıda çekirdeğe sahip grafik işlemcileri kullanılarak problemin daha küçük alt parçalara ayrılabilmesini sağlayarak hesap süresini kısaltmak mümkün olacaktır.

Bazı iş kalemlerinin taşeronlara yaptırılması durumunda taşeronların çalışma takvimine bağlı olarak aktivitelerin bolluk süreleri serbestçe kullanılamayabilir. Taşeronun yürütmekte olduğu iş kalemlerinin tamamını belirli bir süre zarfında bitirmek zorunda olması durumunda anılan zaman zarfının içinde işlerin yürütülebilmesi için ilk yapılan işlerin geciktirilmesi, son yapılan işlerin ise belirli bir süreyi aşmadan geciktirilmesi gerekebilir. Problem 4'te Duvar işini yürüten taşeron T ve U aktivitelerini gerçekleştirecektir. Duvar işinin tamamı 44 ile 69. günler arasında yürütülebilmektedir. Taşeronun şantiyede en fazla 22 gün çalışma esnekliğinin bulunduğu varsayıldığında zemin kat duvar işi olan T aktivitesi en az 3 gün geciktirilerek 47. gün sonunda başlatılabilir. Bu sayede duvar işinin tamamı 22 gün içinde tamamlanıp bu sürenin sonunda taşeron başka işlerini yürütebilir. Belirtilen kısıt, önerilen yöntemde T aktivitesinin bolluğu 0'dan 5'e saydırılmak yerine 3'ten 5'e saydırılarak en iyi çözümün aranması ile belirtilen kısıt sağlanır. Bu durumda arama uzayı önemli ölçüde

azalacağı için hesaplama süresi kayda değer ölçüde kısılacaktır. Çok daha büyük inşaat projelerinde hesaplama süresini kısaltmak için planlama detayı düşürülerek ölçek olarak gün yerine hafta kullanılabilir. Bu sayede 14 gün olan bolluk süresi 2 hafta olacaktır. Bu şekilde arama uzayı büyük oranda küçültülebilir.

Örnek projelerin hesaplama süreleri sırası ile 20 dakika, 3 saat, 7,7 saat ve 27 dakika olarak gerçekleşmiştir. Hesaplamalar hesapcetveli üzerinde çalışan bir makro üzerinden yapıldığı için çalıştırılabilir dosyaya göre daha uzun sürmüştür. Günümüzde ekran kartlarının işlemcileri yüzlerce işlemciden oluşmaktadır. Ekran kartı üreticilerinin geliştirdikleri yazılımlarla kolaylıkla ekran kartlarına komut gönderilebilmekte ve hesaplamalar ana işlemci yerine ekran kartının işlemcileri üzerinden yürütülebilmektedir. İleri çalışma olarak paralel hesaplamaların ekran kartı üzerinden yürütülerek problemin binin üzerinde parçaya ayrılarak hesaplama süresi daha da kısaltılabilir.

Kaynak dengelemesi yapılırken iş güvenliği ve şantiye koşulları da mutlaka dikkate alınmalıdır. Örneğin kule vinç kullanımını gerektiren işler yürütülürken kule vincin yakınında başka işlerin yürütülmesini engelleyecek kısıtlar konulmalıdır. Aksi halde iş güvenliği tehlikeye girebilecektir [49].

5. SONUÇ

Arama uzayının büyüklüğü aktivite sayısına ek olarak aktivitelerin bolluğu ve şebekenin yapısı da arama uzayı büyüklüğünü etkilemektedir. Bu nedenle büyük problemler için tek işlemci ile gerçekleştirilecek çözümler makul sürenin çok üzerine çıkacaktır. Bu çalışmada literatürden derlenen 3 problem ile 2 katlı betonarme bir yapıya ait iş programının kaynak dengeleme probleminin optimum çözümü arama uzayları paralel hesaplama ile taranarak elde edilmiştir. Bu sayede 36 aktiviteye kadar olan projelerin kesin çözümleri elde edilerek yöntemin başarısı ortaya konmuştur. Arama uzayının tamamının taranması KDP'nin kesin çözümünün elde edilmesini garantilemektedir. Ayrıca yüksek verimli biçimde paralel programlamaya uygun olduğu için paralel hesaplama ile çözüm süresi önemli ölçüde kısaltılmaktadır.

Bu çalışmanın literatüre özgün katkısı kaynak dengeleme problemini arama uzayını tarayarak kesin çözümünü bulan bir yöntemin geliştirilmesi ve hesaplamaları eşit parçalara ayırarak paralel programlama yapılmasıdır. İnşaat sektörüne katkısı ise oldukça kolay biçimde uygulanabilecek bir optimizasyon yönteminin geliştirilmesidir. Bu sayede inşaat süresi boyunca boşta kalan personel ve iş makinesi süresi en aza indirilmektedir. Ayrıca inşaatın yürütülmesi için ihtiyaç duyulan en üst kaynak miktarı da azaltılarak maliyetlerin azaltılması sağlanmaktadır. Böylece tüm inşaat firmaları kaynak dengelemesi yaparak inşaatın yürütülmesi sırasında boşta kalacak personel ve iş makinesi sürelerini azaltarak kaynak maliyetlerini önemli ölçüde düşürebileceklerdir. Bu sayede rekabet avantajı kazanabileceklerdir.

Kaynaklar

- [1] Ahbab, C., Daneshvar, S., ve Celik, T. (2019). Cost and Time Management Efficiency Assessment for Large Road Projects Using Data Envelopment Analysis. *Teknik Dergi*, 30(2), 8937-8959.
- [2] Kolisch, R., ve Padman, R. (2001). "An integrated survey of deterministic project scheduling." *Omega*, 29(3), 249-272.
- [3] Li, H., ve Demeulemeester, E. (2016). A genetic algorithm for the robust resource leveling problem. *Journal of Scheduling*, 19(1), 43-60.
- [4] Tarasov, I., Haït, A., ve Battaïa, O. (2020). A Generalized MILP Formulation for the Period-Aggregated Resource Leveling Problem with Variable Job Duration. *Algorithms*, 13(1), 6.
- [5] Li, H., ve Dong, X. (2018). Multi-mode resource leveling in projects with mode-dependent generalized precedence relations. *Expert Systems with Applications*, 97, 193-204.
- [6] Abadi N.S., Bagheri N. ve Assadi M., (2018). Multiobjective model for solving resource-leveling problem with discounted cash flows. *International Transactions in Operational Research*, 25(6), 2009-2030.
- [7] Doulabi Hossein Hashemi, S., Seifi, A., ve Shariat, S. Y. (2011). "Efficient hybrid genetic algorithm for resource leveling via activity splitting". *Journal of Construction Engineering and Management*, 137(2), 137-146.
- [8] Harris, R. B. (1990). Packing method for resource leveling (PACK). *Journal of Construction Engineering and Management*, 116(2), 331-350.
- [9] Hiyassat, M. A. S. (2001). Applying modified minimum moment method to multiple resource leveling. *Journal of Construction Engineering and Management*, 127(3), 192-198.
- [10] Rieck, J., Zimmermann, J., ve Gather, T. (2012). "Mixed-integer linear programming for resource leveling problems." *European Journal of Operational Research*, 221(1), 27-37.
- [11] Neumann, K., Schwindt, C., ve Zimmermann, J. (2012). "Project scheduling with time windows and scarce resources: temporal and resource-constrained project scheduling with regular and nonregular objective functions." Springer Science & Business Media.
- [12] Hegazy, T. (1999). Optimization of resource allocation and leveling using genetic algorithms. *Journal of construction engineering and management*, 125(3), 167-175.
- [13] Son, J., ve Skibniewski, M. J. (1999). "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach." *J. Constr. Eng. Manage.*, 125(1), 23-31.
- [14] Leu, S. S., Yang, C. H., ve Huang, J. C. (2000). "Resource leveling in construction by genetic algorithm-based optimization and its decision support system application." *Autom. Constr.*, 10(1), 27-41.

- [15] Zheng, D. X., Ng, S. T., ve Kumaraswamy, M. M. (2003). "GA-based multiobjective technique for multi-resource leveling." *Bridges*, 10(40671), 29.
- [16] El-Rayes, K., ve Jun, D. H. (2009). "Optimizing resource leveling in construction projects." *J. Constr. Eng. Manage.*, 135(11), 1172-1180.
- [17] Christodoulou, S. E., Ellinas, G., ve Michaelidou-Kamenou, A. (2009). "Minimum moment method for resource leveling using entropy maximization." *J. Constr. Eng. Manage.*, 136(5), 518-527.
- [18] Ponz-Tienda, J. L., Yepes, V., Pellicer, E., ve Moreno-Flores, J. (2013). "The resource leveling problem with multiple resources using an adaptive genetic algorithm." *Automation in Construction*, 29, 161-172.
- [19] Li, H., Xiong, L., Liu, Y., ve Li, H. (2017). "An effective genetic algorithm for the resource levelling problem with generalised precedence relations." *International Journal of Production Research*, 1-22.
- [20] Qi, J. X., Wang, Q., and Guo, X. Z. (2007). "Improved particle swarm optimization for resource leveling problem." In *IEEE International Conference on Machine Learning and Cybernetics*, 2, 896-901.
- [21] Li, Z., Wuliang, P., and Zhongliang, Z. (2010). "An ant colony system for solving resource leveling problem." In *IEEE Int. Conf. Intell. Comp. Tech. and Autom. (ICICTA)*, 1, 489-492.
- [22] Geng, J. Q., Weng, L. P., and Liu, S. H. (2011). "An improved ant colony optimization algorithm for nonlinear resource-leveling problems." *Comput. Math. Appl.*, 61(8), 2300-2305.
- [23] Tran, H. H., and Hoang, N. D. (2014). "A novel resource-leveling approach for construction project based on differential evolution." *J. Constr. Eng.*, <http://dx.doi.org/10.1155/2014/648938>.
- [24] Xu, X., Hao, J., ve Zheng, Y. (2020). Multi-objective artificial bee colony algorithm for multi-stage resource leveling problem in sharing logistics network. *Computers & Industrial Engineering*, 142, 106338.
- [25] Prayogo, D., ve Kusuma, C.T. (2019). Optimization of resource leveling problem under multiple objective criteria using a symbiotic organisms search. *Civil Engineering Dimension*, 21(1), 43-49.
- [26] Prayogo, D., Cheng, M. Y., Wong, F. T., Tjandra, D., ve Tran, D. H. (2018). Optimization model for construction project resource leveling using a novel modified symbiotic organisms search. *Asian Journal of Civil Engineering*, 19(5), 625-638.
- [27] Erzurum, T. ve Bettemir, Ö.H. "Kaynak Dengeleme Problemlerinin Arama Uzayının Belirlenmesi Determination of Search Domain of Resource Leveling Problem", Uluslararası Katılımlı 7. İnşaat Yönetimi Kongresi, pp. 437-453, 6-7 Ekim 2017 Samsun Türkiye.
- [28] Toğan, V., ve Eirgash, M. A. (2018). Time-Cost Trade-Off Optimization with a New Initial Population Approach. *Teknik Dergi*, 30(6).

- [29] Karaa, F. A., ve Nasr, A. Y. (1986). Resource management in construction. *Journal of construction engineering and management*, 112(3), 346-357.
- [30] Takamoto, M., Yamada, N., Kobayashi, Y., Nonaka, H., and Okoshi, S. (1995). "Zero-one quadratic programming algorithm for resource leveling of manufacturing process schedules." *Systems and Computers in Japan*, 26(10), 68-76.
- [31] Easa, S. M. (1989). "Resource leveling in construction by optimization." *J. Constr. Eng. Manage.*, 115(2), 302-316.
- [32] Hariga, M., and El-Sayegh, S. M. (2010). Cost optimization model for the multiresource leveling problem with allowed activity splitting. *Journal of Construction Engineering and Management*, 137(1), 56-64.
- [33] Gather, T., Zimmermann, J., and Bartels, J. H. (2011). "Exact methods for the resource levelling problem." *Journal of Scheduling*, 14(6), 557-569.
- [34] Rieck, J., ve Zimmermann, J. (2015). "Exact methods for resource leveling problems." In *Handbook on Project Management and Scheduling Vol. 1* (pp. 361-387). Springer International Publishing.
- [35] Mattila, K. G. ve Abraham, D. M. (1998). Resource leveling of linear schedules using integer linear programming. *Journal of Construction Engineering and Management*, 124(3), 232-244.
- [36] Erzurum T., ve Bettemir Ö.H. (2018). Optimum or Near-Optimum Resolution of Resource Leveling Problems with Spreadsheet Application. 5th International Project and Construction Management Conference (IPCMC 2018), pp. 1285-1299.
- [37] Bettemir, Ö.H., Erzurum T. (2019), "Comparison of resource distribution metrics on multi-resource projects", *Journal of Construction Engineering, Management & Innovation*, 2(2), pp. 93-102.
- [38] Erzurum T. (2019), "Kaynak Dengeleme Probleminin Optimum veya Yakın Optimum Çözülmesi", İnönü Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi.
- [39] El-Rayes, K. ve Kandil, A. (2004). Distributed Computing for the Optimization of Large-Scale Construction Projects.
- [40] Kandil, A. ve El-Rayes, K. (2005). Parallel computing framework for optimizing construction planning in large-scale projects. *Journal of computing in civil engineering*, 19(3), 304-312.
- [41] Kandil, A. ve 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.
- [42] Kandil, A., El-Rayes, K. ve El-Anwar, O. (2010). Optimization research: Enhancing the robustness of large-scale multiobjective optimization in construction. *Journal of Construction Engineering and Management*, 136(1), 17-25.
- [43] Sayar, A., ve Ergün, U. (2014). Fonksiyonel Programlama Dilleri ile Paralel Programlama. *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 3(2), 1-17.

- [44] Gordon, M. I., Thies, W., ve Amarasinghe, S. (2006). Exploiting coarse-grained task, data, and pipeline parallelism in stream programs. *ACM SIGARCH Computer Architecture News*, 34(5), 151-162.
- [45] Sönmez, M., Dikmen, S. Ü., ve Akbıyıklı, R. (2020). Time-cost relationships for superstructure projects in Turkey. *Teknik Dergi*, 31(2).
- [46] Stevens, J. D. (1990). *Techniques for Construction Network Scheduling*, McGraw-Hill, New York.
- [47] Mutlu, M. Ç. (2010). A branch and bound algorithm for resource leveling problem, Yüksek Lisans Tezi, Fen Bil. Ens., ODTÜ.
- [48] Bettemir, Ö. H. (2009). Optimization of time-cost-resource trade-off problems in project scheduling using meta-heuristic algorithms, Doktora Tezi, Fen bil. Ens., ODTÜ.
- [49] Oral, M., Bazaati, S., Aydinli, S., ve Oral E., (2018). Construction Site Layout Planning: Application of Multi-Objective Particle Swarm Optimization. *Teknik Dergi*, 29(6), 8691-8713.

EK-1 Seri bağı 3 aktiviteli hattın kombinasyon sayısı

$$Seri_3 = \frac{(n+1)(n+2)}{2} + \frac{n(n+1)}{2} + \frac{(n-1)n}{2} + \frac{(n-2)(n-1)}{2} + \dots + \frac{2*3}{2} + \frac{1*2}{2}$$

$$Seri_3 = \frac{1}{2} \left[(n+1)(n+2) + \underbrace{n(n+1) + (n-1)n + (n-2)(n-1) + \dots + 2*3 + 1*2}_{n \text{ terim}} \right]$$

$$Seri_3 = \frac{1}{2} \left[(n+1)(n+2) + \sum_{i=0}^n i(i+1) \right]$$

$$Seri_3 = \frac{1}{2} \left[(n+1)(n+2) + \sum_{i=0}^n (i^2 + i) \right]$$

$$\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

$$Seri_3 = \frac{1}{2} \left[(n+1)(n+2) + \frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \right]$$

$$Seri_3 = \frac{1}{2} \left[(n+1)(n+2) + \frac{n(n+1)}{6} [(2n+1)+3] \right]$$

$$Seri_3 = \frac{1}{2} \left[(n+1)(n+2) + \frac{n(n+1)}{6} [2n+4] \right]$$

$$Seri_3 = \frac{1}{2} \left[(n+1)(n+2) + \frac{n(n+1)(n+2)}{3} \right]$$

$$Seri_3 = \frac{1}{6} [3(n+1)(n+2) + n(n+1)(n+2)]$$

$$Seri_3 = \frac{1}{6} [(n+1)(n+2)(n+3)]$$

EK-2 4 aktiviteli seri hat.

$$Seri_4 = \frac{(n+1)(n+2)(n+3)}{6} + \frac{n(n+1)(n+2)}{6} + \frac{(n-1)n(n+1)}{6} + \frac{(n-2)(n-1)n}{6} + \dots + \frac{2*3*4}{6} + \frac{1*2*3}{6}$$

$$Seri_4 = \frac{1}{6} \left[(n+1)(n+2)(n+3) + \underbrace{n(n+1)(n+2) + (n-1)n(n+1) + (n-2)(n-1)n + \dots + 2*3*4 + 1*2*3}_{n \text{ terim}} \right]$$

$$Seri_4 = \frac{1}{6} \left[(n+1)(n+2)(n+3) + \sum_{i=1}^n i(i+1)(i+2) \right]$$

$$Seri_4 = \frac{1}{6} \left[(n+1)(n+2)(n+3) + \sum_{i=1}^n (i^3 + 3i^2 + 2i) \right]$$

$$\sum_{i=1}^n i^3 = \frac{n^2(n+1)^2}{4}, \quad \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}, \quad \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

$$Seri_4 = \frac{1}{6} \left[(n+1)(n+2)(n+3) + \frac{n^2(n+1)^2}{4} + \frac{n(n+1)(2n+1)}{2} + n(n+1) \right]$$

$$Seri_4 = \frac{(n+1)(n+2)(n+3)}{6} + \frac{n(n+1)}{6} \left[\frac{n(n+1)}{4} + \frac{(2n+1)}{2} + 1 \right]$$

$$Seri_4 = \frac{(n+1)(n+2)(n+3)}{6} + \frac{n(n+1)}{24} [n^2 + n + 4n + 2 + 4]$$

$$Seri_4 = \frac{(n+1)(n+2)(n+3)}{6} + \frac{n(n+1)}{24} [n^2 + 5n + 6]$$

$$Seri_4 = \frac{(n+1)(n+2)(n+3)}{6} + \frac{n(n+1)(n+2)(n+3)}{24}$$

$$Seri_4 = \frac{(n+1)(n+2)(n+3)}{24} [4 + n]$$

$$Seri_4 = \frac{(n+1)(n+2)(n+3)(n+4)}{24}$$

