Pamukkale Univ Muh Bilim Derg, 23(5), 559-565, 2017

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Karesel atama problemi için yeni bir özuyarlamalı paralel güçlü tabu-arama algoritması

A self-adaptive parallel robust tabu-search algorithm for the quadratic assignment problem

Tansel DÖKEROĞLU[1*]

[1]Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Türk Hava Kurumu Üniversitesi, Ankara, Türkiye.
tdokeroglu@thk.edu.tr

Geliş Tarihi/Received: 31.01.2016, Kabul Tarihi/Accepted: 08.09.2016
doi: 10.5505/pajes.2016.88262
* Yazışılan yazar/Corresponding author
Araştırma Makalesi/Research Article

Öz

Bu çalışma ile Karesel Atama Problemi (KAP) olarak bilinen ve çok sayıda konum ve tesis içeren örnekler için en iyi çözümleri hala bulunamamış olan NP-zor bir kombinatoriyal problem için yeni bir paralel sezgisel algoritma önerilmektedir (paralel-tabu-KAP algoritması). İki safhası bulunan paralel-tabu-KAP algoritması, genetik algoritma safhasında efendi işlemcide bulunan popülasyon üzerinde sezgisel tabu-arama algoritmasının parametrelerini jenerasyonlar ile eniyilerken, tabu-arama safhasında işçi işlemciler üzerinde verilen problemin sonucunu farklı başlangıç noktaları ile eniyilemektedir. Yerel takılmaları, aramaya başka noktalardan yeniden başlayarak engelleme özelliğine sahip olan paralel-tabu-KAP algoritması, tek işlemci ile çalışan ve parametreleri statik olarak önceden tanımlanmış olan versiyonlarına göre daha iyi sonuçlar elde etmektedir. Yüzün üzerindeki bençmark problem ile yapılan deneyler sonucunda, ortalama %0.05'lik bir sapma elde edilmiştir. Bu sonuçlar, paralel-tabu-KAP algoritmasının kendi sınıfındaki sezgisel algoritmalar içerisinde KAP'ın çözümü için önerilen en iyi algoritmalar arasında olduğunu göstermektedir.

Anahtar kelimeler: Tabu-arama, Paralel, Karesel atama problemi, Genetik

Abstract

With this study, we propose a novel parallel robust tabu-search algorithm (Parallel-Tabu-QAP) for the NP-Hard Quadratic Assignment Problem (QAP)of whose instances having large number of location and facilites have not been reported to be solved exactly so far. Parallel-Tabu-QAP algorithm that has two phases optimizes the parameters of tabu-search algorithm in its first phase by using genetic algorithms through generations. The individuals that have parameters of the tabu-search are optimized in a population that is located on a master processor. In the second phase, slave processors optimize the solution of the problem by restarting their search process in case of stagnations. With its stagnation prevention and parallel optimization talents, parallel-tabu-QAP algorithm is observed to obtain better results than its sequential and statically parameter-tuned counterparts. The algorithm has 0.05% deviation for the benchmark tests performed on more than 100 problem instances. These experimental results show that the parallel-tabu-QAP algoritm is one the best performing techniques in its heuristics algorithms class when compared with state-of-the-art QAP algorithms.

Keywords: Tabu-search, Parallel, Quadratic assignment problem, Genetic

1 Giriş

Karesel atama problemi (KAP), her konum için farklı kurulum maliyetleri olan tesislerin minimum toplam maliyet ile eldeki lokasyonlara atanması işlemidir [1]-[3]. Kurulum ve gerekli nakliyat için seçilecek en uygun tahsisat planı ile tesisler arasındaki malzemelerin akış miktarlarının en aza indirilmesi hedeflenmektedir. Her ne kadar, tesis yeri seçimi KAP'ın en bilinen örneklerinden birisi olsa da, seyyar satıcı, kutu paketleme, maksimum klik, çizelge bölümleme, etkin ulaşım sistemleri, daktilo klavye tasarımı, kablolama ve veri yerleştirme gibi problemler de KAP ile tasarlanabilen problemler arasında yer almaktadır [4]-[7].

KAP'ı n boyutundaki problem büyüklükleri için formal olarak ifade etmek gerekirse, A, B, C'nin (n x n) boyutundaki üç matris olduğu kabul edildiğinde; denklem (1)'deki a_{ik}, tesis i'den tesis k'ye akış miktarını göstermektedir.

$$A = (a_{ik}) \tag{1}$$

Denklem (2)'deki b_{jl}, lokasyon j'den lokasyon l'ye olan mesafeyi (ulaşım masrafını) göstermektedir,

$$B = (b_{jl}) \tag{2}$$

Denklem (3)'teki c_{ij}, tesis i'nin lokasyon j'ye konumlandırılmasının masrafıdır.

$$C = (c_{ij}) \tag{3}$$

KAP'ın Koopmans-Beckmann formu, denklem (4)'teki gibidir.

$$\min_{\phi \in S_n} = \left(\sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik}\, b_{\phi(i)\phi(k)} + \sum_{i=1}^{n} c_{i\phi(i)}\right) \tag{4}$$

S_n, 1,...,n' e kadar olan sayıların permütasyonudur. Her bir $a_{ik}b_{\phi(i)\phi(k)}$ ifadesi, lokasyon $\phi(i)$ 'deki tesisten lokasyon $\phi(k)$ 'deki tesise olan ulaşım masrafıdır.

KAP büyük boyutlu örnekleri (n=40 ve daha fazlası) için halen çözülememiş problemleri barındırmaktadır. Bu özelliği ile endüstri ve bilgisayar mühendislikleri açısından ilginç bir araştırma alanı olup, birçok bilim insanının ilgisini çekmeye devam etmektedir.

Bu çalışma ile birlikte Güçlü Tabu-Araması (GTA) olarak bilinen ve KAP'ı en iyi çözen algoritmalardan birisi olan sezgisel bir algoritma, paralel bir ortamda eş zamanlı olarak farklı başlangıç noktalarından başlatılarak daha iyi sonuçları daha kısa sürelerde bulacak şekilde paralel hesaplama

ortamlarını kullanarak iyileştirildi. Önerilen algoritma, yüksek derecedeki hesaplama karmaşıklıklarından dolayı makul sürelerde çözülemeyen NP-Zor problemler için uygulanan en son teknolojik metotları bir araya getirmektedir. Çoklu başlangıç, sezgisel algoritma parametrelerinin eniyilenerek arama yapılması ve paralel hesaplama bu tekniklerin en önemlileridir. GTA'nın en iyi algoritmalardan birisi olmasına rağmen, büyük KAP için saatlerce süren aramalar yaptığı ve yine de en iyi sonuca ulaşamadığı bilinmektedir. Bu açıdan bakıldığında sezgisel eniyileme algoritmalarının performanslarının ölçeklenebilir paralel uygulamalar ile daha da iyileştirilebileceği değerlendirilmektedir.

Önerilen paralel-tabu-KAP algoritması, tek işlemci ile çalışan GTA'ya göre daha iyi sonuçlar elde etmekte olup, 100'den fazla problem üzerinde yapılan deney sonuçlarından elde edilen bulgulara göre bu sınıf algoritmalar arasında KAP'ı en iyi çözen algoritmalardan birisidir.

Makalenin ikinci bölümünde; yapılan çalışmaya benzerlik gösteren ve deneylerde karşılaştırılan güncel tabu-arama algoritmaları özetlenmektedir. Üçüncü bölümde; önerilen paralel-tabu-KAP algoritmasının bileşenleri tanıtılmaktadır. Dördüncü bölümde; deney ortamı, bençmark problemleri ve elde edilen sonuçların değerlendirilmesi yapılmıştır. Son bölümde; önerilen algoritmanın genel bir değerlendirmesi ve gelecekte yapılabilecek çalışmalar ile ilgili bilgi verilmiştir.

2 Önceki çalışmalar

Bu bölümde, KAP'ın çözümü için geliştirilmiş tabu-arama tabanlı ve sezgisel bazı algoritmalar hakkında bilgi verilmektedir. Bugüne kadar geliştirilmiş olan algoritmalardan önerilen algoritma ile benzerliği olan (tabu-arama tabanlı), sezgisel ve en iyi sonuçları veren algoritmalar arasında bulunanlar özetlenmiştir.

Güçlü Tabu-Arama (GTA) algoritması Taillard tarafından 1991 yılında KAP'a uygulanmış en etkin sezgisel yapay zeka algoritmalardan birisidir. Önceki tabu-arama algoritmalarına göre daha basit olarak tasarlanmış ve az parametreyle çalışması sağlanmıştır [8]. Çözülemeyen KAP'lar için paralel GTA versiyonu ile yeni en iyi çözüm permütasyonları elde edilmiştir. Bu çalışmada, GTA'nın son versiyonu üzerinde iyileştirmeler yapılarak sonuçları sunulmaktadır. "Iterated tabu search" (ITS) algoritması, yoğunlaşma ve farklılaştırma tekniklerini bir araya getirerek tabu-arama algoritmasına yeni bir boyut kazandırmıştır [11]. Yoğunlaştırma ile mevcut çözümün yakın komşularını ziyaret etmek hedeflenirken, farklılaşma ile yerel en iyi çözümlerin oluşturabileceği darboğazlardan kurtulmak hedeflenmiştir. Çözümlerin daha hızlı hesaplanmasına yönelik de yeni bir yöntem önerilmiştir.

JRG-DivTS algoritması 2006 yılında James ve arkadaşları tarafından geliştirilmiş çoklu başlamalı bir tabu-arama algoritmasıdır [12]. Arama sırasında sonucun iyileşmediğinin görüldüğü durumlarda tabu-araması yeni parametreler ile tekrar başlatılmaktadır. Aramalar esnasında tabu-arama algoritmasının temel versiyonlarından birisi kullanılmaktadır.

Yapılan güncel bir çalışmada, Fescioglu ve Kokar [13] geri besleme ve öz kontrol mekanizmalarını kullanarak oluşturdukları bir yazılım bileşenini tabu-arama algoritmasına entegre ederek yeni bir algoritma geliştirilmiştir. Burada iki reaksiyon stratejisi kullanılmıştır. İlk strateji tabu-aramasını kontrol edilecek hedef sistem olarak görmekte ve parametrelerini kontrol etmektedir. İkinci stratejide de tabu-arama parametreleri farklılaştırmayı sağlayacak şekilde ayarlanmaktadır. Bu algoritmanın bizim geliştirdiğimiz algoritmadan farkı yönleri; çoklu başlangıç uygulamaması ve parametreleri farklılaştırma mekanizması olarak kullanması görülebilir.

James ve arkadaşları [14] 2009 yılında KAP için CPTS algoritmasını önermektedir. CPTS işlemciler arasında iş birliği yapan paralel bir algoritma olup global bir referans kümesi üzerinden yoğunlaşma ve farklılaşma mekanizmalarını takip ederek eniyileme yapmaktadır. Böylece paralel hesaplama ortamlarından faydalanılarak arama süreçlerinin hızlandırılması ve iyileştirilmesi sağlanmaktadır ve KAP için birçok algoritmanın sonucuna göre daha iyi olduğu iddia edilen sonuçlar elde etmektedir. Drezner [15] tarafından geliştirilen GA-C/TS algoritması ile tabu-arama algoritması için daha esnek hareketlilik sağlayan bir mekanizma oluşturularak sonuçların iyileştirilmesi sağlanmıştır. Bu algoritma melez bir çalışmadır. Dökeroglu [2] tarafından geliştirilen paralel melez bir algoritma ile öğrenme ve öğretme tabanlı sezgisel algoritmalar tabu-araması ile birlikte uygulanarak problemlerin büyük bir çoğunluğu optimal olarak çözülmüştür.

Geliştirilen algoritmada kullanılan genetik operatörler NP-Zor kombinatoriyal eniyileme problemlerinde 1970'lerden beri etkin olarak kullanılmaktadır [16],[17]. Doğrusal zamanlı çalışma süreleri ile üssel çözüm karmaşıklığına sahip problemleri en iyi (ya da en iyiye çok yakın) şekilde çözebilmektedirler. Kolay tasarım özelliklerinden dolayı genetik algoritma sezgisel GTA algoritmasının karmaşık parametre uzayının eniyilenmesini sağlamak için kullanılmıştır.

3 Önerilen algoritma, paralel-tabu-KAP

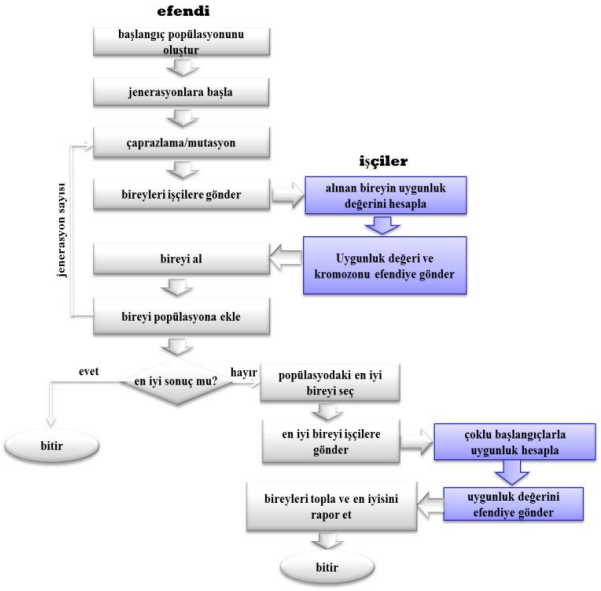
Bu bölümde, önerilen Paralel-Tabu-KAP algoritmasının bileşenleri ve mantıksal akış sırası sunulmuştur. Algoritmanın içerisinde iki ana bileşen bulunmaktadır. Genetik algoritma safhası ve GTA algoritması safhası.

Efendi işlemci üzerinde çözüm kromozomlarından oluşan bir popülasyon bulunmaktadır. Genetik algoritma safhası, efendi işlemci üzerinde bulunan bu popülasyon üzerinde çalışmaktadır. Seçilen bireyler, işçi işlemcilere gönderilerek efendi işlemci tarafından belirlenen parametreler ile GTA algoritması çalışmakta ve sonuçlar tekrar efendi işlemciye gönderilmektedir. Jenerasyonlar sonunda, en iyi çözümü erişilememiş olan problem için çoklu başlatmalar içeren GTA safhası başlatılır. Bu safhada efendi işlemcide bulunan en iyi birey seçilerek tüm işçi işlemcilere gönderilir ve işlemciler birbirlerinden farklı başlangıç noktalarından başlayarak aramalarını yaparlar. Bu işlemler sırasında yeniden başlama sayısı beş olarak verilmiştir. Genetik safhasında kullanılan popülasyon sayısı, jenerasyon sayısı, çaprazlama oranı, mutasyon oranı gibi parametreler Tablo 1'de sunulmuştur. Bu değerler, daha önceki çalışmalarımızdan elde ettiğimiz en iyiye yakın parametrelerden oluşmaktadır.

Tablo 1: Önerilen algoritmanın parametreleri.

Parametre	Büyüklük
Popülasyon sayısı	40
Jenerasyon sayısı	50
Çaprazlama oranı	%20
Mutasyon oranı	%1
Yeniden başlama sayısı	5
İşlemci sayısı	128

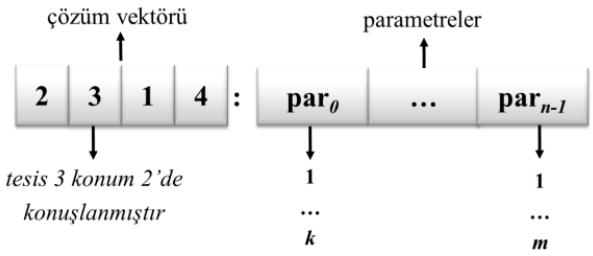
Şekil 1'de algoritmanın akış diyagramı detaylı olarak verilmiştir.



Şekil 1: Önerilen algoritmanın akış diyagramı.

3.1 Kromozom yapısı

KAP çözümü için önerdiğimiz algoritmanın kromozom yapısı Şekil 2'de sunulmuştur. Bu yapı, iki bölümden oluşmaktadır. Sol tarafta çözüm vektörünü içeren permütasyon, sağ tarafta ise GTA'nın parametrelerinin olduğu vektör bulunmaktadır. Sağ tarafta dört adet parametre bulunmaktadır. Bu parçanın genleri birer parametreyi temsil etmekle birlikte alabilecekleri değerlerin aralıkları daha önceden tanımlanmıştır. Tablo 2'de bu aralıklar verilmiştir.



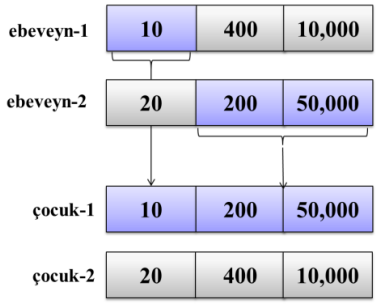
Şekil 2: Kromozom yapısı.

Tablo 2: Tabu-arama algoritması için eniyileme yapılan parametrelerin aralıkları (n=problemin büyüklüğü).

Parametrenin adı	Parametre aralığı	En iyi çalışan parametre aralığı
Hata yapma sayısı	[100x*n*,10000x*n*]	[9000x*n*,10000x*n*]
Tabu görev süresi alt limiti	[2x*n*, 10x*n*]	[8x*n*, 9x*n*]
Tabu görev süresi üst limiti	[11**n* , 20*n]	[11x*n*, 12x*n*]
İstek değeri	[n^2,10xn^2]	[n^2,2xn^2]

3.2 Çaprazlama operatörü

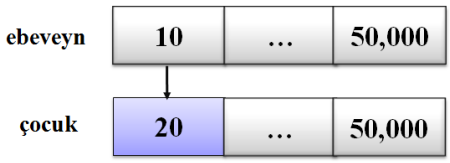
Çaprazlama operatörü, popülasyondan rastgele seçilen iki adet ebeveyn kromozom üzerinde çalışır. Bu kromozomlar, belirlenen oranlara göre ikiye ayrılır ve ayrılan parçalar diğer kromozomun parçası ile birleştirilerek iki yeni çocuk elde edilir. Şekil 3'te çaprazlama işleminin nasıl gerçekleştiği ile ilgili açıklayıcı bir resim sunulmuştur. Ebeveyn 1'in kesilen sol parça ile ebeveyn 2'nin kesilen sağ parça birleşmek suretiyle çocuk 1 kromozomunu oluşturmaktadır. Bu işlem sonucunda geçerli ve yeni bir çözüm kromozomu oluşturulabilmektedir.



Şekil 3: Çaprazlama işlemi örneği.

3.3 Mutasyon operatörü

Mutasyon operatörü, popülasyon içerisinden seçilen ebeveyn kromozomun bir genini (algoritmanın parametresini) rastgele seçer ve seçilen gen üzerinde daha önce tanımlanmış olan sayı aralığında yeni bir sayı üretilerek bir çocuk kromozom oluşturur. Şekil 4, seçilen ebeveynden mutasyon işlemi uygulanarak yeni bir çocuk kromozom oluşturma sürecini göstermektedir. Kromozomdaki ilk gen seçilerek ebeveynde 10 olan ilk parametre değeri 50 ile değiştirilerek yeni bir kromozom (yeni bir parametre kümesi) oluşturulmuştur.

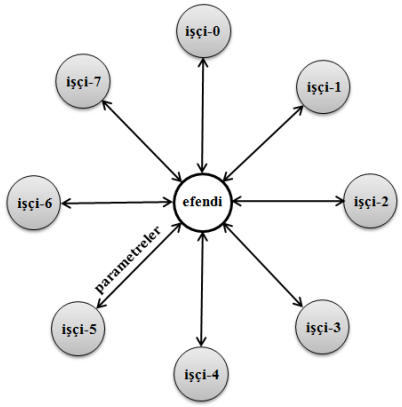


Şekil 4: Mutasyon işlemi.

Paralel-tabu-KAP algoritması, çaprazlama ve mutasyona girecek olan ebeveynleri seçerken rastgele seçim yöntemini uygulamıştır. Bu yöntemde popülasyondaki tüm bireylerin birbiri ile eşleşme ihtimali aynı olmaktadır. Böylelikle daha farklı bireyler birbirleri ile birleşerek çok farklı yeni nesil çözümler oluşturma imkânına sahip olmaktadır.

3.4 Algoritmanın iletişim topolojisi

Paralel-tabu-KAP algoritmasındaki işlemciler, efendi ve işçi işlemciler olmak üzere ikiye ayrılmışlardır. Efendi işlemci her kuşak başlangıcında, işçilere tabu-arama algoritmasının parametrelerini gönderir ve işçiler kendilerine verilen parametreler ile tabu-algoritmasını çalıştırdıktan sonra sonucu efendi işlemciye geri gönderirler ve bir sonraki işi alırlar. Efendi işlemci, gelen sonuçlara göre elindeki parametreleri popülasyondaki bireyler üzerinde günceller, popülasyondaki kötü sonuç veren bireyleri havuzdan çıkarır. Paralel uygulamalarda, (İng. *master-slave*) olarak bilinen ve birçok uygulamada etkin şekilde kullanılan bu topoloji, orta sayıda (50-1000 arasında) işlemcilerin bulunduğu yüksek performanslı bilgisayarlarda ölçeklenebilir bir iletişim altyapısı sunmaktadır. Şekil 5'te, Paralel-tabu-KAP algoritmasının işlemciler arasında gerçekleştirdiği iletişimin topolojisi görülmektedir. Bu konfigürasyonda tek efendi işlemci bulunmakta ve işçi işlemciler jenerasyonlar boyunca efendi işlemciye hizmet etmektedir.



Şekil 5: Sekiz işçi ve bir efendi işlemciden oluşan iletişim topolojisi

3.5 Güçlü tabu-arama algoritması bileşeni

Güçlü Tabu-Arama algoritması (GTA) Taillard [8]'ın 1991 yılında önerdiği ve halen KAP için en etkin çözümleri bulan algoritmalardan birisidir. GTA'nın dört önemli parametresi bulunmaktadır. Hata yapma sayısı, GTA daha iyi bir sonuç elde edemezse farklılaşma (diversification) sürecini uygulayacağı limit değerini tanımlar. Tabu görev süresi alt ve üst limitleri (tabu tenure), GTA'nın daha önce yaptığı değişikliklerin "*tabu*" olarak kabul edildiği alt ve üst süre limitlerini belirler ve o değişikliklerin bu süreler içerisinde tekrar yapılmasına engel olur. İstek değeri (aspiration value), tabu statüsünde bulunan değişiklikler üzerinde istenen bazı düzenlemeler yapılabilmesi durumunda değişikliğe izin veren kuraldır. Aramayı yeniden başlatma tekniği birçok sezgisel algoritma tarafından başarı ile uygulanan bir yöntemdir. Arama sürecinin yerel en iyilere takılması ve ilerleyememesi durumunda, farklı başka noktalardan aramaya tekrar başlanması etkin olarak sonuçları iyileştirebilmektedir.

4 Deney ortamı, elde edilen karşılaştırma sonuçları ve tartışmalar

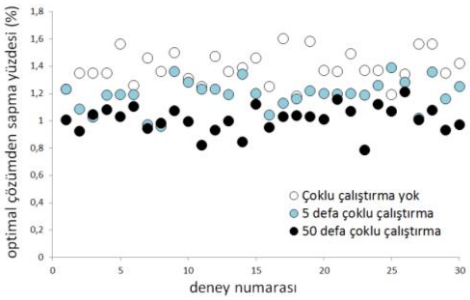
Deneyler sırasında QAPLIB [10] olarak bilinen standart bir bençmark problem kütüphanesi kullanıldı. Söz konusu kütüphanede bu problem üzerinde çalışan bilim adamlarının kullandığı 134 adet örnek problem bulunmakta olup bu çalışmada problemlerin tümü ile deneyler yapıldı. Problemler, Stutzle tarafında uygulanan dört kategori esas alınarak sınıflandırıldı [9]. Bu dört sınıf, rastgele oluşturulmuş örnekler (tip-1), ızgara (grid) tabanlı mesafeler içerenler (tip-2), gerçek hayat örnekleri (tip-3) ve gerçek hayat benzeri (tip-4) örneklerdir.

Deneyler, her birinde 2 CPU bulunan 46 düğümlü ve toplam 368 işlemcisi bulunan ODTÜ Bilgisayar Mühendisliği'nin yüksek performanslı bilgisayarı üzerinde gerçekleştirildi. Bu hesaplama ortamında toplamda 736 GB RAM kapasitesi bulunmakta olup, düğümler arasında iki adet 24 kapılı yüksek hızlı Gbps Ethernet anahtarları bulunmaktadır. Bu iletişim altyapısı, işlemciler arasından 8Gbps'lık bir haberleşme ortamı sağlanmaktadır. Uygulama, C++ ve MPI kütüphaneleri ile geliştirildi. Sonuçların güvenilirliğini artırmak için her problem için deneyler on defa yenilenerek ortalama sonuçları alındı. Deney sonuçları önceki çalışmalar bölümünde bahsedilen, JRG-DivTS, SC-Tabu, CPTS, ITS ve GA/C-TS algoritmalarının elde ettikleri ortalama sonuçlar ile karşılaştırıldı.

GTA iki genin yerini değiştirerek yeni kromozomlar üreten ve böylelikle arama yapan bir teknik uygulamaktadır. Bundan faydalanarak, deneyler sırasında her kromozomun uygunluk maliyeti (fitness value) yeni baştan hesaplanmadan sadece iki genin yer değiştirmesinden ortaya çıkan fark (delta) basit bir şekilde hesaplanmıştır. Bu teknik arama hızını büyük oranda artıran bir yöntemdir.

4.1 GTA çoklu başlama performans deneyleri

Arama işlemlerinin farklı başlangıç noktalarından (rastgele olabilir) başlanarak defalarca tekrarlanması, eniyileme süreçlerinde kullanılan etkin metotlardan birisidir. Bu metot sayesinde yerel takılmaların önüne geçilebilmekte ve problem uzayının farklı bölgelerinin aranması sağlanabilmektedir. Tabu-arama algoritması da, başka noktalardan başlatılarak daha iyi sonuçlar elde edilebilecek bir özelliğe sahiptir. Önerdiğimiz paralel algoritma kapsamında Tabu-arama algoritmasının bu özelliğinden de faydalanıldı. Şekil 6'da, tai100a problemi için çoklu başlangıç olmayan, 5 defa ve 50 defa çoklu başlangıç yapılarak elde edilen deney sonuçları verilmiştir. Tek işlemci ile 30 farklı deney sonucunun ortalaması alındığında, 50 defa yeniden başlangıç uygulayan yöntem ile bir defa başlangıç uygulayan yönteme göre %27'lik bir iyileşme elde edilmiştir.



Şekil 6: tai100a problemi üzerinde çoklu başlama metodunun GTA çözüm kalitesine etkileri.

4.2 Paralel-Tabu-KAP algoritmasının QAPLIB bençmarkları ile yapılan deney sonuçları

Bu bölümde, QAPLIB bençmark kütüphanesinde bulunan 134 problem ile yapılan deney sonuçları sunulmuştur. EK'te bulunan Tablo 3'te Paralel-tabu-KAP algoritmasının hatasız olarak çözdüğü problemlerin sapma yüzdeleri (%0.00) ve algoritmanın çalışma süreleri dk. olarak verilmiştir. Bu tabloya bakıldığında, Paralel-tabu-KAP algoritmasının 118 problemi hatasız olarak çözdüğü görülmektedir. Hata yüzdeleri, bençmark kütüphanesinde rapor edilen en iyi çözümlerle karşılaştırılarak bulunmuştur. Örneğin tai25a için kütüphanede en iyi çözüm 100 olarak verildiğinde ve deney sonucunda 100.01 gibi ortalama bir sonuç bulunduğunda (100.01-100)/100 şeklinde bir hesaplama ile %0.01'lik bir sapma rapor edilmiştir. Tüm hesaplamalar bu yöntem ile yapılmıştır.

Hatasız bulunan deney sonuçlarında 100 boyutlu bazı problemlerin dahi çözülebildiği görülmektedir. sko100e, sko100f ve tai100b bu örneklerdendir. Bu büyük problemler için 75 dk. civarında çalışma süreleri gözlemlenmiş olup üssel olarak tüm ihtimallerin deneneceği bir algoritma ile yıllar sürecek çözüm yöntemlerine göre çok daha etkin bir yaklaşım geliştirilmiştir. Diğer orta büyüklükteki problemler için algoritma çalışma sürelerinin 1-5 dk. arasında olduğu görülmektedir. 128 işlemci ile yapılan deneyler esnasında

işlemci sayısı ile doğrusal olarak artan seviyelerde arama yapılabildiği görülmüştür. Diğer bir deyişle, tek işlemci kullanarak GTA ile bir dakikada on milyon deneme yapılabilirken 128 işlemci ile bu oran bir dakikada 128 milyon deneme sayısına erişilmektedir. Paralel-tabu-KAP algoritması bu anlamda paralel algoritmaların önemli bir özelliği olan ölçeklenebilirliğin güzel bir uygulamasıdır.

Stutzle tarafından tanımlanan dört tip KAP problem seti üzerinde yapılan deneyler sonucunda ortalama %0.063'lük bir sapma elde edilmiştir. Tek işlemcili GTA ile yapılan deneylerde bu sapma oranı %0.217 olarak görülmektedir. EK'te Tablo 4, 5, 6 ve 7'de paralel-tabu-KAP algoritmasının bu problemler üzerindeki performansı ve JRG-DivTS, SC-Tabu, CPTS, ITS ve GA/C-TS algoritmaları ile karşılaştırma sonuçları sunulmuştur. Elde edilen sonuçlardan, paralel-tabu-KAP algoritmasının tip-1 ve tip-2 problemleri için literatürdeki en iyi üç algoritmadan birisi olduğu, tip-3 ve tip-4 problemleri için en iyi algoritmalar ile aynı sonuçları elde ettiği görülmektedir. On defa yapılan bu deney sonuçlarına göre önerilen algoritma güvenilirlik seviyesi yüksek ve güçlü bir paralel sezgisel algoritma olarak görülmektedir.

4.3 Hız artışı ve ölçeklenebilirlik analizleri

Önerilen paralel-tabu-KAP algoritması, paralel hesaplama tekniklerinden faydalandığı için, hız artışı ve ölçeklenebilirlik deney sonuçlarının da sunulması faydalı olacaktır. Paralel ortamında çalıştırılan paralel-tabu-KAP algoritması, eklenen işlemci sayısı ile doğru orantılı olarak bir hızlanma sağlamaktadır. Bu değerlendirme yapılırken, en iyi çözümü elde etmek için üretilen yeni çözüm sayısı ve bunların kaç defa hesaplandığı göz önüne alınmıştır. *Master-slave* (efendi-işçi) mimarisine uygun olarak gerçekleştirilen mesajlaşmalar ile 256 işlemciye kadar %5'lik bir iş yükü artışı olduğu ve hesaplamalarda %95'e varan hızlanmalar elde edilmiştir. Daha anlamlı olarak ifade etmek gerekirse, eklenen her işlemci ile birlikte yaklaşık işlemci sayısına doğru orantılı olarak arama sürecinde bir hızlanma olduğu görülmüştür.

KAP için hesaplama karmaşıklığı $O(n^2)$'dir. Önerilen algoritmada, bu hesaplamaları eklenen işlemci sayısı ile doğru orandaki bir miktar kadar daha fazla yapmak mümkün olmaktadır. Örneğin, problem büyüklüğü n=10 olan bir hesaplamada, 100 işlemcinin olduğu bir paralel hesaplama için problem O(n) gibi davranmaktadır. Bu, KAP'ın doğrusal bir hesaplama gibi yapılabilmesi için gerekli olan işlemci adedini göstermektedir.

Eklenen işlemci sayısına bağlı olarak algoritmanın ölçeklenebilirliği efendi-işçi mesajlaşma topolojisine bağlı olarak 256 işlemci için ölçeklenebilir bir seviye kaldığı görülmektedir. İşlemcilerin mesajlaşması her ne kadar ilave bir yük getirmiş olsa da, işlemciler arasında paylaşılan verinin çok büyük olmaması (1KByte seviyelerinde) bu sayıdaki işlemci mimarileri için algoritmanın ölçeklenebilir olduğunu göstermektedir.

5 Sonuçlar

Bu çalışma ile KAP için paralel hesaplama ortamlarında kendi parametrelerini efendi işlemci üzerindeki popülasyon yardımı ile ayarlayan (eniyileyen) ve çoklu başlangıçlar ile dar boğazlara takılmayı engelleyen evrimsel bir paralel Güçlü Tabu-Arama algoritması geliştirilmiştir. Paralel-tabu-KAP algoritması, kendi sınıfındaki sezgisel algoritmalar arasında çalışma zamanı ve elde ettiği çözüm kalitesi açısından QAPLIB problem kütüphanesi üzerinde yapılan deneylere göre en iyi algoritmalardan birisi olarak değerlendirilmektedir. Problemler üzerinde yaklaşık olarak ortalama %0,05'lik bir sapma ile çalışmakta olup, işlemci sayısının artırılabileceği paralel hesaplama ortamlarında ölçeklenebilir tasarımı ile daha iyi sonuçlar bulabilme potansiyeline sahiptir.

Gelecekte yapılacak çalışmalarda, KAP için hiper-sezgisel olarak adlandırılan ve birçok sezgisel algoritma ile aynı problemi CUDA paralel hesaplama ortamında daha etkin olarak çözmeye çalışan bir algoritma geliştirilmesi planlanmaktadır.

6 Teşekkür

Çalışmamıza değerleri fikirleri ile katkıda bulunan hakemlerimize teşekkür ederiz.

7 Kaynaklar

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

[2] Dokeroglu T. "Hybrid teaching-learning-based optimization algorithms for the quadratic assignment problem". *Computers & Industrial*, 85, 86-101, 2015.

[3] Glover, F. "Tabu search part II". *ORSA Journal on Computing*, 2(1), 4-32, 1990.

[4] Lstiburek M, Stejskal J, Misevicius A, Korecky J, El-Kassaby, YA. "Expansion of the minimum-inbreeding seed orchard design to operational scale". *Tree Genetics & Genomes*, 11(1), 1-8, 2015.

[5] Burkard RE, Karisch SE, Rendl F. "QAPLIB a quadratic assignment problem library". *European Journal of Operational Research*, 55(1), 115-119, 1991.

[6] Steinberg L. "The backboard wiring problem: A placement algorithm". *SIAM Review*, 3(1), 37-50, 1961.

[7] Rossin DF, Springer MC, Klein BD. "New complexity measures for the facility layout problem: An empirical study using traditional and neural network analysis". *Computers and Industrial Engineering*, 36(3), 585-602, 1999.

[8] Taillard E. "Robust taboo search for the quadratic assignment problem". *Parallel Computing*, 17(4-5), 443-455, 1991.

[9] Stutzle, T. "Iterated local search for the quadratic assignment problem". *European Journal of Operational Research*, 174(3), 1519-1539, 2006.

[10] Burkard RE, Karisch SE, Rendl F. "QAPLIB a quadratic assignment problem library". *European Journal of OperationalResearch*, 55(1), 115-119, 1991.

[11] Misevicius, A. "An implementation of the iterated tabu search algorithm for the quadratic assignment problem". *OR Spectrum*, 34(3), 665-690, 2012.

[12] James, T., Rego, C., & Glover, F. Multistart tabu search and diversification strategies for the quadratic assignment problem. IEEE Transactions on Systems, Man, and Cybernetics-part a: systems and humans, 39(3), 579-596, 2009.

[13] Fescioglu-Unver N, Kokar MM. "Self controlling tabu search algorithm for the quadratic assignment problem". *Computers & Industrial Engineering*, 60(2), 310-319, 2011.

[14] James T, Rego C, Glover F. "A cooperative parallel tabu search algorithm for the QAP". *European Journal of Operational Research*, 195(3), 810-826, 2009.

[15] Drezner Z. "The extended concentric tabu for the quadratic assignment problem". *European Journal of Operational Research*, 160(2), 416-422, 2005.
[16] Goldberg D. Genetic algorithms in search, optimization and machine learning. 1st edition, New York, NY, USA, Addison-Wesley, 1989.
[17] Türkbey O. "A genetic algorithm using the local search heuristic in facilities layout problem: A memetic algorithm approach". *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 8(2), 265-271, 2002.

EK: Deney sonuçları.

Tablo 3: Optimal çözümleri bulunan QAPLIB problemlerinin istatistiksel veriler.

Problemin adı	En iyi çözüm	Kaç defa buldu	Standart sapma (%)	Zaman (dk.)	Problemin adı	En iyi çözüm	Kaç defa buldu	Standart sapma (%)	Zaman (dk.)
bur26a	5426670	10	0.00	1.4	lipa60a	107218	10	0.00	3.1
bur26b	3817852	10	0.00	1.4	lipa60b	2520135	10	0.00	3.1
bur26c	5426795	10	0.00	1.4	lipa70a	169755	10	0.00	3.6
bur26d	3821225	10	0.00	1.4	lipa70b	4603200	10	0.00	3.6
bur26e	5386879	10	0.00	1.4	lipa80a	253195	10	0.00	4.1
bur26f	3782044	10	0.00	1.4	lipa80b	7763962	10	0.00	4.1
bur26g	10117172	10	0.00	1.4	lipa90a	360630	10	0.00	4.6
bur26h	7098658	10	0.00	1.4	lipa90b	12490441	10	0.00	4.6
chr12a	9552	10	0.00	0.7	nug14	1014	10	0.00	0.8
chr12b	9742	10	0.00	0.7	nug15	1150	10	0.00	0.9
chr12c	11156	10	0.00	0.7	nug16a	1610	10	0.00	0.9
chr15a	9896	10	0.00	0.9	nug16b	1240	10	0.00	0.9
chr15b	7990	10	0.00	0.9	nug17	1732	10	0.00	1.0
chr15c	9504	10	0.00	0.9	nug18	1930	10	0.00	1.0
chr18a	11098	10	0.00	1.0	nug20	2570	10	0.00	1.1
chr18b	1534	10	0.00	1.0	nug21	2438	10	0.00	1.2
chr20a	2192	10	0.00	1.0	nug22	3596	10	0.00	1.2
chr20b	2298	10	0.00	1.1	nug24	3488	10	0.00	1.3
chr20c	14142	10	0.00	1.1	nug25	3744	10	0.00	1.4
chr22a	6156	10	0.00	1.2	nug27	5234	10	0.00	1.4
chr22b	6194	10	0.00	1.2	nug28	5166	10	0.00	1.5
chr25a	3796	10	0.00	1.4	nug30	6124	10	0.00	1.6
els19	17212548	10	0.00	1.1	rou12	235528	10	0.00	0.6
esc16a	68	10	0.00	0.9	rou15	354210	10	0.00	1.1
esc16b	292	10	0.00	0.9	rou20	725522	10	0.00	1.1
esc16c	160	10	0.00	0.9	scr12	31410	10	0.00	0.7
esc16d	16	10	0.00	0.9	scr15	51140	10	0.00	0.9
esc16e	28	10	0.00	0.9	scr20	110030	10	0.00	1.1
esc16f	0	10	0.00	0.9	sko42	15812	10	0.00	2.2
esc16g	26	10	0.00	0.9	sko49	23386	10	0.00	2.9
esc16h	996	10	0.00	0.9	sko56	34458	10	0.00	2.9
esc16i	14	10	0.00	0.9	sko64	48498	10	0.00	3.3
esc16j	8	10	0.00	0.9	sko72	66256	10	0.00	3.7
esc32a	130	10	0.00	1.7	sko81	90998	10	0.00	4.2
esc32b	168	10	0.00	1.7	sko90	115534	10	0.00	4.6
esc32c	642	10	0.00	1.7	sko100e	149150	10	0.00	76.0
esc32d	200	10	0.00	1.7	sko100f	149036	10	0.00	76.0
esc32e	2	10	0.00	1.7	ste36a	9526	10	0.00	1.9
esc32f	2	10	0.00	1.7	ste36b	15852	10	0.00	1.9
esc32g	6	10	0.00	1.7	ste36c	8239110	10	0.00	1.9
esc32h	438	10	0.00	1.7	tai12a	224416	10	0.00	0.7
esc64	116	10	0.00	3.3	tai12b	39464925	10	0.00	0.7
esc128	64	10	0.00	6.7	tai15a	388214	10	0.00	0.9
had12	1652	10	0.00	0.7	tai15b	51765268	10	0.00	0.9
had14	2724	10	0.00	0.8	tai17a	491812	10	0.00	1.0
had16	3720	10	0.00	0.9	tai20a	703482	10	0.00	1.1
had18	5358	10	0.00	1.0	tai20b	122455319	10	0.00	1.1
had20	6922	10	0.00	1.1	tai25a	1167256	10	0.00	1.4
kra30a	88900	10	0.00	1.6	tai25b	344355646	10	0.00	1.3
kra30b	91420	10	0.00	1.6	tai30a	1818146	10	0.00	1.6
kra32	88700	10	0.00	1.7	tai30b	637117113	10	0.00	1.9
lipa20a	3683	10	0.00	1.1	tai35a	2422002	10	0.00	1.9
lipa20b	27076	10	0.00	1.1	tai35b	283315445	10	0.00	1.9
lipa30a	13178	10	0.00	1.6	tai40b	637250948	10	0.00	2.1
lipa30b	151426	10	0.00	1.6	tai50b	458821517	10	0.00	2.1
lipa40a	31538	10	0.00	2.1	tai60b	608215054	10	0.00	3.1
lipa40b	476581	10	0.00	2.1	tai64c	1855928	10	0.00	3.4
lipa50a	62093	10	0.00	4.1	tai80b	818415043	10	0.00	14.2
lipa50b	1210244	10	0.00	4.1	tai100b	1185996137	10	0.00	75.2

Tablo 4: Önerilen paralel tabu-KAP algoritmasının tip-1 KAP üzerinde güncel en iyi algoritmalar ile karşılaştırılması.

		JRG-DivTS		SC-Tabu		CPTS	ITS	Paralel-tabu-KAP	
Problemin Adı	En iyi Çözüm	Standart sapma (%)	Zaman (dk.)	Standart sapma (%)	Zaman (sn.)	Standart sapma (%)	Standart sapma (%)	Standart sapma (%)	Zaman (dk.)
tai20a	70382	0.00	0.2	0.246	0.001	0.00	0.00	0.00	1.1
tai25a	1167256	0.00	0.2	0.239	0.03	0.00	0.00	0.00	1.3
tai30a	1818146	0.00	1.3	0.154	0.07	0.00	0.00	0.00	1.6
tai35a	2422002	0.00	4.4	0.280	0.18	0.00	0.00	0.00	1.8
tai40a	3139370	0.222	5.2	0.561	0.20	0.148	0.220	0.270	30.3
tai50a	4941410	0.725	10.2	0.889	0.23	0.440	0.410	0.470	37.6
tai60a	7205962	0.718	25.7	0.940	0.41	0.476	0.450	0.480	45.6
tai80a	13499184	0.753	52.7	0.648	1.00	0.570	0.360	0.540	60.7
tai100a	21059006	0.825	142.1	0.977	1.99	0.558	0.300	0.389	75.6
Ortalama Sonuçlar		0.360	26.88	0.548	0.45	0.243	0.193	0.238	28.4

Tablo 5: Önerilen paralel tabu-KAP algoritmasının tip-2 KAP üzerinde güncel en iyi algoritmalar ile karşılaştırılması.

		JRG-DivTS		GA/C-TS		CPTS	Paralel-tabu-KAP	
Problemin adı	En iyi çözüm	Standart sapma (%)	Zaman (dk.)	Standart sapma (%)	Zaman (dk.)	Standart sapma (%)	Standart Sapma (%)	Zaman (dk.)
sko42	15812	0.00	4.0	0.00	1.2	0.00	0.00	2.2
sko49	23386	0.08	9.6	0.009	2.1	0.00	0.00	2.6
sko56	34458	0.02	13.2	0.001	3.2	0.00	0.00	2.9
sko64	48498	0.00	22.0	0.00	5.9	0.00	0.00	3.4
sko72	66256	0.006	38.0	0.014	8.4	0.00	0.00	3.7
sko81	90998	0.016	56.6	0.014	13.3	0.00	0.00	4.2
sko90	115534	0.026	89.6	0.011	22.4	0.00	0.00	4.6
sko100a	152002	0.027	129.2	0.018	33.6	0.00	0.047	76.2
sko100b	153890	0.008	106.6	0.011	34.1	0.00	0.056	76.2
sko100c	147862	0.006	126.7	0.003	33.8	0.00	0.043	76.2
sko100d	149576	0.027	123.5	0.049	33.9	0.00	0.044	76.2
sko100e	149150	0.009	108.8	0.002	30.7	0.00	0.00	76.3
sko100f	149036	0.023	110.3	0.032	35.7	0.003	0.00	76.5
Ortalama sonuçlar		0.012	72.1	0.013	19.9	0.00	0.015	37.0

Tablo 6: Önerilen paralel tabu-KAP algoritmasının tip-3 KAP üzerinde güncel en iyi algoritmalar ile karşılaştırılması.

		JRG-DivTS	ITS	GA-C/TS	SC-TABU	Paralel-tabu-KAP	
Problemin adı	En iyi çözüm	Standart sapma (%)	Standart sapma (%)	Standart sapma (%)	Standart sapma (%)	Standart sapma (%)	Zaman (dk.)
kra30a	88900	0.00	0.00	-	0.714	0.00	1.6
kra30b	91420	0.00	0.00	0.00	0.178	0.00	1.6
kra32	88700	0.00	-	0.00	-	0.00	1.7
ste36a	9526	0.00	0.04	-	-	0.00	1.9
ste36b	15852	0.00	0.00	0.005	-	0.00	1.9
ste36c	8239110	0.00	0.00	0.00	-	0.00	1.9
esc32b	168	0.00	0.00	0.039	-	0.00	1.7
esc32c	642	0.00	0.00	0.00	-	0.00	1.7
esc32d	200	0.00	0.00	0.00	-	0.00	1.7
esc32e	2	0.00	0.00	0.00	-	0.00	1.7
esc32g	6	0.00	0.00	-	-	0.00	1.7
esc32h	438	0.00	0.00	-	-	0.00	1.7
esc64a	116	0.00	0.00	0.00	-	0.00	3.3
esc128	64	0.00	0.01	0.00	-	0.00	6.7
Ortalama sonuçlar		0.00	0.00	0.004	0.446	0.00	2.2

Tablo 7: Önerilen paralel tabu-KAP algoritmasının tip-4 KAP üzerinde güncel en iyi algoritmalar ile karşılaştırılması.

		JRG-DivTS		ITS		CPTS	SC-TABU	Paralel-tabu-KAP	
Problemin adı	En iyi çözüm	Standart sapma (%)	Zaman (dk.)	Standart sapma (%)	Zaman (sn.)	Standart sapma (%)	Standart sapma (%)	Standart sapma (%)	Zaman (dk.)
tai20b	122455319	0.00	0.2	0.00	0.01	0.00	0.00	0.00	1.1
tai25b	344355646	0.00	0.5	0.00	0.01	0.00	0.07	0.00	1.4
tai30b	637117113	0.00	1.3	0.00	0.01	0.00	0.00	0.00	1.6
tai35b	283315445	0.00	2.4	0.02	0.08	0.00	0.059	0.00	1.9
tai40b	637250948	0.00	3.2	0.01	02	0.00	0.00	0.00	2.1
tai50b	458821517	0.00	8.8	0.02	0.5	0.00	0.002	0.00	3.1
tai60b	608215054	0.00	17.1	0.04	1.7	0.00	0.00	0.00	3.3
tai80b	818415043	0.006	58.2	0.23	3.0	0.001	0.003	0.00	4.1
tai100b	1185996137	0.056	118.9	0.14	6.66	0.00	0.014	0.00	76.2
Ortalama sonuçlar		0.070	23.4	0.051	1.35	0.00	0.009	0.00	10.5