

Mathematical Programming with C#.NET

Yunus DEMİR^{1*}

¹Department of Industrial Engineering, Bursa Technical University
yunus.demir@btu.edu.tr, ORCID: 0000-0003-3868-1860

Abstract: Mathematical programming (optimization) is the expression of decision problems with mathematical equations. In this study, tools used for solving mathematical models are introduced. In particular, solution tools that should serve as embedded in various C#.Net software projects, namely software libraries, are mentioned. These libraries are effective tools for building and solving mathematical models like writing ordinary .NET code. In addition, providing access to more than one solver with the same syntax brings these tools to the fore in mathematical optimization. In the continuation of the study, the Google OR-Tools library, which is offered free of charge, is mentioned in detail. Solvers in the Or-Tools library and some solvers accessible by this library are tested with a sample problem and their solution speeds are presented comparatively. It is thought that this study, which is intended to contribute to the Turkish literature, will be beneficial for both practical life practitioners and theorists.

Keywords: Optimization, mathematical programming, mathematical programming solvers

C#.NET ile Matematiksel Programlama

Özet: Matematiksel programlama (optimizasyon), karar problemlerinin matematiksel denklemler ile ifade edilmesidir. Bu çalışmada, matematiksel modellerin çözümü için kullanılan araçlar tanıtılmıştır. Özellikle çeşitli C#.Net yazılım projeleri içinde gömülü bir şekilde hizmet vermesi gereken çözüm araçlarına yani kütüphanelere (API) değinilmiştir. Bu kütüphaneler, sıradan .NET kodu yazar gibi matematiksel modellerin oluşturulması ve çözülmesi konusunda etkili araçlardır. Ayrıca aynı sözdizimi ile birden fazla çözücüye erişim sağlanması, matematiksel optimizasyon konusunda bu araçları öne çıkarmaktadır. Çalışmanın devamında ücretsiz sunulan, Google OR-Tools kütüphanesinden detaylı bir şekilde bahsedilmiştir. Or-Tools kütüphanesinde yer alan çözücüler ve yine bu kütüphane tarafından erişilebilir bazı çözücüler örnek bir problem ile test edilmiş ve çözüm hızları karşılaştırmalı bir şekilde sunulmuştur. Türkçe literatüre katkı sağlaması amaçlanan bu çalışmanın, hem pratik hayat uygulayıcılar hem de teorisyenler açısından faydalı olacağı düşünülmektedir.

Anahtar Kelimeler: Optimizasyon, matematiksel programlama, matematiksel programlama çözücüleri

Reference to this paper should be made as follows (bu makaleye aşağıdaki şekilde atıfta bulunulmalı):

Demir Y., 'Mathematical Programming with C#.NET', Elec Lett Sci Eng , vol. 17(2), (2021),96-104.

1. GİRİŞ

Matematiksel bir model, daha iyi kararlar vermek veya gerçek durumu daha iyi anlamak için kullanılabilecek gerçek bir durumun matematiksel temsilidir. Matematiksel modeller; statik/dinamik, doğrusal/doğrusal olmayan, tam sayılı, deterministik/stokastik olmak üzere birçok farklı şekilde sınıflandırılmaktadır [1].

Matematiksel optimizasyonda kullanılan birincil araç, matematik programlama (optimizasyon) çözücüleridir. Kullanıcılar, matematiksel modeller oluşturmak için çok çeşitli programlama veya modelleme dillerini kullanabilirler [2]. Farklı çözücülerin kendi arabirimleri bulunmaktadır (örneğin, IBM'in Cplex stüdyosu). Modeli girmek için IDE (*Integrated Development Environment*) veya arayüzler için belirli sözdizimini kullanabilir ve ardından çözmek için seçilen çözücü çağrılabilir. Hepsinin arkasındaki mantık aynı olsa da, bu diller veya sözdizimleri genellikle bir çözücüden diğerine farklılık gösterir. Eğer problemin birden fazla çözücü ile çözülmesi gerekiyorsa, çözücüye özgü gösterimleri ortak bir dile dönüştürülebilecek genel bir

* Yunus DEMİR; Tel.:0224 300 3831, yunus.demir@btu.edu.tr

forma ihtiyaç olacaktır. Bu dönüştürme, bazı cebirsel modelleme dilleri (*Algebraic modeling language - AML*) kullanılarak yapılmaktadır [3].

Cebirsel modelleme dilleri, kullanıcıların optimizasyon problemlerini üst düzeyde (değişkenler, kısıtlamalar, amaç vb. tanımlayarak) kolayca temsil etmelerini sağlar. Bunlar problemleri çözmez, çözüm için programlama dillerinden çağrılabilen API (*Application Programming Interface*) olarak adlandırılan bir dizi rutinden oluşan kütüphanelerdir. Verileri kitaplığa aktarmanın birkaç yolu bulunmaktadır, bunlar:

Giriş dosyaları aracılığıyla: Matematiksel modelleri bir çözücünden veya platformdan diğerine geçirmek için kullanılan standart bir biçimdir [4]. MPS (*Mathematical Programming System*), LP (*Linear Programming*), MPI (*Math Programming Interface*) gibi birçok farklı giriş dosya formatları bulunmaktadır. Detaylı bilgi için atıfta [5] verilen linke tıklayınız.

MPS, IBM tarafından geliştirilmiştir. Yaygın olarak kabul edilen bir standarttır ve çoğu çözücü tarafından desteklenmektedir. Ancak çok okunabilir değildir. MPS formatı sütun yönelimli (*column-oriented*) bir formattır. Problemler, satır (kısıt) yerine sütun (değişken) tarafından belirlenmektedir [6].

MPI, dosya formatı, LINDO Systems tarafından matematik programlama modellerini temsil etmek için taşınabilir bir format olarak geliştirilmiştir. Lineer modellerden lineer olmayan modellere kadar tüm matematik programlarını saklayabilmektedir. MPS dosyalarında olduğu gibi, MPI formatı skaler tabanlıdır. En büyük dezavantajı, bir LINGO modeli MPI formatına dönüştürüldüğünde modeldeki tüm setlerin kaybolmasıdır. Bu nedenle, bir modeli kaydederken, yerel LINGO formatında kaydedilmesi yani her zaman KAYDET komutunu kullanılmalıdır (LINGO IDE için).

LP dosya formatı IBM tarafından geliştirilmiştir. MPS gibi doğrusal ve karesel modellerin gösterimi için başka bir standart dosya formatıdır. Ancak, LP formatındaki denklemler standart cebirsel format kullanılarak gösterilir. Bu da LP dosyalarının okunmasını ve yorumlanmasını MPS dosyalarından çok daha kolay hale getirmektedir [7]. Problemleri, satır yönelimli (*row-oriented*) bir biçimde girmek için şekillendirilmiş CPLEX'e özgü bir dosya formatıdır. Başka bir deyişle, LP formatı, problemleri kısıtlar açısından girmenizi sağlamaktadır [8].

Bir IDE aracılığıyla: Entegre geliştirme ortamları (IDE), yeniden düzenleme, otomatik tamamlama ve derleme hatalarının düzeltilmesi gibi genel görevleri otomatikleştiren araçlardır. Bu araçların iki amacı vardır: geliştirici hızını artırmak ve geliştirici hatalarını azaltmak. Bu araçlar yaygın olarak kullanılır (Örn: Visual Studio) [9]. Matematiksel modellerin çözümünde kullanılan önemli araçlardan olan IDE'ler modellerinin derlenmesini/yürütülmesini başlatma ve izleme yeteneğine sahip genel bir metin düzenleyicidir. Bu araçlar ile çözücülerin seçimini kolaylaştırır ve parametrelerin dosya bazında yönetilmesi sağlanır [10]. Bu alanda hizmet veren birçok ticari (Gams, Lindo, Aimms, Ampl...) ve ücretsiz (Lpsolve, Gusek) IDE bulunmaktadır.

API aracılığıyla: API, modeli bellekte oluşturmak, çözmek ve sonuçları döndürmek için bir programlama dilinden çağrılabilen bir dizi rutindir. Birçok olası görevi gerçekleştirmek ve çeşitli seçenekler belirlemek için birçok API rutini vardır. Bu tip araçları kullanmak, tıpkı normal .net kodunu yazar gibi matematiksel programlar oluşturma ve çözme imkânı sağlamaktadır. Ayrıca problem oluşturduktan sonra, onu çözmek için birçok çözücü kullanılabilir [11].

Bu çalışmada C# .NET ile kullanılabilecek AML'ler tanıtılmış ve sonrasında Google OTR-Tools ile bir model kodlanmış ve OR-Tools tarafından desteklenen çözücüler ile çözülmüştür. Sonuçlar karşılaştırmalı bir şekilde sunulmuştur.

2. C# 'DAN ÇAĞRILABİLEN BAZI API'LER

Bu bölümde öncelikle matematiksel optimizasyon dünyasında birbiri ile karıştırılabilen çözücü, matematiksel modelleme dili ve matematiksel modelleme ara yüzü kavramlarına açıklama getirilecektir.

Matematiksel programlama (optimizasyon) çözücüleri, aşırı derecede büyük problemlerin modellenmesi ve çözülmesi için kullanılırlar [12]. Doğrusal, karma tam sayılı, doğrusal olmayan veya kısıt programlama problemlerini çözebilecek gelişmiş algoritmalar içerirler. Ticari çözücülere: Snopt, Baron, Cplex, Gurobi gibi birçok örnek verilebilir. Ücretsiz çözücülerden bazıları ise GLPK, Scip, Clp şeklindedir. Bir çözücü, problem tanımlarını bir tür genel formda alır ve çözümlerini hesaplar. Bahsedilen genel forma matematiksel modelleme/programlama dili denir. Modelleme dilleri çözücü ile kullanıcı arası bağlantıyı kurmaktadır. Ticari modelleme dillerine örnek olarak: Gams, Lingo, Ampl, Cplex, ücretsiz modelleme dillerine örnek olarak: GLPK, Soplex, lp-solve verilebilir. Çözücüler benzer türdeki diğer problemlere kolayca uygulanabilecek şekilde geliştirilmektedirler [13]. Çözücülere modelleme ara yüzleri aracılığı ile ulaşılabilmektedir. Bu ara yüzler tek başına bir bilgisayar programı (GAMS, Lingo,...) veya bir yazılım kütüphanesi (Gurobi, Cplex, Lp-Solve...) şeklinde olabilmektedir. Ara yüzler (giriş bölümünde AML olarak tanıtılan kavram) modeli çözemez, sadece çözücüye özgü gösterimleri ortak bir dile dönüştürerek çözücülere erişimi sağlar. Yazılım kütüphanelerine çeşitli programlama dillerinden (C#, python,...) erişilebilmektedir. Bu yöntem, projelerine gömülü bir şekilde optimizasyon hizmeti sunmak isteyen yazılımcılar tarafından tercih edilir ve doğrudan uygulama kodundan çözücüye erişimi sağlar.

Bu bölümde C#. NET ile kullanılabilecek matematiksel optimizasyon kütüphanelerinden bazıları açıklanmıştır.

2.1. Optano

OPTANO modelleme, OPTANO Platformunun bir parçasıdır. OPTANO Modelleme, matematiksel programlar oluşturmaya ve bunları çözücülere göndermeye yardımcı olan bir .net-API'dir. MIT lisansı¹ ile dağıtılan Optano, doğrusal modelleri, Karma tam sayılı modelleri ve bazı sınırlamalarla quadratic modelleri çözebilir. OPTANO modelleme, tüm .net platformlarında (Windows, Linux ve Mac) kullanılabilir. GLPK, Z3 gibi ücretsiz çözücüleri ve Gurobi, CPLEX ve Xpress gibi çeşitli ticari çözücüleri desteklemektedir [14]. Kullanımı için .Net Framework 4.6.1 ve sonrası sürüm olmalıdır. Nuget üzerinden Optano kütüphaneleri yükleyebilmektedir. Desteklenen çözücülerin kullanılabilmesi için ilgili çözücüye ait kütüphaneler ayrıca indirilmelidir. GLPK çözücüsünü kullanmak için küçük bir düzenleme yapılmalıdır: GLPK çözücüsüne ait kütüphane <http://winglpk.sourceforge.net/> sitesinden indirilir. İnen dosya C sürücüsüne yapıştırıldıktan sonra proje referanslarına Win64² dosyasında yer alan "libglpk-cli.dll", proje referanslarına eklenmelidir. Proje özelliklerinden "Derleme" sekmesinden Platform hedefi x64 olarak değiştirilmelidir. Çeşitli örnek projelere atıfta [15] verilen adresten erişilebilmektedir.

¹ https://tr.wikipedia.org/wiki/MIT_Lisans%C4%B1

² İşletim sisteminiz 64 bit ise

2.2. LP-Solve

LP_Solve bir karma tam sayılı doğrusal programlama çözücüsüdür. Revize simplex metodu ve Dal-Sınır yöntemini kullanmaktadır. Model boyutunda bir sınırlama bulunmamaktadır. C, C++, Pascal, Delphi, Java, VB, C#, VB.NET, Excel, ... gibi farklı dillerden ve uygulamalardan kütüphane olarak da çağrılabilir. Daha genel bir ifadeyle harici kitaplıkları (Windows altında DLL'ler, Unix/Linux altında paylaşılan kitaplıklar (.so)) çağırabilen herhangi bir programlama dili LP_Solve API'yi çağırabilir. Ayrıca bir sürücü programı aracılığıyla AMPL, MATLAB, O-Matrix, Scilab, Octave, R'den de çağrılabilir. LP_Solve, ANSI C ile Eindhoven Teknoloji Üniversitesi'nde Michel Berkelaar tarafından geliştirilmiştir. Linux ve WINDOWS gibi birçok farklı platformda derlenebilmektedir [16].

2.3. Extreme Optimization (Numerical Libraries)

Bu kütüphane, doğrusal cebir, karmaşık sayılar, sayısal entegrasyon ve türev alma, denklem çözme, optimizasyon, rassal sayılar, regresyon, ANOVA, istatistiksel dağılımlar, hipotez testleri dahil olmak üzere çok çeşitli sayısal teknikleri kapsayan geniş bir algoritma seti sunmaktadır. NET Framework dillerinden herhangi biri ile uygulama geliştirilebilmektedir. .NET framework 3.5 ve sonrası sürümleri gerektirmektedir [17].

2.4. MOSEK

MOSEK, doğrusal, karışık tam sayılı doğrusal, karesel, karışık tam sayılı karesel, karesel kısıt, konik matematiksel optimizasyon problemlerinin çözümü için bir yazılım paketidir. MOSEK, son derece verimli bir iç nokta algoritması kullanarak büyük ölçekli doğrusal, dışbükey karesel kısıt ve konik programları çözmek için etkili bir araçtır. MOSEK, bir optimizasyon problemini çözmek için birden fazla CPU'dan (veya birden fazla çekirdekli bir CPU'dan) yararlanabilmektedir [18]. MOSEK çözücü, C, C#, Java, MATLAB, Python ve R dilleri için arayüzleri bulunmaktadır [19]. AMPL ve GAMS gibi başlıca modelleme sistemleri MOSEK ile uyumlu hale getirilmiştir. Akademik lisans ile ticari olmayan kullanımlar için sınırsız kullanım imkânı sınırlanmıştır. MOSEK'in en güçlü yanı, sürekli lineer, karesel ve konik problemler için son teknoloji dâhili nokta (*interior-point*) optimize edicisidir. MOSEK, bilgisayarınızda yerel bir lisansa sahip olmadan problemlerinizi uzaktan çözebileceğiniz optimizasyon sunucu hizmeti de sağlamaktadır [20].

2.5. Google OR-Tools

OR-Tools, kombinatoriyal optimizasyon problemleri için, çok sayıda olası çözüm arasından bir en iyi çözümü bulmaya çalışan, için açık kaynaklı bir yazılımdır. OR-Tools, C++ ile yazılmıştır, ancak Python, Java veya C# ile de kullanılabilir. Nuget üzerinden kolayca yüklenebilir. Yüklenmesi ile ilgili detaylara atıfta [21] verilen linkten erişilebilir. Kullanılabilmesi için Windows 10 64-bit (x86_64) işletim sistemi, Microsoft Visual Studio 2019 veya Visual Studio Community 2019 (.Net Framework 4.5.2 sürümü ve sonrası) gerekmektedir. OR-Tools, çeşitli çözümleri çağırarak bir CMD olarak hizmet sunmaktadır. Yansın en çok bilinen kombinatoriyal optimizasyon problemlerine özgün çözümler de sunmaktadır. Bunlar:

- Atama problemi
- Rotalama problemleri (gezgin satıcı, araç rotalama...)
- Kutu paketleme problemleri
- Şebeke akış problemleri (maksimum akış, minimum maliyet akışı...)

- Çizelgeleme problemleri (personel çizelgeleme, atölye çizelgeleme)

CMD olarak OR-Tools, doğrusal, tam sayılı ve kısıt optimizasyon, olmak üzere üç farklı sınıfta hizmet sunmaktadır. Bunlar alt başlıklar halinde açıklanmıştır:

2.5.1. Doğrusal optimizasyon

OR-Tools, doğrusal optimizasyon problemlerini çözmek için iki alternatif sunmaktadır:

GLOP: Google'ın açık kaynak olarak sunulan in-house doğrusal çözücüsüdür.

Google Apps Komut Dosyasında Doğrusal Optimizasyon Hizmeti: Geliştiricilerin doğrusal optimizasyon problemlerini çözmek için işlev çağrılarını yapmasına imkan tanır. Tüm değişkenlerin gerçek değerler alabileceği saf doğrusal optimizasyon problemleri için GLOP'u kullanır. Herhangi bir değişken tamsayı olarak sınırlandırılmışsa SCIP kullanır.

2.5.2. Tam sayılı optimizasyon

OR-Tools, tam sayılı programlama problemlerini çözmek için iki ana araç sunmaktadır:

MIP Çözücü: Standart dal ve sınır tekniklerini kullanan bazı üçüncü parti MIP çözücü için bir platformdur.

CP-SAT Çözücü: SAT (*satisfiability*) yöntemlerini kullanan bir kısıt programlama çözücüsüdür. Tam sayılı optimizasyon problemlerinin çözümünde de kullanılabilir. Hesaplama hızını artırmak için CP-SAT çözücüsü tamsayılar üzerinde çalışmaktadır. Bu, optimizasyon probleminin yalnızca tamsayılar kullanarak tanımlanması gerektiği anlamına gelmektedir. Tamsayı olmayan terimlerle kısıtlamaları olan bir problem, tüm terimlerin tamsayı olması için önce bu kısıtlamaları yeterince büyük bir tamsayı ile çarpılması gerekmektedir. Örnek için atıfta [22] verilen bağlantıya tıklayınız. CP-SAT çözücüsü, değişkenlerin çoğunun 0-1 tipinden olan problemler için daha uygundur [23].

Hem tamsayı hem de 0-1 değişkenleri olan tipik karma tam sayılı modeller için, iki çözücü arasında genellikle net bir hız farkı yoktur, bu nedenle seçiminiz kişisel tercihlere bağlıdır. Google ayrıca AI Workshop aracılığıyla bir MILP çözücüyü bir bulut API'si sunar. Detaylar için atıfta [24] verilen bağlantıya başvurunuz. OR-Tools tarafından desteklenen çözücüler aşağıda kısaca tanıtılmıştır:

CLP (Coin-OR Linear Programming): Kâr amacı gütmeyen bir eğitim vakfı olan COIN-OR tarafından geliştirilmiştir. Mart 2004' kurulan topluluk, modellerin, algoritmaların ve son teknoloji hesaplamalı araştırmaların geliştirilmesini ve dağıtımını hızlandırmak için yöneylem araştırması yazılımı geliştirmeyi amaç edinmiştir. COIN-OR projesi ilk olarak 2000 yazında Atlanta'daki 17. Uluslararası Matematiksel Programlama Sempozyumu (ISMP) konferansında kamuya duyurulmuştur. CLP, C++ ile yazılmış açık kaynaklı bir doğrusal programlama çözücüsüdür. Öncelikle çağrılabilir bir kitaplık olarak kullanılması amaçlanmıştır, ancak bağımsız bir yürütülebilir sürüm de mevcuttur. CLP Eclipse Public License (EPL)³ kapsamında açık kaynak kodu olarak yayınlanmıştır. Kod öncelikle, şimdi IBM Research'ten emekli olan John J. Forrest tarafından yazılmıştır.

³ https://en.wikipedia.org/wiki/Eclipse_Public_License

CBC (Coin-OR Branch and Cut): CBC, C++ ile yazılmış açık kaynaklı bir karma tamsayı programlama çözücüsüdür. Çağrılabilir bir kitaplık olarak veya bağımsız bir yürütülebilir dosya kullanılarak kullanılabilir. AMPL, GAMS, AIMMS ve Excel gibi birçok uygulama aracılığı ile çağrılabilir [25].

GLPK (GNU Linear Programming Kit): GLPK paketi, büyük ölçekli doğrusal programlama, karma tam sayılı programlama ve diğer ilgili sorunları çözmek için tasarlanmıştır. ANSI C'de yazılmış ve çağrılabilir bir kitaplık şeklinde düzenlenmiş bir dizi rutindir [26]. GLPK, Moskova Havacılık Enstitüsü'nden Andrew O. Makhorin tarafından geliştirilmiştir. İlk sürümü Ekim 2000'de yayınlanmıştır. Tamsayı olmayan problemler için revize edilmiş simpleks yöntemini ve primal-dual iç nokta yöntemini ve (karma) tamsayı problemleri için Gomory'nin karışık tamsayı kesimleriyle birlikte dal-ve-sınır algoritmasını kullanmaktadır [27].

SCIP (Solving Constraint Integer Programs): SCIP şu anda karışık tam sayılı programlama ve karışık tam sayılı doğrusal olmayan programlama için ticari olmayan çözücülerden biridir. Çözüm sürecinin tam kontrolüne ve çözücünün tüm detaylarına erişime izin verir. SCIP'in de bileşeni olduğu SCIP Optimizasyon Paketinin diğer üyeleri şu şekildedir:

SoPlex: Doğrusal programlama çözücüsü

PaPILO: Tamsayı ve doğrusal optimizasyon için paralel ön çözüme

ZIMPL: Matematiksel programlama dili

UG: Karışık tam sayılı (doğrusal ve doğrusal olmayan) programlar için paralel çerçeve

GCG: Genel dal-kesme ve fiyat çözücü [28]

BOP (Binary Optimizer): BOP, 0-1 problemleri için geliştirilmiştir. Tamsayı değişkenleri için destek yavaş ve deneyseldir. 0-1 modellerde CPLEX/Gurobi ile rekabetçi bir çözücüdür. Doğrusal programlama gevşemesi kötü ise BOP üstündür. Doğrusal programlama gevşemesi çok iyiye ve/veya probpem neredeyse tek modüllüyse Gurobi / CPLEX üstün gelmektedir [29].

SAT(Satisfiability): Birçok "rastgele" sezgisel yöntem ile SAT; değişken dallanma sırası (sırayla, ters, rastgele), dal seçimi (her zaman doğru, her zaman yanlış, en iyi objektif, rastgele, ...) ve farklı çözücü parametrelerini denemektedir [29].

3. GOOGLE OR-TOOLS İLE BİR UYGULAMA

Bu bölümde bir örnek problem (atölye tipi çizelgeleme problemi) Google OR-Tools yardımı ile modellenmiştir. Daha sonra oluşturulan model farklı boyuttaki test problemleri için, OR-Tools tarafından çağrılabilen çözücüler ile çözdürülmüş ve karşılaştırmalı sonuçlar paylaşılmıştır.

Bu bölümde önce ele alınan problem tanıtılmış daha sonra modele ait bilgiler ve model formülasyonu sunulmuştur.

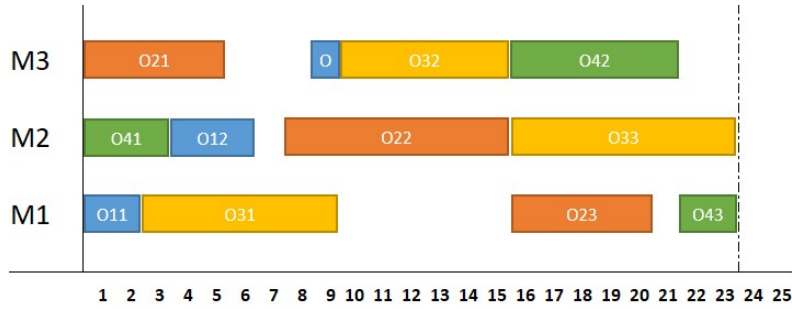
3.1. Problemin Tanımı

Atölye tipi çizelgeleme problemi, m makine ve işlenmeyi bekleyen n işten oluşmaktadır. Her iş önceden belirlenmiş sırada ve önceden bilinen sürede m makinenin tamamında veya bir kısmında işlenmektedir. Amaç maksimum yayılımı (Cmax) minimize etmektir. Problemin anlaşılması için bilgileri Tablo 1'de verilen dört iş, üç makineli küçük bir örnek paylaşılmıştır.

Tablo 1. Örnek probleme ait veriler

İş (Rota)	M1	M2	M3
1 (1-2-3)	2	3	1
2 (3-2-1)	4	8	5
3 (1-3-2)	7	8	6
4 (2-3-1)	2	3	6

Verilen örnek için optimum çizelge, Şekil 1’de sunulmuştur.



Şekil 1. Örnek problem için optimum çizelge

3.2. Problemin matematiksel modeli

Bölümün bu kısmında modelde kullanılan indis, parametre ve karar değişkenleri tanımlanmış ve model formülasyonu verilmiştir.

İndisler

- i, f: İş indeksi $i, f \in \{1, 2, \dots, n\}$
 j, g: Operasyon indeksi
 k: Makine indeksi $k \in \{1, 2, \dots, m\}$

Parametreler

- p_{ik} : i işinin k operasyonunun işlem süresi
 O_k : k makinesinde işlenen operasyon seti

Karar değişkenleri

- C_{ij} : i işinin j operasyonunun tamamlanma zamanı
 x_{ijfg} : i işinin j operasyonu, f işinin g operasyonundan sonra yapılırsa, 1
 Aksi takdirde, 0

Model formülasyonu

$$C_{ij} \geq p_{ij} \quad \forall i, \forall j = 1 \quad (1)$$

$$C_{ij} \geq C_{ij-1} + p_{ij} \quad \forall i, \forall j > 1 \quad (2)$$

$$C_{ij} \geq C_{fg} + p_{ij} - M * (1 - x_{ijfg}) \quad \forall i, j \text{ ve } \forall f, g \in O_k \quad (3)$$

$$C_{fg} \geq C_{ij} + p_{fg} - M * (x_{ijfg}) \quad \forall i, j \text{ ve } \forall f, g \in O_k \quad (4)$$

$$C_{max} \geq C_{ij} \quad \forall i, j \quad (5)$$

$$\text{Min. } Z = C_{max} \quad (6)$$

Modelde (1) numaralı denklemde tüm işlerin ilk operasyonlarının tamamlanma zamanları en az işlem süreleri kadar olmaları sağlanıyor. (2) numaralı denklemde, aynı işe ait operasyonlardan, öncül operasyon tamamlanmadan ardıl operasyonun başlamasına izin verilmiyor. (3) ve (4) numaralı denklemlerde aynı makineye atanmış operasyonlar sıralanmaktadır. (6) numaralı denklem amaç fonksiyonudur ve maksimum yayılımın (C_{max}) minimize edileceğini ifade etmektedir.

3.3. Sayısal sonuçlar

Oluşturulan model Google Or-Tools kütüphanesi yardımı ile C# programlama dili ile kodlanmış ve Intel Core i5-8250U CPU, 1.60GHz özelliklerinde bir bilgisayar ile çözdürülmüştür. Google OR-Tools tarafından çözücülerden altısı ile çeşitli boyuttaki 12 problem çözdürülmüş ve optimum çözüm için CPU zamanları Tablo 2’de paylaşılmıştır.

Tablo 2. Karşılaştırmalı sonuçlar

Problem Boyutu		Çözücüler					
İş (n)	Mak. (m)	GUROBI	CBC	SCIP	BOP	SAT	CPSAT
8	3	1,6	3,55	13,23	0,34	0,24	0,32
9	3	1,76	5,79	17,77	-	0,28	0,48
9	4	0,19	3,85	0,87	0,22	0,16	0,36
10	3	1,12	345,68	89	-	0,39	2,42
10	4	0,75	29	1,44	0,32	0,17	0,39
11	3	35,32	-	-	-	21,10	8,82
12	3	141,82	-	-	-	146,17	58,83
12	4	9,4	-	225	-	6,25	4,14
13	3	-	-	-	-	-	366,37
13	4	12,94	-	-	-	17,33	9,23
13	5	4,7	-	-	-	3,52	2,89
14	3	-	-	-	-	-	-

Her bir problem için model, en fazla 600 saniye koşturulmuştur. Sonuçlara bakıldığında Google OR-Tools kütüphanesinde yer alan ücretsiz SAT ve özellikle CPSAT çözücüsünün, ticari Gurobi çözücüsüyle rekabet edebildiği görülmektedir.

4. SONUÇ

Bu çalışmada matematiksel modellerin çözümünde kullanılan araçlara yönelik bir araştırma yapılmıştır. Öncelikle bu alanda kullanılan ve sıklıkla birbiri ile karıştırılan kavramlara açıklık getirilmiştir. Ayrıca C#.Net den çağrılabilen matematiksel çözücü kütüphaneleri tanıtılmıştır. Bu kütüphaneler, projeye gömülü bir şekilde optimizasyon çözümleri elde etmek isteyen yazılımcılar tarafından oldukça tercih edilmektedir. Sıradan bir .Net kodu yazar gibi birden fazla çözücüye erişme imkanı sağlaması, bu kütüphanelerin tercih edilmesindeki en önemli faktörlerdendir. Çalışmada Google OR-Tools kütüphanesine detaylı bir şekilde yer verilmiş ve bu kütüphane ile çağrılabilen bazı çözücüler detaylı bir şekilde açıklanmıştır. Son olarak OR-Tools kullanarak bir uygulama yapılmıştır. OR-Tools ile çağrılabilen bazı çözücüler, çözüm hızları bakımından test edilmiş ve sonuçlar karşılaştırmalı bir şekilde sunulmuştur. Google OR-Tools kütüphanesinde yer alan ve ücretsiz bir şekilde erişilebilen SAT ve CPSAT çözücülerinin

özellikle düşük bütçeli projelerde, karma tam sayılı problemlerin çözümü konusunda iyi bir alternatif olabileceğini göstermektedir.

References

- [1] Winston, W. L., & Goldberg, J. B. (2004). Operations research: applications and algorithms (Vol. 3).
- [2] gurobi.com/resources/mathematical-optimization-web-page/
- [3] <https://or.stackexchange.com/questions/1504/what-is-the-purpose-of-libraries-like-pyomo-and-google-or-tools>
- [4] <https://www.lindo.com/downloads/PDF/LINGO.pdf>
- [5] https://www.gurobi.com/documentation/9.1/refman/model_file_formats.html
- [6] https://www.rpi.edu/dept/math/math-programming/cplex66/sun4x_56/doc/refman/html/appendixE3.html
- [7] <https://www.lindo.com/downloads/PDF/LINGO.pdf>
- [8] https://www.rpi.edu/dept/math/math-programming/cplex66/sun4x_56/doc/refman/html/appendixE.html
- [9] G. C. Murphy, M. Kersten, and L. Findlater. How are Java software developers using the Eclipse IDE? IEEE Software, 23(4):76–83, July 2006. doi: 10.1109/MS.2006.105.
- [10] <https://www.gams.com/latest/docs/gamside/contents.htm#:~:text=The%20GAMS%20IDE%20is%20a,monitored%20in%20the%20process%20window.&text=The%20IDE%20also%20facilitates%20the,a%20file%20by%20file%20basis>
- [11] <http://lpsolve.sourceforge.net/5.5/>
- [12] Küçükdeniz, D. D. T. Uygulamalı Optimizasyon Teknikleri.
- [13] <https://en.wikipedia.org/wiki/Solver>
- [14] <http://docs.optano.net/modeling/current/userDoc/supportedsolvers.html>
- [15] <https://github.com/OPTANO/optano.modeling.examples/tree/master/Source>
- [16] <http://lpsolve.sourceforge.net/5.5/>
- [17] <https://www.extremeoptimization.com/Documentation/Default.aspx>
- [18] https://www.gams.com/latest/docs/S_MOSEK.html#MOSEK_INTRODUCTION
- [19] Dahl, J., & Andersen, E. D. (2021). A primal-dual interior-point algorithm for nonsymmetric exponential-cone optimization. Mathematical Programming, 1-30.
- [20] <https://solve.mosek.com/web/index.html>
- [21] <https://developers.google.com/optimization/install/dotnet>
- [22] https://developers.google.com/optimization/cp/integer_opt_cp
- [23] <https://developers.google.com/optimization/mip/mip>
- [24] <https://aihub.cloud.google.com/u/0/p/products%2F03a54ca4-f9ba-489b-bbb3-b6ca8c22c5cf>
- [25] <https://projects.coin-or.org/Cbc>
- [26] <https://www.gnu.org/software/glpk/>
- [27] https://en.wikipedia.org/wiki/GNU_Linear_Programming_Kit
- [28] <https://www.scipopt.org/>
- [29] http://aixia2015.unife.it/wp-content/uploads/slides/invited/Perron-OR_at_Google.pdf