Pamukkale Univ Muh Bilim Derg, 24(5), 898-905, 2018

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Karesel atama problemleri için tavlama benzetimi paralelleştirme yöntemlerinin karşılaştırılması

Comparison of simulated annealing parallelization methods for quadratic assignment problems

Selahattin AKKAŞ[1*], *Kadir KAVAKLIOĞLU*[2]

[1]Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Pamukkale Üniversitesi, Denizli, Türkiye.
sakkas@pau.edu.tr

[2]Makine Mühendisliği Bölümü, Mühendislik Fakültesi, Pamukkale Üniversitesi, Denizli, Türkiye.
kadir.kavaklioglu@pau.edu.tr

Geliş Tarihi/Received: 09.06.2017, Kabul Tarihi/Accepted: 14.12.2017
doi: 10.5505/pajes.2017.46794
* Yazışılan yazar/Corresponding author
Araştırma Makalesi/Research Article

Öz

Karesel atama problemi (KAP), NP-hard sınıfındaki en zor kombinatoryal optimizasyon problemlerinden birisidir. Problemin zorluğundan dolayı birçok araştırmacı bu tip atama problemini çalışılmaktadır. Bu çalışmada tavlama benzetimi yöntemi MATLAB platformunda paralelleştirilerek iyi bilinen bir KAP Kütüphanesi olan QAPLIB'den alınan 36 örnek problemi çözmek için kullanılmıştır. Değişik paralelleştirme yöntemlerinin performansları kullanılan problemler için karşılaştırılmıştır. Sonuç olarak seri tavlama benzetimi yöntemiyle karşılaştırıldığında, paralel yöntemlerin uygun parametreler kullanıldığında daha hızlı sonuç verdiği görülmüştür.

Anahtar kelimeler: Karesel atama problemi, Paralel programlama, Tavlama benzetimi, Optimizasyon

Abstract

Quadratic assignment problem (QAP) is one of the most difficult combinatorial optimization problems in the NP-hard class. Due to the difficulty of the problem, many researchers have been studying this type of assignment problem. In this work, simulated annealing method is parallelized on MATLAB platform and is used to solve 36 problems from QAPLIB which is a well-known QAP library. The performance of different parallelization methods is compared for the problems used. As a result, when compared with the serial simulated annealing method, it is seen that the parallel methods give faster results when the appropriate parameters are used.

Keywords: Quadratic assignment problem, Parallel programming, Simulated annealing, Optimization

1 Giriş

Karesel atama problemi ilk Koopmans ve Beckman tarafından 1957 senesinde tanıtılmıştır [1]. Ekonomik faaliyetlerin modellenmesi için önerilen KAP, en fazla tesis yerleşimi problemlerinde kullanılmaktadır. Bunun dışında kampüs içerisindeki binaların yerleşimi, hastane birimlerinin yerleşimi, daktilo tuşlarının yerleşimi, elektronik bileşenlerin yerleşimi gibi problemlerin çözümünde KAP'dan faydalanılmıştır.

Problemde n tesis n konuma atanmaya çalışılmaktadır. Tesis i ve j arasındaki akış f_{ij}, k ve p konumları arasındaki uzaklık d_{kp} şeklinde tanımlanırsa, KAP Denklem (1)'deki optimizasyon problemi olarak ifade edilebilir [2].

$$\min_{x} \sum_{i,j=1}^{n} \sum_{k,p=1}^{n} f_{ij} d_{kp} x_{ik} x_{jp}$$

$$kısıtlar: \sum_{i=1}^{n} x_{ij} = 1, \quad 1 \le j \le n \qquad (1)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad 1 \le i \le n$$

$$x_{ij} \in \{0,1\}, \quad 1 \le i,j \le n$$

Denklem (1)'de, x çözümlerden birisi olup hangi konuma hangi tesisin atandığını göstermektedir. Kısıtlar bir konuma bir tesisin atanmasını ve bir tesisin tek bir konuma atanmasını sağlar.

Problem optimal olarak çözülmesi en zor olan NP-hard sınıfında yer almaktadır. Özel oluşturulmuş problemler dışında, en iyi çözümü garanti eden algoritmalar $N \le 36$'ya kadar çözebilmektedir. Boyutu daha büyük olan problemler için çeşitli sezgisel yöntemler geliştirilmiştir [3]. Sezgisel yöntemlerden en bilinenleri genetik algoritmalar, tabu araması ve tavlama benzetimi algoritmalardır.

Tavlama benzetimi yönteminin sürekli problemler için paralelleştirilmesi Onbaşoğlu ve diğ. [4] ile Ferreiro ve diğ. [5], ayrık problemler için de Laursen [6] ile Holmqvist ve diğ. [7] çalışmalarında detaylıca incelenmiştir.

Son zamanlarda KAP için tavlama benzetimi yöntemi diğer yöntemlerle karşılaştırma için [3],[8]-[10] veya başka yöntemlerle hibritlenerek [11]-[14] kullanılmaktadır. Paralel uygulamaları üzerinde çok fazla durulmamaktadır. Son yıllarda yapılan KAP için paralel tavlama benzetimi kullanan çalışmalar incelendiğinde sadece Paul'un GPU üzerinde çalışan tavlama benzetimi türevi bir çalışması vardır [15].

Bu çalışmada tavlama benzetimi yönteminin değişik paralelleştirme yöntemleriyle paralelleştirilerek, sabit iterasyonla çalıştırmak yerine, hangi paralelleştirme yönteminin bilinen en iyi sonuca daha hızlı ulaştığı anlaşılmaya çalışılmıştır. Çalışmada hata oranı 0.01'den az olması için gereken süre ve iterasyon sayılarındaki değişimler incelenmiştir. Hata oranı Denklem (2) kullanılarak

belirlenmiştir. Denklemde c bulunan çözüm değeri ve c_{best} problem için bilinen en iyi değerdir.

$$Q = \frac{C - C_{best}}{C_{best}} \quad (2)$$

Çalışmanın ilk kısmında tavlama benzetimi yöntemi detaylıca tanıtılmış, KAP'a nasıl uygulandığı anlatılmış, paralelleştirme yöntemleri açıklanmıştır. Sonraki kısımda deney ortamı açıklanmış ve sonuçlar verilmiştir. Son kısımda ise tartışma ve önerilerden bahsedilmiştir.

2 Tavlama benzetimi

Tavlama benzetimi yöntemi aşırı ısıtılmış metallerin termal soğuması benzetimine dayalı optimizasyon yöntemidir. Bir metal yeterince ısıtıldığında, erimiş metaldeki atomlar serbestçe hareket etmektedir. Sıcaklık azaldıkça atomlar düzenli hale girme eğilimindedir ve uygun sıcaklık profili kullanıldığında, son durumda kristaller minimum içsel enerjiye sahiptirler. Kristallerin dizilimi soğuma oranına bağlı olarak değişmektedir. Hızlı soğuma gerçekleştirildiğinde metalde kusurlar görülebilmektedir. Metalin yüksek sıcaklıklarda bir süre bekletilip, sonra yavaş soğutulması işlemine tavlama denilmektedir [16].

Yöntem kısaca şu şekilde çalışmaktadır: Rasgele bir başlangıç çözümü x ve belirlenen sıcaklık değeri T ile başlanılır. KAP için 2 konumdaki tesisler yer değiştirilerek yeni bir çözüm x' üretilir ve bulunan yeni çözümün maliyeti $f(x')$ ile bir önceki çözümün maliyeti $f(x)$ karşılaştırılır. Yeni çözüm daha iyiyse çözüm kabul edilir. Eğer çözüm daha kötüyse, ardışık iki çözüm Denklem (3)'te verilen koşula göre yeni çözüm kabul edilir ve sıcaklık düşürülür: $T = T * \alpha$. Tavlama benzetiminde kötü çözümlerin kabulünün olasılıksal olarak değişmesine Markov zinciri denilmektedir.

$$e^{-\left(\frac{f(x)-f(x')}{T}\right)} > rand(0,1) \quad (3)$$

Burada, α soğuma katsayısıdır. α değeri (0,1) aralığındadır. α çok küçük olduğunda hızlı soğuma, 1'e çok yakın olduğunda ise yavaş soğuma olmaktadır. Maksimum iterasyon sayısına ulaşınca veya sıcaklık belirli bir değerin altına düşünce algoritma sonlandırılır. Sıcaklığın bu şekilde değişmesine üstel soğuma denilmektedir. Tavlama benzetimi algoritması Şekil 1'de özetlenmiştir.

```
x ← rasgele başlangıç çözümü oluştur
T ← başlangıç sıcaklığı belirle
α ← soğuma katsayısı belirle
while sonlandırma kriteri sağlanmıyorsa do
    i ← 1
    while i < aynı sıcaklıkta deneme sayısı do
        x' ← x çözümünde rasgele 2 değeri yer değiştir
        if f(x') < f(x) then
            x ← x'
        else if e^(-((f(x)-f(x'))/T)) > rand(0,1) then
            x ← x'
        end if
        i ← i + 1
    end while
    T ← T * α
    if T < 1 then
        T ← başlangıç sıcaklığı
    end if
end while
```

Şekil 1: Tavlama benzetimi algoritması.

3 Paralel tavlama benzetimi

Tavlama benzetimi, sıralı çalışan bir sezgisel yöntem olduğundan tekrarlı yapısını bozmadan paralelleştirilmesi oldukça zordur [17].

Tavlama benzetimini paralelleştirme yöntemleri aşağıda verilmiştir.

3.1 Uygulama bağımlı paralelleştirme

Maliyet fonksiyonunun hesaplanması bu yöntemde işlemcilere paylaştırılabilir [5]. Ancak hesaplama yeterince karmaşık olmadığında, paralelleştirme işleminin maliyeti hesaplama maliyetini geçmektedir. Bu sebeple uygulama bağımlı paralelleştirme yönteminin uygunluğuna probleme göre karar vermek gerekmektedir.

3.2 Arama uzayı paylaştırma

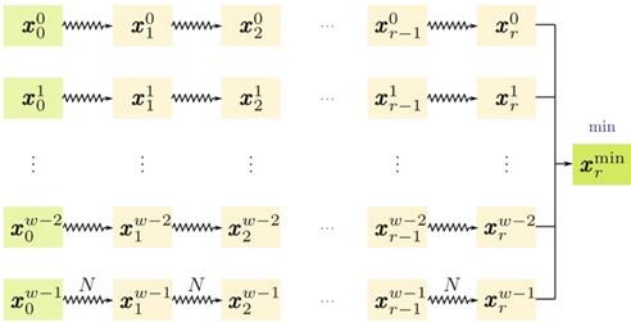
Arama uzayını alt uzaylara bölerek her paralel işlemin kendi uzayında en iyi değeri bulması sağlanır. Arama işlemi tamamlandığında bulunan sonuçlardan en iyi olanı seçilir.

3.3 Çoklu Markov zinciri yöntemi

Tavlama benzetimini en doğal şekilde paralelleştirme yöntemidir. Çoklu Markov zincirleri birbirleri arasında haberleşmeden çalıştırılır, belirli aralıklarla veya her zincirin arama işlemi tamamlandıktan sonra zincirler arası haberleşme sağlanır. Haberleşme durumuna göre ikiye ayrılır.

3.3.1 Asenkron

Tavlama benzetimi yöntemini paralelleştirmek için en temel yöntem, günümüz işlemcilerdeki çok çekirdek yapısından faydalanarak her çekirdek üzerinde eş zamanlı tavlama benzetimi algoritması çalıştırılır. Her çekirdek aramasını tamamladıktan sonra bulunan değerlerden en iyi olanı seçilir. Bu yöntemde çekirdekler aynı başlangıç çözümleriyle veya rasgele başlangıç çözümleriyle başlatılabilir. Şekil 2'de süreç görsel olarak ifade edilmiştir.

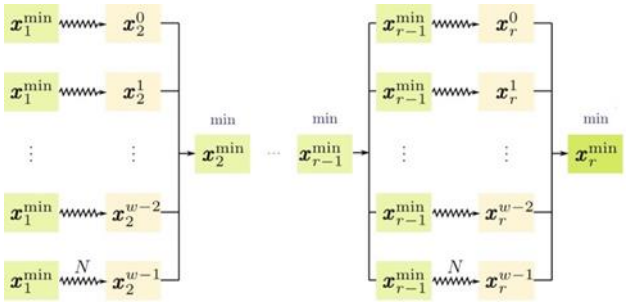


Şekil 2: Asenkron paralel tavlama benzetimi [5].

3.3.2 Senkron

Bu yöntemde her bir çekirdek üzerinde rasgele başlangıç çözümleriyle tavlama benzetimi algoritması çalıştırılır. Önceden belirlenen sıcaklık değerine veya iterasyona ulaşıldığında haberleşme yapılır. O andaki mevcut çözümler karşılaştırılır ve sonuçlardan en iyi olanı yeni başlangıç çözümü olarak seçilir ve sıcaklık azaltılarak aynı işlem tekrarlanır. Şekil 3'te süreç görsel olarak ifade edilmiştir.

Haberleşme yapıldığında sonuçlardan en iyisi belirlenir ve diğ.ne gönderilir. Bu işlem yapıldığında bir haberleşme maliyeti oluşmaktadır. Bu yüzden haberleşme belirli aralıklarla yapılarak haberleşme maliyetinin azaltılması istenebilir.



Şekil 3: Senkron paralel tavlama benzetimi [5].

4 Deneyler

Bu çalışmadaki testlerde iki adet altışar fiziksel çekirdekli Intel Xeon E5-2620 2.0 GHz işlemcisi olan 32 GB ram bulunan bir iş istasyonu kullanılmıştır. Sistemde 12 fiziksel çekirdek olduğundan ve MATLAB fiziksel çekirdek sayısı kadar paralel işleme izin verdiğinden eş zamanlı 12 paralel işlem yapılmıştır. MATLAB'da paralel hesaplama motorlarına işçi (worker) adı verildiğinden sonraki kısımlarda işçi kelimesi kullanılmıştır.

Kullanılan örnek problemler QAPLIB [18] sayfasından alınmıştır. Sayfada problemlerle ilgili detaylı bilgiler bulunmaktadır. Bu çalışmada 3 farklı örnek grubundan 36 örnek kullanılmıştır. Kullanılan örnekler ve özellikleri aşağıdaki gibidir:

Nugent: Nugent ve diğ. [19] tarafından önerilen, boyutları 12 ile 30 arasında değişen 15 örnek bulunmaktadır. Uzaklık matrisi Manhattan uzaklıklarının dikdörtgensel ızgara üzerine yerleştirilmesi ile oluşturulmuştur. Örnek boyutu $n = \{14,16,17,18,21,22,24,25\}$ olan örnekler daha büyük örneklerin belirli satır ve sütunlarının silinmesi ile elde edilmiştir.

Skorin: Skorin [20] tarafından önerilen, boyutları 42 ile 100 arasında değişen 13 örnek bulunmaktadır. Mesafeler diktörtgenseldir ve uzaklıklar sözde rasgele sayılardır [21].

Lipa: Li ve Pardalos [22] tarafından önerilen, boyutları 20 ile 90 arasında değişen 16 örnek bulunmaktadır. Bilinen optimum değerlerden asimetrik örnekler oluşturulmuştur. Bu örnekler kendi içerisinde a ve b şeklinde ikiye ayrılmaktadır. Her boyut için 2 problem vardır. Bu problemlerde mesafe matrisi aynı, akış matrisleri farklıdır. Bu çalışmada problem boyutuna bağlı olarak değişimi izlemek adına örneklerden (b grubu) kullanılmıştır.

Testlerde tavlama benzetimi için soğuma katsayısı olarak $\alpha = 0.9$, aynı sıcaklıktaki deneme sayısı olarak problem boyutunun yarısı $(n/2)$ ve maksimum iterasyon sayısı olarak problemin 3000 katı belirlendiğinde iyi sonuçlar alınmıştır. Başlangıç sıcaklığı olarak $T = 0.005 * f(x_{best})$ şeklinde alındığında Hussin ve Stützle [8] makul sonuçlar verdiğini belirtmişlerdir. Burada x_{best} aramada bulunan en iyi değerdir. Ancak probleme göre maliyet fonksiyonunun değeri değişmekte; bazılarında başlangıç sıcaklığı kötü çözüm kabulü açısından düşük, bazılarında yüksek gelmektedir. Bu yüzden oran 0.005 yerine her problem için başlangıç sıcaklığı oranı her problem için ızgara araması yöntemi yardımıyla belirlenmiştir. Her oran değeri için 10 deneme yapılmış, iterasyon sayılarının ortalaması en az olan değer o problem için başlangıç sıcaklığı oranı olarak belirlenmiştir. İlk sıcaklık 100 rasgele çözümün ortalaması ile belirlenen oranın çarpımı şeklinde elde edilmiştir. Algoritma çalışırken Sıcaklık 1'in altına düştüğünde de yeniden ısıtma işlemi uygulanmıştır. Seri tavlama benzetiminde belirlenen parametreler paralel yöntemlerde değiştirilmeden kullanılmıştır.

Çalışmada ilk olarak maliyet hesaplama fonksiyonunun paralelleştirilmesinin performansa etkileri denenmiştir. Ancak paralelleştirilme işlemi gerçekleştirildiğinde, büyük boyuttaki problemler için bile hesaplama süresinin kısalmadığı, aksine arttığı gözlemlenmiştir. Buradan maliyet fonksiyonunu paralelleştirilmesi işlemi maliyetinin hesaplama işleminin maliyetinden fazla olduğu, KAP için uygun olmadığı anlaşılmıştır. Arama uzayı paralelleştirilmesi de uzayın çok geniş olması (boyutu n olan problem için n! farklı çözüm) sebebiyle, 12'ye bölünse bile hala çok büyük olması, tüm çözümlerin hesaplanmasının mümkün olmaması sebebiyle denenmemiştir.

Tüm problemler için seri (paralel olmayan), işçilerin birbiriyle haberleşmediği asenkron ve belirli aralıklarla haberleşme yapılıp tüm işçilerin bulunan en iyi çözümü ara başlangıç çözümü olarak kabul ettiği senkron algoritmalar her problem için 100 defa çalıştırılmış ve ortalamaları alınmıştır. Sonuçlar hesaplama süresi ve iterasyon sayısı bakımından karşılaştırılmıştır.

Asenkron çalışan algoritmada 12 tane asenkron tavlama benzetimi eş zamanlı çalıştırılmış, işçilerden bir tanesi istenilen hata oranına ulaşınca, diğer işçilerdeki işlemler sonlandırılmıştır.

Senkron algoritmada birinci işçi ana işçi olarak belirlenmiştir. Her işçi rasgele bir çözümle başlatılmıştır. Belirli aralıklarla diğer işçiler buldukları çözümü 1. işçiye göndermekte, 1. de kendisine gönderilen çözümlerden en iyisini bulup diğer işçilere bu çözümü göndermektedir. Kullanılan örnek problemler için hep en iyi çözüm üzerinden gidildiğinde bazı problemlerde iyi çözümler verse de genel olarak seri tavlama benzetimi yönteminden daha kötü sonuç verdiği, hatta maksimum iterasyon sayısına ulaştığı halde hedeflenen hata oranına ulaşılamadığı görülmüştür. Örnek olarak Tablo 1 incelendiğinde Lipa problemlerinde senkron yöntem tüm örnekler için hedef hata değerine ulaşamamıştır. En iyi çözümden gitmenin yerel minimumlara takılmalara sebep olduğu anlaşılmıştır.

Tablo 1: Lipa problemleri seri ve senkron yöntem karşılaştırma.

	Seri		Senkron	
Problem	Süre (s)	Hata (Q)	Süre (s)	Hata (Q)
lipa20b	0.28	0.0000	18.69	0.0645
lipa30b	1.28	0.0000	66.75	0.1000
lipa40b	4.63	0.0000	161.33	0.1261
lipa50b	12.32	0.0000	363.49	0.1489
lipa60b	34.69	0.0000	664.37	0.1634
lipa70b	82.91	0.0000	1112.16	0.1687
lipa80b	265.67	0.0009	1774.62	0.1751
lipa90b	443.12	0.0008	2679.61	0.1768
Ortalama	105.61	0.0002	855.13	0.1404

Senkron algoritmada her zaman en iyi çözümden gitmenin iyi olmadığı anlaşılınca, genetik algoritmayı anımsatması

nedeniyle evrimsel senkron olarak isimlendirdiğimiz bir algoritma denenmiştir. Bu algoritmada 100 iterasyonda bir haberleşme yapılmış, haberleşme yapılırken her işçinin bulduğu çözümler sıralanarak en iyi 6 çözüm, diğer çözümlerden rasgele 2 tanesi ve rasgele oluşturulan 4 yeni çözüm ara başlangıç çözümü olarak kullanılmıştır. Haberleşme 100 iterasyondan daha seyrek yapıldığında kullandığımız bazı problemlerde hiç haberleşme yapılmadan sonuca ulaşılacağı, bu durumda da yöntemin asenkron yöntemden farkı kalmayacağı için 100 iterasyonda bir haberleşme yapılmıştır. Son olarak evrimsel senkron yönteminin haberleşme sıklığı arttırıldığında çözüm üzerine etkilerini incelemek için 25 ve 50 iterasyonda bir haberleşme de yapılmış, 100 iterasyonda bir yapılan haberleşmeyle karşılaştırılmıştır.

5 Sonuçlar

Tablo 2'de Nugent problemleri için yöntemlerin değerleri verilmiştir. Ortalama sonuçlara bakıldığında Nugent problemleri için en iyi hesaplama süresi asenkron yöntemde alınmıştır. Asenkron yöntem sürede %46.24 iyileşme, evrimsel senkron 100 yöntemi ise %14.68 iyileşme sağlamıştır. İterasyon sayısında iyileşmeye bakıldığında ise asenkron yöntem %72.88 iyileşme, evrimsel senkron yöntemi ise %84.94 iyileşme sağlamıştır.

Şekil 4 ve Şekil 5 incelendiğinde nug12'den nug18'e kadar olan problemlerde evrimsel senkron yönteminin iterasyon yönünden en iyi, ancak hesaplama süresi olarak en kötü sonuçları vermiştir. Bunun sebebi olarak haberleşme maliyeti verilebilir.

Tablo 3'te Skorin problemleri için yöntemlerin değerleri verilmiştir. Ortalama sonuçlara bakıldığında, hesaplama süreleri bakımından Skorin problemleri için en iyi sonuçlar yine asenkron yöntemde alınmıştır. Asenkron yöntem %79.39, evrimsel senkron 100 yöntemi ise %42.31 iyileştirme sağlamıştır. İterasyon sayısı olarak bakıldığında da paralel yöntemler iterasyonu oldukça azaltmaktadır. Asenkron yöntem %82.13, evrimsel senkron 100 ise %80.51 iyileşme sağlamıştır.

Tablo 2: Nugent problemleri için karşılaştırma.

	SERİ		ASENKRON			EVRİMSEL SENKRON			
Problem	Süre	İterasyon	Süre	İyileşme(%)	İterasyon	İyileşme(%)	Süre	İyileşme(%)	İterasyon	İyileşme(%)
nug12	0.43	2446	0.39	9.76	1174	52.02	0.45	-6.07	248	89.86
nug14	0.32	1606	0.27	16.33	673	58.07	0.46	-41.54	225	85.99
nug15	0.23	991	0.22	7.48	394	60.28	0.45	-92.96	207	79.11
nug16a	0.46	1936	0.28	39.44	623	67.81	0.48	-5.01	258	86.68
nug16b	0.48	2019	0.31	35.63	634	68.61	0.50	-5.30	269	86.68
nug17	0.39	1453	0.27	31.18	441	69.65	0.49	-24.55	239	83.55
nug18	0.78	2809	0.37	52.18	698	75.14	0.59	23.78	411	85.37
nug20	0.90	2850	0.43	51.78	787	72.38	0.59	34.58	365	87.19
nug21	1.07	3049	0.40	62.25	584	80.85	0.58	45.60	341	88.81
nug22	1.10	3040	0.45	58.56	612	79.88	0.62	43.56	357	88.26
nug24	1.26	3062	0.50	60.66	672	78.05	0.68	45.84	445	85.47
nug25	0.79	1715	0.44	44.43	464	72.96	0.63	20.48	383	77.66
nug27	5.29	10703	0.99	81.33	1161	89.15	1.57	70.36	1577	85.27
nug28	4.25	8423	0.85	80.02	956	88.65	1.32	69.04	1159	86.24
nug30	1.76	3081	0.66	62.64	625	79.71	1.02	42.31	681	77.90
Ortalama	1.30	3279	0.45	46.24	700	72.88	0.70	14.68	478	84.94

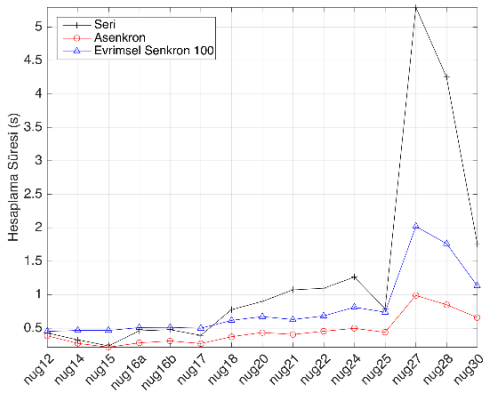
Tablo 3: Skorin problemleri için karşılaştırma.

	SERİ		ASENKRON			EVRİMSEL SENKRON			
Problem	Süre	İterasyon	Süre	İyileşme(%)	İterasyon	İyileşme(%)	Süre	İyileşme(%)	İterasyon	İyileşme(%)
sko42	5.85	5344	1.33	77.21	836	84.35	1.90	67.59	875	83.63
sko49	11.48	7520	2.58	77.49	1328	82.34	4.23	63.12	1756	76.65
sko56	26.69	13008	4.65	82.59	1952	84.99	6.55	75.46	2130	83.63
sko64	24.19	7966	3.94	83.72	1111	86.05	4.71	80.51	1056	86.74
sko72	31.43	7917	5.56	82.32	1237	84.38	7.13	77.32	1240	84.34
sko81	47.05	8692	7.75	83.53	1250	85.62	9.44	79.93	1250	85.62
sko90	67.62	9361	14.13	79.10	1771	81.08	19.37	71.35	2007	78.56
sko100a	67.63	7002	18.02	73.36	1683	75.97	22.65	66.52	1783	74.54
sko100b	123.56	12752	25.70	79.20	2410	81.10	39.29	68.20	2889	77.34
sko100c	106.34	11130	25.71	75.82	2406	78.38	32.91	69.05	2394	78.49
sko100d	63.41	6641	15.42	75.68	1473	77.81	20.77	67.24	1626	75.51
sko100e	119.00	12492	19.84	83.32	1926	84.58	26.66	77.59	1968	84.25
sko100f	75.84	7942	16.10	78.77	1503	81.07	24.50	67.70	1800	77.34
Ortalama	59.24	9059	12.36	79.39	1607	82.13	16.93	71.66	1752	80.51

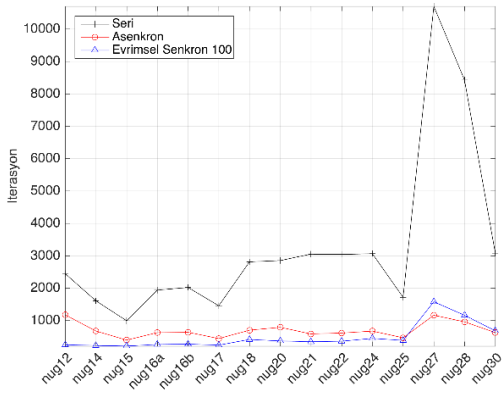
Şekil 6 ve Şekil 7 incelendiğinde Skorin problemleri için paralel yöntemlerin iterasyon sayılarının birbirine çok yakın olmasına rağmen, hesaplama sürelerindeki farklılık aynı şekilde haberleşme maliyetine bağlanabilir. Tablo 4'te ise Lipa problemleri için yöntemlerin değerleri verilmiştir. Ortalama sonuçlara bakıldığında Lipa problemleri için en iyi hesaplama süresi önceki problemlerde olduğu gibi asenkron yöntemde alınmıştır. Asenkron yöntem sürede %70.00, evrimsel senkron 100 yöntemi ise %50.08 iyileşme sağlamıştır. İterasyon sayısındaki iyileşmeye bakıldığında ise asenkron yöntem %80.90, evrimsel senkron yöntemi ise %80.87'lik bir iyileşme sağlamıştır.

Lipa problemleri için süre ve iterasyondaki iyileşmeler grafik olarak Şekil 8 ve Şekil 9'da görülebilmektedir. Ancak burada önceki problemlerin aksine en iyi iterasyon sayıları asenkron yöntem tarafından elde edilmiştir.

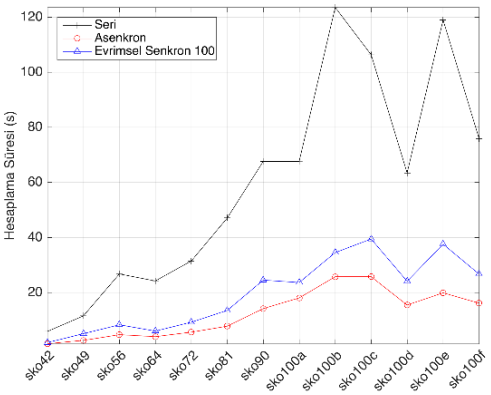
Sonuç olarak 36 probleme bakıldığında paralel yöntemler hesaplama sürelerinde ve iterasyonda oldukça belirgin iyileştirmeler sağlamışlardır. Paralel yöntemler içinde de asenkron yöntem daha iyi sonuçlar vermiştir.



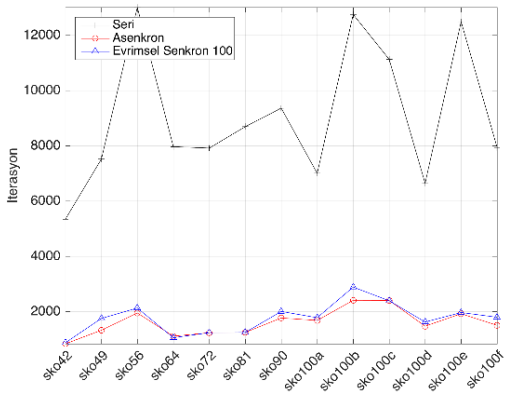
Şekil 4: Nugent problemleri için hesaplama süreleri.



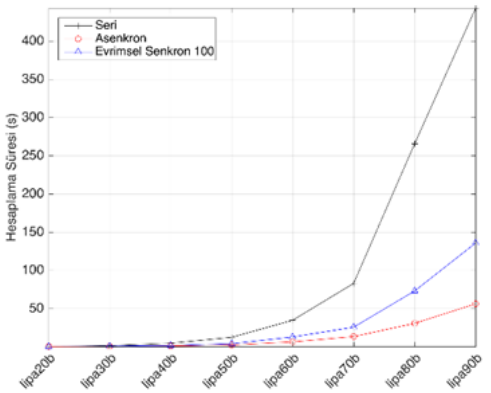
Şekil 5: Nugent problemleri için iterasyon sayıları.



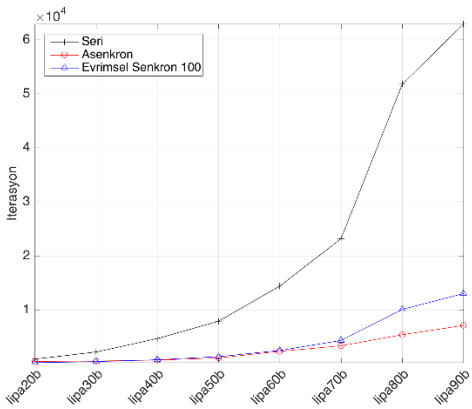
Şekil 6: Skorin problemleri için hesaplama süreleri



Şekil 7: Skorin problemleri için iterasyon sayıları.



Şekil 8: Lipa problemleri için hesaplama süreleri



Şekil 9: Lipa problemleri için iterasyon sayıları

5.1 Evrimsel senkron yönteminde haberleşme sıklığının çözüme etkileri

Evrimsel senkron yönteminin haberleşme sıklığı etkisini incelemek için 100 iterasyonda haberleşmeye göre 50 ve 25 iterasyonda haberleşme sonuçları karşılaştırılmıştır. Nugent problemleri için sonuçlar Tablo 5, Skorin problemleri için sonuçlar Tablo 6 ve Lipa problemleri için sonuçlar Tablo 7'de verilmiştir. Haberleşme sıklığının arttırmanın bazı problemler için olumlu sonuçlar verdiği, ancak ortalama sonuçlara bakıldığında olumlu etkisinin fazla olmadığı görülmüştür.

6 Tartışma ve öneriler

Sonuçlardan da görüldüğü gibi paralel yöntemler kullanılan örnek problemler için seri tavlama benzetimine göre çok daha iyi sonuçlar vermiştir. Bu çalışmada en iyi sonuçlar asenkron paralel tavlama benzetiminde alınmıştır. Çözümle ilgili bilgilerin olmadığı problemler için de belirli bir iterasyon kadar çalıştırılarak bu sonuçlardan en iyi olanı kullanılabilir. Evrimsel senkron yönteminde seri tavlama benzetimine göre çok daha iyi sonuçlar alınmasına rağmen asenkron yöntem kadar başarılı değildir.

Tablo 4: Lipa problemleri için karşılaştırma.

	SERİ		ASENKRON			EVRİMSEL SENKRON 100			
Problem	Süre	İterasyon	Süre	İyileşme (%)	İterasyon	İyileşme (%)	Süre	İyileşme (%)	İterasyon	İyileşme (%)
lipa20b	0.28	876	0.27	4.18	431	50.83	0.49	-76.60	227	74.10
lipa30b	1.28	2177	0.57	55.65	507	76.69	0.74	42.23	401	81.58
lipa40b	4.63	4670	1.04	77.57	676	85.52	1.45	68.79	745	84.05
lipa50b	12.32	7881	2.30	81.32	1083	86.26	3.43	72.15	1327	83.16
lipa60b	34.69	14385	6.40	81.54	2310	83.94	8.95	74.19	2460	82.90
lipa70b	82.91	23154	13.28	83.98	3326	85.63	22.41	72.97	4327	81.31
lipa80b	265.67	51750	30.65	88.46	5378	89.61	68.67	74.15	10060	80.56
lipa90b	443.12	62888	56.19	87.32	7101	88.71	120.69	72.76	13009	79.31
Ortalama	105.61	20973	13.84	70.00	2602	80.90	28.35	50.08	4070	80.87

Tablo 5: Nugent problemleri için evrimsel senkron yöntem karşılaştırması.

	EVRİMSEL SENK. 100		EVRİMSEL SENKRON 50			EVRİMSEL SENKRON 25			
Problem	Süre	İterasyon	Süre	İyileşme (%)	İterasyon	İyileşme (%)	Süre	İyileşme (%)	İterayon	İyileşme (%)
nug12	0.45	248	0.45	1.37	209	15.73	0.45	0.45	194	21.77
nug14	0.46	225	0.47	-2.68	209	7.33	0.46	0.47	197	12.67
nug15	0.45	207	0.45	-0.68	177	14.73	0.45	0.47	183	11.47
nug16a	0.48	258	0.46	4.54	220	14.73	0.48	0.51	247	4.46
nug16b	0.50	269	0.47	6.72	281	-4.46	0.50	0.51	252	6.23
nug17	0.49	239	0.45	7.85	217	9.41	0.49	0.50	232	3.03
nug18	0.59	411	0.54	9.53	411	0.00	0.59	0.61	401	2.49
nug20	0.59	365	0.59	-1.30	425	-16.44	0.59	0.67	438	-19.93
nug21	0.58	341	0.57	1.76	348	-1.91	0.58	0.63	368	-7.77
nug22	0.62	357	0.58	6.45	336	6.02	0.62	0.68	403	-12.82
nug24	0.68	445	0.68	0.97	436	2.02	0.68	0.81	519	-16.63
nug25	0.63	383	0.68	-8.20	432	-12.79	0.63	0.74	435	-13.58
nug27	1.57	1577	1.78	-13.41	1756	-11.32	1.57	2.02	1762	-11.75
nug28	1.32	1159	1.35	-2.66	1120	3.41	1.32	1.76	1362	-17.54
nug30	1.02	681	1.04	-2.36	660	3.16	1.02	1.13	661	2.90
Ortalama	0.70	478	0.70	0.53	482	1.98	0.70	0.80	510	-2.33

Tablo 6: Skorin problemleri için evrimsel yöntem karşılaştırması.

	EVRİMSEL SENK. 100		EVRİMSEL SENKRON 50			EVRİMSEL SENKRON 25			
Problem	Süre	İterasyon	Süre	İyileşme (%)	İterasyon	İyileşme (%)	Süre	İyileşme (%)	İterasyon	İyileşme (%)
sko42	1.90	875	1.78	5.92	884	-0.97	1.79	5.77	873	0.29
sko49	4.23	1756	4.61	-8.84	1808	-2.96	5.01	-18.49	1787	-1.74
sko56	6.55	2130	6.93	-5.86	2014	5.45	8.24	-25.85	2149	-0.87
sko64	4.71	1056	5.35	-13.42	1123	-6.34	5.98	-26.90	1203	-13.90
sko72	7.13	1240	7.49	-5.09	1246	-0.48	9.18	-28.74	1498	-20.81
sko81	9.44	1250	12.37	-31.01	1582	-26.52	13.47	-42.61	1670	-33.60
sko90	19.37	2007	23.74	-22.54	2319	-15.52	24.50	-26.46	2336	-16.38
sko100a	22.65	1783	24.63	-8.76	1807	-1.32	23.61	-4.24	1763	1.14
sko100b	39.29	2889	38.71	1.48	2759	4.52	34.48	12.24	2664	7.79
sko100c	32.91	2394	35.85	-8.93	2748	-14.77	39.41	-19.75	2854	-19.23
sko100d	20.77	1626	23.53	-13.31	1630	-0.22	24.06	-15.84	1681	-3.38
sko100e	26.66	1968	30.27	-13.52	2152	-9.35	37.53	-40.75	2615	-32.86
sko100f	24.50	1800	23.74	3.12	1793	0.42	26.66	-8.81	1900	-5.54
Ortalama	16.93	1752	18.38	-9.29	1835	-5.24	19.53	-18.50	1922	-10.70

Tablo 7: Lipa problemleri için evrimsel yöntem karşılaştırması.

	EVRİMSEL SENK. 100		EVRİMSEL SENKRON 50			EVRİMSEL SENKRON 25			
Problem	Süre	İterasyon	Süre	İyileşme (%)	İterasyon	İyileşme (%)	Süre	İyileşme (%)	İterasyon	İyileşme (%)
lipa20b	0.49	227	0.49	0.14	195	14.10	0.51	-4.70	192	15.53
lipa30b	0.74	401	0.76	-2.38	388	3.24	0.83	-12.09	411	-2.56
lipa40b	1.45	745	1.42	2.12	682	8.46	1.48	-2.64	758	-1.74
lipa50b	3.43	1327	3.69	-7.63	1338	-0.79	4.01	-16.97	1612	-21.46
lipa60b	8.95	2460	11.14	-24.43	2800	-13.82	12.78	-42.71	3444	-40.01
lipa70b	22.41	4327	25.35	-13.12	4975	-14.98	25.60	-14.24	4804	-11.01
lipa80b	68.67	10060	72.45	-5.50	9803	2.56	73.00	-6.31	9742	3.17
lipa90b	120.69	13009	130.93	-8.49	13099	-0.69	135.75	-12.48	12887	0.94
Ortalama	28.35	4070	30.78	-7.41	4160	-0.24	31.75	-14.02	4231	-7.14

Mevcut haliyle haberleşmenin hangi aralıklarla olması gerektiği, hangi sonuçların devam edeceği, hangileri yerine rasgele yeni çözümler üretileceği gibi belirlenmesi gereken parametreler ortaya çıkmaktadır. Bu da şu anki sonuçlar için daha az parametre olduğundan asenkron yöntemi avantajlı kılmaktadır. Asenkron yöntemin seri yönteme göre belirlenmesi gereken tek ekstra parametresi aynı anda kaç paralel tavlama benzetimine karar vermektir. Burada da sistemin izin verdiği kadar başlatılması sonuç başarısını arttıracaktır.

Gelecek çalışmalar olarak daha iyi bir birleştirme stratejisi geliştirerek evrimsel yöntemin başarısını arttırmaya çalışılabilir. Örneğin, haberleşme her 100 iterasyonda bir değil; yüksek sıcaklıklarda çok az, sıcaklık azaldığı zamanlarda daha sık haberleşmenin yapılması gibi yaklaşımlar denenerek asenkron yöntemden daha iyi sonuçlar verip vermeyeceği incelenebilir. Ayrıca çalışmadaki hesaplamalar grafik işlem birimi üzerinde yapılarak da genişletilebilir.

7 Kaynaklar

[1] Koopmans TC, Beckmann M. "Assignment problems and the location of economic activities". *Econometrica*, 25(1), 53-76, 1957.

[2] Loiola EM, de Abreu NMM, Boaventura-Netto PO, Hahn P, Querido T. "A survey for the quadratic assignment problem". *Europian Journal of Operational Research*, 176(2), 657-690, 2007.

[3] Paul G. "Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem". *Operations Research Letters*, 38(6), 577-581, 2010.

[4] Onbaşoğlu E, Özdamar L. "Parallel simulated annealing algorithms in global optimization". *Journal of Global Optimization,* 19(1), 27-50, 2001.

[5] Ferreiro AM, García JA, López-Salas JG, Vázquez C. "An efficient implementation of parallel simulated annealing algorithm in GPUs". *Journal of Global Optimization*, 57(3), 863-890, 2013.

[6] Laursen PS. "Parallel heuristic search-introductions and a new approach". *Solving Combinatorial Optimization Problems in Parallel*. Editors: Ferreira A, Pardalos P. Lecture Notes in Computer Science, 1054, 248-274, 1996.

[7] Holmqvist K, Migdalas A, Pardalos PM. *Parallelized Heuristics for Combinatorial Search*. Editors: Migdalas A, Pardalos PM, Storøy S. Parallel Computing in Optimization, 269-294, Boston, MA, USA, Springer, 1997.

[8] Hussin MS, Stützle T. "Tabu search vs. simulated annealing as a function of the size of quadratic assignment problem instances". *Computers & Operation Research*, 43, 286-291, 2014.

[9] Said GAENA, Mahmoud AM, El-Horbaty ESM. "A comparative study of meta-heuristic algorithms for solving quadratic assignment problem". *International Journal of Advanced Computer Science and Applications*, 5(1), 1-6, 2014.

[10] Karami M, Niroomand S, Mirzaei N, Vizvari B. "Analysis and performance measurement of existing solution methods of quadratic assignment problem". *International Journal of Electronics, Mechanical and Mechatronics Engineering*, 3(2), 557-570, 2014.

[11] Ghandeshtani KS, Mollai N, Seyedkashi SMH, Neshati MM. "New simulated annealing algorithm for quadratic assignment problem". *ADVCOMP 2010: The Fouth International Conference on Advanced Engineering Computing and Applications in Sciences*, Florence, Italy, 25-30 October 2010.

[12] Wang JC. "A multistart simulated annealing algorithm for the quadratic assignment problem". *3rd International Conference on Innovations in Bio-Inspired Computing and Applications*, Kaohsiung, Taiwan, 26-28 September 2012.

[13] Kovac M. "Solving quadratic assignment problem in parallel using local search with simulated annealing elements". *The International Conference on Digital Technologies 2013*, Zilina, Slovakia 29-31 May 2013.

[14] Kaviani MA, Abbasi M, Rahpeyma B, Yusefi MM. "A hybrid tabu search-simulated annealing method to solve quadratic assignment problem". *Decision Science Letters*, 3(3), 391-396, 2014.

[15] Paul G. "A GPU implementation of the simulated annealing heuristic for the quadratic assignment problem". Computing Research Repository, arXiv: 1208.2675, 2012.

[16] Rao SS. *Engineering Optimization Theory and Practice*. 4th ed. New Jersey, USA, Wiley, 2009.

[17] Chen H, Flann NS, Watson DW. "Parallel genetic simulated annealing: a massively parallel SIMD algorithm". *IEEE Transactions on Parallel and Distributed Systems*, 9(2), 126-136, 1998.

[18] Burkard RE, Çela E, Karisch SE, Rendl F. "QAPLIB". http://anjos.mgi.polymtl.ca/qaplib/ (09.05.2017).

[19] Nugent CE, Vollmann TE, Ruml J. "An experimental comparison of techniques for the assignment of facilities to locations". *Operations Research*, 16(1), 150-173, 1968.

[20] Skorin-Kapov J. "Tabu search applied to the quadratic assignment problem". *ORSA Journal of Computing*, 2(1), 33-45, 1990.

[21] Burkard RE, Karisch SE, Rendl F. "QAPLIB A quadratic assignment problem library". *Journal of Global Optimization,* 10(4), 391-403, 1997.

[22] Li Y, Pardalos PM. "Generating quadratic assignment test problems with known optimal permutations". *Computational Optimization and Applications*, 1(2), 163-184, 1992.