



Makespan and energy based virtual machine scheduling in cloud systems

Alper Kızıllı^{1*}, Korhan Karabulut²

¹Department of Computer Engineering, Faculty of Engineering, Yasar University, 35100, Bornova, Izmir, Türkiye

²Department of Software Engineering, Faculty of Engineering, Yasar University, 35100, Bornova, Izmir, Türkiye

Highlights:

- Experiment-based energy model
- Tradeoff between energy and time metrics
- Effects of server load on both energy and time metrics

Keywords:

- Energy Consumption
- Makespan
- Genetic Algorithm
- Task Scheduling

Article Info:

Research Article

Received: 15.11.2022

Accepted: 25.08.2023

DOI:

10.17341/gazimmfd.1189807

Correspondence:

Author: Alper Kızıllı

e-mail:

alper.kizil@yasar.edu.tr

phone: +90 5397 657 992

Graphical/Tabular Abstract

Cloud computing is a newly emerging computing concept that is only strengthened by the advent of worldwide internet infrastructure. Although, these datacenters are giant, have a limited set of resources and potentially infinite demand for these resources and these resources needed to be mapped that is most cost effective and time efficient. This problem is known in literature as cloud resource scheduling problem, and it is proven to be NP-Hard. This study proposes two genetic algorithm-based solvers that will work on the VM Task assignment level and try to optimize makespan and energy consumption while assigning these user tasks and tries to determine if there is a tradeoff between these two metrics in different host loads (different number of VMs per host machine). Figure A explains how the proposed system is going to work. Users in the system submit their jobs in the form of tasks. These tasks are then assigned to virtual machines running on physical server machines by the proposed 2 genetic algorithms.

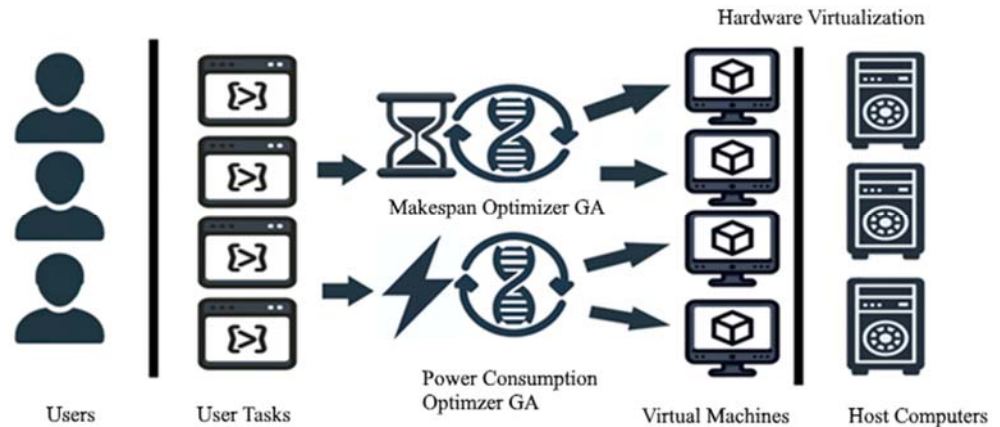


Figure A. General layout of cloud scheduling problem & GA Task Assignment solvers

Purpose: To find if there is a tradeoff between energy and makespan metrics and how much host load affects those metrics while scheduling user tasks on virtual machines in a cloud setting.

Theory and Methods: An energy model is devised based on running Cinebench R23 rendering workload on AMD Ryzen 2700X Zen+ processor and obtaining power consumption based on how it is reported by CPU to the operating system by disabling DVFS, setting voltage fixed voltage and alternating CPU clock frequency between 600 – 3400 MHz values and using 3200Mhz frequency as a baseline, two GA variants are developed. One is optimizing makespan while other optimizing energy consumption on the said datacenter while scheduling the user tasks, all the while alternating number of virtual machines per host, thus changing the host load.

Results: Makespan and energy consumption metrics are closely related and there does not seem to be a visible tradeoff between the two metrics. In the full or near full host load, it is found that energy consumption optimizer GA performs significantly better (10-13%) compared to makespan optimizer in same scenarios.

Conclusion: Energy optimizer algorithm performed significantly better in the experiment scenario. While this is not a conclusive evidence, thus further experiments are required, in general host load has a direct effect on comparing two optimizers.



Bulut sistemlerinde toplam tamamlanma ve enerji tabanlı sanal makine çizelgelemesi

Alper Kızıl^{1*}, Korhan Karabulut²

¹Yaşar Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 35100, Bornova, İzmir, Türkiye

²Yaşar Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, 35100, Bornova, İzmir, Türkiye

Ö N E Ç İ K A N L A R

- Deneysel tabanlı bir enerji modeli
- Enerji ve zaman ölçütleri arasındaki ödünleşim
- Sunucu yoğunluğunun enerji / toplam süre ölçütlerine etkisi

Makale Bilgileri

Araştırma Makalesi

Geliş: 15.11.2022

Kabul: 25.08.2023

DOI:

10.17341/gazimmfd.1189807

Anahtar Kelimeler:

Enerji tüketimi,
toplam bitiş süresi,
evrimsel algoritma,
sanal makine çizelgeleme

Ö Z

Bulut bilişim, internet altyapısının son yıllarda dünya çapında gelişmesiyle önem kazanmış yeni bilişim paradigmalardandır. Hem büyük şirketlere hem de gelişme aşamasındaki küçük ve orta ölçekli şirketlere hem de bağımsız geliştiricilere kendi donanım altyapılarını güncel ve işler tutma maliyetlerinden kurtaran, kaynak kullanımında esneklik sağlayan, veri kayıplarını minimize edebilmeleri için ek olanak sağlayan bir teknolojidir. Gelecekte de bulut bilişime olan talebin artacağı ortadadır. Bu tür veri merkezleri yapıları gereği oldukça yüklü miktarda enerji tüketimi yapmaktadırlar, bu tüketimde yapılacak en küçük tasarruflar bile bu devasa merkezlerin çok önemli miktarda enerji tasarrufu yapmalarını sağlayacaktır. Literatüre baktığımızda da Yeşil bilişim (Green Computing) yıldan yıla gitgide önem kazanmaktadır. Bulut Çizelgeleme problemi, kısıtlı sayıda bulut kaynağının teoride sınırsız sayıda olabilecek kullanıcı talebine en uygun, en iyi çözümün bulunmasını amaçlayan NP-Zor olduğu kanıtlanmış bir problemdir. Bu çalışmada, öncelikle bir sunucu bilgisayarı için deneye dayalı bir iş yükü / güç tüketimi modeli önerilmiş, sonra da toplam bitiş süresi ve enerji tüketimi eniyileme yapan iki genetik algoritma, farklı sunucu yüklerinde bu ölçütler üzerinden kıyaslanmıştır. Sonuçta bu iki kriterin birbirleriyle yakın ilişkide olduğu görülmüş, ayrıca enerji kriterini eniyilemenin tam ya da tama yakın sunucu yüklerinde, zaman kriteri eniyilemeye göre %10 – %13 arasında daha olumlu bir etkisi olduğu saptanmıştır. Bu sayede, yüksek sunucu yüklerinde, enerji eniyilemenin amaç fonksiyonu olarak kullanılmasını ile ciddi oranda enerji tasarrufunun mümkün olabileceği gösterilmiştir.

Makespan and energy based virtual machine scheduling in cloud systems

H I G H L I G H T S

- An experiment-based energy model
- Tradeoff between energy and time metrics
- Effects of server load on time and energy metrics

Article Info

Research Article

Received: 15.11.2022

Accepted: 25.08.2023

DOI:

10.17341/gazimmfd.1189807

Keywords:

Energy consumption,
makespan,
evolutionary algorithm,
virtual machine scheduling

ABSTRACT

Cloud computing is one of the newest computing paradigms that emerged after worldwide development of Internet infrastructure. It is a technology that saves both large companies and small and medium scale companies as well as independent developers from the cost of keeping infrastructure hardware up to date, and operational while also providing flexibility on resource use as well providing additional opportunity to minimize data losses. While in the future, it is evident that demand for cloud computing will be on the rise. These kinds of datacenters, due to their nature, consume large amounts of energy and even the savings on smallest scales will enable these gigantic centers to save a significant amount of energy in total. If we have a look at the literature, we can see green computing is gaining immense popularity over the years. The Cloud Scheduling problem is a proven problem to be NP-Hard, aiming to find the best solution for a limited number of cloud resources, which could theoretically be serving an unlimited number of user demands. In this study, firstly, an experimental workload / power consumption model is proposed for a server computer, and then two genetic algorithms optimizing makespan and energy consumption are compared on these metrics at different server loads. As a result, it has been seen that these two criteria are closely related to each other, and it has been determined that optimizing the energy criterion has a more positive effect between 10% and 13% compared to the time criterion optimization at full or near full server loads. In this way, it has been shown that significant energy savings can be achieved by using energy optimization as an objective function at high server loads.

1. Giriş (Introduction)

Bulut bilişim, günümüzde farkında olarak ya da olmadan internete erişimi olan her bireyin kullandığı en önemli bilgi teknolojilerindedir. Birçok şirket, kendisine ait bir bilgi işlem altyapısı bulundurur ancak bu bilgi işlem altyapısını yaptığı iş hacmine göre ölçeklendirmek, bu altyapının kurulumu ve bakımını yapacak personel bulundurmak, yetiştirmek ve bulunduğu ülkedeki kanuni düzenlemelere uygun, kullanıcı talebini karşılayabilir halde tutmak ciddi bir maliyet getirmektedir. Tahmin edilebildiği gibi bu büyük şirketler için daha kolay idare edilebilir bir sorun iken, daha küçük çaplı, startup dediğimiz başlangıç aşamasındaki şirketler için ciddi bir problemdir. Bulut bilişim, startup ve daha büyük kapsamlı şirketlerin, bireysel uygulama geliştiricilerin bu problemine kiralama yöntemiyle çözüm bulma hedefiyle ortaya çıkmıştır.

Amazon Web Services 2006 yılında ilk çıkan bulut bilişim servis sağlayıcısı olmuş, bunu 2008 yılında Microsoft Azure ve Google Cloud daha sonraları IBM ve Oracle izlemiştir. Ayrıca Çin şirketleri Tencent ve Alibaba bu pazardaki diğer belirli başlı şirketlerdir. Bununla birlikte, Amazon, Microsoft ve Google üçlüsü piyasanın %65'ine hükmetmektedir. [1]

Bulut bilişim, sunucu, depolama, ağ, veri tabanı, işlem gücü, uygulama geliştirme platformu ve yazılımların internet altyapısı üzerinden kullanıcının talebine göre ön tanımlı bir fiyatlandırma şeması üzerinden sunulmasını ifade eder. Bulut bilişim servis sağlayıcıları bu gerekli altyapıyı dünyanın çeşitli bölgelerinde kurarak kiralamaya açarlar ve servise üye olan kullanıcılar talep ettikleri altyapıyı bir web uygulaması üzerinden belirleyerek kiralarlar.

Bulut bilişimin kullanıcı açısından en önemli avantajı maliyetlerdeki azalmadır. Şirketler, donanım ve yazılım satın alma, bu donanım ve yazılımın kurulumu, işletilmesi, işletilirken harcanacak elektrik ve soğutma maliyeti ve ayrıca bu kurulum ve bakımı yapacak teknik personel maliyetinden tasarruf ederler.

Ayrıca yine bulut bilişim servislerinin sunduğu "Disaster Recovery" (Felaket Kurtarma) araçları, verilerin çok sayıda uzak konumda yedeklenebilmesini ve bu yedeklerin gerektiğinde geri yüklenilebilmesini sağlar.

Ölçeklendirilebilme de bu tarz bir hizmetin sunduğu diğer bir artıdır. Altyapıyı servisin en yoğun kullanılacağı kısa zaman dilimine göre kurmak bu altyapının diğer zamanlarda atıl kalmasına sebep olur. Bunun yerine bulut servis sağlayıcılar ihtiyaç kadar kaynak kiralamaya izin verirler. Böylece talep yüksek olduğunda kiralanan

kaynak miktarı arttırılabilir, talep tekrar düştüğünde de kiralanan kaynak tekrar iade edilebilir.

Bir diğer avantaj da bu tarz veri merkezlerinde donanım çok sık güncellendiği için kullanıcıya her zaman en iyi performansı garanti etmeleridir.

Bunların yanında bu teknolojinin dezavantajları da bulunmaktadır. Bunlardan en önemlisi veri güvenliğidir. Bu tarz bir bulut bilişim sistemi kullanıldığında şirket verileri üçüncü bir taraf ile paylaşılmaktadır. Bu yüzden banka ve devlet kurumları daha maliyetli ve daha uğraştırıcı olmasına rağmen kendi bilgi teknoloji altyapılarını ve bunların uzak sunucu yedeklerini kurarlar.

Ayrıca donanıma ve zaman zaman da yazılıma müdahale olanağı olmadığı için yaşanabilecek problemleri kullanıcı kendi çözemeyebilir. Böyle durumlarda bulut bilişim servisi ile iletişime geçmek gerekir.

Amazon, Microsoft ve Google gibi şirketlerin üçüncü parti kişi ve şirketlerin hizmetine açtığı bu servisler Umumi (Public) Bulut servisleri olarak adlandırılırlar. Eğer bir şirket sadece kendi alt şirketlerine ya da kendi müşterilerine hizmet vermek amaçlı bir bulut bilişim altyapısı kurduysa buna da Hususi (Private) Bulut bilişim servisi olarak adlandırılır. Yine eğer bir şirket operasyonlarını yürütmek için bu iki hizmeti birleştiriyorsa bu da Karma (Hybrid) Bulut bilişim olarak ifade edilir.

Bulut Bilişim servisleri temel olarak 3 farklı model olarak hizmet vermektedirler.

1.1. Bir Servis olarak Altyapı (Infrastructure as a Service- IaaS)

Alt seviye donanım ve yazılımın Bulut Bilişim servis sağlayıcı tarafından, üst katmandaki esas uygulama, bu uygulamaya ait veri ve ayrıca uygulama altyapısının (Framework & Library) kullanıcı tarafından yürütüldüğü modeldir. IaaS modeli Şekil 1'de gösterilmiştir.

Bulut Bilişim servisi üzerinde oluşturulan sanal makineler (SM) bu kategoriye girmektedir.

1.2. Bir Servis olarak Platform (Platform as a Service- PaaS)

Donanım altyapısı yanında, yazılım altyapısı (framework, programlama dili, kütüphaneler, araçlar vb.) de bulut servis sağlayıcısı tarafından sağlanıyorsa, bu modele PaaS modeli



Şekil 1. Infrastructure as a Service Modeli (Infrastructure as a Service Model)

denilmektedir. Bu model genelde mobil ve web tabanlı uygulama geliştirmede kullanılmaktadır. Kullanıcı sadece uygulaması ve bu uygulamanın oluşturduğu verilerin saklanmasıyla ilgilenir. Google App Engine, Amazon AWS Beantalk bu servisin tipik örnekleridir. PaaS modeli Şekil 2’de gösterilmiştir.

1.3. Bir Servis Olarak Yazılım (Software as a Service- SaaS)

Kullanıcının yazılım geliştirme kısmıyla ilgilenmediği, sadece bulut servis sağlayıcıdan ücreti karşılığında bir çevrimiçi hizmet satın aldığı modeldir. Google’ın G Suite hizmeti ve Microsoft’un Office 365 hizmetleri bu modele tipik birer örnektir. SaaS modeli Şekil 3’de verilmiştir.

1.4. Bir Servis Olarak Fonksiyon (Function as a Service- FaaS)

Geleneksel yaklaşımlardan farklı olarak, enerji tüketimi ve maliyeti azaltmanın bir yolu da kullanılmayan kaynakların devre dışı bırakılmasıdır. Sunucusuz hesaplama (Serverless computing) modeli olarak da geçen bu modelde, diğer geleneksel servis yöntemlerinden farklı olarak, geliştirici, kodunu durumsuz (stateless) yani bir kaynağa statik olarak bağlanmayacak şekilde bir konteyner içerisinde aktarır. Bu konteyner, belli koşullara bağlı olarak, bir kaynağa gerektiğinde dinamik bir şekilde bağlanır. Bu koşullar bir veri tabanı olayı, sunucuya dışarıdan gelen bir istek (kullanıcının programda bir butona tıklaması gibi), sunucuya bir dosya alışverişi ya da çizelgelenmiş bir olay olabilir. Bu modelin en büyük artışı, geliştiricinin uygulaması için gereken kaynağı önceden tahmin etmek zorunda olmaması ve bulut sağlayıcı üzerinde faturalandırmanın tam olarak kullanılan kaynak miktarı üzerinden yapılmasıdır. Genellikle Bir servis olarak Yazılım modeliyle ya da Bir Servis olarak Platform modelleriyle birlikte önerilmektedir. Bu tarz servisin örneği olarak Amazon AWS Lambda, Azure Serverless ya da Google Cloud Functions verilebilir.

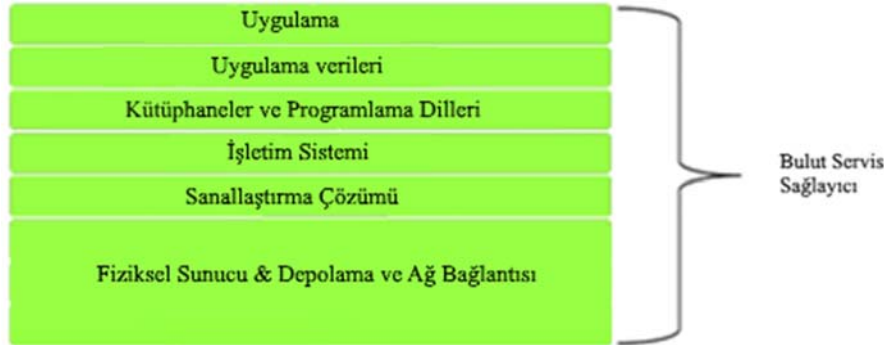
Kullanıcı hangi servis modelini seçerse seçsin bir bulut sağlayıcısı müşterilere daima ancak kurulumunu ve işletmesini yaptığı sınırlı bir kaynak sunabilir, bunun yanında müşterilerden bu bulut kaynaklarına potansiyel olarak sınırsız bir talep olabileceği göz önüne alınırsa bu kaynakların verimli kullanılmasının ne kadar önemli olduğu ortaya çıkmaktadır. Bu tarz veri merkezlerinde verimlilik ölçütlerinde de en başta enerji verimliliği gelmektedir.

2016 yılında yayınlanmış Amerikan Enerji Bakanlığı ve Berkeley Ulusal laboratuvarı ortak raporuna göre [2], bulut bilişim veri merkezleri yüksek seviye (Hyperscale) veri merkezleri olarak tanımlanmaktadır. Bu tip veri merkezleri 35 – 40 km² ve üzeri merkezler olarak sınıflandırılmışlardır. Raporunda ayrıca bu tip veri merkezlerinin enerji verimliliği en yüksek veri merkezleri olduğu ve veri merkezi alanındaki 2010 yılından itibaren neredeyse bütün büyümenin bu merkezler sayesinde olduğu kaydedilmiştir. Raporunda 2014 yılında ABD’de veri merkezlerinin elektrik kullanımının 70 milyar kWh olduğu, bunun bütün ABD’nin aynı yıldaki elektrik kullanımının %1,8’ine denk geldiği vurgulanmış ve 2020 yılında tahmini 73 milyar kWh enerji harcayacağı tahmin edilmiştir ve enerji verimliliği yüksek bu veri merkezlerine yönelimin önemi vurgulanmıştır.

Dünyanın gün geçtikçe artan bilişim ihtiyacı da düşünüldüğünde, bilgisayar bilimlerinde de “Green Computing” ya da Yeşil Hesaplama son zamanlarda bir hayli önem kazanmıştır. Clarivate Web of Science [3] veri tabanına baktığımızda bilgisayar ve bilişim temalı dergiler ve konferanslarda bu alanlarda yapılan yayınların gün geçtikçe arttığını görmekteyiz. “Green Computing”, “Power consumption”, “Energy Saving”, “Energy Consumption” anahtar kelimeleriyle arama yapıldığında bu durum açıkça görülmektedir. 2005 yılında bu anahtar kelimelerle sadece 2132 yayın eklenmişken, 2019 yılında 8634 yayınlara ulaşıldığını görmekteyiz. Yine aynı veri tabanında “Green Computing” ile ilgili ilk yayınların 2008 yılında eklendiğini



Şekil 2. Platform as a Service Modeli (Platform as a Service Model)



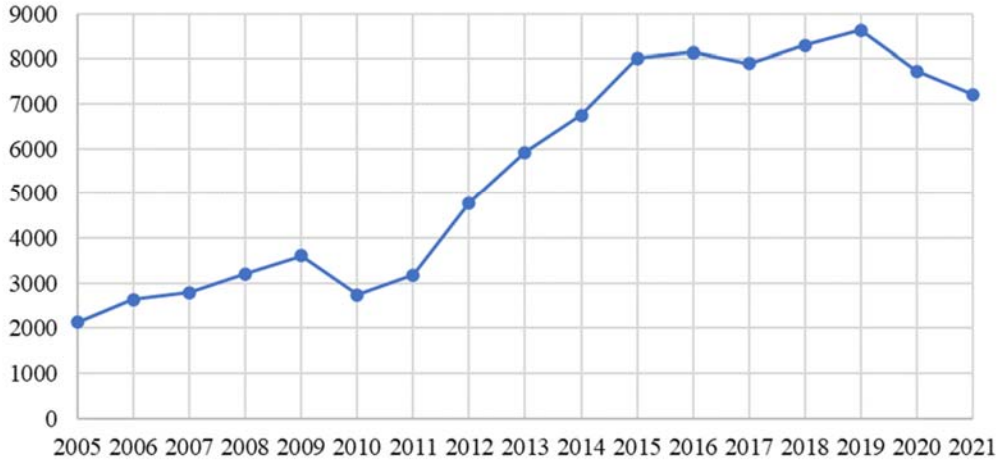
Şekil 3. Software as a Service Modeli (Software as a service Model)

ve bu konuda sadece 4 yayın olduğu ve 2018 yılına geldiğimizde toplam 467 yayın yapıldığını görmekteyiz. Literatürdeki bu çalışmalar, en büyük veri merkezlerindeki enerji tüketiminin düşürülmesinden [4], en küçük çapta nesnelerin interneti cihazlarına [5] ve mikro PC'lere [6] kadar geniş bir yelpazeyi kapsamaktadır. Küresel ısınma, elektronik cihazların sayısındaki artış gibi etkilerin de bu tür çalışmaları arttıracığı ve Yeşil hesaplamının daha önem kazanacağı açıktır. Bu durumu anlatan çalışmalara da [7 – 8] örnek olarak verilebilir. Şekil 4 ve Şekil 5, belirtilen anahtar kelimeler kullanılarak Web of Science veri tabanında yapılan aramalarda çıkan toplam yayın ve makale sayılarını belirtmektedir.

2. Problem Tanımı (Problem Definition)

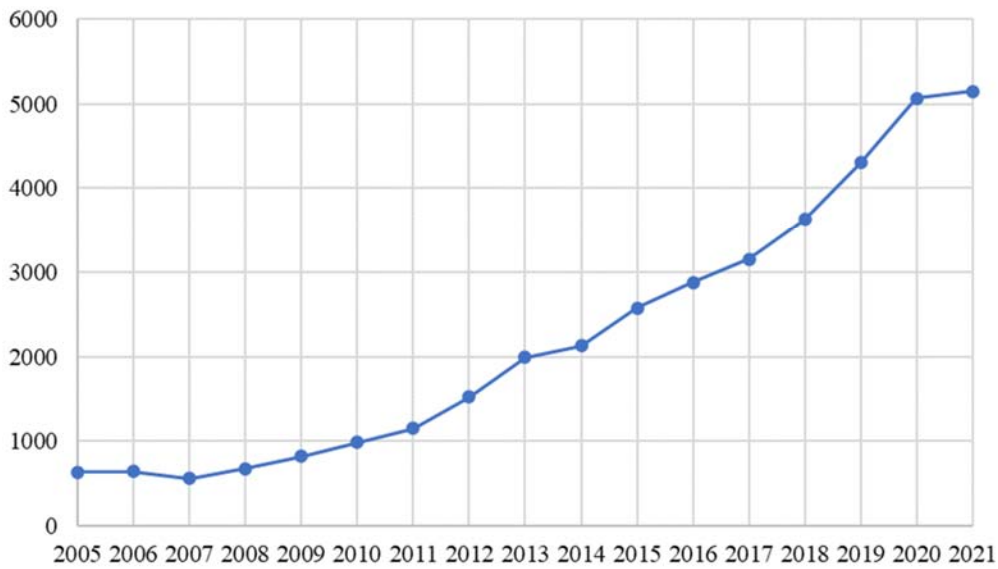
Bulut bilişim, sınırlı kaynak (CPU / Fiziksel Makine / Ağ / Depolama / Veri tabanı / Uygulama olarak ifade edilebilir) ve bu kaynağa karşılık gelecek talebin önceden bilinmemesi sebebiyle kaynakların verimli kullanılmasını zorunlu kılmaktadır. Bu yüzden bu arayış literatürde ciddi bir araştırma konusu yaratmıştır. Bu sınırlı sayıdaki kaynağın müşterilere adil ve yeterli dağıtılabilmesi problemi literatürde Bulut Kaynak Çizelgelemesi ("Cloud Resource Scheduling") problemi olarak geçmektedir. Bu problemde amaç M adet kullanıcı işinin, N adet mevcut Bulut kaynağına Z adet uygunluk kriteri çerçevesinde

Clarivate Web of Science Veritabanındaki Yayınlar (Makale + Konferans yayınlarının toplam sayılarının yıldan yıla değişimi)



Şekil 4. Clarivate [3] Web of Science Veri tabanında kayıtlı Enerji temalı Bilgisayar Bilimleri Yayınları Toplamı (Total number of publications based on energy related theme in Computer Science in Clarivate Web Index)

Clarivate Web of Science Veritabanındaki Yayınlar (Makale sayılarının yıldan yıla değişimi)



Şekil 5. Clarivate [3] Web of Science Veri tabanında kayıtlı Enerji temalı Bilgisayar Bilimleri Dergi Makaleleri (Number of journal articles with energy theme in computer Science journals in Clarivate web index)

atanması şeklinde ifade edilir. Bu çalışmada bulut kaynağı olarak fiziksel Makineler (FM), kullanıcı işleri olarak da sanal makineler ve bu sanal makineler üzerinde çalıştırılacak kullanıcı işleri ("Task" ya da Bilgisayar Programları) ele alınacaktır. Uygunluk kriteri olarak da işlerin toplam tamamlanma süresi (makespan) ve Kilovat – saat cinsinden harcanan enerji olarak ele alınmıştır. Bulut çizelgeleme problemi Şekil 6'daki gibi görselleştirilebilir. Şekil 6'da verilen kısaltmaların açıklamaları Tablo 1'de verilmektedir.

Makalenin geri kalanında kullanılacak semboller Tablo 1'de verilmiştir.

Tablo 1. Bulut Çizelgeleme Problemi Değişkenleri (Cloud scheduling problem variables)

Sembol	Tanımı
u	Kullanıcı görevleri
w	Toplam görev sayısı
sm	Sanal makine
m	Toplam Sanal Makine Sayısı
fm	Fiziksel makine (Sunucu)
n	Fiziksel makine (Sunucu) sayısı
Z	Eniyileme kriteri
i	Toplam eniyileme kriteri sayısı

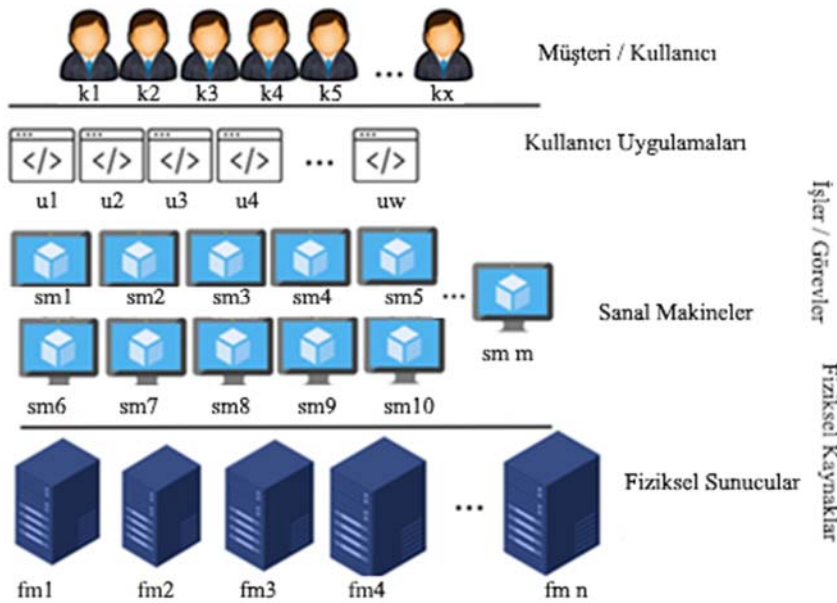
Bu problem NP-zor olduğu kanıtlanmış bir problemdir [9] ve var olan algoritmalar yüksek boyutlu problem örneklerini çözmekte verimsiz kalmaktadır. Ayrıca çizelgeleme problemi doğası gereği dinamik yapıda, yani yeni görevler geldikçe atama gerektiriyor da olabilir. Bu yüzden ilk uygun (first fit) veya en uygun (best fit) gibi algoritmalar ve ilk olarak en kısa iş (shortest job first) ya da ilk olarak en uzun iş (longest job first) gibi algoritma kombinasyonları kabul edilebilir çözümler bulabilirlerken, stokastik çözümleyiciler de literatürde bu alandaki yüksek boyutlu problem örneklerini çözmek için kullanılmışlardır. Bu stokastik çözümleyicilerden en popülerlerinden bir tanesi de genetik algoritmalarıdır. Genetik algoritmalar, farklı problemlerde başarısı kanıtlanmış, iyi sonuçlar verebildiği literatürde çokça gösterilmiş algoritmalarıdır. Genetik algoritmalar, John Henry Holland tarafından ilk defa tanımlandığı 1975 yılından bu yana, literatürde hem teorik problemler hem de gerçek hayat problemlerinin

çözümünde sıklıkla kullanılmıştır. Teorik problem çözümlerine birkaç örnek, çok boyutlu sırt çantası problemi [10] (multidimensional knapsack problem), araç rotalama problemi [11] (vehicle routing problem), atölye çizelgeleme problemi [12] (job shop scheduling), karesel atama problemi [13] (quadratic assignment problem), kutulama problemi [14] (bin packing problem), gezgin satıcı problemi [15] gibi problemler verilebilir.

Genetik algoritmalar ayrıca gerçek hayat problemlerinin çözümlerinde de sıkça kullanılmışlardır, bu çözümlere örnek olarak, moleküler biyolojide şekil tabanlı ilaç üretimi [16], bilgisayar grafiklerinde, 3d modellemeye optimal kamera yerleşimi bulma [17], insan hareket tanımlama problemi için oluşturulan derin öğrenme algoritmasında kullanılacak özelliklerin seçilmesi için [18], inşaat mühendisliğinde, su dağıtım şebekesinde kurulum maliyet optimizasyonu [19], elektrik- elektronik mühendisliğinde, elektrikli arabaların süperkapasitörlerinin kalan ömürlerinin tahminlenmesi [20], kısa dönemli rüzgar santrali güç üretimi tahminlenmesi [21], elektrik şebekesi dizaynı [22], tıp alanında, bireylerde kalp hastalıklarını tahminleyen bir yapay sinir ağını optimize etmede [23], yaşlılarda femur kemiğindeki kırıkları tespit eden evrişimli sinir ağına ait süper parametrelerin optimizasyonu [24] gibi çalışmalar verilebilir.

Ayrıca, bulut çizelgeleme problemde de literatürde kullanım örnekleri vardır. [25 – 27]. Bunun dışında, Genetik algoritmalar Montaj hattı çizelgeleme gibi daha farklı kapsamdaki çizelgeleme problemlerinin çözümünde de çokça kullanılmışlardır. [28] ayrıca yine üretilen farklı çizelgeleme algoritmaları için de kıyas algoritması olarak da kullanılmışlardır [29]. Ayrıca bu problemin çok farklı şekillerde ortaya çıkması (iş akışı ya da birbirinden bağımsız görevler, homojen ya da heterojen görevler, bu görevlerin işlemci ya da network ağırlıklı işler olabilmesi, network topolojilerindeki farklılık vs.) sonucu kıyaslama yapılabilecek evrensel bir veri kümesi bulunmadığı sonucuna varılmıştır.

Bu çalışmada 2 adet temel eniyileme kriteri bulunmaktadır. Bunlar zaman ve enerjidir. Buna ilave 2 adet de zamanla ilgili yan kriter; her bir iş için ortalama bekleme süresi ve ortalama bitiş süreleri hesaplanmıştır.



Şekil 6. Bulut Çizelgeleme Problemi (Cloud scheduling problem)

2.1. Zaman ile ilgili ölçütler (Time related Metrics)

Ortalama Bekleme Süresi: Bekleme süresi, bir kullanıcı görevi ya da programın sanal makineye aktarıldıktan sonra (T_s) çalışmaya başlanana T_e kadar geçen süreyi ifade eder. Ortalama bekleme süresi tüm görev ya da programların sürelerinin ortalamasını ifade eder. Eş. 1'de ortalama bekleme süresi formülü tanımlanmıştır.

$$\text{Ortalama Bekleme süresi} = \left(\sum_{i=1}^w T_{e_i} - T_{s_i} \right) \div w \quad (1)$$

Ortalama Bitiş Süresi: Bitiş süresi bir görev ya da programın başlangıçtan (T_0) çalışmayı bitirdiği ana kadar geçen süreyi ifade eder. Ortalama bitiş süresi ise tüm görev ya da programların sürelerinin ortalamasını ifade eder. Eş. 2'de ortalama bitiş süresi matematiksel olarak tanımlanmıştır.

$$\text{Ortalama Bitiş süresi} = \left(\sum_{i=1}^w T_{f_i} - T_0 \right) \div w \quad (2)$$

Toplam Bitiş Süresi (Makespan): Sisteme eklenen bütün görev ya da programların bitiş süresini ifade eder. Tek bir görevin sanal makinede geçirdiği süre aşağıdaki gibi ifade edilir. Eş. 3'te bir görev için gereken işleme süresi verilmiştir.

$$\text{İşleme süresi} (T_{E_u}) = IL_u \div MIPS_{SM_m} \quad (3)$$

Formüldeki IL_u u numaralı görevin toplam komut sayısını, $MIPS_{SM_m}$ ise m numaralı sanal makinenin MIPS (Millions of Instructions per second) cinsinden CPU gücünü ifade eder. Bu formülü kullanarak toplam bitiş süresi (Makespan) Eş. 4'te verilen aşağıdaki matematiksel ifade gibi tanımlanır.

$$\text{Makespan} = (T_{E_u}) \in \{E_0 \dots E_{u-1}\} \quad (4)$$

2.2. Enerji ile ilgili Ölçütler (Energy related Metrics)

Belirli bir çizelgenin harcaması tahmin edilen enerji miktarını ifade eder. Birimi Joule olarak belirtilir. Bu değer 3600'e bölünerek kWh cinsinden de belirtilebilir ve bölgedeki enerji maliyeti biliniyorsa parasal birime çevrilerek maliyet de hesaplanabilir.

Bir bilgisayar, işlemci, anakart, bellek, depolama birimi ve görüntüleme birimi gibi elemanlardan oluşur ve bu elemanlar içerisinde eğer görüntü işlemcisi aktif olarak kullanılmıyorsa, enerji tüketiminin büyük kısmı işlemci tarafında gerçekleşir. Intel [30] Z690 anakart çipi için tam yükte 6W güç tükettiğini belirtmiştir. Yine ünlü bir RAM modülü üreticisi Crucial [31] tipik bir DDR4 bellek kitinin 3 W harcadığını ifade etmektedir. Yine dünyaca ünlü SSD ve Hard Disk üreticisi Western Digital [32] ürün kataloğunda tipik bir sunucu

HDD'sinin 6W civarında güç tükettiğini belirtmektedir. Bunların aksine tipik bir sunucu işlemcisi olan AMD EPYC 773X dünyaca ünlü SPEC [33] benchmark'ına göre tam yükte 447 W güç tüketmektedir. Dolayısıyla bu çalışmada bir sunucu bilgisayarın enerji tüketimi işlemcinin enerji tüketimi olarak kabul edilmiştir.

Güç tüketimi hesaplanması işlemciden işlemciye değişebilir. İşlemcinin hangi saat hızında çalıştığı, kaç işlem biriminden oluştuğu, üretildiği silikon teknolojisi, üretildiği silikon kalitesi gibi çok sayıda etmen işlemcinin harcadığı gücü belirlemektedir. Bu sebeple bu çalışmada sunucu işlemcisi elde bulunmaması sebebiyle bir son tüketici işlemcisi olan AMD Ryzen 2700X üzerinde CPU voltajı sabit (1,212V) tutularak, yük / güç tüketimi tablosu farklı frekans değerlerinde çıkarılmıştır. Bu işlemci 8 çekirdekli bir işlemcidir. Dolayısıyla tek çekirdek tam yükte çalıştığında, Windows görev yöneticisi beklendiği şekilde %12 İşlemci kullanımı göstermektedir. Bu deneylerde işlemci yükünün ve işlemci çalışma frekansının güç tüketimine etkisi incelenmiş ve elde edilen değerler Tablo 2'de verilmiştir.

İşlemcinin çalışma saat frekansının güç tüketimine doğrudan etki edeceği düşünüldüğünde bir tane sabit frekans değerinin seçilip güç tüketiminin buna göre hesaplanması gerekliliği ortaya çıkmıştır. Bunun için 3200 Mhz değeri seçilmiştir. Saat hızının güç tüketimine etkisi ayrı bir çalışmada değerlendirilebilir. Yine boştaki güç tüketimi bu çalışmada kullanılmamıştır, sunucuya %10 yük geleceği varsayılmıştır. Bu varsayımlar ile kullanıma göre oluşan sonuçlar Tablo 3'de gösterilmiştir.

Tablo 3. İşlemci Kullanım- Güç Tüketimi Tablosu (CPU load – power consumption table)

İşlemci Kullanımı (%)	Wh Enerji Tüketimi
%10,0 – %12,5	29,3
%12,6 – %37,5	41,5
%37,6 – %50,0	53,6
%50,1 – %62,5	62,8
%62,6 – %75,0	68,4
%75,1 – %87,5	74,6
%87,5 – %94,0	80,0
%94,1 – %100	85,4

Kullanıcı görevleri sanal makineler üzerinde çalışacağı ve bu sanal makineler de fiziksel makine (sunucu) üzerinde çalışacağı için güç hesabının sunucu düzeyinde yapılması gerekmektedir. Bunun için öncelikle sunucunun hangi zaman aralıklarında hangi sanal makineleri çalıştıracağı belirlenmelidir. Bu çalışmada her sunucunun kapasitesi kadar sanal makine çalıştırabileceği varsayılmıştır. Kapasite üzerine çıkılması mümkün olsa da bu sanal makinelerin

Tablo 2. Ryzen2700X yük - frekans - güç tüketimi (Ryzen 2700X load - frequency – power consumption)

CPU Frekans	Boşta (%0-%5)	1 çekirdek	2 çekirdek	3 çekirdek	4 çekirdek	5 çekirdek	6 çekirdek	7 çekirdek	8 çekirdek
600 MHz	10.3	14	18	20.5	22	23.66	24	26	27
800 MHz	10.8	15	20	23.3	25.5	27.35	28	30.2	31.5
1000 Mhz	11	16	21	24.7	27.75	29	31.7	33	35
1200 Mhz	11.2	17	23	27.2	30.5	32.3	35	37	39
1400 Mhz	11.3	17.4	25	30	33.65	36.1	39.2	40.2	43
1600 Mhz	11.3	17.9	25.5	32	36	38.6	41	44.4	47.3
1800 Mhz	12	19.65	28	35.3	39.5	41.3	46.6	50	53
2000 Mhz	12.2	21.5	31.7	39	45	48.6	53	57	60
2200 Mhz	12.2	22.5	33.8	41.5	48.8	52.5	57.3	61.35	64.5
2400 Mhz	12.5	23.6	35.45	44.6	50.7	55.2	60.45	65.7	68.5
2600 Mhz	12.6	25	37.4	47.2	53.7	58.7	63.4	69.3	72.5
2800 Mhz	12.7	26.5	38.6	49.5	57.6	62.3	67.5	72.7	76.6
3000 Mhz	12.8	28.1	40	51.6	60.7	65.6	71	76.6	81
3200 Mhz	12.8	29.3	41.5	53.6	62.8	68.4	74.6	80	85.4
3400 Mhz	12.8	29.2	43.4	56	65.7	71.2	77	84.6	88.6

MIPS kapasitelerini düşürecek ve sanal makine üzerindeki işlerin uzamasına sebep olacaktır ve bu kapasite düşüşü ek hesaplama gerektirecektir. Bu nedenle bu kısım bu çalışma kapsamına alınmamıştır. Belirli bir $t=i$ anındaki işlemci (Fiziksel Makine – Sunucu) yükü / CPU kullanımı aşağıdaki formülle hesaplanmaktadır. Eş. 5'te fiziksel makineye ait kapasitenin matematiksel tanımı verilmiştir. Ayrıca Eş. 6'da bir fiziksel makineye ait maksimum kapasite tanımlanmıştır.

$$Capacity_{FM} = CoreCount_{FM} * MIPS_{FM} \quad (5)$$

$$U_{FM_{t=i}} = \frac{\sum_{i=0}^{a-1} MIPS_{SM_i}}{MaxCapacity_{FM}} \quad (6)$$

Burada $CoreCount_{FM}$ fiziksel sunucunun işlemcisinin çekirdek sayısını, $MIPS_{FM}$ fiziksel sunucunun MIPS kapasitesini, a değeri bu fiziksel makine üzerinde t anında çalışan sanal makine sayısını, $MIPS_{SM}$ ise çalışmakta olan sanal makinelerin her birine atanan MIPS değerini ifade eder. $U_{FM_{t=i}}$ değerine tabloda karşılık gelen Watt değeri belirtilen zaman aralığındaki güç tüketimi olarak ifade edilir. Burada zaman birimi saniye Watt birimi ise saatlik tüketime karşılık gelmektedir bu nedenle bu değer 3600'e bölünmesi Wh değerini verir.

Buna göre tek bir sunucu için toplam güç tüketimi E_{FM} formülü t delta zaman aralığı (t ve $t-1$ arasında geçen süre için)

$$E_{FM} = Power_{FM_{t=t_i-t_{i-1}}} = U_{FM_{t_i-t_{i-1}}} * \frac{(t_i-t_{i-1})}{3600} * CPU \text{ Wattage} \quad (7)$$

E_{FM} değeri, 1000'e bölünerek kWh değeri bulunur.

3. Literatür Özeti (Literature Review)

Bulut Kaynak Çizelgeleme problemi ile ilgili literatürde hem tek kriter hem de çoklu kriter içeren çalışmalar görülmektedir. Ayrıca çizelgeleme, kullanıcı görevlerinin çizelgelenmesi, sanal makinelerin fiziksel makine/sunucular üzerinde çizelgelenmesi ya da fiziksel makine/sunucuların veri merkezinde çizelgelenmesi gibi konuları içerebilir.

Bu çalışmada kullanıcı görevlerinin çizelgelenmesiyle ilgilenildiği için bu kısımda bu alanda literatürde yapılan çalışmalardan bahsedilecektir. Burada yine yapılan çalışmaya göre birbirinden bağımsız görevler/programlar ya da birbirleriyle bağlantılı (Workflow) görev/programlar olarak 2 temel kategoriye ayrılırlar. Bu alanda yapılan çoğu çalışma kaynak olarak işlemci ve maliyet üzerinde durmuştur. Bu çalışmalarda Parçacık Sürü Optimasyonu, Karınca Kolonisi, Yarasa Algoritması, Genetik Algoritmalar gibi meta sezgisel yöntemler sıklıkla kullanılmıştır.

2022 yılındaki çalışmalarında Xia et. al. [34] çoklu kriter genetik algoritmayla birbiriyle bağlantılı kullanıcı görevlerini (Workflow) toplam tamamlama süresi ve enerji tüketimi kriterleri açısından incelemiştir. Bu çalışmada LCS (Longest Common Subsequence) adını verdikleri tekniği klasik GA mutasyon ve çaprazlama operatörlerine eklemişler, ayrıca algoritmanın başlangıç kısmında daha uzun işlerin daha güçlü sanal makinelere atandığı bir sezgisel algoritma kullanmışlardır. CloudSim kütüphanesi üzerinde Montage, CyberShake, Epigenomics, and LIGO Inspiral Analysis isimli Workflow tabanlı veri setleriyle deneylerini gerçekleştirmişlerdir. Sonuçlarını literatürde sık kullanılan beş algoritmayla karşılaştırmışlardır. Makespan olarak 3 veri setinde, enerji tüketimi olarak da 8 veri setinde bulunan en iyi sonuçları geliştirmişlerdir. 2022 yılındaki çalışmalarında Liang vd., [35] tüm fiziksel sunucular

arasında yük dağılımını eşitleyerek enerji tüketimini azaltmayı hedefleyen bir genetik algoritma önermişlerdir. CloudSim üzerinde rastgele oluşturulmuş bir veri seti üzerinde yaptıkları deneyde önerdikleri algoritma ile, literatürden FFD ve yine literatürden bir başka genetik algoritma ile elde edilen sonuçları karşılaştırmışlar ve önerdikleri IGGA algoritmasının daha dengeli bir yük dağıtımını yapması sayesinde enerji tüketimini diğer iki algoritmaya kıyasla önemli ölçüde azalttığını görmüşlerdir.

Yine 2022 yılında yapılan başka bir çalışmada Li vd. [36] heterojen bir veri merkezinde maliyet düşürüp, performansı aynı anda arttırmaya yönelik bir sanal makine çizelgeleme algoritması önermişlerdir. Bu iki kriter arasında bir ağırlıklı ortalamaya dayanan bir formül yaratarak çoklu kriter fonksiyonunu tek kriter düşürmüşlerdir. Farklı karakteristiklere sahip bir veri setinde kendi algoritmaları, Greedy, Round-Robin ve NSGA II algoritmalarıyla kıyaslama yapmışlardır. Kendi algoritmaları ve NSGA II'nin kaynakları en verimli kullanan algoritmalar olduğu kendi algoritmalarının ayrıca daha yüksek performansa sahip olduğu sonucuna varmışlardır.

2021 yılında yapılan bir başka çalışmada Motlagh vd. [37], birbirine bağımlı (Workflow) görev çizelgeleme problemini güvenilirlik arttırarak ve makespan ölçütünü azaltmak amacıyla Frog-leaping algoritmasını genetik algoritmaya başlangıç popülasyonu yaratma amacıyla eklemişlerdir. Rastgele oluşturulmuş bir veri seti yanında, CyberShake ve Montage veri setlerinde literatürden seçtikleri beş farklı algoritmayı ve kendi algoritmalarını kıyaslamışlar ve güvenilirliği belirli durumlarda %43'e kadar arttırmışlardır.

2016 yılında Zhu vd. [38] tarafından yapılan çalışmada, yine birbiriyle bağlantılı görevlerin çizelgelenmesi probleminde çoklu kriterlere sahip bir genetik algoritma önermektedir. Çalışmada kullanılan ölçütler toplam tamamlama süresi ve kullanıcı maliyeti olarak belirlenmiştir. Deneylerde Montage, CyberShake, Epigenomics, LIGO Inspiral Analysis and SIPHT veri setleri ve bunun yanında kendilerinin rastgele oluşturduğu bir veri seti kullanılmış, NSGA- II algoritmasına dayanan kendi algoritmaları EMS-C'yi literatürden beş farklı algoritmayla kıyaslamışlardır.

2015 yılında Shojaifar vd. [39] Durağan Durum (Steady State) genetik algoritmasına çaprazlama ve uygunluk değeri hesaplama kısımlarında bulanık mantık kullanarak sisteme gelen her görevi kendisine en uygun işlemci, bellek ve bant genişliğine sahip makineye atamaya dayanan bir algoritma önermiştir. Bu uygulamada kullanıcı görevleri/programları birbirlerinden bağımsız varsayılmıştır. Rastgele oluşturulan bir veri kümesinde algoritmalarını Karınca Kolonisi, Bulanık Sinir Ağı ve literatürden bir başka genetik algoritma tabanlı yaklaşımla kıyaslamışlardır. Sonuçta kendi yaklaşımlarının diğer algoritmalarından zaman ve maliyet olarak daha iyi çalıştığını saptamışlardır.

4. Deneysel Metot (Experimental Method)

Tüm deneyler AMD Ryzen 5900X üzerinde kurulu 4 Çekirdek / 8 izlekli 16 GB bellek ayrılmış bir VMware sanal makinesinde çalıştırılmış, işletim sistemi olarak Windows 10 kullanılmıştır. Uygulama Java 18.0.2 x64 platformlarında derlenmiş ve çalıştırılmıştır.

Bu çalışmada bir kullanıcı, bir veri merkezi ve bu veri merkezinde on adet fiziksel makine (sunucu) olduğu varsayılmıştır. Fiziksel makinelerin birbirleriyle aynı olduğu varsayılmış ve her birinde 5000 MIPS'lik 10 adet işlemci çekirdeği olduğu varsayılmıştır. Bu çalışmada sanal makine sayısının değişiminin veri merkezindeki toplam enerji tüketimini ne oranda etkilediği araştırılmıştır. Sanal makineler her biri eşit kapasitede (5000 MIPS) ve tek bir çekirdekte çalışmaktadır. Sunucu sayısının sabit olması dolayısıyla sanal makine

sayısındaki değişim doğrudan her bir fiziksel makinedeki yükü doğrudan etkileyecektir. Deneylerde kullanılan konfigürasyon Tablo 4'te belirtilmiştir.

Tablo 4. Deneylerde kullanılan SM konfigürasyonları (VM configurations used within the Experiments)

SM Sayısı	Veri merkezindeki toplam SM kapasitesi	Ortalama FM yükü
20 SM	100 SM	%20
40 SM	100 SM	%40
60 SM	100 SM	%60
70 SM	100 SM	%70
80 SM	100 SM	%80
90 SM	100 SM	%90
100 SM	100 SM	%100

Kullanıcıya ait 1000 Görev olduğu düşünülmüştür. Bu görevlerin komut uzunlukları 2500 ve 400000 komut arasında değişmektedir. Görev uzunlukları Java'nın Random fonksiyonuyla belirlenmiş ve dağılımı Tablo 5'te gösterilmiştir.

Kullanıcı sanal makinelerinin fiziksel sunuculara atanmasında En Uygun (Best Fit) algoritması kullanılmıştır. Algoritma bütün sanal

makineleri oluşturulma sırasına göre seçip, atama anında kapasitesi en az dolu olan fiziksel makineye atamaktadır. Her sunucu için kapasite başlangıçta eşittir ve 10 SM olarak belirlenmiştir. Deneylerde 2 farklı genetik algoritma çalıştırılmıştır, bu algoritmaların birincisi toplam tamamlama süresi kriterine göre eniyileme yapmaktadır. Bu algoritma GA-Makespan (Toplam Tamamlama süresi) olarak isimlendirilecektir. Diğer genetik algoritma ise enerji metriğine göre en iyileme yapmaktadır. Bu algoritma da GA-Power olarak belirtilecektir. Algoritmalarda tek değişiklik eniyileme fonksiyonundadır. Kullanılan tüm parametreler aynıdır. Kullanılan parametreler Tablo 6'da belirtilmiştir.

Her bir algoritma her bir deney konfigürasyonu (SM sayısı) için 10 defa çalıştırılmıştır ve çıkan sonuçların ortalaması alınmıştır. Deney parametreleri Tablo 7'de özetlenmiştir.

4.1. Deney sonuçları (Results of the Experiment)

Sunucu kullanım kapasitesine göre bitirme süresi / enerji tüketimi ilişkisini gösteren sonuçlar Tablo 8'de özetlenmiştir. Deneyde de görüldüğü üzere fiziksel makinaya atanan SM sayısının artması yani kapasitenin daha iyi kullanılması ortalama bitirme süresi, ortalama bekleme süresi, toplam bitiş süresi (Makespan) ölçütlerine ve bunun yanında aynı zamanda enerji ölçütüne olumlu yansımıştır. Dolayısıyla

Tablo 5. Deneylerde kullanılan görevlerin uzunluklarına göre dağılımı (Distribution of tasks used within the experiments based on their instruction lengths)

RND SEED	<=25000	100,000 - 200,000	200,000 - 400,000	Toplam
RND100	266	265	469	1000
RND101	231	245	524	1000
RND102	245	267	488	1000
RND103	229	260	511	1000
RND104	249	242	509	1000
RND105	246	241	513	1000
RND106	252	269	479	1000
RND107	253	255	492	1000
RND108	221	239	540	1000
RND109	251	258	491	1000
Ortalama	244	254	502	1000
Yüzde	%25	%25	%50	%100

Tablo 6. Deneylerde çalıştırılan GA parametreleri (GA parameters used within the experiments)

GA Parametreleri		
Gösterim	Vektör	GörevID,smID
Çaprazlama tipi	Tek nokta Liste Çaprazlama	%100 olasılık
Mutasyon tipi	5 adet SM Swap	%100 olasılık
Ebeveyn Seçimi	İkili Turnuva Seçilimi	%85 olasılık
Popülasyon Boyutu	100 Birey	
Seçkin Birey Sayısı (Elitizm)	2 Birey	
Sonuçlanma kriteri	150 iterasyon	

Tablo 7. Deneylerde çalıştırılan bulut veri merkezi parametreleri (Cloud datacenter parameters used within the experiments)

Deney Parametreleri	
Veri merkezi sayısı	1
Kullanıcı Sayısı	1
Sunucu / Fiziksel Makine sayısı	10
Sanal Makine sayısı	20-40-60-70-80-90-100
Kullanıcı Görevleri / Programları	1000
Sunucu MIPS Gücü	5000 MIPS
Sunucu Çekirdek sayısı	10
SM MIPS Gücü	5000 MIPS
Görev/Program komut sayısı	Rastgele 2500-400,000

Tablo 8. Sunucu kullanım kapasitesine göre bitirme süresi / enerji ilişkisi (Relationship between cloud server load / energy consumption)

GA Varyantı	Sunucu Kapasite Kullanımı	Toplam bitiş süresi / Makespan (s)	Ortalama bekleme süresi(s)	Ortalama Çalışma süresi (s)	Ortalama Bitiş süresi (s)	Enerji Tüketimi Wh	SM başına ortalama Program Sayısı
GA_Makespan	%20.00	710.250	326.227	13.111	339.338	139.574	50.00
GA_Makespan	%40.00	396.339	163.142	13.337	176.479	102.914	25.00
GA_Makespan	%60.00	293.570	109.328	13.593	122.921	75.756	16.67
GA_Makespan	%70.00	258.589	93.206	13.686	106.893	67.243	14.29
GA_Makespan	%80.00	235.405	80.726	13.547	94.273	60.480	12.50
GA_Makespan	%90.00	219.421	72.207	13.808	86.014	55.924	11.11
GA_Makespan	%100.00	206.984	65.483	13.884	79.367	52.474	10.00
GA_Power	%20.00	890.620	333.742	13.307	347.049	138.474	50.00
GA_Power	%40.00	435.598	162.731	13.268	175.999	100.017	25.00
GA_Power	%60.00	308.122	107.498	13.421	120.920	70.800	16.67
GA_Power	%70.00	267.639	91.297	13.268	104.564	61.608	14.29
GA_Power	%80.00	232.254	79.297	13.247	92.544	54.988	12.50
GA_Power	%90.00	217.734	70.006	13.203	83.210	49.756	11.11
GA_Power	%100.00	208.613	62.516	13.229	75.745	45.763	10.00

Tablo 9. İki genetik algoritmanın farklı sunucu yüklerine göre makespan ve enerji değerleri (Makespan and energy consumption values of two GA variants based on different host loads)

Sunucu Kapasite Kullanımı	Makespan(s)			Enerji (Wh)		
	GA MKS	GA PWR	% Fark	GA MKS	GA PWR	% Fark
%20.00	710.250	890.620	20.3	139.574	138.474	0.8
%40.00	396.339	435.598	9.0	102.914	100.017	2.8
%60.00	293.570	308.122	4.7	75.756	70.800	6.5
%70.00	258.589	267.639	3.4	67.243	61.608	8.4
%80.00	235.405	232.254	-1.4	60.480	54.988	9.1
%90.00	219.421	217.734	-0.8	55.924	49.756	11.0
%100.00	206.984	208.613	0.8	52.474	45.763	12.8

buradan çıkartılabilecek ilk sonuç çizelgeleme algoritması olarak genetik algoritma kullanıldığı durumda sunucu yoğunluğunun artırılması (daha fazla SM yaratılması), her iki ana ölçüde de olumlu yansımaktadır. Bu hesaplama sunucuların boşa enerji tüketimini hesaba katmamaktadır. Boşa enerji tüketimi durumunda daha farklı bir sonuç çıkması olasıdır.

Sunucu kullanım kapasitesine göre bitirme süresi / enerji tüketimi ilişkisini gösteren sonuçlar Tablo 8'de özetlenmiştir. Deneyde de görüldüğü üzere fiziksel makinaya atanan SM sayısının artması yani kapasitenin daha iyi kullanılması ortalama bitirme süresi, ortalama bekleme süresi, toplam bitiş süresi (Makespan) ölçütlerine ve bunun yanında aynı zamanda enerji ölçütüne olumlu yansımasıdır. Dolayısıyla buradan çıkartılabilecek ilk sonuç çizelgeleme algoritması olarak genetik algoritma kullanıldığı durumda sunucu yoğunluğunun artırılması (daha fazla SM yaratılması), her iki ana ölçüde de olumlu yansımaktadır. Bu hesaplama sunucuların boşa enerji tüketimini hesaba katmamaktadır. Boşa enerji tüketimi durumunda daha farklı bir sonuç çıkması olasıdır.

Tamamlama süresi ve Enerji tüketimini en iyileyen iki genetik algoritmanın karşılaştırılması Tablo 9'da yapılmıştır. İki Genetik algoritma kıyaslandığında da toplam tamamlama süresini eniyilemeyi hedefleyen genetik algoritma %70 kapasite doluluk oranına kadar daha iyi gözükürken, %70 kapasite doluluk üzerinde iki GA varyantı arasında bir fark kalmadığı gözlemlenmiştir. Görülen farklar çok küçük farklardır ve ciddi bir sonuç çıkartmak için yetersizdir. Enerji ölçütü incelendiğinde ise, her durumda enerji eniyilemeyi hedefleyen genetik algoritmanın daha başarılı çalıştığı söylenebilir. Burada ayrıca sunucu yük oranı arttıkça yüzdelik farkın da arttığı görülmektedir. %100 ve %90 doluluk oranlarında %10 üzerinde ciddi bir fark görülmektedir.

5. Sonuçlar ve Tartışmalar (Results and Discussions)

Bulut bilişim günümüzde internet altyapısının dünya çapında gelişmesi ve yaygınlaşması sonucu ortaya çıkan hem bireysel yazılımcılar hem de küçük çaplı şirketler için önemli maliyet tasarrufu ve büyük şirketlerle daha adil bir zeminde rekabet şansı tanıyan çok önemli bir teknolojidir. Bulut veri merkezleri sayesinde küçük yatırımcılar giriş maliyeti dediğimiz maliyetleri düşünmeden, o anki ihtiyaçlarına göre kaynak kiralayabilirler. Ayrıca bu model doğası gereği her şirketin kendi altyapısını bulundurduğu modele göre daha verimli ve enerji dostu olduğu da yapılan araştırmalarda ortaya çıkmıştır. Dünya çapında son yıllardaki karbon ayak izinin düşürülmesi trendi de düşünüldüğünde, bu tür veri merkezlerinin daha genişleyeceği ve sayılarının artacağı açıktır. Literatüre baktığımızda da enerji verimliliği konularının son yıllarda önem kazandığını görmekteyiz. Bulut bilişimde, bütün servis modellerinde ve bütün katmanlarda bu makalede de bir kısmına değinildiği üzere, bu tarz çalışmalar için oldukça açık bir alan mevcuttur. Bu iyileştirmeler veri merkezi seviyesinde, sunucu seviyesinde ya da sanal makine üzerindeki çizelgelemeyi değiştirerek yapılabilir.

Bu çalışmada, ilk olarak bir enerji tüketim modeli deneysel olarak elde edilmiştir. Daha sonra sunucu üzerindeki yoğunluğu değiştirerek aynı genetik algoritma enerji ve toplam süre kriterlerini ayrı ayrı optimize edecek şekilde çalıştırılmıştır. Sonuçta enerji ve toplam süre ölçütlerinin birbirleriyle yakın bir ilişki içerisinde olan iki ölçüt olduğu, birisini eniyilemenin diğerini de iyileştirdiği hatta ortalama bekleme süresi ve ortalama bitiş süresi gibi yan ölçütlere de olumlu etkilediği görülmüştür. Bunun yanında, enerji ölçütünü eniyilemenin tam ya da tama yakın sunucu yüklerinde, zaman ölçütü eniyilemeye göre %10 – %13 arasında daha olumlu bir etkisi olduğu saptanmıştır. Böylelikle yüksek sunucu yüklerinde, enerji eniyilemenin amaç fonksiyonu olarak kullanılmasın ile ciddi oranda enerji tasarrufunun

mümkün olabileceği gösterilmiştir. Gelecekteki çalışmalarda boştaki enerji tüketiminin dikkate alınması, farklı işlemci frekanslarının zaman ve enerji tüketimine etkisi ya da farklı kriterleri de göz önüne alarak çoklu kriterli bir genetik algoritma oluşturulması düşünülebilir. Ayrıca yine bu alanda sık kullanıldığını fark ettiğimiz parçacık sürü optimizasyonu (particle swarm optimization), karınca koloni optimizasyonu (ant colony optimization), guguk kuşu arama algoritması (Cuckoo search algorithm) gibi algoritmaların benzer problemlerde başarılı sonuçlar verdiği bilinmektedir, bunun yanı sıra benzetimli tavlama algoritması (simulated annealing) ya da tabu araması gibi klasik üst sezgisel algoritmalar da bu problem için denenebilir. Bu algoritmaları birleştirerek melez bir algoritma üretme ve çözümü daha ne kadar iyileştirdiği de incelenebilir.

Bu makalede AMD Ryzen tabanlı bir işlemci için güç modeli oluşturulmuştur. Bu işlemci ailesi, özellikle işlem birimi sayısı ve enerji tüketimi verimliliğinin özellikle talep gören noktalar olduğu sunucu bilgisayarları pazarında son 2 yılda kullanımı önemli miktarda artan AMD Epyc işlemcilerle aynı çekirdek mimarisini taşımaktadır. Dolayısıyla, güç tüketimi 2 işlemci modeli arasında çok benzer olacaktır. Belirlenen güç modeli aynı şekilde rakip şirket Intel'e ait Xeon ailesi işlemciler için de benzer yöntem ile ölçüm yaparak hesaplanabilir ve iki mimari birbirleriyle kıyaslanabilir. Ayrıca Sunucu marketinde ARM mimarisine sahip Nvidia Grace işlemcileri de yaygınlaşmaktadır. Ancak bu işlemciler tüketici marketine açık olmadığı için karşılaştırma şu an için mümkün gözükmemektedir.

Ayrıca, grafik işlemci (GPU) tabanlı hesaplama da yapay zekâ uygulamalarıyla birlikte yaygınlaşan bir diğer alandır. Bu çalışmada sadece merkezi işlem birimi (CPU) tüketimi baz alınmıştır. Bunun sebebi, grafik işlem biriminin her türlü hesaplamaya uygun bir yapısı olmayışıdır. Ancak, sahne modelleme (rendering) ya da yapay sinir ağı eğitime gibi özel iş yükleri düşünülerek, grafik işlem birimi üzerinde de benzer bir çözümlenmesi yapılabilir. Son olarak, bu çalışmada işletim sistemine ait çözümlenmesi algoritmasının sonuca etkisi de hesaplanmamıştır. Bu deneyler Windows ve Linux ortamlarında tekrarlanarak, işletim sistemi çözümlenmesinin güç tüketimine etkisi de incelenebilir.

Kaynaklar (References)

1. Synergy Research Group. Q2 Cloud Market Grows by 29% Despite Strong Currency Headwinds; Amazon Increases its Share. <https://www.srgresearch.com/articles/q2-cloud-market-grows-by-29-despite-strong-currency-headwinds-amazon-increases-its-share>. Yayın tarihi Temmuz, 28 2022. Erişim tarihi Kasım, 5, 2022.
2. Shehabi A., Smith S., Sartor D., Brown R., Herrlin M., Koomey J., Masanet E., Homer N., Azevedo I., Lintner W., United States Data Center Energy Usage Report, Berkeley National Laboratory, Orlando, 2016.
3. Clarivate. Clarivate Web of Science Document Search. <https://www.webofscience.com/wos>. Erişim tarihi Ağustos, 15 2022.
4. Haitao Y., Bi J., Zhou M., Qing L., Ammari A., Biobjective Task Scheduling for Distributed Green Data Centers, *IEEE Transactions On Automation Science And Engineering*, 18 (2), 731-742, 2021.
5. Othman S., Almalki F., Chakraborty C., Sakli H., Privacy-preserving aware data aggregation for IoT-based healthcare with green computing technologies, *Computers and Electrical Engineering*, 101 (2), 108025, 2022.
6. Medeni I., Virtual Private Server Or Micro Pcs: Which Is Better For The Learning Management System Decision?, 3rd International Conference on Education and Social Sciences (INTCESS), Istanbul - Türkiye, 2016.
7. Öztürk A., Ümit K., Medeni İ., Üçüncü B., Caylan M., Akba F., Medeni T., Green ICT (Information and Communication Technologies): A Review of Academic and Practitioner, *International Journal of e Business and e Government Studies*, 3 (1), 1-16, 2011.
8. Lannelongue L., Grealey J., Inouye M., Green Algorithms: Quantifying the Carbon Footprint of Computation, *Advanced science*, 8 (12), 2021.
9. Zhan Z.-H., Liu X.-F., Gong Y.-J., Zhang J., Chung H. S.-H., Li Y., Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches, *ACM Computing Surveys*, 47 (4), 1-33, 2015.
10. Chu P. C., Beasley J. E., A Genetic Algorithm for the Multidimensional Knapsack Problem, *Journal of Heuristics*, 4 (1), 63-86, 1998.
11. Prins C., A simple and effective evolutionary algorithm for the vehicle routing problem, *Computers & Operations Research*, 31 (12), 1985-2002, 2004.
12. Pezzella F., Morganti G., Ciachetti G., A genetic algorithm for the Flexible Job-shop Scheduling Problem, *Computers & Operations Research*, 35 (10), 3202-3212, 2008.
13. Tosun U., Dokeroglu T., Cosar A., A robust Island Parallel Genetic Algorithm for the Quadratic Assignment Problem, *International Journal of Production Research*, 51 (14), 4117-4133, 2012.
14. Quiroz-Castellanos M., Cruz-Reyes L., Torres-Jimenez J., Gómez C., Huacuja H., Alvim A., A grouping genetic algorithm with controlled gene transmission for the bin packing problem, *Computers & Operations Research*, 55 (1), 52-64, 2015.
15. Ha Q., Deville Y., Pham Q., A hybrid genetic algorithm for the traveling salesman problem with drone, *Journal of Heuristics*, 26, 219-247, 2020.
16. Wang R., Ying G., Lai L., LigBuilder: A Multi-Purpose Program for Structure-Based Drug Design, *Molecular modeling annual*, 6, 498-516, 2000.
17. Olague G., Mohr R., Optimal camera placement for accurate reconstruction, *Pattern Recognition*, 35 (4), 927-944, 2002.
18. Yılmaz A., Güzel M., Bostancı E., Askerzade I., A Novel Action Recognition Framework Based on Deep-Learning and Genetic Algorithms, *IEEE Access*, 8, 00631-100644, 2020.
19. Prasad T., Park N., Multiobjective Genetic Algorithms for Design of Water Distribution Networks, *Journal of Water Resources Planning and Management*, 130 (1), 73-82, 2004.
20. Zhou Y., Wang Y., Wang K., Kang L., Peng F., Wang L., Pang J., Hybrid genetic algorithm method for efficient and robust evaluation of remaining useful life of supercapacitors, *Applied Energy*, 260, 2020.
21. Shahid F., Zameer A., Muneeb M., A novel genetic LSTM model for wind power forecast, *Energy*, 223, 2021.
22. Zhu J., Optimal reconfiguration of electrical distribution network using the refined genetic algorithm, *Electric Power Systems Research*, 62 (1), 37-42, 2002.
23. Arabasadi Z., Alizadehsani R., Roshanzamir M., Computer aided decision making for heart disease detection using hybrid neural network-Genetic algorithm, *Computer Methods and Programs in Biomedicine*, 141, 19-26, 2017.
24. Beyaz S., Açıcı K., Sümer E., Femoral neck fracture detection in X-ray images using deep learning and genetic algorithm approaches, *Joint Diseases and Related Surgery*, 31 (2), 175-183, 2020.
25. Cui H., Li Y., Liu X., Ansari N., Liu Y., Cloud service reliability modelling and optimal task scheduling, *IET Communications*, 11 (2), 161-167, 2016.
26. Mohammad S., Javanmardi S., Saeid A., Cordeschi N., FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method, *Cluster Computing-The Journal of Networks Software Tools and Applications*, 18 (2), 829-844, 2015.
27. Li H., Wang B., Yuan Y., Zhou M., Fan Y., Xia Y., Scoring and Dynamic Hierarchy-Based NSGA-II for Multiobjective Workflow Scheduling in the Cloud, *IEEE Transactions on Automation Science and Engineering*, 19 (2), 982-993, 2022.
28. Kılınççı Ö., Assembly line balancing problem with resource and sequence-dependent setup times (ALBPRS), *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38 (1), 557-570, 2023.
29. Sarac T., Ozelik F., A matheuristic algorithm for multi-objective unrelated parallel machine scheduling problem, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 38 (3), 1953-1966, 2023.
30. Intel Corporation. Intel® Z690 Chipset. <https://www.intel.com/content/www/us/en/products/sku/218833/intel-z690-chipset/specifications.html>. Erişim tarihi Kasım, 5 2022.
31. Crucial. How Much Power Does Memory Use? <https://www.crucial.com/support/articles-faq-memory/how-much-power-does-memory-use>. Erişim tarihi Kasım, 5 2022.

32. Western Digital. Product Brief: WD Red Pro NAS HDD. https://documents.westerndigital.com/content/dam/doc-library/en_us/assets/public/western-digital/product/internal-drives/wd-red-pro-hdd/product-brief-western-digital-wd-red-pro-hdd.pdf. Erişim tarihi Kasım, 5 2022.
33. Standard Performance Evaluation Corporation. First Quarter 2022 SPECpower_ssj2008 Results. https://www.spec.org/power_ssj2008/results/res2022q1/. Yayın Tarihi Mart, 24 2022. Erişim tarihi Kasım, 5 2022.
34. Xia X., Qiu H., Xu X., Zhang Y., Multi-objective workflow scheduling based on genetic algorithm in cloud environment, *Information Sciences*, 606 (1), 38-59, 2022.
35. Liang B., Liu R., Dia D., Design of Virtual Machine Scheduling Algorithm in Cloud Computing Enviroment, *Journal of Sensors*, 2022, 2022.
36. Li J., Zhang R., Zheng Y., QoS-aware and multi-objective virtual machine dynamic scheduling for big data centers in clouds, *Soft Computing*, 1-14, 2022.
37. Aida A. M., Movaghar A., Rahmani A. M., A new reliability-based task scheduling algorithm in cloud computing, *International Journal of Communication Systems*, 35 (3), 2022.
38. Zhu Z., Zhang G., Li M., Liu. X., Evolutionary multi-objective workflow scheduling in cloud, *IEEE Transactions on parallel and distributed Systems*, 27 (5), 1344-1357, 2015.
39. Shojafar M., Javanmardi S., Abolfazli S., Cordeschi N., FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method, *Cluster Computing*, 18 (2), 829-844, 2015.